



POLITECNICO DI TORINO

Master Degree course in Cybersecurity

Master Degree Thesis

Retrieval-Augmented Social Media Intelligence: Detecting and Reporting of High-Risk Communication Patterns using Large Language Models

Supervisors

Prof. Andrea ATZENI

Prof. Paolo DAL CHECCO

Candidate

Simona BERTÈ

OCTOBER 2025

*To those who are always
running: you can have
everything, just not all at
once.*

*If the sun rose in an instant,
there would be no dawn.*

Summary

The growing prevalence of social media has generated massive amounts of digital data, becoming a primary source for OSINT and SOCMINT. The proliferation of high-risk content poses a critical challenge for intelligence teams, who require advanced tools to effectively identify and analyze these phenomena. Large Language Models (LLMs), as part of the broader integration of Artificial Intelligence into the intelligence cycle, offer significant opportunities to automate and enhance analytical processes, enabling faster and more efficient management of the vast data available on social networks.

This work develops a system based on Retrieval-Augmented Generation (RAG) technology, designed to support intelligence teams in the automated analysis of Twitter profiles. The system identifies communication patterns linked to high-risk phenomena and generates preliminary reports that guide further investigation.

A knowledge base framework was designed for four risk categories (terrorism and extremism, cybercrime and hacking, hate speech and cyberbullying, mental health), with initial implementation focused on terrorism, containing multiple documents that describe distinctive linguistic and communicative features such as keywords, hashtags, and emojis that correspond to specific subcategories. These documents are derived from sources including intelligence reports, academic papers, and behavioral studies. Tweets are collected, preprocessed, and classified as belonging to one of the four categories or neutral. The system then identifies the most prevalent category for the overall profile being analyzed and further associates each analyzed tweet with specific subcategories within that category by comparing each tweet to the relevant documents, measuring the semantic similarity between tweets and documents. The final structured report, generated by the LLM, justifies the classification by providing evidence based on the identified communication patterns distinctive to each subcategory.

Validation through terrorism-related profiles demonstrates the system’s capability in identifying relevant communication patterns and generating coherent preliminary reports. Performance depends on prompt design, the quality of the RAG knowledge base, and the underlying LLM. The system provides intelligence teams with an efficient tool for the preliminary assessment of potentially critical profiles, contributing to the automation of intelligence processes. The RAG-enhanced approach enables contextualized and transparent analysis, supporting decision-making by providing both classifications and textual evidence.

Acknowledgements

A chi, quando il mondo sembrava fermarsi, ha trovato sempre il modo di rimetterlo in moto per me.

Contents

1	Introduction	9
2	State of the Art	13
2.1	Open Source Intelligence (OSINT)	13
2.1.1	OSINT and the Intelligence Disciplines	13
2.1.2	Benefits and Challenges of OSINT	14
2.1.3	Parties Interested in OSINT Information	14
2.1.4	OSINT Tool and Techniques	16
2.2	Social Media Intelligence (SOCMINT)	17
2.2.1	Parties Interested in SOCMINT Information	18
2.2.2	Challenges of SOCMINT	18
2.2.3	Social Media Profiling	18
2.3	AI and Large Language Models	30
2.3.1	Machine learning (ML)	30
2.3.2	Deep learning (DL)	31
2.3.3	Explainable AI (XAI)	31
2.3.4	Generative Artificial Intelligence (Generative AI)	32
2.3.5	Natural Language Processing (NLP)	33
2.3.6	Large Language Models (LLMs)	33
2.3.7	AI Agents	41
2.4	AI and OSINT	41
2.4.1	AI and SOCMINT	47
2.5	Large Language Models and OSINT	54
2.5.1	Retrieval-Augmented Generation (RAG) for OSINT	60
2.6	Conclusions	65
3	Technologies and Tools	67
3.1	Social Media Scraping	67
3.1.1	Scraping Methods	67
3.1.2	Challenges and Solutions in Social Media Scraping	68
3.1.3	Choice of Strategy and Motivations	72
3.1.4	Comparison of Leading Social Media Scraper APIs (2025)	73
3.2	Data Preprocessing	79
3.3	Text Classification Techniques using LLMs	81

3.4	The Cheshire Cat AI Framework	82
3.4.1	Cheshire Cat Architecture	83
3.4.2	Message	87
3.4.3	Plugin	88
3.5	Large Language Models and Embedding Models	89
3.5.1	Large Language Model	89
3.5.2	Embedding Model	93
4	Legal and Ethical Considerations	95
4.1	The Legal Status of Social Media Data	95
4.1.1	Personal Data	95
4.1.2	Special Categories of Personal Data	96
4.1.3	Legal Basis for Processing Personal Data and Special Categories of Personal Data	96
4.1.4	Intelligence and Law Enforcement Context	97
4.1.5	Data Protection Principles and Implementation Requirements	99
4.2	Legal Risks of Web Scraping	100
4.2.1	Key Aspects to Consider	100
4.2.2	Opinions and Case Studies	102
4.3	AI Usage in Sensitive Contexts: Key Legal and Ethical Challenges	106
4.3.1	Core Problematic Areas	106
4.3.2	Legal Requirements	108
5	System Architecture and Pipeline	111
5.1	Overview of the Pipeline	111
5.2	Choice of LLM and Embedder	112
5.3	RAG Documents: Role and Structure	113
5.3.1	Document creation	114
5.3.2	Document Ingestion	116
5.4	X Profile Input and Tweet Extraction	119
5.4.1	Tweet Volume Configuration Considerations	120
5.5	Data Cleaning and Preprocessing	121
5.6	Preliminary Classification Module	121
5.7	Retrieval-Augmented Generation Module	125
6	Evaluation and Testing	133
6.1	Evaluation Objectives	133
6.2	Dataset Creation and Preparation	134
6.2.1	Dataset for Preliminary Classification	134
6.2.2	Dataset for RAG Evaluation	135
6.3	Evaluation of Preliminary Classification	140
6.3.1	Performance Metrics	140
6.3.2	Experimental Design	141
6.3.3	Testing Methodology	142
6.3.4	Qualitative Analysis	145

6.3.5	Final consideration	146
6.4	Evaluation of the RAG Pipeline	146
6.4.1	Performance Metrics	146
6.4.2	Ingestion Evaluation	149
6.4.3	Retrieval Evaluation	150
6.4.4	Generation Evaluation	155
6.4.5	Profile Configuration Impact Analysis	170
6.4.6	Final consideration	172
7	Conclusions	175
7.1	Conclusion	175
7.2	Future Work	175
A	User Guide	179
A.0.1	Prerequisites	179
A.0.2	Installing Cheshire Cat AI	179
A.0.3	Starting Cheshire Cat	180
A.0.4	Initial LLM and Embedder Configuration	180
A.0.5	Installing the Report Generator Plugin	182
B	Developer Guide	185
B.0.1	document_filter.py	185
B.0.2	profiling.py	186
B.0.3	prompt.py	187
B.0.4	plugin.json	187
B.0.5	requirement.txt	187
	Bibliography	189

Chapter 1

Introduction

A single social media post, word, emoji, or hashtag can be leveraged for purposes far more significant than one might expect, both implicitly, as a coded language conveying deeper meanings, and explicitly. In today’s digital landscape, social media platforms represent more than simple spaces for interaction and entertainment. They have evolved into complex ecosystems where information flows with unprecedented volume and speed, fundamentally transforming how individuals and groups interact and express opinions.

This evolution poses substantial challenges for intelligence, law enforcement, and security agencies, particularly in detecting and profiling harmful behaviors. High-risk communications can originate from a wide spectrum of actors: terrorist and extremist networks recruiting followers or coordinating operations, cybercriminals orchestrating or selling illicit activities and products, individuals spreading hate, or people experiencing mental health crises with the potential for harmful outcomes. The ability to identify and analyze these actors and their content is critical. Early detection can prevent attacks, disrupt criminal networks, safeguard vulnerable individuals, and ensure public safety in both digital and physical domains.

In response to these challenges, Open Source Intelligence (OSINT) and Social Media Intelligence (SOCMINT) have emerged as indispensable disciplines, enabling systematic collection, analysis, and interpretation of publicly available online data. Subsequently, the rapid advancement of Artificial Intelligence (AI) technologies has opened unprecedented possibilities for automating and enhancing these intelligence processes. These sophisticated tools can identify subtle patterns, extract relevant insights from large datasets, or generate context-aware reports, potentially revolutionizing the accuracy, scalability, and efficiency of intelligence operations.

As of 2025, Italy’s Information System for the Security of the Republic, alongside several international agencies, is actively expanding SOCMINT capabilities. This includes recruiting specialists dedicated to analyzing social media data for national security purposes, from conducting in-depth reporting to continuous monitoring of security-relevant issues, often with the support of automation tools. These efforts underscore the strategic importance of harnessing digital intelligence to navigate the complexities of the modern information environment.

Despite growing academic and operational interest, the application of Large Language Models (LLMs) and Retrieval-Augmented Generation (RAG) systems to intelligence driven social media profiling remains largely unexplored. Although existing research within OSINT and SOCMINT contexts has extensively examined approaches based on traditional AI, including Machine Learning (ML) and Deep Learning (DL), and more recently explored general applications of LLMs, a significant gap persists in understanding how these advanced AI systems can be integrated with expert domain knowledge.

This research seeks to bridge this gap by exploring the potential of RAG systems to enhance intelligence analysis through the strategic combination of AI-driven automation with domain-specific expertise. The study aims to design, develop and evaluate, in the form of a Proof of Concept (PoC), an innovative pipeline in which a RAG system, powered by a LLM, incorporates knowledge of already known high-risk communication patterns on social media, contributed by psychologists, sociologists, terrorism analysts, and cybercrime specialists. More specifically, the research investigates whether such a hybrid, multidisciplinary system can effectively identify high-risk profiles on X (formerly Twitter) by comparing user tweets with the knowledge embedded in the RAG system, and by generating detailed profiling reports. These reports are intended not only to highlight the rationale behind each assessment and trace correspondences with known patterns, but also to serve as a decision support tool for intelligence and law enforcement teams, rather than as a replacement for human judgment. The research question is of both theoretical and practical relevance: theoretically, as it addresses the intersection of multiple disciplinary fields; and practically, as it offers tangible benefits for the effectiveness of intelligence and law enforcement operations.

For the sake of terminological consistency, this thesis adopts the name “X” when referring to the social media platform previously known as Twitter, regardless of the publication date of the referenced content. The term “tweet” is used to refer to user-generated posts on X. Although the platform now refers to them simply as “posts”, the term “tweet” is retained here for clarity and consistency.

The experimental research methodology is structured around two main phases: system development and system evaluation. Prior to development, the system’s knowledge base is enriched with documents containing the descriptions of frequently used keywords, hashtags, and emojis associated with high-risk communication. These documents were specifically created for this research to mimic expert-curated intelligence materials, drawing on multiple sources such as intelligence reports, academic papers, and behavioral studies. This knowledge base forms the foundation of the RAG system, which is implemented using the Cheshire Cat AI framework. The system functions as an intelligent chatbot that, given a X profile URL, automatically collects and analyzes tweets, and generates comprehensive profiling reports. The evaluation of the system follows a dual approach: quantitative assessment, using classical classification and retrieval metrics; qualitative analysis, comparing generated reports against established ground truth data to assess the reliability and relevance of the profiling outputs. For testing purposes, tweets are sourced from publicly available datasets or synthetically generated content, avoiding real-time scraping during evaluation. Nevertheless, the operational pipeline is designed to function as a chatbot capable of receiving X profile URL and collecting tweets via

carefully selected scraping API.

Finally, the thesis is structured as follows:

- Chapter 2 presents the state of the art, introducing key concepts in AI, including ML, DL, LLMs, RAG, and AI agents, as well as the fundamentals of OSINT and SOCMINT. Particular attention is given to existing research on the integration of AI and LLM into OSINT and SOCMINT, with a focus on the embedding RAG within intelligence workflows. The chapter also explores high-risk social media profiling from both linguistic and behavioral perspectives, drawing on insights from academic and governmental research.
- Chapter 3 describes the technologies and tools used in this work, including social media scraping techniques, data preprocessing, LLM-based classification methods, and the Cheshire Cat AI framework, encompassing LLMs and embedders.
- Chapter 4 addresses legal and ethical considerations of the research, covering topics such as legal risks of social media scraping, data protection regulations, the use of AI in sensitive contexts for automating decision-making, and ethical implications of profiling, both from the perspective of researchers and of intelligence and law enforcement professionals.
- Chapter 5 introduces the system architecture and experimental pipeline, providing a detailed description of the methodological setup.
- Chapter 6 presents and analyzes the results of the conducted evaluations.
- Chapter 7 concludes the thesis, highlighting the main findings, discussing limitations, and proposing directions for future research.

Chapter 2

State of the Art

This chapter presents a comprehensive overview of the application of Artificial Intelligence (AI), with particular attention to the role of Large Language Models (LLMs), in the field of Open Source Intelligence (OSINT), focusing especially on Social Media Intelligence (SOCMINT).

2.1 Open Source Intelligence (OSINT)

Open Source Intelligence (OSINT) refers to all information that is publicly available and can be legally accessed without violating copyright or privacy laws [1].

Publicly available and open data becomes OSINT when it undergoes systematic processing and analysis to generate actionable intelligence, then distributed to relevant actors within the intelligence framework [2]. This process is known as the **intelligence cycle**, based on planning, collection, processing, analysis, dissemination and evaluation [3].

Although OSINT is gaining increasing attention today, it is not a new area of interest for intelligence operations. Even before the advent of the World Wide Web, intelligence services relied on offline OSINT, monitoring newspapers, radio broadcasts, television [4].

However, the emergence and rapid expansion of the Internet have dramatically increased the volume, variety, and accessibility of online sources, including social media, search engines, websites, online forums, company directories, and databases. Specifically, the term SOCMINT refers to the sub-disciplines focused on data extracted from social media [1].

During the collection of OSINT sources, analysts may occasionally encounter classified or restricted information. This type of content is referred to as NOSINT, in contrast to OSINT. Although it falls outside the scope of open-source intelligence in legal terms, it is often still considered during intelligence assessments [1].

2.1.1 OSINT and the Intelligence Disciplines

OSINT is just one of several intelligence collection disciplines. Other key categories include human intelligence (HUMINT), signals intelligence (SIGINT), measurement and

signature intelligence (MASINT), and geospatial intelligence (GEOINT). These disciplines often overlap, with one frequently complementing or enabling another.

HUMINT, one of the oldest forms of data collection, involves gathering or confirming information through human sources, using overt, covert, or clandestine methodologies.

SIGINT involves intercepting and exploiting signals, remotely collected via ground stations, aircraft, or satellites. It is divided into three sub-disciplines: communication intelligence (COMINT), electronic intelligence (ELINT), and foreign instrumentation signals intelligence (FISINT).

MASINT analyzes equipment emissions, such as radar, acoustic, and nuclear signals, within the electromagnetic spectrum to detect information patterns not yet exploited.

GEOINT combines satellite or aircraft imagery and geospatial information to support decision-making in military and civilian contexts [5].

2.1.2 Benefits and Challenges of OSINT

OSINT provides several advantages. Relying on publicly available information reduces the risks associated with other forms of intelligence, such as deploying spy satellites or operating human assets in hostile environments. It is also typically more cost-effective. Furthermore, OSINT is highly accessible and constantly updated, making it a flexible resource for a wide range of intelligence contexts.

However, OSINT also presents challenges. First, it generates a huge amount of data that must be processed and analyzed to extract meaningful insights. A further critical issue concerns source reliability. OSINT often needs to be cross-verified with classified or otherwise trusted information to ensure accuracy and credibility. Human intervention remains essential, as analysts must validate the output of automated tools against reliable sources, a process that demands significant time and resources. Finally, although OSINT is based on publicly available data, its use must still adhere to applicable legal frameworks. Compliance with national and international legal standards is therefore not only a matter of ethics, but also a prerequisite for the operational legitimacy of OSINT activities [1].

2.1.3 Parties Interested in OSINT Information

OSINT serves a wide range of actors across public and private sectors.

Government, International Organizations, and Law Enforcement Agencies

Governments use OSINT for national security, counterterrorism, cyber surveillance, gauging public opinion, and informing both domestic and foreign policy. It also enables near real-time monitoring of foreign media, including translation of broadcasts and online content.

International organizations use OSINT to support peacekeeping operations. Humanitarian organizations rely on OSINT to guide relief efforts and safeguard supply chains, often by analyzing online platforms to predict future threats.

Law enforcement agencies use OSINT to prevent and investigate crimes such as sexual abuse, identity theft, and online fraud. By analyzing digital footprints, law enforcement agencies can detect emerging threats and monitor transnational criminal networks [1].

Business Corporations Businesses use OSINT to investigate new markets, monitor competitors, plan marketing activities, and anticipate factors that may affect their current operations and future growth.

In addition to financial uses, corporations use OSINT for cybersecurity, detecting potential data leaks, identifying vulnerabilities, and developing threat intelligence strategies. By combining external OSINT with internal data, organizations strengthen their cyber-risk management policies, protecting financial interests, brand reputation, and customers [1].

Black Hat Hackers and Criminal Organizations Malicious actors leverage OSINT to gather intelligence on targets, conduct reconnaissance, and support social engineering attacks [1].

Privacy-Conscious People Individuals concerned with privacy use OSINT to assess their digital footprint, understand potential vulnerabilities, and protect themselves against identity theft and online surveillance [1].

Terrorist Organizations Terrorists use OSINT sources to plan attacks, gather intelligence on targets, recruit members via social media, and spread their propaganda across the world [1].

Open Source Intelligence in Cybersecurity OSINT is a powerful methodology in cybersecurity, enabling professionals identify potential threats and vulnerabilities while improving their ability to protect digital assets effectively. Passive OSINT, which involves the collection of data without direct interaction with the target, allows experts to gather valuable insights without alerting potential adversaries. In contrast, Active OSINT, which involves direct interaction and engagement with the target, such as participating in online forums or extracting information through interactive means, empowers professionals to access information beyond what is readily available and gain deeper understanding of potential risks.

The applications of OSINT in cybersecurity are extensive. Dark Web OSINT, for example, uncovers potential hidden threats by monitoring illicit marketplaces, forums, and encrypted communication channels, providing early insights into emerging cyber-criminal activities. It also enables experts to evaluate the external attack surface in order to identify vulnerabilities visible to potential attackers. Additionally, OSINT supports the analysis global events that may impact digital security; understanding geopolitical landscape helps anticipate potential cyber threats arising from international tensions, ensuring a more comprehensive and adaptive cybersecurity posture.

OSINT enhances incident response and digital forensics by providing intelligence on attack patterns and IoCs. It also strengthens defenses against social engineering and targeted attacks by gathering relevant information on potential targets. OSINT helps additionally organizations evaluate their digital exposure and maintain alignment with data security regulations.

Integration with automation, machine learning, and tools such as Google Dorks allows for faster, more precise collection and analysis of relevant data, helping to identify emerging threats in real-time. OSINT further fosters collaboration and information sharing between organizations and public institutions, enhancing collective resilience against cyberattacks.

By integrating OSINT into cybersecurity workflows, organizations can anticipate and respond to threats more effectively, adopting a proactive and comprehensive approach to defending against the constantly evolving digital risk landscape.

2.1.4 OSINT Tool and Techniques

A wide range of tools and techniques support OSINT operations throughout the intelligence cycle, enabling investigators to collect, process, and analyze information more efficiently.

Effective tools are essential in OSINT operations to reduce the workload of investigations. For a comprehensive list of OSINT tools, refer to [6].

Examples of OSINT techniques include:

- Social media investigation, collecting information about individuals, groups, or organizations from publicly available social media content.
- Reverse image search, identifying the origin or context of images across the web.
- Scraping, automatically collecting large volumes of data from websites.
- Geolocation and mapping, determining the physical location from digital footprints, images, or social media posts.
- DNS and domain analysis, mapping domains, subdomains, and associated infrastructure to understand potential attack surfaces.
- WHOIS and certificate analysis, investigating domain ownership and SSL/TLS certificates to uncover registration details and potential links between entities.
- Vulnerability and exploit database search, searching for reported vulnerabilities and current exploits.
- Advanced web search techniques, performing targeted searches to uncover relevant information beyond standard queries.
- Metadata analysis, extracting information from files and documents, to identify origins, authors, or modifications.
- IoT and exposed services discovery, using tools to identify connected devices and publicly exposed services.
- Breach data analysis, consulting leaked credential databases to identify compromised accounts and potential attack vectors.

- Blockchain analysis, tracing transactions on public blockchains to support cyber-crime investigations.
- Network footprinting, collecting publicly available information on IP addresses, servers, and network services to assess external exposure.
- Monitoring and alerts, tracking changes on websites, social media accounts, or other digital assets to detect emerging threats or relevant activity.

2.2 Social Media Intelligence (SOCMINT)

In today's digital era, it is increasingly rare to find an Internet user without at least one social media account. According to recent studies, approximately 94.2 percent of global Internet users are active on social media. The most recent data report that there are 5.24 billion active social media user identities around the world, an increase of 4.1 percent over the preceding twelve months [7].

Social media platforms host vast amount of information, including political views, religion beliefs, ethnicity, country of origin, frequently visited locations, multimedia content, such as personal images and videos, relationship status, social activities, home and work addresses, educational and professional histories, significant event dates, such as birth, graduation, job transitions, and social interactions.

The collection and analysis of such information fall within the scope of **Social Media Intelligence (SOCMINT)**.

SOCMINT encompasses a wide range of nuances, particularly concerning the types of information that can be collected and the diversity of platforms that can be leveraged for intelligence purposes. It is important to note that SOCMINT includes all social media platforms, not only social networking sites like Facebook and LinkedIn. Information can be found on media-sharing sites like Instagram, image-sharing like Pinterest, video-sharing like Youtube, forums like Reddit, microblogging platforms like X, social gaming platforms like Xbox Live.

Online investigators aim to collect a wide range of social media content types, including profile information, posts, comments, replies, multimedia content, social interactions, and associated metadata. Metadata, in particular, can include valuable contextual details such as the date, time, and geolocation of a post or uploaded file, contributing significantly to the analytical value of the content.

The accessibility of this information varies depending on privacy settings: some content is publicly accessible, while other data may be restricted and require specific authorization to access. The classification of social media content as OSINT remains a topic of ongoing debate, as discussed further in Section 4. Although most social media platforms require user registration to access full site content, surveys indicate that users still expect to have some degree of privacy for their online activities, even when sharing content publicly. Nevertheless, the prevailing view among security experts is that information voluntarily shared on publicly accessible platforms qualifies as OSINT [1].

2.2.1 Parties Interested in SOCMINT Information

Although SOCMINT is commonly associated with investigations targeting individuals or groups involved in criminal activities, its applications extend far beyond traditional law enforcement contexts. SOCMINT can play a vital role in a wide range of investigations, including counter-terrorism, human trafficking, organized crime, child sexual exploitation, population density, economic and market analysis, disaster prediction and response, health and disease monitoring, drug use, emerging trends, and cyber-crime.

This list is by no means exhaustive, but it illustrates the diverse potential of SOCMINT across multiple sectors. Although government and law enforcement agencies remain primary users, private companies and non-governmental organizations also leverage SOCMINT to support strategic decision-making, improve marketing, ensure brand reputation, and identify emerging social and economic trends [8].

2.2.2 Challenges of SOCMINT

While OSINT faces significant legal and ethical challenges, these are significantly amplified within the scope of SOCMINT. Social media platforms present a particularly complex landscape for intelligence gathering, where issues of data volume, veracity, reliability, user privacy, and technical access limitations become even more pronounced.

One of the primary challenges in SOCMINT is the overwhelming volume of data. With billions of active users across platforms, extracting relevant information from this vast data pool becomes a difficult task. Many platforms aim to protect user identities, complicating investigations efforts. The presence of fake accounts and automated bots further complicates the identification of reliable sources.

The proliferation of disinformation, such as propaganda, AI-generated content, fake news, and coordinated influence operations, raises concerns about the integrity and reliability of the collected intelligence.

SOCMINT also raises serious ethical and privacy concerns. Investigators risk crossing legal and moral boundaries, especially when engaging in surveillance facilitated by governments, data brokers, or the platforms themselves.

Technical defenses such as CAPTCHAs, WAFs, and API limitations are implemented by social media platforms to prevent automated scraping. These measures pose additional challenges for SOCMINT analysts, especially those without access to professional tools or advanced automation. Attempting to bypass such defenses may also contravene legal and platform-specific regulations.

Lastly, covert research often necessitates identity masking, which is resource-intensive and ethically questionable.

In summary, SOCMINT inherits all of OSINT's challenges but intensifies them due to the vast, dynamic, and tightly controlled nature of social media. Balancing operational needs with legal and ethical constraints remains a core tension [9].

2.2.3 Social Media Profiling

Human Behavior and its Relevance in the Digital Era The study of human behavior is crucial for understanding the mechanisms that drive both constructive and

harmful social dynamics. Human motivations are often described using Maslow’s hierarchy of needs: physiological, safety, love, esteem, and self-actualization [10,11]. Fulfillment of these needs generally correlates with positive social behaviors, while unmet needs can lead to harmful actions.

Human behavior emerges from a complex interplay of personal experiences, intrinsic psychological traits, and external factors such as environmental conditions, crises, socio-economic pressures, or significant life events [11–13]. Personality traits significantly influence how individuals perceive the world, make decisions, and engage in social interactions, often reflected in observable behavioral patterns, including those expressed online [14,15]. Understanding these patterns provides a foundation for analyzing human behavior in digital contexts, where individual psychology and collective social dynamics increasingly intersect.

From Human behavior to Social media Behavior In the digital sphere, social media platforms act both as mirrors of personal identity and as amplifiers of ideas, emotions, and intentions. They provide a space without precedents for individuals to express themselves, consciously or unconsciously revealing aspects of their motivations, beliefs, and personality through content, interactions, and network structures.

Online behavior can be considered a digital manifestation of broader human behavioral principles. Online interactions are shaped not only by individual motivations and traits, typical aspects of traditional human behavior, but also by the affordances of the platforms, related to anonymity, virality, algorithmic recommendation, which can influence communication style, emotional expression, and even tendencies to risk.

Social Media Profiling: Applying Behavioral Insights Social media profiling, a core component of SOCMINT, applies the principles of behavioral analysis to the traces individuals leave online. It involves the collection and analysis of data from posts, interactions, and network connections of social media platforms to create detailed user profiles [16]. The purpose is to identify patterns in the data that reveal details about the target’s beliefs, interests, habits, personality traits, and social relationships, ultimately compiling these elements into a comprehensive profile, equivalent to a virtual representation of each user [17].

Building on insights into online human behavior, recent studies have explored how individuals express deviant, risky, or socially significant behaviors on social media. The following subsection synthesizes key findings from the literature on behavioral patterns in four specific domains of interest: terrorism and extremism, cybercrime and hacking, depression and suicide, hate speech and cyberbullying. These areas cover a wide range of motivations, from ideology and personal turmoil to organized crime and individual acts of aggression.

Examining research across these areas helps identify recurring tactics, communicative strategies, and linguistic markers that can enhance understanding of online human behavior. Despite their differences, these domains share identifiable online signatures in language use, interaction patterns, and thematic content, making them especially suitable for behavioral analysis and profiling in social media contexts.

Extremism and Terrorism

Extremism refers to holding extreme beliefs or ideologies that deviate significantly from mainstream or moderate views, whereas terrorism is a tactic employed by extremist individuals or groups to achieve their goals through violence, fear, and intimidation.

The roots of extremism and terrorism are complex, involving ideological, social, and psychological factors. Through a multidisciplinary approach that combines psychological profiling and sociological analysis, experts examine how extremist ideologies, such as religious or political extremism, interact with socioeconomic issues like marginalization, trauma, and political exclusion, creating feelings of hopelessness. Extremist groups offer a sense of belonging and purpose to isolated individuals, while personal traits such as narcissism and impulsivity can increase the likelihood of radicalization.

The use of social media by terrorist and extremist organizations has been widely studied by academics and international institutions, including UNICRI, INTERPOL, and the Institute for Strategic Dialogue. Social media has become a central tool for terrorists and extremist organizations, used for propaganda dissemination, radicalization, recruitment, psychological warfare, information sharing, disinformation, operational coordination, weapon tutorials and incitement to data mining and fundraising [18, 19]. Despite ongoing efforts by platforms to identify and remove extremist content, these actors continue to exploit these channels with increasing sophistication.

Although different groups may adopt distinct approaches, several tactics are commonly employed and understanding these is a key part of countering their influence.

Social media differs from traditional media in interactivity, reach, frequency, usability, immediacy, and permanence. Unlike the traditional “one-to-many” model, social media platforms enables anyone to create, share, and access content, enabling direct and continuous engagement. This capacity to proactively reach target audiences makes social media a powerful tool for influence, recruitment, and mobilization. In some ideological contexts, especially within jihadist frameworks, participation in “electronic jihad” is considered a legitimate form of warfare, conferring the symbolic status of mujahid on online participants [18].

Terrorists use both mainstream social networking platforms, such as Facebook, Instagram, X, as well as encrypted communication channels like Telegram and Whatsapp, as well as online video games. A common strategy involves initially publishing content on Telegram, followed by its redistribution on more public platforms like X, which remain widely used platforms for disseminating extremist material. These ecosystems are often interconnected: public accounts often link to private or encrypted spaces for deeper ideological indoctrination and even operational guidance [20]. Encrypted platforms enhance operational security by enabling confidential communication and reducing exposure to surveillance [21].

A primary aim is psychological warfare, aiming to spread anxiety, fear, intimidation and hatred among target populations, while fostering admiration and solidarity among sympathizers, particularly among young or marginalized individuals [21]. Carefully crafted narratives provide a sense of belonging, purpose, and heroism, which serve as powerful tools in the radicalization process. Groups frequently publish press releases, battlefield photographs, video of combat operations, and eulogies for fallen members to

provoke strong emotional responses and encourage identification with the their cause [18]. While violence remains a recurring theme, significant emphasis is also placed on education, indoctrination, and ideological reinforcement [22].

Beyond propaganda and recruitment, terrorist groups use online platforms for direct operational purposes, including distribution of instructional materials, combat training, coordination of attacks, and sharing of tactical information [18]. Content often includes URLs linking to external sites hosting videos, documents, or other propaganda [20]. Platforms like YouTube have been used to demonstrate bomb-making techniques, weapons maintenance, and battlefield strategies [18].

Disinformation is another key component of extremist online strategies. For example, during the COVID-19 pandemic, terrorist and violent extremist groups exploited social media to disseminate conspiracy theories, attributing the origin of the virus to governments or framing it as a “soldier of Allah” sent to punish non-believers [23].

Fundraising represents another area in which social media proves useful to terrorist organizations, allowing extremist groups to solicit financial support from sympathizers [24].

To avoid detection, terrorists employ a range of sophisticated strategies, such as the use of fake identities and personal data, as well as the choice of content types less likely to be flagged by automated monitoring systems [18]. Images are particularly favored as they are less susceptible to keyword-based monitoring and censorship systems [22].

Linguistic and communicative strategies Understanding how extremist and terrorist groups use social media requires not only an analysis of platform strategies, but also a careful examination of language, rhetorical styles, symbols, and multimedia elements. These linguistic and communicative patterns are crucial both for understanding the groups’ messaging and for identifying relevant content.

Research on these linguistic and communication patterns has been conducted by law enforcement and intelligence organizations, as well as academic researchers. Law enforcement and intelligence teams generally have access to classified information and advanced technological tools, allowing them to combine multidisciplinary expertise with covertly gathered data. It is assumed that a significant portion of this data is restricted from public disclosure. In contrast, academic researchers rely mostly on publicly available datasets, often collected via web scraping from platforms such as X, and frequently employ a mix of qualitative and quantitative methods within broader cultural and sociopolitical frameworks [25]. Their studies often focus on well-known terrorist groups, particularly Islamist extremist groups such as ISIS, for which there is a substantial amount of accessible data.

Regarding the work of law enforcement and intelligence agencies, the Bureau of Justice Assistance, an U.S. Department of Justice, published the glossary *Investigating Terrorism and Criminal Extremism: Terms and Concepts* [26]. Intended as a tool for criminal justice professionals, it provides definitions of terminology, phrases, activities, symbols, organizations, and names relevant to extremist movements. It includes terms specific to certain groups, legal terms reinterpreted by extremists, and descriptions of common activities and tactics. The glossary also documents significant organizations, movements, and publications relevant to understanding terrorism and extremism in the United States.

Although the edition covers the period 2005-2009, it demonstrates the agencies interest in compiling such resources; it is reasonable to assume that more recent, even if not public, versions exist for intelligence purposes.

Regarding the studies conducted by academic researches, a notable example is a 2017 work by the Program on Extremism at George Washington University, one of the leading research centers on all forms of extremism, which examined changes over time among English-language Islamic State (IS) sympathizers on X. Based on 845,646 tweets produced by 1,782 pro-IS accounts between February 15, 2016 and May 1, 2017, the study analyzed username dynamics, URLs, and hashtags, also providing examples of significant tweets to illustrate the group’s language [27]. Another study from the same program focused on White Nationalists, identifying the top 10 hashtags and most recurrent themes [28].

A separate comparative study analyzed a dataset of 105 pro-ISIS X users and 91 randomly selected users, both collected between 2015 and 2016. The pro-ISIS dataset on Kaggle, *How ISIS Uses Twitter*, was originally compiled by Fifth Tribe and contained over 17,000 tweets from 112 accounts that met specific criteria such as extremist keywords, imagery, and network connections, additionally validated by two independent psychologists. The random sample dataset, matched in size to the pro-ISIS one, was generated via the X API. The findings revealed that ISIS supporters employed significantly more third-person plural pronouns, but fewer first-person singular and second-person pronouns, compared to the random group. They also used more vocabulary related to death, certainty, and anger, as well as a higher proportion of words with six or more letters. Furthermore, their tweets contained more negative language and tone. The research also provided a list of the most frequently used words and hashtags [29]. Other research using the same Kaggle dataset has focused on identifying keywords, determining the most frequent words and hashtags, and exploring the main topics discussed.

Another study also examined ISIS related content, listing some of the most frequent hashtags and keywords in each group’s messages. That analysis used two main datasets: a subset of a larger dataset originally collected by a research group at Texas A&M University, the *Alfifi* dataset, during 2014-2015, and user timelines collected in December 2021 via the X API, comprising approximately 10 million tweets from 6,173 accounts [30].

Beyond ISIS-related research, some studies have investigated the online activity of extremist or malign actors unconnected to the group. For example, one analysis examined X activity by bad actors affiliated with Iraqi militias, using three datasets totaling approximately 16,000 tweets [31]. The analysis identified three main themes: public shaming and personal attacks, particularly targeting Iraqi journalists Ahmed Al-Bashir, Stevan Nabil, and Saad Al-Bazaz, who were frequently mentioned in hashtags; legal threats and misinformation accusations, often expressed through hashtags such as #Expel_Al-Shariqiya and #Basrah_expels_Al-Shariqiya; and glorification of Shiite figures and promotion of conspiracy narratives, including references to Abu Mahdi Al-Muhandis, killed by U.S. forces in Iraq in 2020, accompanied by calls for retaliation against U.S. forces and opposition to U.S. interests. Another study focused on two Turkish groups, the FETO organization and the PKK [32]. It examined their network structures, forms of hashtag activism, key actors, and activism processes and listing the most frequently used words and hashtags, along with their occurrence frequencies. A more recent work analyzed the

first 2,848 tweets posted in English by Hamas between March 2015 and March 2018, using thematic content analysis to examine the socio-political and religious narratives at the core of Hamas’s online public diplomacy during its first three years on X [33].

Even though much of the research focuses on English-language posts, it is important to note that there are also studies dedicated to content in other languages. For example, regarding Arabic language content, one of the earliest examples is a 2017 study based on a dataset of over 1.9 million messages posted on X by more than 25,000 ISIS members [34]. This study highlighted significant patterns in ISIS propaganda efforts and identified the most frequently used words in tweets, primarily related to violence, theology, and sectarian themes. Another study examined jihadist propaganda in Arabic, mapping recurrent communication patterns and frequently used terms, mostly boasting of ISIS’s alleged victories or threatening enemies, with a clear Shiite-Sunni sectarian dimension. The analysis showed signs of coordination and the frequent usage of keywords such as Laden, Al-Zawahiri, Al-Awlaki, Al-Zarqawi, often used alongside hashtags such as #Al-Maqdissi, and #Abu-Qumama [35].

Regarding the use of emojis by criminal and violent extremists, an example of research is the Joint Intelligence Bulletin (JIB) [36], made by the Central Florida Intelligence Exchange (CFIX), San Diego Law Enforcement Coordination Center (SD-LECC), and the Utah Statewide Information and Analysis Center (SIAC). The bulletin outlines the emojis most commonly used in contexts of human trafficking, gang activity, drug sales, and violent extremism. In these settings, emojis are used to signal support for illicit activities, promote group membership or association in online communities, and issue calls for violence. For example, ISIS has used emojis without accompanying text to share links Telegram channels in order to avoid suspension. One such channel, promoting ISIS activity in Libya, used only the scissors emoji; another used the truck emoji to symbolize vehicle ramming attacks, acting as a hyperlink to the respective channel. Violent extremists have also used emojis to identify attack methods, such as the truck, knife, or bomb emoji to represent respectively vehicle ramming, edged-weapon attacks, improvised explosive devices (IEDs). Potential targets are denoted by emojis symbolizing churches, mosques, or synagogues. Foreign Terrorist Organizations (FTOs) have used the index finger pointing up emoji to digitally represent the finger of Tawheed, while the syringe or pill emoji denotes chemical or biological attacks. White Supremacist Extremists (WSEs) often use the fire emoji or the “OK” hand sign to symbolize white power, whereas anarchist extremists frequently employ the encircled “A”, raised fist, and black flag emojis. Another example comes from the Institute for Strategic Dialogue (ISD), which reports that pro-Islamic State alternative news outlets used a code of 43 emoji to describe different types of attacks, distinguish between official and unofficial messaging, and identify the group’s enemies. This coded communication enables messages to be shared with followers in ways that are more difficult for outsiders or automated systems to detect, thereby enhancing resilience and evasion on social media platforms [37].

Overall, linguistic and communicative analyses, ranging from lexical choices and hashtags to coded emoji usage, provide critical insights into extremist behavior on social media. They also highlight why terrorism represents a particularly well-studied category, with abundant, structured data available to both academic and intelligence communities.

Cybercrime and Hacking

Cybercrime refers to criminal activities carried out using computers and the Internet, while hacking is the act of exploiting vulnerabilities or weaknesses in computer systems to gain unauthorized access to data, networks, or devices. Hackers' motivations vary widely, from curiosity and experimentation to malicious purposes.

The psychological profiling of cybercriminals and hackers is a relatively new discipline compared to the well-established study of terrorists. Nonetheless, research is growing, employing multidisciplinary approaches that integrate criminology, law, and psychology to account for personal traits, criminal expertise, social factors, and motivational drivers. Cybercriminals and hackers often display distinctive psychological characteristics, such as fearlessness, lack of empathy, and boldness, that shape their actions. Their motivations can often be classified using the MICE framework: Money (e.g., data theft, cyber fraud), Ideology (politically or socially driven attacks), Compromise (coercion or blackmail), and Ego (thrill-seeking, reputation building, testing skills against individuals or organizations). Desires for power, control, and revenge further shape cybercriminal behavior. Some individuals completely disregard legal and ethical norms, even risking their reputation within the cyber community. Collectively, these elements form a complex psychological profile underlying cybercrime [38, 39].

The use of social media by cybercriminals and hackers, as for terrorists, has been studied by academics and international institutions. However, in the context of this research, it has been more difficult to find detailed and comprehensive studies focusing specifically on behaviors and content patterns of cybercriminal and hacker on social media. This may be due to the decentralized, loosely organized, and fluid nature of cybercriminal groups, which often operate individually or in small, transient teams. Unlike terrorist organizations, which tend to have more structured and public propaganda efforts, cybercriminal activities are often less visible and more fragmented, focusing on encrypted communication channels and private platforms, making their activities harder to monitor. Additionally, their tactics, tools, and platforms evolve rapidly to avoid detection, further complicating research efforts. Moreover, cybercriminals have diverse motivations, resulting in varied and less predictable behaviors compared to terrorist groups. Finally, terrorism has historically received greater political and institutional attention, leading to more resources and studies dedicated to that field, while cybercrime research remains more technical, fragmented, and less accessible, partly because cybercrime and hacking are relatively more recent phenomena. It is assumed, however, that intelligence teams and law enforcement agencies possess more extensive and detailed information not publicly available.

Social media has become a central tool for cybercriminals and hackers, used for social engineering attacks, malware distributions, online scams, public extortion, credential and vulnerabilities sharing, illicit service promotion, hacktivism, recruitment, networking, and disinformation campaigns.

As with terrorist actors, social media offers cybercriminals and hackers unique advantages for a broad spectrum of illicit activities. Its vast reach, low entry barriers, and built-in trust between users enable not only large-scale malware dissemination but also the rapid sharing of stolen credentials, vulnerabilities, and illicit tools. The informal and

interactive nature of social media platforms facilitates social engineering, online scams, and public extortion, while the viral dynamics amplify disinformation campaigns and hacktivist actions. These platforms serve also as low-cost marketplaces and recruitment hubs, connecting offenders with accomplices such as money mules or technical collaborators, and as a public showcase directing potential customers to more concealed Dark Web services [40–42]

Social engineering is a core activity, leveraging platform trust dynamics to manipulate users into disclosing sensitive data or performing risky actions. Techniques include phishing links masked as promotional offers [43], NFT airdrop scams [42], and recruitment schemes targeting vulnerable demographics such as youths seeking quick income [44]. A high-profile example of attack occurred in July 2020, when X accounts of over a hundred users were hijacked to tweet Bitcoin-doubling requests [45].

Social media platforms are also widely exploited as a vectors for malware distribution, often through malicious attachments, infected links, or compromised applications via posts and comments. Empirical research shows that social media contains up to 20% more potential malware delivery methods than other online sources [40]. The trusted nature of connections increases the probability to click on malicious links, facilitating large-scale infections, amplified by chain exploitation mechanism. Financial fraud alone accounts for significant revenue, with estimates exceeding \$290 million annually across social media platforms [40]. More technically sophisticated actors also manipulate social media to propagate remote access trojans (RATs) and similar payloads. For example, on a Russian-language cybercriminal forum, a method was shared whereby an attacker downloads a popular YouTube video, embeds a RAT, re-uploads it, and then places a download link in the description to spread the malware at scale [46].

Organized crime groups take advantage of the trust people place in well-known platforms to attract money mules with job-like ads promising easy money, sometimes even showing fake proof of past payments. Once someone responds, middlemen may guide them through the process, collecting personal information and bringing them into money-laundering schemes [44].

Cybercriminals also use social media to teach and advertise hacking skills. Public channels and videos disseminate tactics, techniques, and procedures (TTPs). Channels on platforms like VKontakte and Odnoklassniki disseminate compromised financial information, tutorials, and commercial malware offerings [46]. Large-scale analyses of cybercriminal activity channels reveal widespread sharing of compromised credentials, exploit kits, and black-hat hacking resources, with significant portions of distributed files containing additional malware or phishing payloads [40, 46, 47].

While most cybercriminal groups operate covertly, some hacktivist collectives, such as Anonymous, maintain a strong public presence to coordinate actions, promote socio-political causes, and conduct operations like doxing or anonymous blogging [48, 49]. Beyond hacktivism, some actors use social media to claim responsibility for cyberattacks, amplifying their reputational standing or ideological message [40]. Social media platforms also facilitate the formation of temporary groups among individuals with shared interests. These groups can coordinate various malicious operations such as distributed denial-of-service (DDoS) campaigns, recruitment of technical collaborators, and networking with

criminal accomplices [42, 44].

Disinformation campaigns conducted by hackers and state-sponsored actors exploit social media’s amplification mechanisms to manipulate public opinion, destabilize institutions, and influence geopolitics. Tactics include bot networks, fake news outlets, and staged leaks, often combined with typosquatting, geofenced advertising, and multi-stage redirection to evade detection [41, 42]. Such operations increasingly blur the lines between cybercrime, hacktivism, and information warfare.

Linguistic and communicative strategies As in terrorism studies, understanding the language of hackers and cybercriminals on social media, including keywords, topics, codewords, and communication strategies, can be crucial to interpret content or attribute posts to specific actors, as occurred when linguistic traces in Chinese were linked to the WannaCry ransomware. However, available research reveals a scarcity of detailed analyses in this domain. This scarcity can be attributed to several factors: cybercriminal activities, as previously described, are often less publicly visible, more fragmented, and rely on encrypted communication channels and private platforms, making them harder to monitor; publishing specific codewords or emoji meanings risks alerting the actors, who could quickly change their lexicon; many insights available to intelligence teams are maybe derived from classified sources, including dark web infiltration and private communications, that cannot be disclosed. In contrast, linguistic profiling of terrorist groups is more widely available, because many groups have been dismantled and their propaganda was public-facing, allowing researchers to study it without compromising operations.

A major component of cybercriminal activity is social engineering, which aim to manipulate users into performing risky actions, such as downloading files or disclosing sensitive information. Linguistically, most of these attacks align with Cialdini’s persuasion principles [50]: reciprocity (“gift for your feedback”), scarcity (“offer ends in 2 hours”), authority (“official notice”), commitment and consistency (“just complete this quick step”), liking and consensus (“thousands already joined”). Common markers include urgency (“urgent”, “immediately”, “24 hours only”), emotionally adjectives, excessive punctuation (“!!!”), irregular capitalization, grammatical errors, frequent use of modal verbs, and formulaic openings and closings [51]. Keyword analyses of phishing messages have identified recurring high-risk terms, such as money, investment, free, invoice, verification, and request [52, 53].

These generic strategies translate effectively to social media, where speed of interaction, perceived authenticity from personal connections, and platform-specific language styles, including emojis, hashtags, and abbreviations, enhance their impact. Although research on social engineering is extensive, studies focusing specifically on linguistic manipulation in social media attacks remain limited. A study on Facebook scams show that curiosity, urgency, and personal tone are common lures (“Most people can’t watch this video for more than 25 seconds. It’s unbelievable!”, “I’m deeply shocked. And, what’s more, ashamed! Check out these recently leaked government documents.”, “Facebook started closing fake and duplicate accounts. Update your account soon. This is serious, I lost my old account :(”, “Check this out. I watched it at least 20 times.”) [54]. Other

research highlights message features that influence user susceptibility, such as spelling and grammar quality, message length, inclusion of images or videos, number of likes, and overall organization [55]. Analyses of spam vocabulary on X and Facebook reveal frequent use of enticing or financial terms. On X, terms like amazing, hear, watch, hunt, and win are common, while on Facebook, words such as money, marketing, and free frequently appear [56].

When promoting illegal services, cybercriminals often mimic legitimate advertising, using organized lists, prices, features, guarantees, and technical jargon to signal credibility. Evasion techniques include coded terminology, such as words derived from colors or shapes of illicit items (e.g., “four-piece set” for bank card, phone, USB, and ID copy), multilingual characters, masking illicit content outside the main text, such as in usernames, media, hashtags, or poll options, and hiding offers in popular hashtags. Linguistic obfuscation is common, employing codewords, emojis, abbreviations, or unconventional formatting to hide contact information. For further communication, cybercriminals prefer end-to-end encrypted messaging platforms, whose contacts informations are embedded as URL or IDs account into text. Such patterns have been documented in detail in [40, 57]

In hacktivism, understanding the general jargon used by hackers and cybercriminals, as well as the most prevalent topics, the specific terminology employed by each group, and the meaning of key words and emojis, is crucial for accurately interpreting social media posts. Resources like DarkOwl’s Glossary of Darknet Terms, which compiles slang and acronyms from underground forums, instant messaging services, and cybersecurity research, provide valuable support for these analyses. In fact, it is plausible that darknet terms also appear in social media posts, as highlighted by DarkOwl’s CEO [58]. However, it is not sufficient to know only general terms: linguistic analysis tailored to individual groups is essential to distinguish between different hacktivist collectives. Recent research analyzing around 20,000 X and Telegram messages from 35 hacktivist accounts revealed that language analysis can identify group clusters, shared tactics, and potential links to state-sponsored actors. Topics included cyberattacks against countries such as Israel, Ukraine, Russia, Iran, India, Spain, and the USA, operations targeting militant organizations, and leaks of sensitive information tied to major geopolitical events. Analysis showed that groups like JustEvil and NoName057 share writing styles with Cyber Army of Russia Reborn, suggesting overlapping operators and potential state sponsorship. Shifts in writing style, as observed in the IT Army of Ukraine during 2022, indicate that accounts can be sold, repurposed, or operated by new actors [59].

Several studies focus on Anonymous. For example, a research examined the latest 1,500 tweets from the six most influential Anonymous accounts, reporting most present topics and keywords. Most common topics are politics, particularly US and UK issues, including Brexit, Donald Trump, and net neutrality. Other common themes include leak culture, Wikileaks, major data breaches, activism, protests, and movements like Occupy. Some accounts referenced specific operations, such as “Op Icarus”, and figures like Edward Snowden [48]. A separate study highlighted the group’s return to social media following George Floyd’s murder, again showing recurring topics and keywords [49]. Regarding the use of emoji by Anonymous members, a recent study found that Anonymous users generally use emojis in a similar way to regular X users, primarily to

convey emotions. While some emojis have developed group-specific associations, such as expressing admiration or infatuation for notable members, there is no evidence that emojis are being used as a coded language or to send secret messages [60].

Regarding the usage of hashtags, although there are no studies specifically analyzing which hashtags are most used by hackers or cybercriminals, it has been observed that one way hackers attempt to exploit social media is by adding hashtags to make their content more viral.

Overall, linguistic and communicative analysis remains an essential but underexplored tool for understanding cybercriminal and hacktivist behavior on social media.

Depression and Suicide

Nowadays, many individuals experiencing depression or suicidal ideation share their feelings and mental state on social media as a form of emotional relief and self-expression. These users often post content that reflects sadness, emptiness, or frustration, ranging from melancholic citations, personal reflections, images or status updates that reveal their inner struggles, to suicide notes. Some users may withdraw from interactions, limiting comments or messages, while others search support by openly expressing their emotions. Detecting depressive and suicide-related content online is crucial for early intervention, potentially saving lives, providing timely medical support, protecting vulnerable populations, and maintaining healthy online communities. Social media platforms have recognized this need and implemented mechanisms to report accounts potentially engaging in self-harm or suicidal behavior. Similarly, mental health professionals have identified suicidal ideation through posts, including both photos and explicit statements, enabling timely support for clients with serious mental illness [61].

Linguistic studies, both older and more recent, and whether based on social media or other generic written materials, have consistently revealed recurring patterns in the language of people experiencing depression or suicidal thoughts. Such individuals tend to use more first-person pronouns and fewer third-person pronouns, indicating heightened self-focus and social withdrawal [62–65]. Their writing often expresses intense negative emotions, such as sadness, hopelessness, and anger, and may include references to physical symptoms like insomnia, fatigue, and appetite changes [62, 66, 67]. In online contexts, activity patterns frequently shift toward late night posting and reduced social engagement, with fewer posts, replies, and interactions [68]. Many also disclose personal circumstances behind their distress, including relationship breakdowns, family conflict, work stress, and even specific details about therapy or medication [62, 69]. Suicidal ideation can appear explicitly (“I want to die”, “kill myself”) or in more veiled forms (“better off without me”, “go to sleep forever”).

While there are clear links between writing style and the presence of depression or suicide ideation, regardless of age or gender, there is still little research on how language patterns differ among specific subtypes, such as postpartum depression, seasonal affective disorder, or severe clinical depression. Some studies have explored postpartum depression [70, 71], age or race related differences in depression [72, 73], or suicidal ideation in specific life contexts, but such works are limited and often based on small datasets. This

area remains emerging, with significant potential for future work that combines linguistic analysis with behavioral and contextual data to improve detection and intervention strategies.

Hate and Cyberbullying

Cyberbullying and online hate speech are pervasive issues on social media platforms, where anonymity, rapid content dissemination, and algorithmic amplification can amplify the harm caused to individuals and communities. Cyberbullying involves repeated hostile, intimidatory, or aggressive behavior with the purpose to harm or humiliate the victim, often exploiting digital features such as comment threads, private messages, and public tagging posts. Online hate speech involve expressions that incite violence, discrimination, or hostility against individuals or groups based on characteristics such as race, religion, gender, sexual orientation, or disability. Detecting these behaviors online is crucial not only for preventing psychological harm but also for countering the escalation into offline violence [74].

Research has extensively examined hate speech and cyberbullying from a linguistic perspective, highlighting distinctions between subtypes. A key distinction emerges between directed and generalized hate speech, focusing on different types of prejudice, from sexism and racism, to nationalism and ageism [75]. Directed hate speech is more personal, informal, and aggressive, often involving name calling and greater use of language signaling authority and influence. Generalized hate speech tends to focus on broader targets, often employing lethal terms such as “murder”, “exterminate”, and “kill”, along with quantity markers like “million” and “many”. Examples include frequently used terms targeting disability (“retards”, #Retard) and religious slurs (“Jihadis”, “Zionazi”, “Muzzie”). Hashtag analysis illustrates these tendencies: for instance, #whitepower (ethnicity-based hate) is associated with terms like #whitepride, #whitegenocide, “nazi”, and #kkk, whereas #nomuslimrefugees (religion-based hate) occurs with #stopislam, #muslimban, and #americafirst [76].

Other studies have examined hate speech against specific categories. For migrants, discourse often constructs migrants as a threat and proposes violent or eliminatory “solutions” [77]. For LGBTQ communities, analyses identify recurring online hate topics [78]. Misogynistic attacks against candidate are frequently characterized by abusive hashtags, memes, caption and sounds, such as hashtag composed by the name of the candidate and abusive words (#marshablackburnisatraitor, #laurenboebertdiarrhea) [79].

Moreover, the use of veiled language, sarcasm, neologisms, and visual codes makes the identification and understanding of hostile messages more complex, as these elements require knowledge of cultural context and the internal dynamics of online communities to be correctly interpreted. To address this, several specialized glossaries have been developed to document and explain coded language, slang, and symbols used in hate speech and cyberbullying. Examples include contemporary online slang such as “Mog”, to surpass someone in attractiveness or status, common in incel forums, and “Bash Board”, an online forum dedicated to targeted ridicule [80]. The Anti-Defamation League (ADL) maintains databases of symbols, codes, and phrases frequently used by white supremacist and other hate groups [81]. Similarly, Hatebase, a now retired joint project of the Sentinel

Project for Genocide Prevention and the Dark Data Project, served for nearly a decade as a multilingual, usage-based repository of slurs, variants, and contextual notes [82].

Visual symbols and emojis play a significant role in coded hostility. A recent study [83] showed how seemingly innocuous emojis can be weaponized: the milk emoji, especially with #rawmilk, can imply white supremacist ideas linked to “Nordic ancestry”; the kiwi emoji has been used as a transphobic symbol; the airplane emoji can signal calls for deportation; and the lizard emoji is associated with antisemitic conspiracy theories about global control. Emojis are also exploited in cyberbullying as covert harassment tactics, often unnoticed by adults, highlighting the importance of emoji literacy in online safety [84].

2.3 AI and Large Language Models

Artificial Intelligence (AI) refers to the simulation of human intelligence by machines, typically implemented through computer programs designed to enable them to “think” and “learn from experience”, thereby performing tasks that once required human intervention [85].

To achieve this, AI systems are trained on large volumes of data, which allow the underlying models to perform tasks commonly associated with human intelligence, such as understanding natural language, recognizing patterns, making decisions, and adapting based on experience [86–88].

2.3.1 Machine learning (ML)

Machine learning (ML), a subfield of AI, focuses on the development of algorithms that can autonomously solve tasks by learning from data, without being explicitly programmed [88,89]. In essence, ML involves building models that learn and improve their performance over time with minimal human intervention.

The core objective of ML is to enable algorithms to identify patterns and relationships within data so that they can make predictions or perform tasks based on historical information [90]. In this sense, ML algorithms are at the heart of many AI-powered systems, enabling them to appear “intelligent” in their operations.

Several learning paradigms exist within ML, depending on the nature of the data and the task at hand. The three main categories are: Supervised, Unsupervised and Reinforcement Learning. Supervised Learning is the most commonly used approach, particularly in commercial applications. Here, algorithms are trained on labeled datasets, where the correct output is already known, to learn the mapping from inputs to outputs. Once trained, the model can classify new, unseen inputs or predict future outcomes [88,91]. Moreover, in the Unsupervised Learning approach, the algorithm is provided with unlabeled data and must discover hidden structures, patterns, trends, or groupings. This is useful for tasks like clustering, dimensionality reduction, or association [88,92]. The third approach, Reinforcement Learning, involves an agent that learns optimal actions through interactions with an environment. By receiving feedback in the form of rewards or penalties, the agent incrementally improves its decision-making policy to maximize cumulative reward over time [88,93,94]

Despite their methodological differences, ML algorithms share a common goal: to process data in ways that enable classification, regression, or clustering, often by determining effective decision boundaries [95].

2.3.2 Deep learning (DL)

Deep learning (DL) is a more advanced subset of ML, characterized by algorithms that rely on artificial neural networks with multiple layers of interconnected nodes. The term deep refers to the presence of these multiple hidden layers, which allow the system to model complex data representations and automatically detect correlations and patterns in large datasets [88, 91, 96].

What distinguishes DL from traditional ML approaches are several key factors. While ML methods are often effective with smaller, structured datasets and may rely on manual feature engineering, DL methods typically require large amounts of data and significant computational resources, such as GPUs, but have the advantage of automatically learning relevant features directly from raw data. Moreover, while many ML algorithms are competitive on structured data using models such as decision trees, random forests, or support vector machines, DL excels at handling unstructured and high-dimensional data, such as images, audio, and natural language. As a result, DL frequently outperforms traditional ML in complex tasks like image recognition, natural language understanding, and speech recognition [97, 98].

At the core of DL lie Artificial Neural Networks (ANNs), computational models inspired by biological neurons [88, 99]. A standard deep neural network is composed of an input layer, multiple hidden layers, and an output layer. Each hidden layer receives input from the layer before it, processes the data through certain calculations, and then sends the result to the following layer. This next layer performs similar computations and passes its output onward. This chain of processing continues until the final output layer produces the ultimate result or prediction. These hidden layers enable the network to learn hierarchical representations, from low-level features, such as edges in images, to high-level concepts, like faces or objects, through a nested architecture. Thanks to these properties, DL excels in processing various forms of high-dimensional data: from one-dimensional data, like signals and texts, to multidimensional data such as images, video, or audio [88, 97, 100].

2.3.3 Explainable AI (XAI)

The ability of ML and DL algorithms to solve complex tasks has been a crucial factor in the advancement of AI. However, the lack of transparency and accountability in these so-called “black-box” models introduces significant risks. A major concern is the potential consequence of delegating high-stakes decisions to systems whose internal reasoning cannot be understood or audited. This opacity, often accompanied by biases embedded in the training data, is one of the main barriers to the widespread adoption of AI technologies.

The essential need to interpret such decisions and justify algorithmic decisions has given rise to a new area of research known as **Explainable AI (XAI)**. The goal of XAI

is to develop a set of tools and models capable of providing clear and understandable explanations for AI-driven decisions. This helps users not only to comprehend and trust AI systems but also to identify potential biases, validate decisions, and ensure compliance with ethical and legal requirements [101].

2.3.4 Generative Artificial Intelligence (Generative AI)

Generative Artificial Intelligence (Generative AI) is a subfield of AI that focuses on the autonomous creation of new content, such as text, images, audio, video, or code, based on patterns learned from training data. Unlike traditional ML or DL models that are primarily designed for classification or prediction tasks, Generative AI aims to produce original data instances that resemble the distribution of the training dataset [102, 103].

This technology is grounded in advances in deep generative models (DGMs), a class of DL models that are trained to understand and replicate complex, high-dimensional probability distributions [88, 104, 105]. By learning these distributions, DGMs can generate new samples that are statistically similar to training data [88, 106]. Unlike discriminative models, which focus on modeling the relationship between input features and output labels, generative models attempt to capture the underlying structure of the data itself [107]. This makes DGMs fundamentally different in their purpose and behavior, as their primary goal is the probabilistic generation of new data, rather than decision-making on existing data, such as classification, regression or clustering [88, 105, 108].

While early DGMs such as Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs) were foundational, modern Generative AI increasingly relies on large-scale training using self-supervised learning. This is a hybrid approach that starts with a small set of labeled data (supervised learning) and continues with a large volume of unlabeled data (unsupervised learning). This allows to train very large datasets without the need for difficult complete labeling [88, 109].

A key milestone in the development of Generative AI was the introduction of the Transformer architecture, proposed in the influential paper *Attention Is All You Need* by researchers from Google Brain and the University of Toronto in 2017 [110]. Originally designed to enable efficient parallel computation on GPUs, the Transformer architecture marked a significant shift in the design of neural networks for sequence modeling. At the core of the Transformer lies the self-attention mechanism, which enables the model to capture long-range contextual dependencies more effectively than previous approaches based on recurrence or convolution [111]. The Transformer language model architecture is composed of an encoder and a decoder, each built from stacks of identical layers. Each encoder layer includes a multi-head self-attention sub-layer followed by a position-wise feed-forward network. The decoder layers extend this structure by adding a third sub-layer that attends to the encoder's output. Positional encodings are also incorporated to provide information about the order of tokens, compensating for the lack of recurrence.

In Generative AI systems, the application interface serves as the main point of interaction between the user and the underlying model. A central interaction technique in this context is prompting, which is a distinctive feature of Generative AI applications. Prompting allows end users to communicate with the model, often through natural language, in order to instruct it to generate specific types of content, such as text, images,

or other data forms. The form of the prompt can vary depending on the application. For example, text-to-image systems interpret textual descriptions to produce visual outputs, whereas image-to-image models rely on an input image to guide the generative process. The concept of prompting first emerged in the context of Large Language Models (LLMs) and has since become a general mechanism for interacting with a wide range of generative systems [88, 112, 113].

By design, outputs from Generative AI models are probabilistic. This means that for the same input prompt, a Generative AI application may produce different outputs each time it is prompted, while still generating valid and relevant results. Additionally, different prompts can lead to the same desired outcome. This variability has given rise to the field of prompt engineering, which focuses on designing and refining prompts to improve the quality, consistency, and controllability of generated outputs [88, 112].

2.3.5 Natural Language Processing (NLP)

Before introducing the concept of Large Language Models (LLMs), it is essential to understand the role of **Natural Language Processing (NLP)**, a subfield of AI that enables machines to interpret, process, and generate human language. NLP allows computers to convert raw textual or spoken input into machine-readable formats and then generate appropriate responses in natural language. This makes NLP foundational for tasks such as speech recognition, machine translation, sentiment analysis, and conversational AI [114].

By automating the interpretation of language, NLP has significantly expanded the scope of tasks that were once reliant on human judgment. It enables machines to extract meaning from unstructured data, detect sentiment and emotional tone, and derive insights from large volumes of text with efficiency and consistency [115].

A particularly important aspect of NLP is the ability to measure semantic similarity between texts, which plays a key role in understanding language at scale. This involves converting linguistic input into numerical representations, allowing algorithms to assess how closely two pieces of text relate in meaning. Semantic similarity underpins a variety of practical applications, such as identifying equivalent user queries in online forums, detecting plagiarism, recommending related news articles, and clustering social media posts for content analysis and user profiling [116].

The continuous development of such methods has laid the groundwork for modern Large Language Models (LLMs), which build upon and extend these foundations to achieve unprecedented performance in natural language understanding and generation.

2.3.6 Large Language Models (LLMs)

In the field of Generative AI, the emergence of **Large Language Models (LLMs)** has marked a significant technological advancement.

Language models (LMs) are computational systems designed to understand and generate human language. They predict the likelihood of word sequences and producing coherent text based on a given input [117–120].

Early pretrained neural LMs, which set the stage for modern LLMs, leverage neural networks and are trained on large-scale datasets using GPU-based computation. Among

early Transformer-based pre-trained LMs (PLMs), three main architectural categories can be identified: encoder-only, decoder-only, and encoder-decoder models. Encoder-only PLMs consist solely of an encoder component and were primarily developed for language understanding tasks, such as text classification. Examples include BERT and its variants. Decoder-only PLMs, such as GPT-1 and GPT-2 developed by OpenAI, focus on language generation tasks. These models laid the foundation for more advanced LLMs like GPT-3 and GPT-4. Encoder-decoder PLMs combine both encoder and decoder modules into a single architecture, enabling them to perform both language understanding and generation tasks. Notable examples include T5, mT5, MASS, and BART [111]. Comprehensive overviews of early PLMs can be found in [121, 122].

More recently, advances in transformer-based architectures have significantly extended the capabilities of LMs. LLMs are large-scale, transformer-based PLMs pretrained on massive, web-scale corpora and contain billions of parameters [111]. LLMs distinguish themselves from earlier PLMs by their scale, extensive pretraining, and emergent capabilities [120, 123–125].

Emergent abilities of LLMs include: in-context learning, the ability to learn new tasks from a few examples provided in the prompt during inference; instruction following, the capacity to generalize to novel tasks by simply following natural language instructions, without the need for fine-tuning or explicit examples; multi-step reasoning, solving complex problems by decomposing them into intermediate reasoning steps, as demonstrated through chain-of-thought prompting [111, 126]

Dominant LLM Architectures

The most prevalent architectures for LLMs are encoder-only, decoder-only, and encoder-decoder models, all of which are typically built upon the Transformer architecture.

Encoder-Only Models allow each attention layer to access all tokens within the input sequence simultaneously. These models are usually pre-trained by intentionally corrupting input sentences, such as by masking random words, and training the model to reconstruct the original content. Encoder-only models are particularly effective for tasks that require comprehensive understanding of the input, including sentence classification, named entity recognition, and extractive question answering.

Decoder-Only Models, also referred to as auto-regressive models, restrict each attention layer to access only the preceding tokens in a sequence. Their pretraining generally involves next-token prediction, where the model learns to generate the subsequent word based on previous context. These models are especially well-suited for tasks involving text generation. The GPT family of models is a prominent example of this architecture.

Encoder-Decoder Models, often called sequence-to-sequence models, combine both components: the encoder can access all input tokens, while the decoder can access only earlier tokens in the output during generation. These models are usually pre-trained involving variations of the objectives used in encoder or decoder models, sometimes with added complexity. Encoder-decoder models are particularly suited for tasks where output generation is conditioned on a specific input, such as summarization, translation, and generative question answering [111].

Large Language Model Families Two prominent families of LLMs have emerged in recent years: the GPT family and the LLaMA family.

The GPT Family (Generative Pre-trained Transformers) consists of decoder-only, Transformer-based models developed by OpenAI, including GPT-1, GPT-2, GPT-3, InstructGPT, ChatGPT, GPT-4, CODEX, and WebGPT. Early versions of GPT are open-source, while more recent iterations, such as GPT-3 and GPT-4, are closed-source. A major milestone in the evolution of LLMs was the release of ChatGPT, a conversational agent capable of performing a wide range of tasks, including question answering, information retrieval, summarization, and more [111].

The LLaMA Family refers to foundation models released by Meta under a custom license that allows research use but restricts commercial applications. This openness has led to widespread adoption within the research community, enabling rapid development of improved open-source LLMs. These models are used both to compete with closed-source systems and to build task-specific LLMs for high-stakes or mission-critical applications [111].

This section focuses on two representative families to illustrate different LLM development paradigms. For detailed coverage of additional families see [111].

Data and modeling techniques

Data Cleaning Data quality plays a critical role in the performance of LMs trained on them. Empirical studies have shown that data cleaning techniques, particularly filtering and deduplication, have a significant impact on model performance. Data filtering removes irrelevant, noisy, or misleading content, mitigates outliers, and balances class distributions. Additional preprocessing may involve removing stop words, punctuation, or other non-informative elements, as well as resolving or excluding ambiguous or contradictory inputs that could confuse the model during training. Deduplication eliminates duplicate or repeated entries, preserving dataset diversity essential for robust NLP model training [111].

Tokenization Tokenization refers to the process of breaking down a sequence of text into smaller parts called tokens. While the simplest form of tokenization splits text based on whitespace, more advanced methods rely on predefined vocabularies or word dictionaries. However, these dictionary-based approaches often face the out-of-vocabulary (OOV) problem, where words not included in the vocabulary cannot be processed effectively.

To address this limitation, modern tokenization techniques are typically based on subword units. These subword-based methods enable the tokenizer to represent rare or previously unseen words, and even multilingual content, by combining smaller components to reconstruct complete words [111].

Three commonly used methods are Byte Pair Encoding (BPE), WordPiece, and SentencePiece. BPE, which originated as a data compression method, merges the most frequent pairs of adjacent bytes in a sequence, keeping common words intact while splitting rare words. WordPiece, starts with a vocabulary that includes all individual characters and incrementally constructs subwords based on frequency statistics, aiming to maximize

the likelihood of the corpus. SentencePiece processes text as a continuous character sequence, which makes it particularly effective for languages without clear word boundaries or for data containing irregular spacing, invented terms, or noisy structure [111].

Positional Encoding Positional encoding refers to a class of techniques used in Transformer models to inject information about the order of tokens in a sequence. Techniques include absolute, relative, rotary, and bias-based encodings, improving sequence generalization and capturing token dependencies [111].

Model Pre-training Pre-training allows LLMs to develop fundamental linguistic competencies that are useful for a wide range of language tasks. This stage typically involves training the model on large-scale, usually unlabeled texts using self-supervised learning strategies. Common objectives include next token prediction, also known as autoregressive modeling, and masked language modeling, where parts of the input are masked and the model learns to predict the missing elements. More recently, the Mixture of Experts (MoE) framework also enables high performance with reduced computational cost [111].

Fine-Tuning and Instruction Tuning To adapt a foundation model to specific tasks, it is often necessary to apply fine-tuning using labeled data, an approach referred to as supervised fine-tuning (SFT). This process can be performed on a single task or extended to multiple tasks simultaneously, improving performance and reducing the need for complex prompt engineering. In some cases, fine-tuning also serves as an alternative to retrieval-augmented generation and allows the integration of new or proprietary data not seen during pre-training. A particularly important variant is instruction tuning, which aims to align the model’s behavior with human expectations when following prompts. Instruction tuning datasets, such as Natural Instructions, typically include task descriptions, illustrative examples, and guidelines on what to avoid. Instruction-tuned models consistently outperform their base foundation models across a range of tasks [111].

Alignment AI alignment ensures models act according to human goals and ethical principles. Although LLMs are pre-trained to predict words, they can sometimes produce unintended outputs, including toxic, harmful, biased, or misleading content, so additional alignment measures are often necessary to further reduce undesirable behaviors [111].

Decoding Strategies Decoding is the process by which pre-trained LLMs generate text from tokenized inputs. The model predicts token probabilities via logits and softmax. Several decoding strategies exist, such as greedy search, beam search, top-K, and top-P sampling. Greedy search selects the most probable token at each step, ignoring all others; it is fast but can lead to less coherent sequences because it doesn’t consider the broader context. Beam search improves on greedy search by keeping the top N most likely sequences at each step; this continues until a set sequence length or end-of-sequence token is reached. Top-k sampling randomly selects the next token from the top k most probable ones. The randomness is influenced by a temperature parameter that ranges from 0 to 1, which controls how much weight is given to high-probability tokens: lower temperatures

make the output more deterministic, higher ones encourage diversity. Top-p sampling, or nucleus sampling, selects tokens from the smallest set whose cumulative probability exceeds a threshold p . Unlike top-k, the number of candidate tokens isn't fixed, allowing for more adaptive and creative outputs. This makes it especially useful when the top-k tokens do not capture enough probability mass [111].

LLM limitations

Despite advances in fine-tuning and alignment, LLMs have inherent limitations:

- **Lack of memory or state:** LLMs are stateless by default and do not retain information from previous prompts, limiting applications that require context persistence or continuity.
- **Probabilistic nature:** Outputs vary across runs for the same input. Temperature and other parameters can modulate randomness, but variability is intrinsic.
- **Outdated knowledge and no real-time data access:** LLMs rely on training data and have no awareness of current events or real-time context unless connected to external tools or updated regularly.
- **Computational cost:** Training and deploying large models require powerful hardware, potentially causing high latency or reduced reliability.
- **Susceptibility to hallucination:** LLMs may produce content that sounds plausible but is factually incorrect due to training on mixed-quality data [111].

Hallucinations A major limitation of LLMs is hallucination, the generation of outputs that lack coherence or factual accuracy. This concept, adapted from psychology, is widely studied in natural language generation. According to the paper *Survey of Hallucinations in Natural Language Generation* [127], hallucinations occur when models produce content that is non-sensical or misaligned with the intended source.

Hallucinations generally fall into two main categories: intrinsic hallucinations, in which the outputs contradict the source material, often introducing factual errors or logical flaws; extrinsic hallucinations, in which the output, while not necessarily incorrect, cannot be verified against the source and often include speculation or unverifiable claims.

The term “source” varies depending on the task. In conversational AI, it may refer to general world knowledge, while in summarization tasks, it usually refers to the input document. Notably, the acceptability of hallucinations can also be context-specific, for example, in creative writing hallucinations can be desirable.

Measuring hallucinations often combines statistical, model-driven approaches, and human evaluation. Despite improvements in automated metrics, human judgment remains crucial for assessing factual accuracy and coherence [111]

Prompt Design and Engineering

A prompt in generative AI is the textual input provided by the user to guide the model’s output. It can range from a simple question to a detailed description and may include instructions, data, or examples. Effective prompts exploit a model’s capabilities while considering its limitations and operational context [111].

Prompt engineering, a rapidly evolving field, focuses on designing effective prompts to achieve specific outcomes.

One of the most influential and widely adopted prompt engineering techniques is Chain of Thought (CoT) prompting. This technique addresses the fact that, while LLMs excel at predicting the next token, they are not inherently equipped for explicit reasoning. CoT guides the model through logical steps to improve reasoning on multi-step tasks such as math problems or logic puzzles [111, 126].

Inspired by the human approach to problem-solving, Tree of Thought prompting enables the model to explore multiple reasoning paths before converging on the most plausible solution, enabling better handling of ambiguity and complex decision-making [111, 128].

The Self-Consistency technique involves prompting the model multiple times with the same query and comparing the results to increase confidence in accuracy, useful where factual precision is critical, such as fact-checking [111, 129].

Reflection involves prompting the model to review and critique its own output. After an initial response is generated, the model is asked to reflect on aspects such as factual correctness, logical consistency, and relevance. This self-evaluation can lead to improved and more refined answers through an iterative cycle of generation and revision [111, 130].

Expert Prompting simulates one or multiple expert perspectives to enhance depth and accuracy [111, 131].

Chains refer to sequentially structured prompts or modules, where each component performs a specific function and passes its output to the next, supporting modular and complex workflows [111].

Rails are a mechanism to constrain and guide LLM outputs using predefined rules or templates. These constraints ensure that the generated responses adhere to safety, accuracy, and formatting standards [111].

Automatic Prompt Engineering (APE) aims to automate the creation and refinement of prompts. It leverages the capabilities of LLMs themselves in a self-referential manner, generating, evaluating, and optimizing prompts recursively. This reduces manual intervention while improving the quality and effectiveness of the prompts, increasing the likelihood of generating desirable outputs [111, 132].

Few-shot, Zero-shot, and Continual Learning LLMs can generalize to tasks without explicit training. This is made possible by few-shot and zero-shot learning, two emergent capabilities observed in large-scale transformer models.

In zero-shot learning, the model performs a task based solely on a natural language instruction, without any examples. This allows rapid deployment of new functionalities without requiring fine-tuning or retraining. Few-shot learning involves providing a handful of examples within the prompt, enabling the model to infer patterns and replicate

them in new contexts. These paradigms reduce the need for expensive labeled datasets and make LLMs highly adaptable to a wide variety of domains and tasks. Zero-shot works best when the task is common and the instructions are clear, while few-shot is especially useful for more nuanced tasks where tone, logic, or style matters, reducing ambiguity and improves output alignment [133].

A significant prompting technique is role prompting, in which the purpose is to improve output tone and style by telling the model who it should act as. This technique instructs the model to adopt a specific voice, level of detail, or technical sophistication.

Additionally, for tasks requiring structured data, like JSON or tabular responses, output formatting instructions could be included.

Another relevant paradigm is continual learning, where a model incrementally incorporates new knowledge over time without forgetting previous information. While traditional LLMs are static after training, various techniques, including parameter-efficient fine-tuning, memory-augmented models, and agent-based tool use, aim to simulate continual learning. In practice, continual learning is essential for adapting to evolving data, such as changes in user behavior or online discourse trends, making it particularly useful in applications like social media profiling [134].

Integrating these prompting and learning paradigms enhances model flexibility, reduces the need for retraining, and improves responsiveness to real-world dynamics.

Augmenting LLMs through external knowledge

Once trained, LLMs can be utilized to generate outputs for a wide range of tasks through basic prompting. However, to fully exploit its capabilities or to overcome certain inherent limitations, it often becomes necessary to enhance the model with external mechanisms [111].

Retrieval-Augmented Generation (RAG) One of the primary shortcomings of pre-trained LLMs is their inability to access up-to-date knowledge or specialized, private, or context-specific information. To address this, **Retrieval-Augmented Generation (RAG)** has emerged as a prominent solution, allowing to reduce hallucinations and improving factuality [135].

The RAG framework comprises three essential stages: Retrieval, Augmentation, and Generation. The process begins by transforming the user’s input into a query used to retrieve pertinent data from a designated knowledge base. This retrieved content is then appended to the prompt and passed to the model for final output generation. Specifically, the retriever is responsible for retrieving relevant content from external sources such as document repositories or databases. It typically employs vector-based search, where both queries and documents are embedded in a shared space to identify the most semantically aligned results. The generator produces the final output by incorporating the retrieved information. It uses the augmented input to generate responses that are coherent, accurate, and tailored to the query’s context [111, 136].

Other External Tools Beyond retrieval, LLMs can interface with various external tools, such as APIs, service endpoints, or databases, to extend their capabilities [111].

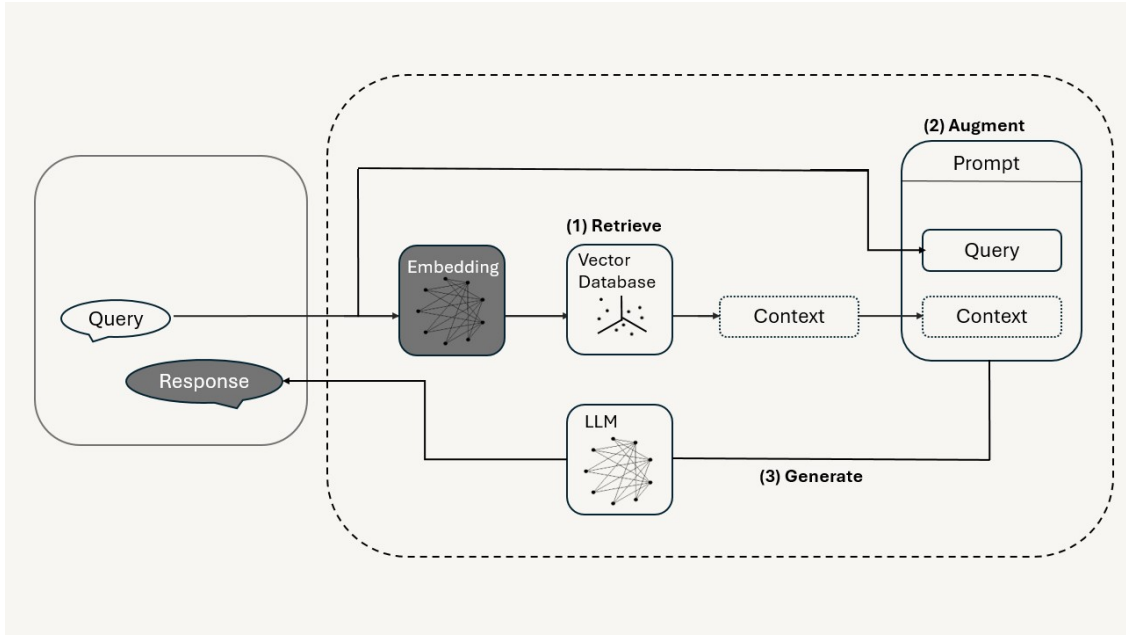


Figure 2.1. Retrieval-Augmented Generation Workflow (Source: [Medium](#))

Knowledge Graphs (KGs) represent a particularly powerful class of structured external tools. They encode real-world entities and their relationships in graph-based form, allowing for efficient and semantically rich access to factual knowledge. When integrated with LLMs, KGs can serve multiple roles: improving factual consistency, enabling precise entity linking, disambiguating references, and supporting reasoning over structured data [137, 138].

Future directions

Recent developments in LLMs suggest several promising directions. While early progress followed the “bigger is better” approach, their high costs, energy demands, and infrastructure requirements have motivated research into smaller, more efficient models (SLMs) that perform well on specific tasks at a fraction of the cost. At the architectural level, Transformers remain dominant, but their attention mechanisms struggle with long contexts, limiting applications such as RAG. Emerging post-attention paradigms aim to overcome these limitations and handle longer inputs more efficiently. Even within attention-based models, techniques like Mixture of Experts (MoE) allow training very large models while activating only relevant components at inference, balancing scale and efficiency. The evolution toward multi-modal models, capable of processing text, images, audio, and video, promises broader applications. Better utilization and augmentation of LLMs is also a key focus. Techniques such as advanced prompt engineering, tool integration, and context-aware analysis can reduce issues like hallucinations, enabling more reliable

and personalized applications, from recommendations to customer service. Finally, as LLMs see wider adoption, security and ethical considerations become paramount. Ensuring robustness against attacks, preventing misuse, and addressing bias are essential for responsible deployment in sensitive real-world contexts [111].

2.3.7 AI Agents

AI Agent An AI agent is generally defined as an autonomous system capable of perceiving its environment through sensors, evaluating its current state, making decision and taking appropriate actions from a set of available options. AI Agents often incorporate multiple forms of AI, such as machine learning, computer vision, and natural language processing, to operate effectively.

Key characteristics of AI agents include autonomy, which allows them to act independently toward predefined goals without constant supervision; perception and interaction, which enable them to gather information from their environment using sensors, cameras, or digital inputs; learning, which allows them to improve their behavior over time through methods like reinforcement learning or supervised training; and integration with external systems such as APIs, databases, or third-party tools, which expands their capabilities and allows them to deliver better results.

LLM agents LLM agents extend traditional AI agents by integrating a large language model with retrieval, reasoning, memory, and tool use, enabling them to autonomously complete complex, multi-step tasks.

One of their key abilities is retrieval, which allows them to access up-to-date information in real time, often via RAG. Equally important is reasoning. Through advanced prompting techniques, it is possible to guide agents in how to make decisions. Memory plays a crucial role as well. Agents can retain short-term memory (such as recent conversation history), long term-memory (storage of structured information), and recall task-specific details, allowing them to adapt to users, maintain continuity, and personalize their responses. The fourth element, tool use, allow the LLM agents to interact with external systems, databases, or APIs.

By combining these pillars, LLM agents move beyond simple Q&A systems to autonomous assistants, able to interact with diverse systems and manage sophisticated processes from start to finish. Their performance is primarily determined by the quality and scale of the underlying language model.

LLM agents are widely used in AI chatbots, virtual assistants, content generation software, and other applied tools. Most modern AI agents are LLM-based, leveraging the reasoning and language capabilities of large language models for diverse applications.

2.4 AI and OSINT

AI and OSINT: a bidirectional relationship As outlined in Section 2.1.2, OSINT operations face significant challenges due to the large volume, heterogeneity, and variable reliability of open source data.

To overcome these limitations and improve the overall intelligence process, the integration of AI has emerged as a key critical strategy. Rather than functioning solely as a remedial tool, AI actively enhances key stages of the OSINT cycle, from data collection to analysis and dissemination.

By automating essential tasks, AI reduces the burden of managing massive and diverse datasets, accelerates investigative procedures, increases scalability, reduces analysts' workload, and improves the accuracy and efficiency of intelligence workflows. Consequently, AI facilitates the extraction of actionable insights from raw data that would otherwise be overwhelming or inaccessible [139].

At the same time, the relationship between AI and OSINT is bidirectional. The vast amount of open source data available online can be leveraged to train models, improving their accuracy and adaptability across a wide range of security and intelligence applications. In this reciprocal dynamic, OSINT not only benefits from AI, but also contributes to its development by supplying real-world, high-volume datasets [139].

Challenges The integration of AI systems into OSINT operations presents significant methodological challenges, extra to the challenges purely related to traditional OSINT. As emphasized by the RAND Corporation, the AI effectiveness within the intelligence cycle cannot be assessed solely through conventional performance metrics such as algorithmic accuracy. Instead, the assessment must focus on the AI system's actual contribution to the overall intelligence process [3]. This distinction is fundamental: even a AI system with 99% technical accuracy may still be inadequate if its outcomes are inadequate or, more critically, lead to significant or catastrophic errors. This perspective emphasizes the importance of evaluating AI tools not only in terms of technical performance but also in terms of operational relevance and impact within intelligence workflows [3].

AI-specific challenges include bias in algorithms and training data, which can skew analyses or inaccurate predictions, potentially compromising the reliability of the intelligence gathered [140]. Moreover, while the vast quantity of online data is a well-known challenge in OSINT, AI-driven systems may inadvertently process misleading or false information at scale. This amplifies the need for human validation and sophisticated verification methods to ensure the accuracy and trustworthiness of extracted intelligence [141]. Developing algorithms that can effectively keep pace with the dynamic nature of online information presents a significant challenge in AI-powered OSINT [142].

Another critical consideration is the risk of overreliance on AI technology. While automation can enhance efficiency, human expertise remains essential for interpreting nuanced information, understanding cultural contexts, and making informed decisions [143]. Striking the right balance between automated processes and human expertise is a persistent challenge. Finally, the susceptibility of AI systems to adversarial attacks presents a unique operational risk. Malicious actors may attempt to manipulate algorithms by feeding deceptive data or exploiting vulnerabilities in automated collection processes. Developing robust defenses against such threats is an ongoing concern in securing OSINT systems, powered by AI [144].

Application Sectors Current research on the integration of AI into OSINT focuses primarily on domains such as security operations, law enforcement, intelligence services, and cyber threat intelligence. However, the application of such techniques is gradually expanding into a broader range of areas, including governmental and non-governmental organizations, public health monitoring, emergency response, defense, penetration testing, and cyber incident response [139].

Despite its strategic relevance, research in the defense domain remains limited, largely due to the classified and sensitive nature of the data involved. Consequently, much of this work may not be publicly available or published through conventional academic channels. Moreover, several key areas, such as penetration testing and cyber-incident response, remain underexplored in current research [139].

Data sources OSINT applications leverage a wide variety of data sources, including social media platforms, which represent the largest direct source of data, as well as publicly available OSINT datasets, online news outlets, websites, blogs, forums, Domain Name System (DNS) records, the Dark Web, and online marketplaces. Unsurprisingly, most research efforts tend to focus on sources that are more readily accessible and structured, often prioritizing ease of use over diversity of data [139].

Although textual and image-based OSINT dominate in current practice, recent research has explored alternative modalities, such as audio data. Despite being relatively underutilized, audio sources offer significant potential in specific operational contexts, particularly within military intelligence. These efforts typically operate within the processing and analysis phase of the intelligence cycle, transforming raw audio inputs into structured, machine-readable information [145].

One notable example involves extracting actionable intelligence from military audio data. The study, focused on Pakistan’s defense sector, proposes a novel AI-method to identify key entities, such as location, rank, operation, date, and weapon, from transcribed military conversations. The authors created a custom dataset comprising 2,000 annotated transcriptions and evaluated four NLP models for their entity recognition performance: DistilBERT (via Hugging Face Transformers), spaCy, NLTK, and Stanford CoreNLP. The results demonstrate the effectiveness of AI-based methods in processing audio-based OSINT and highlight their growing relevance in modern military intelligence workflows [145].

Limitations and Research Gaps Despite the growing body of research on AI-powered OSINT, existing studies exhibit several limitations continue to hinder its scalability and generalizability.

One of the most prominent issues is the monolingual nature of existing systems, which limits their applicability in diverse operational environments. Incorporating multilingual capabilities represents a crucial direction for future research, enabling tools to operate effectively on a global scale. This is particularly relevant in domains where threat actors operate across multiple regions and languages, such as cyber threat intelligence, human trafficking, disaster monitoring, and sentiment analysis [139].

Several studies have already identified this limitation and proposed multilingual enhancements in their future research plans, particularly in the field of cyber threat intelligence [146], human trafficking [147], natural disaster monitoring [148], sentiment analysis [149]; few recent studies, such as [150], have already implemented multilingual capability.

Another recurring limitation lies in the over-reliance on single data sources. Many current studies base their analyses exclusively on one type of data, due to its accessibility. While this facilitates experimentation, it also narrows the analytical scope and risks producing biased or incomplete insights [139].

Many studies plan to incorporate multiple data sources in future work, which could significantly broaden the monitoring capabilities of AI-based OSINT applications. Increasing data diversity may also support the development of more comprehensive OSINT suites, capable of combining various AI techniques across different analytical tasks [139]. For example, some studies propose incorporating additional platforms such as Facebook, alongside the commonly used X data, to improve event tracking and cyber threat detection [151]. Others aim to enhance their misinformation detection models by including data sources beyond X [152], or by increasing dataset diversity to improve fake news identification [153]. Additional efforts suggest integrating various social media channels into cyber intelligence systems to further strengthen analytical capabilities [154].

Security remains a critical concern, particularly in scenarios involving data poisoning or fake accounts, which can compromise the integrity of AI models [139]. Research on the robustness of AI systems against such adversarial conditions is still in its early stages but represents a critical direction, especially for applications involving cyber event detection [155].

A relatively underexplored area is the application of OSINT-driven AI techniques to penetration testing. This domain, which relies heavily on identifying publicly available information to assess an organization’s attack surface, would benefit significantly from AI-enhanced automation and classification [139].

One notable example addresses this by training classifiers on OSINT data from the National Vulnerability Database to detect SQL injection vulnerabilities within web content and social media [156]. Another example, FastEmbed, utilizes the LightGBM algorithm trained on OSINT data collected from exploit databases to estimate the likelihood of vulnerability exploitation in real-world scenarios [157]. While these examples are promising, the volume of research in this area remains small compared to other domains such as cyber threat intelligence [139].

Finally, a notable limitation is the lack of integration between academic research and existing OSINT tools. Many OSINT tools are focused primarily on data collection, while academic efforts often target tasks such as processing and analysis. As a result, the development of end-to-end systems remains limited [139].

One of the few examples is TwitterOSINT, a platform developed to provide real-time visual representations to intelligence analysts. However, its current capabilities are limited to annotation and do not extend into deeper analytical functionalities [158]. Another example is a recent research, that integrate Nmap and theHarvesterTool [150].

AI in the intelligence cycle: Collection, Processing and Analysis Among the various stages of the intelligence cycle, AI proves most effective in the stages of data collection, processing and analysis. In these areas, AI technologies significantly enhance OSINT activities by automating data acquisition, accelerating the detection of relevant patterns, trends, and anomalies, performing classification, and ultimately supporting timely and informed decision-making [159].

In particular, ML algorithms are commonly employed during the processing and analysis stages. The models most commonly used include traditional supervised learning techniques such as Support Vector Machines (SVM), Naive Bayes, K-Nearest Neighbour (KNN), and Random Forests [139].

For instance, Naive Bayes and KNN have been used to classify online news articles by category, such as people, topics, and locations [160]. Similarly, in another notable example, SVMs have been applied to fake news detection tasks by processing features such as article text, title, type, and publication date. In this approach, Term Frequency-Inverse Document Frequency (TF-IDF), a widely adopted feature extraction technique in text-based OSINT applications, is first employed to extract and weigh relevant terms, which are then classified using SVMs. This method proves particularly valuable in national security contexts by enabling the early identification of misinformation campaigns that may compromise public trust or democratic processes [161].

Beyond traditional approaches, more advanced DL architectures are increasingly adopted in OSINT workflows. Commonly used models include Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Long Short-Term Memory (LSTM) [139].

CNNs are particularly well-suited in tasks involving image classification and visual recognition [139]. A notable example is PicHunt, a tool designed to analyze social media images to extract contextual clues and infer geolocation. The system combines traditional and deep learning methods for image similarity detection, and has proven effective in assisting law enforcement and emergency responders [162].

Hybrid architectures, which combine different algorithms, also show promise. One study proposes a combined CNN-LSTM model to improve the detection of Domain Generation Algorithms (DGA) and malware [163].

Models such as LightGBM also play a crucial role, particularly for high-performance gradient boosting tasks [157].

Research in this field covers a wide range of applications; however, the majority of studies are concentrated in the domains of cybersecurity and sentiment/hate speech detection. Notable applications include cyber threat intelligence, detection of domain generation algorithms (DGAs), cyberattack prediction, malicious domain detection, vulnerability discovery, sentiment analysis, and hate speech identification [139].

In the cybersecurity domain, one notable example involves the automated analysis of over 25,000 cyber intelligence reports, collected from blogs, websites, malware databases, and threat intelligence platforms. These reports are automatically labeled to assist security operations teams in decision making [146]. Similarly, DGA detection, critical for identifying malware activity, is often addressed using deep learning models such as RNNs

and LSTMs, trained on labeled OSINT feeds of benign and malicious domains [164]. Complementing this, another study examines 12 years of DNS logs to train a Random Forest classifier trained on 55 DGA families. This approach integrates Explainable AI (XAI) to produce computable Cyber Threat Intelligence (CTI) that is automated, verifiable, and interpretable [165]. Phishing detection also benefits from the integration of OSINT and AI. One recent study multilingual datasets (English and Arabic) and utilizes tools such as Nmap and theHarvester to extract features like domain names, IP addresses, and open ports. The data is then processed using multiple classifiers, including Decision Tree, Random Forest, SVM, XGBoost, and Multinomial Naïve Bayes [150].

Regarding sentiment and hate speech detection, X is frequently employed as a source of OSINT. In one study, deep learning models outperformed traditional n-gram-based approaches in identifying offensive content [166]. Another approach combines Naïve Bayes for sentiment classification and Latent Dirichlet Allocation (LDA) for topic modeling, focusing on polarised (positive and negative) posts and excluding neutral content [167]. A hybrid approach that combines Swarm Intelligence algorithms, PSO and ACO, with traditional classifiers such as Naïve Bayes and SVM further enhances sentiment classification. TF-IDF is used for feature extraction and optimization, improving accuracy and reducing redundancy [149].

Other application areas, though less extensively explored, include fake news detection, opinion mining, event extraction, natural disaster management, misinformation detection, human trafficking investigations, and pandemic monitoring.

For example, in fake news detection, a comparative study evaluates multiple feature extraction methods and classification algorithms, finding that TF-IDF combined with Linear SVM yields the highest accuracy [161]. In opinion mining, a hybrid deep neural network combining Bidirectional Gated Recurrent Units (BGRU) and LSTM layers enhances sentiment analysis, particularly in dealing with contextual nuance and class imbalance [168]. In the area of event extraction, supervised machine learning is used to extract events from Arabic tweets, demonstrating the potential of AI for real-time event detection in multilingual social media contexts [169]. For natural disaster management, the Senti-SAIL tool was applied to German X data during the 2013 Central European flood, providing situational insights through sentiment analysis that supported emergency response efforts [148]. Misinformation detection is another growing area. One study applies NLP techniques, such as those provided by NLTK, to detect discrepancies in sentiment and semantics between tweets from verified sources and unverified users. This unsupervised method enables large-scale identification of disinformation campaigns [152]. AI has also proven valuable in human trafficking investigations by extracting actionable intelligence from unstructured media reports. An enhanced BERT-based Question Answering model processes open-source news to identify relevant cases and entities, addressing the challenges posed by limited structured data [147]. Finally, in pandemic monitoring, lessons from COVID-19 have informed the development of AI-driven methodologies for outbreak detection and management. One proposed framework outlines six phases for integrating AI into public health systems to support real-time, data-driven decision-making in future pandemics [170].

AI in the intelligence cycle: Planning and Dissemination In contrast, the application of AI in the planning and dissemination phases of the intelligence cycle remains relatively limited. Although AI has the potential to support these phases, by suggesting optimal data collection strategies or generating tailored intelligence reports, its deployment must be approached cautiously. Cybersecurity concerns and ethical implications related to automated decision-making impose significant constraints that must be carefully addressed in the design and implementation of AI systems within these contexts [159].

An notable example is EPIWATCH, an AI-based OSINT early warning system that collects and processes vast amounts of multilingual OSINT data from news media and publicly available online sources worldwide, providing early warnings about potential outbreaks [171]. Another study proposes a model that gathers intelligence from social media and generates alerts when information relevant to national security is detected [172].

Some researchers have identified the development of AI capabilities for planning and dissemination as a promising future direction. For example, in a study focused on detecting the online sale of narcotics on X, the authors plan to automate the generation of reports summarizing their model’s findings, formatted according to the reporting templates required by agencies such as the Food and Drug Administration (FDA) and the Drug Enforcement Administration (DEA) [173]. Similarly, the CyberDetect project aims to enhance its cyberbullying detection AI by incorporating an alert system to notify relevant authorities when bullying is identified [174].

2.4.1 AI and SOCMINT

Main Social Media Data Sources Among OSINT data sources, social media platforms stand out for their richness, immediacy, and user-generated nature, offering unique opportunities for intelligence extraction, especially in SOCMINT operations.

X is the most widely used source in research, particularly for cyber threat intelligence and sentiment analysis, due to its relatively open data access policies and the availability of real-time information via APIs. Other platforms such as Facebook and Reddit are used less frequently, while video platforms like YouTube are mainly used when audiovisual content analysis is needed [139].

However, the accessibility of X data has introduced a methodological bias in the literature: platforms like Facebook, YouTube, and especially newer or less accessible services, such as Discord, remain underrepresented [139].

It is important also to observe that, the landscape of social media data access is rapidly evolving. X policies have changed dramatically, introducing significant costs for API access and more stringent limitations. Emerging platforms in SOCMINT analysis include Telegram, LinkedIn and TikTok.

AI-based Techniques in SOCMINT AI is essential for analyzing the vast, heterogeneous, and fast-moving data generated on social media. SOCMINT workflows increasingly rely on AI-driven techniques.

Data acquisition is typically performed through web scraping or platform APIs, capturing not only text but also images, videos, and metadata such as geolocation, timestamps, and user information. This raw data is then preprocessed to reduce noise, filter irrelevant content, remove duplicates, and standardize formats for suitable for analysis.

A key components of AI in SOCMINT is NLP, which enables a deep understanding of textual content by classifying sentiments, identifying prevailing themes through topic modeling techniques, and inferring user intentions. Moreover, NLP helps in recognizing potential threats or suspicious behavior by detecting aggressive, radical, or suspicious content.

ML models can identify abnormal behaviors linked to illicit activities such as fraud, human trafficking, or identity theft. Named Entity Recognition automatically finds key actors, organizations, and locations, facilitating the mapping of social networks.

In terms of multimedia, Convolutional Neural Networks (CNNs) analyze images and videos to detect weapons, extremist symbols, or illegal activities. Similarly, Automatic Speech Recognition (ASR) transcribes audio content from videos or live broadcasts, enabling further textual analysis.

Real-time monitoring systems powered by AI generate instant alerts upon detection of specific keywords or suspicious patterns, while predictive models anticipate future events based on emerging social media trends. Advanced integrated intelligence platforms consolidate data from multiple sources, providing a comprehensive situational overview, often visualized through dashboards, graphs, and geo-spatial maps to support timely decision-making [139, 159, 175].

The following sections outline key application areas in which AI techniques have been successfully deployed on social media data to extract actionable intelligence.

Online Language and Content Analysis

Hate Speech and Online Abuse Detection With the exponential rise in user interactions on social platforms, hateful behaviors, ranging from cyberbullying to attacks based on nationality, gender, religion, sexual orientation, have proliferated online. AI offers scalable solutions to protect vulnerable communities and support societal risk monitoring.

Early studies compared traditional machine learning methods, such as Logistic Regression, Random Forests, and SVM, using Bag-of-Words or TF-IDF features, against deep learning models including FastText, CNNs, and LSTMs for hate speech classification on X. Results consistently showed that deep models outperform traditional classifiers in distinguishing racist, sexist, and neutral content [166]. Interestingly, simpler models like MLPs with TF-IDF features sometimes achieved surprisingly high accuracy, highlighting that feature selection can be as important as model depth [176]. More recent work has introduced contrastive learning frameworks, such as Dual Contrastive Learning (DCL), which combine self-supervised and supervised objectives to capture subtle semantic cues in hateful content. By integrating focal loss to address class imbalance, these deep learning models achieve high precision and robustness [177]. Similarly, studies leveraging domain-specific embeddings with BiLSTM and transfer learning with BERT report

strong results, with BERT achieving the highest F1-scores, confirming the effectiveness of large-scale pre-trained models for hate speech detection [178].

Another study applies neural network-based NLP to detect and monitor violent movements on X [116]. The system integrates similarity models with sentiment analysis to identify aggressive tweets, while clustering algorithms and social network analysis reveal groups of users promoting violence and hate. This AI-driven approach enables law enforcement agencies (LEAs) to detect suspicious accounts and mitigate risks of cybercrime and hostile social manipulation (HSM).

Cross-platform Generalization Hate speech detection across multiple platforms, such as YouTube, Reddit, X, Wikipedia, has been explored using models like XGBoost, FFNN, and SVM, with features ranging from LIWC and BoW to TF-IDF, Word2Vec, XLM-RoBERTa, and BERT embeddings. XGBoost performed best overall, though fine-tuning remains critical for broad generalization [179, 180].

Multilingual and Language-Specific Detection Hate speech detection has been explored across multiple languages using diverse computational approaches. In Arabic, hierarchical annotation and both traditional (SVM, NB, LR) and deep models (CNN, LSTM, GRU) with BERT embeddings were evaluated, revealing challenges in minority and ambiguous categories [181]. For morphologically rich Amharic, RNN-GRU models with n-gram and Word2Vec features, along with hate term clustering, highlighted risks to ethnic minorities [182]. Hindi-English code-mixed texts were addressed using pretrained transformers (XLNet, BERT) and traditional ML with subword features, showing difficulties with irregular syntax and semantic drift [183, 184]. Other studies combined multiple classifiers with BERT embeddings for aggression detection in English and code-mixed Hindi, emphasizing the need for context-aware transformer models [185]. Recent transformer-based methods, BERT, LSTM and BiLSTM, achieved language-independent detection across platforms such as X, Facebook, WhatsApp, and Instagram, outperforming state-of-the-art baselines in Bengali, English, German, and Italian [186].

Multi-Label and Multi-Task Classification Different approaches have extended beyond binary classification to detect multiple abusive content categories simultaneously. For instance, the SP-MTL framework combines CNN, LSTM, and GRU architectures to classify hate, aggression, harassment, offensive, racist, and sexist content across multiple datasets, showing that multi-task learning can capture interrelated abusive behaviors [187].

Explainable AI Explainability has become important in high-stakes applications. One approach combines BERT with LIME to highlight which words most influenced classification decisions, enhancing transparency and trust in AI-driven hate speech detection [188].

Cyberbullying detection shares similarities with hate speech classification but often emphasizes real-time monitoring.

Traditional approaches used Bag-of-Words with Random Forest, AdaBoost, or SVM classifiers in a dual-model strategy to optimize detection speed and accuracy [189]. Multimodal methods combine textual and visual information, such as CapsNet for text and ConvNet for images, using late fusion techniques to improve prediction [190]. Hybrid deep learning models like RoBERTa+ViT further enhance performance across multiple

social media platforms [191].

Real time monitoring Finally, real-time streaming frameworks leverage incremental learning to adapt continuously to new social media data. Even simple Bag-of-Words features can scale effectively to platforms like Twitter Firehose, enabling practical detection of related behaviors such as sarcasm, racism, and sexism [192].

Sentiment Analysis and Opinion Mining The rise of user-generated content has intensified interest in sentiment analysis and opinion mining, to detect and interpret emotions, opinions, and attitudes.

Mental Health and Emotional State Detection Suicide detection from social media blogs has been addressed using LIWC features with a dilated-LSTM, though challenges remain in capturing real-time language dynamics [193].

In the context of depression detection, a study employed TF-IDF unigrams/bigrams with Random Forest and PCA, with suggestions to improve performance via feature fusion and neural networks [194].

A more recent system proposed real-time psychological state detection using n-gram features, Bag-of-Words, TF-IDF, and GloVe embeddings, stored in a centralized database for proactive law enforcement support [11].

A comparative study evaluated DistilBERT, RoBERTa, Logistic Regression, SVM, and Naïve Bayes. Transformer-based models significantly outperformed traditional and lexicon-based approaches, especially in handling nuances like sarcasm, though issues with computational cost and domain-specific terminology remain [195].

Multi-modal Sentiment Analysis Comparative study was proposed for multimodal sentiment analysis using deep neural networks involving visual recognition and natural language processing [196].

Explainable and Multilingual Sentiment Analysis A recent study proposed a hybrid sentiment analysis framework combining XLM-RoBERTa for Sinhala and code-mixed content, and BERT-base-uncased for English with domain-specific lexicon correction. Predictions are accompanied by SHAP and LIME explanations, providing token-level interpretability and aspect-wise sentiment visualizations [197].

Hybrid Optimization and Feature Selection Sentiment classification used TF-IDF features optimized via Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO), followed by Naïve Bayes and SVM. This approach improves robustness over lexicon-based models. [149].

Time-Series Sentiment and Topic Modeling In a complementary study, X sentiment was classified with Naïve Bayes, while Latent Dirichlet Allocation (LDA) extracted topics. Excluding neutral data, the system tracked daily emotional trends with implications for cybersecurity and commerce [167].

Aspect-Based Sentiment Analysis (ABSA) ABSA focuses on detecting fine-grained opinions by combining aspect extraction and sentiment classification. Recent studies apply DL and Transformer-based models to improve accuracy and handle implicit aspects, complex sentences, and multilingual content. Challenges remain in robustness, context understanding, and real-time applications [198].

Fake News and Misinformation Detection In recent years, detecting misinformation and rumors in real time on social media has become a critical challenge due to the unmoderated and decentralized nature of social media platforms.

Traditional machine learning approaches, based on TF-IDF features and SVMs have supported early fake news detection using textual and metadata cues, particularly in national security contexts [161]. Other researches proposed framework to identify misinformation using deep learning LSTM, outperforming the baseline and existing models [199]. FakeBERT, is an example of fake news detection framework with a BERT-based deep learning approach, which combines different parallel blocks of the single-layer deep CNN and filters with the BERT [200].

Multilingual Detection Several studies extend AI-based misinformation detection to less-researched languages. One applied a hybrid approach to Spanish texts, combining linguistic features with ML classifiers, and achieved better performance than transformer baselines like BETO and RoBERTa-BNE. The analysis identified markers of false rumors, such as emotional language and reduced information density, while BERT-based topic modeling revealed diverse deception strategies. This shows how combining feature-based ML with transformers can improve both accuracy and interpretability in multilingual misinformation detection [201].

Similarly, SomBERTa, a transformer trained specifically for Somali, was fine-tuned on newly annotated social media datasets for fake news and toxicity detection, outperforming multilingual baselines. This approach demonstrates replicable methods for linguistically diverse AI systems [202].

Multi-modal Detection To address misinformation spread via text and images, enhanced Transformer architectures have been employed for multimodal semantic understanding, significantly outperforming unimodal models [203]. Building on this, MCOT integrates cross-modal attention, contrastive learning, and optimal transport to align text and image embeddings, improving multimodal detection [204].

Graph-based Detection More recent researches focused on the usage of graph-based methods, for example employing standard GNN, a Graph Attention Network (GAT), and a Bidirectional GCN (BiGCN), to capture user engagement patterns and social context [205].

Radicalization and Extremism Detection Terrorist groups have effectively used the Internet as a tool for propaganda and recruitment. As a result, online radicalization has become a major concern. In this context, digital traces on social networks serve as valuable indicators for identifying radical or extremist behaviors.

Early works for radicalization detection applied supervised machine learning using stylometric, temporal, and sentiment-based features to classify pro-jihadist tweets, combining sentiment analysis, like CoreNLP and RNN, with traditional machine learning classifiers such as SVM, Naive Bayes, and AdaBoost [206]. Other studies adopted one-class classification approaches, using One-Class SVM and KNN, for distinguishing hate-promoting from neutral content [207]. The emergence of deep learning and especially Transformer-based models has advanced this field. For instance, OSINT frameworks leveraging BERT exploit bidirectional attention to produce semantically rich embeddings,

improving text classification for terrorist threat detection on social networks [208].

Beyond purely linguistic features, several works integrate psycholinguistic and emotional dimensions. Hybrid approaches combine linguistic models, such as TF-IDF and word2vec embeddings, with psychological profiling using LIWC, achieving the best results with Random Forests and Neural Networks, and highlighting the superiority of embeddings over n-gram features [209].

More recent studies adopt a multimodal perspective, merging semantic similarity, via pre-trained embeddings against radical lexicons, with emotional statistics derived from emotion dictionaries. Concatenating these features enhances classification, with models such as Logistic Regression and Linear SVM showing promising performance [210].

At a broader level, knowledge graph-based systems have been proposed to integrate heterogeneous data sources for terroristic investigation. These architectures combine semantic web technologies, graph databases, such as Neo4j, and AI-driven reasoning, enabling law enforcement to fuse social media data with official reports for criminal network analysis [211]. In parallel, computer vision techniques have enhanced OSINT capabilities beyond text. PicHunt, for example, leverages supervised deep CNNs to detect visually similar images across social networks, outperforming traditional feature-based methods (ORB, RANSAC). By retrieving modified or altered images in real time, the system aids investigators and emergency responders in extracting contextual clues, inferring geolocation, and uncovering manipulated visual evidence [162].

Multilingual and Language-Specific Detection Given the growing need for multilingual analysis of extremist content, recent research has introduced a framework that leverages a Bi-LSTM model for binary classification and the Transformer-based DistilBERT-multi pre-trained model for multiclass classification. This approach has demonstrated substantial improvements in multilingual text classification, particularly in detecting extremism and radicalization within social networks [212]. Another recent research presented a framework for Arabic radicalization detection, combining a hybrid CNN-Bi-LSTM framework with a top Arabic transformer model (AraBERT) through a weighted ensemble strategy [213].

Cyber Threat Management (CTM)

Cyber Threat Management is a generic discipline focused on identifying, analyzing, monitoring, and mitigating cyber threats throughout their entire lifecycle, combining strategic intelligence, operational detection, and incident response to guarantee protection. OSINT sources, particularly social media platforms, enable the collection of large volumes of information that helps detect emerging threats, identify malicious actors, and understand attacker behavior before incidents occur.

Cyber Threat Intelligence (CTI) A notable example of classical ML in the context of CTI is CySecAlert, which uses active learning to train a relevance classifier for cybersecurity-related tweets. Texts are vectorized using TF/TF-IDF and grouped with streaming clustering algorithms, reducing the need for manual annotation [155]. A similar system, based on DL, combines CNN and BiLSTM: the CNN identified tweets relevant to cybersecurity, while the BiLSTM performed Named Entity Recognition (NER) to extract

software, vulnerabilities, and threat actors, generating alerts and IoCs [214]. Another approach employs a cascaded CNN framework: the first model identifies cybersecurity-related tweets, while the second categorizes them into eight threat types, including DDoS, ransomware, and 0-day exploits. This cascade improves the accuracy and granularity of OSINT-driven threat extraction [215]. Recent studies also show that AI and NLP can generate country-level threat indices by analyzing social media posts on cyberattacks and electronic warfare. Keyword-based techniques combined with CNNs detect anomalies and explain their underlying causes [216].

Beyond social media platforms, CTI can also be extracted from dark web sources like hacker forums, which provide rich metadata and diverse TTPs. One system automatically analyzes and classifies forum content using ML and DL, with RNN-GRU achieving the best performance. Interactive visualizations support analysts in exploring the data for proactive threat detection [217]. A more recent study analyzed darknet markets to identify popular social media platforms and the intent behind user discussions. Both LSTM and Transformer-based models BERT variants were used for multiclass classification of user intentions [218].

AI has also been applied to mitigate the disclosure of sensitive information on social media. The Tweet-Scan-Post (TSP) system employs a layered classification approach that integrates multiple ML models to identify tweets containing sensitive data (personal, professional, or health-related). A sensitivity scaling mechanism further quantifies the level of exposure, providing a practical tool for government, defense, and private-sector security operations [219]. Similarly, clustering-based AI approaches have been developed to automatically detect national security incidents in real time. By grouping related social media content and integrating the full intelligence cycle these systems reduce the analytical workload and accelerate the transformation of raw data into actionable intelligence [172].

Multi-modal CTI Regarding multi-modal threats intelligence, a recent approach presented a framework that integrates NLP, CNNs, and LSTM networks, combining BERT for text class, ResNet50 for photograph processing, and a hybrid LSTM-3-d CNN community for video content material analysis [220].

Bot Detection A recent study proposes a Multi-scale with Signed-attention Graph Filter for social bot detection called MSGS. Specifically, MSGS utilizes a multi-scale structure to produce representation vectors at different scales, which are then combined using a signed-attention mechanism. Finally, multi-scale representations via MLP after polymerization to produce the final result [221]

Graphical Representation Complementary systems such as TwitterOSINT leverage NLP pipelines (Stanford CoreNLP, Stucco) to extract domain-specific entities and provide real-time visualizations, enabling early detection of threats even before their appearance in official vulnerability databases [154, 158].

Social Engineering (SE) Detection SE attacks on social media platforms mimic normal posts but embed malicious intent, making detection challenging due to subtle semantic content and sentiment nuances.

An example regarding the usage of AI for SE detections is a framework leverages

a Recurrent Neural Network with Long Short-Term Memory (RNN-LSTM). Particularly, posts from corporate and personal Facebook accounts are labeled for sentiment via heuristics (keyword matching, source profiling, social graph analysis), lemmatized, and tokenized. The RNN-LSTM model then learns complex spatial and temporal patterns, enabling classification of five SE attack types. This AI-driven approach captures nuanced dependencies in text, offering a scalable solution for social media SE detection [222].

A recent research, focused specifically on phishing attacks and online scam detection, performed the evaluation of seven ML methods to analyze two phishing datasets, in order to classify the type of websites and establish its normality [223]. Additionally, phishing detection has been performed using fine-tuned BERT-based models, respectively DistilBERT, TinyBERT and RoBERTa, leveraging pre-trained transformer architectures for text understanding [224]. Another study integrates computational psycholinguistic analysis with transformer-based models: using Cialdini’s persuasion principles, influential sentences likely to cause security breaches are identified, with BERT and DistilBERT fine-tuned to detect persuasive strategies in phishing attempts [225].

Hacker Profiling In cybersecurity research, traditional AI technique has been used to explore cybercriminal behavior through profiling.

A recent study applied ML models based on the Big Five (OCEAN) personality traits to predict hacker types, achieving high accuracy in mapping personality clusters to hacker categories such as White Hat and Grey Hat. The approach uses K-means clustering to group individuals with similar personality profiles, aiding in the identification of distinct hacker groups and their potential behaviors. The model was validated on real-world datasets, focusing on hackers motivated by professional or career-related reasons [38].

DL approach LSTM was also used in order to classify fake and real accounts on Instagram, demonstrating considerable better performance in comparison to existing machine and deep learning frameworks [226].

2.5 Large Language Models and OSINT

Following the broader application of AI techniques in OSINT operations, recent advancements have shown the growing relevance of LLMs in this domain.

The ability of LLMs to interpret and generate human language, such as producing coherent reports based on analyzed data, makes them particularly appropriate for OSINT applications. The vast, heterogeneous, and unstructured nature of publicly available information demands advanced contextual and semantic understanding. LLMs excel at recognizing and managing linguistic nuances, including idioms, sarcasm, and culturally specific expressions, and are often inherently multilingual, allowing them to process and generate text across multiple languages without the need for separate models. Moreover, LLMs can be rapidly adapted to different contexts without retraining from scratch, leveraging previously learned knowledge. This contextual transferability allows analysts to apply the same model across diverse OSINT domains, simply by adjusting the prompts or instructions provided to the model. Finally, LLMs provide powerful capabilities for analyzing large datasets, identifying patterns, and extracting actionable intelligence from

unstructured text, thereby enhancing both the efficiency and depth of intelligence analysis [227].

These advantages are largely unique to LLMs and distinguish them from more traditional AI systems. However, the integration of LLMs into OSINT workflows also presents several significant challenges. The first key issue is the verification of the LLM generated information: they can produce plausible but inaccurate content (“hallucination”), introduce subtle narrative biases, and, due their persuasive language, encourage overreliance. The opaque reasoning (the “black box” problem), makes it difficult to trace sources or verify how responses are generated. Additionally the models can be manipulated via crafted inputs, resulting in skewed analysis, and their creative content generation introduces legal and ethical risks, which must be carefully managed [227].

In OSINT research, LLMs have been applied to a wide range of open sources data and sectors. SOCMINT dominates the literature due to the abundance, diversity, and real-time nature of social data. However, sources such as news articles, official documents, and technical reports offer complementary advantages, including higher verifiability and domain specificity. Both categories are presented in the following subsections via representative examples of LLMs usage.

Among the most widely used approaches there is LLM prompting, which allows to guide the model’s behavior to perform different tasks without retraining. This approach leverages the LLM’s ability to understand, transform, and generate text, useful across multiple tasks such as classification, sentiment analysis, information extraction, question-answering, or report generation.

Regarding classification tasks, a recent study applied Mistral, Gemma, and Llama models to classify posts from a dark web hackers’ forum into four cybersecurity categories. Models were tested under zero-shot, few-shot, and fine-tuning settings, with fine-tuning yielding the best performance. The work highlights both the potential of LLMs in handling complex hacker jargon and the value of the newly created expert-labeled dataset, while also noting limitations related to dataset size, language complexity, and computational demands [228]. Another study assessed LLM-based chatbots in binary classification and NER tasks for CTI. Using a dataset of 31,281 labeled tweets, binary classification determines whether text is relevant to cybersecurity, while NER identifies domain-specific entities such as vulnerabilities and affected products. A zero-shot prompting methodology was applied to several commercial and open-source models, including LLaMA, GPT4all, Dolly 2.0, Stanford Alpaca, Alpaca-LoRA, Vicuna, Falcon, and ChatGPT. Results show comparable performance in binary classification, but limitations in NER compared to specialized models [229].

LLMs have also been used to infer psychological dispositions and perform sentiment analysis. For example, GPT-3.5 and GPT-4 were tested to infer Big Five personality traits of social media users from Facebook status updates, in a zero-shot learning scenario. The findings suggest that LLMs can do that without having been explicitly trained, in a more accurate way for women and younger users [230]. In the context of sentiment analysis, in another research GPT-based models analyzed multilingual tweets about elections to detect sarcasm and comedy, and automatically identify propaganda. Results indicate that

GPT performs well in analyzing political tweets, however, limitations remain, including managing diversity and accurately interpreting attitudes on social media [231].

LLMs can also identify linguistic and behavioral patterns in social media. For example, GPT-4 was used to detect community practices in pro-Ukraine english X communities, performing best with chain-of-thought prompts, highlighting the potential of prompt design in practice identification, context-sensitive moderation, and study online communities [232]. Similarly, a recent study addresses the detection and classification of persuasion techniques in parliamentary debates and social media within the Slavic NLP 2025 shared task. The system uses GPT-4o-mini with chain-of-thought prompting to identify 25 persuasion strategies across multiple Slavic languages, showing top performance and highlighting the potential of LLMs for multilingual propaganda and influence analysis [233]. Another study combined LLM pre-annotation with fine-tuning for propaganda detection on social media, using a hierarchical taxonomy of 14 techniques [234].

LLMs have been integrated also into fake news identification tasks. An example is the FND-LLM Framework, which integrate textual, visual, and cross-modal analyses for fake news detection. Particularly, text is processed through LLMs, images through CNN, and video through transformers, in order to identify cross-media inconsistency [235].

LLMs not only understand, but can also transform data. One study combined the associative capabilities of LLMs with hypergraph neural networks, representing users as node and social environments as hyper-edges. Particularly, fragmented user records containing demographic and social attributes are injected into LLM prompts, in order to enrich the raw data with contextual and external knowledge, transforming scattered information into cohesive narratives that capture the essence of individual personalities, interests, and behaviors. These enhanced descriptions, together with three types of social environments, are then encoded into embeddings and integrated into a hypergraph neural network, enabling a more nuanced representation of user traits and their online interactions and improving accuracy and richness compared to traditional methods [14].

LLMs are also used to generate content and structured reports. For instance, a study developed a dataset of social media posts annotated with DSM-5 depressive symptoms, evaluating the explanations of symptoms done by GPT-4.1, Gemini 2.5 Pro, and Claude 3.7 Sonnet under zero-shot and few-shot settings, based on fidelity, accuracy, and structural coherence. Results showed that while LLMs can generate symptom-aware and clinically relevant outputs, though quality varied and human oversight remained crucial [236].

Another study introduced a platform to automatically generate target profiles by integrating fragmented and heterogeneous social media data information, obtained from Facebook, X and LinkedIn. The system combines OpenAI's GPT-4o, used to analyze and convert data, with computer vision models such as YOLOv8 and DeepFace for facial recognition and identity verification, producing structured intelligence reports including demographic and publicly available information. The platform demonstrates LLMs' ability to transform fragmented online data into comprehensive profiles [237].

A second frequent approach involves LLMs as embedders, which transform text into vector representations that capture semantic meaning and linguistic nuances. This approach relies on representation learning and text understanding, enabling downstream

models to perform clustering, anomaly detection, semantic similarity searches, and other advanced data analyses. By converting unstructured text into structured embeddings, LLMs facilitate scalable processing of large volumes of social media content and open-source data.

One study combined GPT-3 embeddings with traditional machine learning classifiers, including SVMs, to detect stress in Reddit posts. The embeddings transformed unstructured social media text into informative numerical representations, with SVMs achieving the best performance. This demonstrates that LLM embeddings enhance the detection of subtle mental health indicators, though challenges remain regarding bias and generalization [238].

Another study evaluated LLM embeddings, RoBERTa, BERT, and OpenAI embeddings, for personality prediction on one million Reddit posts from the PANDORA Big Five dataset. Embeddings fed into a bidirectional LSTM outperformed zero-shot methods and traditional feature engineering, capturing linguistic and emotional patterns for scalable personality assessment. OpenAI embeddings achieved slightly higher accuracy, while RoBERTa provided a computationally efficient alternative. Limitations include reliance on a single dataset and potential biases in the language models [239].

One study proposed Social-LLM, a scalable model for behavioral profiling in social networks. The approach combines user content, primarily from profile descriptions, with network interactions such as retweets and mentions. Using a Siamese architecture, Social-LLM generates user embeddings through unsupervised training, optimizing similarity between connected users while differentiating unconnected ones. Evaluated on seven large-scale X datasets, Social-LLM outperforms methods relying solely on content or network features, facilitating robust behavioral analysis. Limitations include dependency on network structure and biases inherent in social media interactions [240].

Advanced Persistent Threats (APTs) are difficult to detect due to their stealth and ability to mimic normal behavior, especially in highly imbalanced datasets. APT-LLM addresses this by combining LLM embeddings with autoencoder architectures (AE, VAE, DAE) to model normal process behavior and identify anomalies. Evaluated on real-world provenance traces from the DARPA Transparent Computing Program APT-LLM outperforms traditional anomaly detection methods, demonstrating the effectiveness of LLM-based feature extraction in cybersecurity [241].

LLMs are also applied in semi-supervised learning and data augmentation, particularly when annotated data are scarce. These methods expand datasets or generate additional labeled examples, improving coverage of rare classes and model robustness.

For example, one study proposed a semi-supervised framework to detect depressive symptoms in X posts, starting from a small clinician-annotated dataset and leveraging LLM-based zero-shot classification to iteratively harvest additional data from self-disclosed depressed users, producing one of the largest dataset for depression detection. Accuracy improved on curated datasets, though gains were limited on noisier data [242].

An hybrid approach is presented in a recent study, which applied LLMs for data augmentation and multi-class classification to extract citizen-reported issues from tweets. Starting from manually annotated data, GPT-based augmentation generated diverse

paraphrases to balance the dataset, which was then used to fine-tune BERT and BERTweet models. Fine-tuned BERT achieved the highest accuracy, outperforming both BERTweet and zero- or few-shot GPT classification. Limitations include reliance on a single generative model and the assumption of mutually exclusive classes [243].

LLMs can also be fine-tuned for OSINT-specific tasks, adapting parameters to improve performance on targeted objectives. An example of study explored the use of fine-tuned LLMs, including Polyglot, Blossom, and EEEVE, for author profiling tasks such as predicting age and gender from anonymous online texts. Experiments showed that Polyglot combined with parameter-efficient fine-tuning strategies like LoRA and QLoRA achieved strong performance in gender prediction while reducing computational costs compared to full fine-tuning. The findings underscore the forensic value of such models for narrowing suspect pools in online investigations, though challenges remain with explainability, gender bias, and lower accuracy in age prediction [244].

There are also additional examples of LLM applications in OSINT, where data are not extracted from social media. For example, recently was introduced CONTACT, a framework for predicting territorial control using LLMs with minimal supervision. CONTACT is based on two approaches: SetFit, an embedding-based few-shot classifier, and a prompt-tuned BLOOMZ-560m model. Trained on a dataset of news articles on ISIS activity in Syria and Iraq, CONTACT employs prompt-conditioned extraction to identify relevant signals such as military operations, casualties, and locations. Experiments show that BLOOMZ consistently outperforms SetFit, largely due to the integration of explicit label definitions and semantic context within prompts, which enhances generalization in low-resource settings. While CONTACT demonstrates that LLMs can enable structured inference from unstructured OSINT streams, its current limitations include reliance on a small dataset and exclusion of multimodal information [245].

Another study explores the integration of LLMs into OSINT in the context of cybersecurity threat detection. Everything start with the aggregation of a comprehensive dataset from social media, news outlets, and public forums, focusing on content relevant to cybersecurity threats. Then LLMs are integrated into the workflow through a three-stage process: initial screening filtered irrelevant content and binary classification of cybersecurity-related information; NER to identify key elements such as threat actors, malware types, and vulnerabilities; synthesis and reporting intelligence for cybersecurity professionals. Results indicate that LLMs can streamline OSINT workflows, improve the accuracy of threat identification, and produce actionable reports [227].

Beyond these approaches, LLMs support specific OSINT applications because of their language and reasoning capabilities, including multimodal, multilingual, temporal, and geospatial applications.

For instance, multimodal analysis allow to analyze images and text together to understand memes, spot propaganda, find the geographic location of pictures, or pull meaningful details from documents. They can also compare texts in different languages to track global conversations, recognize translated content, and analyze how people feel

about certain topics across different cultures. Temporal analysis uses text understanding and reasoning to reconstruct event timelines, detect trends, and analyze coordinated campaigns. Geospatial analysis enables linking entities and events to specific locations, creating maps of conflicts, monitoring infrastructure, or tracking environmental phenomena. By combining these core abilities with different operational ways, analysts can use LLMs for a wide range of OSINT tasks, making their work more effective and versatile. An example investigates GPT-4V’s performance in analyzing multimodal social media content, considering text, images, and videos. Five representative tasks were tested: sentiment analysis, hate speech detection, fake news identification, demographic inference, and political ideology detection. Combining quantitative benchmarks with qualitative case studies, results show that GPT-4V demonstrates strong joint text-image understanding, contextual and cultural awareness, and robust commonsense reasoning. Its performance highlights the potential of large multimodal models for advancing computational social science and social media research. The model struggles with multilingual and culturally nuanced content, particularly considering regional dialects, idiomatic expressions, and the ever-evolving linguistic trends that shape online discourse. The model also shows deficiencies in effectively analyzing fresh, unprecedented content, which underscores the need for continual learning and adaptation [246]. A more recent research explores how AI can enhance OSINT from social media, focusing on Telegram. Particularly, textual and visual content was automatically collected and classified using GPT-based models, then formatted in STIX 2.1 and integrated into OpenCTI for analysis. Results show that AI-driven classification achieves high accuracy in detecting threat-related content, while OpenCTI integration improves situational awareness by visualizing relationships and patterns. The use of open-source threat intelligence infrastructure enables faster, deeper, and more automated analysis compared to manual methods [247].

Regarding multilingual scenario, a recent study aims to fill the research gap proposing COSMUS, a sentiment analysis framework in Ukrainian social media, where frequent code-switching with Russian poses major challenges. COSMUS is characterized by 12,224 corpus collected from Telegram annotated for both sentiment and language, and benchmark three paradigms: few-shot prompting of LLMs, multilingual models (mBERT), and the Ukrainian-specific UkrRoBERTa. They also test augmentation strategies, finding that LLM-based word substitution improves performance, while back-translation reduces it. Results show that fine-tuning the UkrRoBERT with GPT-4o-driven data augmentations achieves the best accuracy. However, limitations remain: code-switched data is underrepresented, sarcasm and complex emotions remain difficult to detect, and Russian tokens are often misclassified as negative [248].

Also temporal analysis in OSINT can benefit from LLMs by evaluating sequences of complex events reported across multiple news articles over time. In a recent paper, these events, referred to as Temporal Complex Events (TCE), are analyzed and extracted using LLMs combined with RAG. Results show that LLMs with appropriate retrieval mechanisms can effectively capture temporal dynamics from extensive text, supporting timely situational awareness and predictive OSINT analysis [249].

An example of LLM application is OSINT geospatial analysis is GeoLLM. Particularly GeoLLM is a method that extracts geospatial knowledge embedded LLMs, useful for

OSINT tasks. By fine-tuning LLM on prompts constructed with auxiliary map data from OpenStreetMap data, GeoLLM achieves an improvement in performance over baseline methods and performs comparably to satellite-based benchmarks [250].

Finally, it is worth noting an emerging trend represented by LLM-based agents and, more broadly, AI agents. In the context of OSINT, this approach is gaining traction, enabling more flexible, automated, and dynamic intelligence workflows.

2.5.1 Retrieval-Augmented Generation (RAG) for OSINT

The use of LLMs in OSINT offers significant opportunities, but also presents notable challenges, including hallucinations, limited domain knowledge, and opaque reasoning, risks that are particularly critical in sensitive intelligence contexts.

A widely adopted strategy to mitigate these challenges is RAG, which enhances LLMs by dynamically connecting them to external, verified knowledge bases. This approach aims to improve accuracy, traceability, and reliability while mitigating hallucinations and bias. In a February 2024 interview, a Thomson Reuters executive stated that within Westlaw AI-Assisted Research, RAG “substantially minimizes hallucinations to nearly zero” [251]. Likewise, LexisNexis has noted that RAG enables it to “provide accurate and authoritative answers grounded in the closed universe of authoritative content” [252].

However, ensuring the effectiveness of RAG requires continuous attention to security, transparency, and human oversight. Most importantly, its performance depends entirely on the quality, coverage, and maintenance of the knowledge base: if the stored data are incomplete, outdated, or biased, the model’s outputs will be the same.

The limited domain knowledge of generic LLMs, trained on broad, general-purpose data, poses a significant constraint in fields requiring specialized expertise. This limitation is especially critical when dealing with classified or restricted information, as in intelligence scenario, which generic LLMs cannot access. Consequently, practitioners face a trade-off: rely on general-purpose models despite their limited accuracy, or develop custom, task-specific pipelines that are more precise but demand large labeled datasets and extensive optimization. An alternative is RAG, which enriches LLMs with external, authoritative knowledge without the need for resource-intensive custom pipelines. Particularly, a significant application of RAG is in classification tasks, where it leverages the reasoning capabilities of LLMs while dynamically incorporating relevant external knowledge.

This is the approach adopted by TechniqueRAG, a framework introduced in recent research within the context of cyber threat intelligence [253], in order to address the challenge of rapidly evolving attacker behaviors. Security analysts aim to map traces left by attackers to MITRE ATT&CK tactics, techniques, and sub-techniques, but manual mapping is time-consuming and requires expertise. Despite ATT&CK defining over 550 techniques, only approximately 10,000 annotated examples are available publicly, motivating automated labeling of incident reports. TechniqueRAG approach combines three core components: retrievers for candidate extraction based on text similarity; instruction-tuned LLMs to re-rank candidates; a generative model fine-tuned on a minimal set of

text–technique pairs. By relying on generic LLMs for retrieval and ranking, and fine-tuning only the generator, it mitigates data scarcity. Explicit step-by-step reasoning improves relevance assessment and captures hierarchical relationships between techniques and sub-techniques. Experimental evaluations demonstrate that TechniqueRAG significantly outperforms baseline approaches. A main limitation is the lack of large, balanced datasets that connect threat reports with ATT&CK techniques, as creating them requires expert knowledge, leaving coverage gaps that reduce RAG effectiveness.

A complementary line of research focuses on real-time cybersecurity threat detection. Traditional static approaches are increasingly insufficient against dynamically evolving threats. Recent work proposes a novel RAG-based cyber-reasoning system that integrates continuous threat intelligence feeds with LLMs, enhancing real-time detection and response capabilities [254]. Leveraging GPT-4o and RAG techniques, this approach incorporates dynamic, up-to-date data sources that static LLMs cannot access. The framework automates retrieval of diverse intelligence feeds (CVE, CWE, EPSS, KEV) via the Patrowl system, embedding them using all-mpnet-base-v2 and storing them in Milvus for efficient querying. Empirical evaluations reveal significant improvements over baseline GPT-4o in addressing recently disclosed vulnerabilities, KEVs, and high-EPSS-score CVEs. This line of work underscores the potential of combining RAG with continuous threat intelligence for adaptive, proactive cybersecurity. It also highlights the growing shift from static to dynamic analysis, paving the way for more intelligent, automated threat management systems.

A similar framework has been proposed in the context of terrorism detection on X [255]. The approach leverages a RAG model powered by LLaMA3 and external OSINT sources, such as news articles, government reports, and academic publications, to provide broader contextualization of X posts. This enriched context enables the detection of subtle linguistic markers and potential signs of radicalization that might otherwise be overlooked. Evaluation used two datasets: regular non-extremist tweets from an online library and extremist tweets extracted from a Kaggle dataset “ISIS fanboys”. Building on prior research on patterns in language and rhetoric indicative of violent radicalization, the system collects tweets in a vector database, enriches unfamiliar terms via an OSINT pipeline querying DuckDuckGo, and uses the LLM to assess extremist tendencies, comparing predictions to ground-truth labels. Results show that integrating RAG with OSINT can improve classification, though performance depends on the LLM, with Gemma2 outperforming LLaMA3. Limitations include reliance on single-keyword query through DuckDuckGo’s API, a method that, while straightforward, constrains both coverage and depth. Moreover, the retrieved sources may not always be reliable or verifiable, introducing potential noise, misinformation, or outdated content into the system. Expanding the retrieval process to multiple keywords, integrating more advanced OSINT techniques, and incorporating curated or verified intelligence databases would significantly enhance the accuracy, robustness, and trustworthiness of the system.

A recent study proposed a RAG-based framework for detecting and classifying extremist content in Kazakh [256]. For each text, the model retrieves relevant information from external sources, including: legislative standards and legal acts regulating extremist activities; scientific articles and research on extremism, radicalization, and countering

extremist ideologies; sociological and political analyses on social tensions and extremism; studies on extremist groups and their recruitment methods; examples of texts with and without extremist content; historical documents and archival materials. The system uses a prompt to guide the RAG pipeline: it generates vector representations of the text, retrieves the most relevant results, and classifies the text as extremist or non-extremist. A score between 0 and 1 is assigned, where 0 indicates no extremist indicators and 1 indicates strong certainty of extremist content. If extremist content is detected, the model further classifies it according to type of extremism, including political, religious, ethnic/-national, racial, economic, social, youth, and environmental. Using a corpus of 9,600 texts evenly split between extremist and non-extremist content, the model achieved high precision and recall, performed consistently across extremism types, and adapted well to language-specific features, demonstrating potential for automated monitoring and threat prevention in other languages and contexts.

Research on the integration of RAG has also been extended to the mental health domain, particularly for the detection of depression and suicidal ideation. An example is MindWatch [257], a framework designed to support early symptom detection and provide personalized psychoeducation. Using a Reddit dataset of 232,000 posts (200,000 for training, 32,000 for evaluation), transformer-based models, including ALBERT, Bio-Clinical BERT, bi-LSTM, OpenAI GPT-3.5, outperformed traditional approaches in accurately detecting suicidal ideation. For psychoeducation, MindWatch employs RAG, with FAISS for retrieval, sentence transformers for embeddings, and foundation models like LLaMA2-7b-chat for augmentation, generating concise, interpretable reports with tailored recommendations. The system is intended as a supportive tool for early symptom detection and education, not a replacement for professional diagnosis.

Another approach, also in the context of mental health classification of social media posts, is presented in a recent study [258]. The system combines dynamic example retrieval from a vector database with pre-trained sentence embeddings and multi-shot LLM prompting. For each post, the top-K most relevant examples are retrieved and incorporated into prompts for LLMs such as Gemini-1.5-Flash, Gemma-2B-it, and GPT-4o. Results show that RAG prompting outperforms zero- and few-shot methods and achieves performance competitive with instruction-tuned models, without any fine-tuning. The study also highlights the importance of prompt design, showing that careful engineering can significantly affect results. Overall, RAG improves accuracy and efficiency in mental health classification, offering a practical solution for resource-constrained settings.

RAG has also been applied to hate speech detection and countering. In recent research [259], a pipeline was developed to identify and respond to hateful tweets related to the Ukraine war and bio-weapons labs, using a dataset of 500,000 unique tweets. Zero-shot classification was performed with Mistral Instruct, guided by X’s hate speech guidelines. Detected hateful content is addressed via RAG: external news data is chunked, embedded with MPNET, stored in a FAISS vector database, and retrieved to inform prompts, enabling the model to generate contextually accurate and factually grounded counter narratives. This approach reduces hallucinations and produces nuanced responses, but limitations include handling subtle cultural references, regional dialects, and implicit meanings, as well as dependency on training data quality and diversity.

A recent study, also aimed at content moderation, proposed Class-RAG [260], a classification framework that combines LLMs with RAG to enhance content moderation tasks. It dynamically accesses a real-time retrieval library for semantic hotfixing, addressing challenges like subjective policy interpretation, diverse audiences, and slow fine-tuning cycles. The system consists of four main components: an embedding model, a retrieval library divided into negative and positives examples, a retrieval module, and a fine-tuned LLM classifier. Prompts are embedded, compared with indexed examples via FAISS, and the top references are provided to the LLM for classification. Tested against eight obfuscation techniques, Class-RAG shows superior performance, robustness to adversarial attacks, and scalable improvements through library expansion. Advantages include flexibility, transparency, and rapid updates, while limitations involve false positives/negatives, dependence on the base LLM and training data, limited out-of-scope knowledge, and potential misuse such as over-censorship.

A recent study [261] proposed a RAG-based framework for detecting abusive language on social media by integrating psychological knowledge into an LLM. Using the ALDIPF dataset of 505 users, which includes personality traits, emotional reactions, and message content, association rule mining and decision tree analysis extracted rules linking user attributes to class labels. Nine rules generated via the Apriori algorithm were selected based on support, confidence, and lift; five were associated with the Harmful class, and four with the Neutral class. Experiments were conducted using GPT-3.5 Turbo. Experiments showed the framework outperformed baselines, maintained robustness to noise, and demonstrated strong potential for personalized, context-aware detection of abusive language.

Hateful content often spreads through multimodal media such as memes, which can evade traditional text-only or image-only detection. To address this, a multimodal hate detection framework was proposed that integrates OCR to extract embedded text, captioning for visual description, sub-label classification, RAG for context-aware retrieval, and VQA for iterative analysis of symbolic and contextual cues [262]. Experiments on the Facebook Hateful Memes dataset show that this approach outperforms unimodal and conventional multimodal models in accuracy and AUC-ROC. The pipeline extracts text and captions, combines them with the image for analysis, and uses GPT-4 and GPT-3.5 for context-sensitive Q&A to identify hate signals. Q&A interactions are stored in a contextual database, and the RAG-based Hate Detection Module cross-references a knowledge base to generate detailed summaries. Data augmentation further improves robustness, demonstrating the effectiveness of combining RAG with multimodal strategies for nuanced hate detection in memes.

Another significant application of RAG is in fake news and misinformation detection. A recent study [263] proposed a system that classifies textual claims into four categories: Supported, Refuted, Conflicting Evidence/Cherrypicking, or Not Enough Evidence, while providing supporting evidence for each classification. Using RAG and In-Context Learning (ICL), the pipeline retrieves the top three relevant documents from the Averitec dataset via FAISS, converts the claim into a question using an LLM, and extracts concise answers from the documents to form an evidence set. A separate LLM then classifies the claim based on this evidence. The approach demonstrates RAG’s potential for scalable,

evidence-backed fact-checking.

Similarly, another study [264] proposed a framework designed to autonomously collect and process OSINT in response to user queries. To handle diverse, rapidly updating sources and complex formats, the system integrates chain-of-thought reasoning with RAG, allowing the LLM to dynamically access external OSINT tools. The architecture includes a scheduling module for query interpretation and tool coordination, a memory module, based on RAG, for storing and reusing intelligence, and a toolkit module providing access to search engines, Whois queries, and vulnerability scanners. While flexible, the framework depends on tool quality, inherits LLM limitations such as reasoning errors and biases, and requires manual setup for tool integration, which can constrain adaptability and scalability in rapidly evolving domains.

Another valuable application of RAG lies in the automatic generation of reports from specific information.

RAG has been applied to intelligence report generation, which requires integrating dynamic event data into coherent narratives. Manual report creation faces challenges in synthesizing evolving incidents and actors, formulating precise queries, and filling information gaps.

Specifically, the FABULA [265] framework allows analysts to query an Event Plot Graph (EPG) to retrieve relevant event plot points from a knowledge graph, which can then be used to enrich LLM prompts during report generation. Informally, an EPG can be seen as a graph linking people, places, and events into a connected storyline. These points enrich LLM prompts for report generation. FABULA’s RAG-based approach combines narrative structures, LLMs, and knowledge graphs to produce intelligence reports that exhibit high semantic relevance, strong coherence, and low redundancy, as demonstrated through both ROUGE scores and human evaluation.

Similarly, in the criminal investigation domain, the CrimeKGQA system [266] integrates RAG framework with a Neo4j-based crime knowledge graph to generate precise Cypher queries and dynamically retrieves the relevant facts, grounding answers in data rather than hallucinations. Users submit questions that are translated by an LLM into Cypher queries executed on the graph; results are then fed back to the LLM to produce concise, contextually grounded answers. The knowledge graph contains 61,521 nodes across entities such as suspects, victims, locations, events, and evidence, and 105,840 relationships capturing complex interactions. CrimeKGQA can identify people involved in multiple crimes, map connections between entities, and analyze geographical patterns of crime, producing both textual explanations and visualizations of the inferred crime networks.

Another notable application of RAG is in the cybersecurity domain. IDS/IPS in large enterprises can generate hundreds of thousands of alerts per hour, overwhelming security analysts with logs that require deep and rapidly evolving expertise. To address this challenge, CyberRAG [267] is a modular, agent-based RAG framework designed for real-time classification, explanation, and structured reporting of cyber-attacks. At the core of the framework there is a LLM agent orchestrating specialized classifiers, tool adapters, and

an iterative retrieval-and-reason loop over a domain-specific knowledge base. Unlike standard single-pass RAG, CyberRAG allows reasoning over retrieved evidence and dynamic re-querying to refine results. Upon an alert, it identifies attack types, retrieves supporting context, and generates comprehensive reports with mitigation steps. An LLM-powered assistant enables interactive querying for analysts. The framework is extensible, new attack types can be added without retraining the core agent, and the RAG component enriches explanations with relevant knowledge. Evaluations show superior accuracy, robustness, and explainability compared to traditional classifiers, effectively handling noisy or unseen inputs.

Social media platforms provide real-time data for rapid disaster management. The Open-Source Intelligence Disaster Event Tracker (ODET) [268] is a modular platform integrating RAG, with customizable endpoints (APIs, classification, text inference, embeddings) and agents (data collection, classification, embedding, knowledge graph, summarization). ODET used unmodified AI models: GIST-all-MiniLM-L6-v2 for sentence embeddings and Mistral-Nemo Instruct-2407-Q6_K_L for zero-shot classification and other tasks. Datasets included tweets from Hurricane Harvey and the 2023 Turkey earthquake. RAG combines retrieval from ChromaDB embeddings with generative summarization, producing accurate reports, while knowledge graphs organize topics. The system demonstrated high reliability under noisy conditions, and its modular design allows easy integration of new models on standard hardware or cloud platforms.

Finally, it is worth noting the rise of multimodal RAG, significant in context intrinsically multimodal as social media. A survey of multimodal RAG is presented in [269,270].

2.6 Conclusions

This work focuses on SOCMINT, the intelligence derived specifically from social media platforms, with X chosen as the primary platform due to its high volume of textual content and real-time nature. Unlike traditional OSINT approaches, analyzing X allows for timely identification of potentially sensitive or dangerous content.

The main goal of the proposed pipeline is not just to detect threats, but to provide detailed profiling and reporting. Tweets are analyzed and categorized, but the system also explains the reasoning, highlighting specific words, hashtags, emojis, and communication patterns that indicate potential risks. This moves beyond simple classification, offering contextual insights that are valuable for intelligence teams.

To achieve this, the pipeline leverages RAG rather than relying solely on standard LLMs. RAG grounds the model in specialized documents, reducing hallucinations, providing explainable reasoning, and producing structured reports. This makes the system both reliable and interpretable, supporting analysts rather than replacing them.

Overall, this work demonstrates a cohesive approach to SOCMINT, integrating threat detection, explainable profiling, and RAG-based analysis in a single framework that can help identify and understand sensitive content on social media.

Chapter 3

Technologies and Tools

3.1 Social Media Scraping

Social media **scraping** refers to the automated collection and extraction of large volumes of publicly available raw data from social platforms, followed by a process of structuring and preprocessing, aimed at enabling subsequent analysis. This process typically involves gathering user-generated content, metadata, and other relevant information [271–273].

3.1.1 Scraping Methods

There are several approaches to performing social media scraping, such as the use of third-party scraper APIs, official APIs provided by platforms, python scraping libraries combined with proxies, or pre-made social media datasets.

The choice of the most suitable method depends on multiple factors, including ethical and legal considerations, cost, available functionalities, and ease of use, and should align with the specific objectives of the project, whether data collection, competitor analysis, or market research.

A key aspect to consider is the overall complexity of the scraping project, which can be influenced by the number of websites and records to be scraped, the data format and types, the scraping frequency, the required turnaround time and data loading speed, as well as the complexity and structure of the target websites and the presence of anti-scraping measures. These variables directly affect costs, timelines, and the level of expertise required: the more complex a website is to scrape, the more hours will be needed from skilled developers, leading to higher overall expenses.

From a cost perspective, these challenges can be generally divided into two broad categories. The first relates to the initial acquisition of the required data, which is typically a one-off effort. The second concerns the recurring costs of support and maintenance, which may be necessary when continuous monitoring of the platform is required, along with periodic adjustments to the scraping code in order to handle structural changes or updated anti-scraping measures.

3.1.2 Challenges and Solutions in Social Media Scraping

Although social media scraping and web scraping in general are essential techniques for data collection and analysis, they face several challenges that can be broadly categorized into two groups: issues related to the target websites themselves and limitations of the scraping tools and infrastructure.

One major difficulty arises from the nature of modern websites. Many platforms load content dynamically, using technologies such as AJAX (Asynchronous JavaScript and XML), which improve user experience but render traditional HTML scraping ineffective. A possible solution is to use headless browsers like Puppeteer, Selenium, or Playwright to mimic user interactions and render content.

Website structural changes, even small, also pose a significant problem. Scrapers that rely on fixed JavaScript or HTML element paths can break with even minor modifications to the website layout. To mitigate this, adaptable and target-specific parsers should be developed and regularly updated to handle such changes efficiently.

Anti-scraping measures implemented by websites present another set of challenges, including:

- CAPTCHAs, designed to differentiate humans from bots and prevent automated access;
- Robots.txt files, which specify pages disallowed for crawling to protect server resources and avoid network congestions;
- IP blocking, to limit repeated requests from the same IP address;
- Honeypots, which trap bots with hidden links and are used to detect and block unauthorized access;
- Browser fingerprinting, which tracks website visitors through unique device and browser characteristics.

To counter these defenses, scrapers can implement solutions such as CAPTCHA-solving services, respect site policies by using official APIs or obtaining permission, utilize rotating or residential proxies to evade IP blocks, avoid interacting with hidden honeypot elements, and reduce browser fingerprint uniqueness through shared proxies or stealth techniques.

Another challenge is scalability, the need to handle large-scale data scraping efficiently. This requires parallelization and optimization to maximize performance. A useful solution is to adopt asynchronous tools that support concurrent requests.

From a legal and ethical standpoint, scraping public data is not inherently unlawful but must comply with privacy regulations and website terms of service. To minimize legal risks, it is crucial to adhere to site-specific policies and respect the directives stated in robots.txt files (see Section 4 for more details).

The final challenge is related to infrastructure maintenance, as ongoing updates are required to meet evolving needs and manage increasing data volumes. A solution is to use providers that offer built-in proxy and parser support, as well as scalable infrastructure [274, 275].

Social Media Scraper APIs

Social Media Scraper APIs are ready-made tools and scripts designed to facilitate large-scale data extraction from social media platforms. They typically provide an interface with pre-configured templates for major platforms, requiring users to enter only basic parameters such as the target URL, search query, and desired output format. The system then automatically generates the corresponding scraping code customized to needs.

One of the main advantages of these APIs lies in their ease of use. Most solutions operate on a no-code or low-code basis, enabling users without extensive programming skills to perform data extraction efficiently and rapidly. Furthermore, many incorporate built-in measures to bypass common scraping challenge, thus sparing users from having to manage these complexities directly.

From a cost perspective, commercial Social Media Scraper APIs typically follow subscription-based pricing models, often requiring a monthly commitment rather than pay-as-you-go options, which is a less frequent solution. While the initial cost might be higher than a one-time scraping project, ongoing maintenance costs tend to be lower since the service provider manages infrastructure demands, including resource-intensive tasks like running headful browsers. Users only bear the cost of executing the scraper and the subscription fees, which can provide predictable budgeting for long-term projects.

However, these APIs can be less flexible compared to fully custom-built scrapers. Since they rely on pre-configured templates, they may not accommodate highly specific or unconventional data requirements. Additionally, because these services automate much of the scraping process, users often have limited control over customization and optimization.

Regarding ethical and legal considerations, many commercial APIs adhere to the terms of service of social platforms and implement mechanisms to respect user privacy and data usage regulations. Nevertheless, it remains crucial for users to verify compliance with local laws and platform policies, especially when dealing with sensitive or personally identifiable information.

In summary, Social Media Scraper APIs are particularly suitable for developers and organizations needing to collect large volumes of data quickly and with minimal technical overhead. Their combination of ease of use, scalability, and managed infrastructure makes them particularly advantageous in case of projects requiring predictable ongoing costs, though at the expense of customization and potentially higher subscription fees [276].

Official Social Media APIs

Many social media platforms provide official APIs that allow developers to programmatically access platform data in a reliable and authorized manner. These APIs offer a structured and compliant way to connect and interact with social media services, ensuring adherence to each platform's terms of service and relevant data protection regulations.

One of the key advantages of using official APIs is their clear legal standing. Because the data access is performed through channels approved by the platform, developers and organizations can be confident that their activities follow the platform policies and legal requirements, significantly reducing the risk of violations related to unauthorized data

collection.

However, official APIs often come with notable limitations. They are primarily designed for application developers to retrieve data from users who have explicitly granted permission through installed applications. These APIs typically impose strict rate limits on the number of requests per application or user, as well as daily usage limits. Many users and developers also report challenges in using these APIs, specifically in setting up developer accounts and obtaining API keys procedures, highlighting occasional lack of comprehensive support or documentation, and a reluctance from platform providers to simplify or fully open data access.

In terms of cost, official APIs are generally free to use up to certain limits, which makes them accessible to developers and organizations with constrained budgets. Nevertheless, scaling beyond free tiers may require subscription fees or premium plans, depending on the platform's policies.

For businesses managing widely used social media applications or aiming for fully compliant data acquisition, official APIs remain the most effective and legally choice.

Notable examples include Facebook Graph API [277] and X API [278], which enable developers to legally and programmatically retrieve structured data from their platforms.

To use these APIs, developers must first register for a developer account, obtaining API keys and credentials. It is also essential to review the official documentation to understand available endpoints, parameters, and rate limits, as well as to follow the prescribed OAuth-based authentication process to obtain valid access tokens. Once authenticated, requests can be made to the desired endpoints [276].

X Official API The X Official API [278] provides authorized and structured access to X's public data, including posts, users, spaces, DMs, lists, trends, media, and more, enabling developers and researchers to retrieve and manipulate platform information programmatically. This API is considered the most reliable and legally compliant method to access X data, as it operates within the platform's terms of service and data protection regulations.

Access to the X API requires registration through the X Developer Portal, which serves as the central hub for obtaining API credentials and managing applications. A X account is mandatory to apply for developer access, which includes submitting the intended use case. Upon approval, developers receive essential credentials, including API Key and Secret, Bearer Token, Access Token and secret. These credentials are critical for secure and authenticated interactions with the API.

Three are the main pricing tiers. At the lowest level, the Free tier provides highly limited functionality, allowing only 500 posts per month and a maximum of 100 read requests. While this option permits minimal experimentation or very small-scale projects, it is insufficient for most professional, academic, or large-scale analytical purposes. The next tier is the Basic plan, which costs \$200 per month and expands the available request limits and grants access to additional endpoints. It is primarily targeted toward independent developers and smaller organizations that require more robust capabilities than those offered by the Free tier. At the top end, the Pro tier is priced at \$5,000 per month, enabling large-scale integrations and advanced analytics suitable for major companies,

high-volume applications, and complex research projects.

The primary advantage of using the official X API is the assurance of uninterrupted access without the risk of being blocked or banned, as long as all API guidelines and rate limits are respected. This ensures long-term stability and compliance for projects relying on X data.

X offers free access to public data for researchers through the X DSA Researcher Application, aligned with the EU Digital Services Act (Regulation (EU) 2022/2065, DSA) (See 4). To qualify, researchers must satisfy specific eligibility criteria set forth in Article 40, Sections 8 and 12 of the DSA and submit a detailed application through the designated form. However, many members of the X community have reported difficulties in obtaining access, often receiving denials on the grounds that their research does not meet the narrowly defined conditions focusing on systemic risk within the EU. This indicates that official researcher access is restrictive and limited to specific use cases.

Python Libraries with Proxies

Custom scrapers are typically developed using one or more scraping libraries. Python, with its extensive library support and active community, is an excellent choice for scraping tasks of any complexity or scale. It offers a rich selection of open-source libraries and frameworks specifically designed for web scraping, such as Scrapy [279], BeautifulSoup [280], Selenium [281], and Requests [282].

Particularly, Scrapy is a powerful framework ideal for managing large-scale and complex scraping projects, BeautifulSoup is used to parse HTML and XML documents, enabling navigation, Selenium is used to automate web browser interaction from Python, while Requests allows efficient sending of HTTP requests.

One of the main advantages of using Python libraries is the high degree of flexibility and customization they offer. Developers can tailor scrapers to specific data requirements, handle complex website structures, and implement strategies to overcome anti-bot protections. This approach allows complete control over the scraping process and data extraction logic.

From a cost perspective, building a custom scraper involves an initial development investment to create the core scraper framework. Recurring costs arise from running the scraper, such as server or cloud infrastructure expenses, proxy services, and CAPTCHA solving tools, as well as maintenance efforts. Websites frequently update their structure or anti-scraping defenses, requiring regular debugging and adjustments.

Over time, the benefits of this solution become more apparent, as initial setup costs are amortized and ongoing running and maintenance expenses may be lower than purchasing commercial APIs or already scraped datasets.

Regarding ethical and legal considerations, custom scraping carries a higher risk of violating platform terms of service or data privacy regulations if not carefully managed. Unlike official APIs, these solutions require developers to ensure compliance independently, which can be challenging depending on the policies of the target platform and regional laws.

However, they require developers to handle challenges like anti-bot protections. Obviously, in terms of ease of use this approach demands significant programming expertise

and familiarity with web scraping challenges. It is therefore best suited for developers with experience aiming for tailored and flexible data extraction solutions [276,283].

Pre-made Social Media Datasets

Another approach to acquiring social media data is through ready-made social media datasets, sometimes also provided by web scraping service vendors. These datasets consist of information collected from publicly available data across social platforms, eliminating the need for manual data collection.

The main advantage of pre-made datasets is that they simplify data acquisition by removing the need to develop, deploy and maintain custom scraping pipelines. This reduces the technical requirements and speed up the process. Additionally, many providers offer some degree of customization, allowing users to filter datasets based on parameters like geographic location, date range, or specific topics, to better align particular research or business objectives.

However, they may have limitations in terms of customization and may not always reflect the most up-to-date data, depending on the frequency of updates provided by the service.

In terms of cost, purchasing datasets can be more economical and straightforward than developing an in-house scraping solution or subscribing to commercial scraping APIs, especially for one-off projects or initial explorations. However, depending on the data requirements and integration efforts needed, additional costs might arise in order to adapt and process the data.

Ethically and legally these datasets often come with assurances of compliance with platform policies and data protection laws. Still, users should conduct verify the legitimacy and scope of permitted data use.

Pre-made social media datasets are particularly suitable for business users or organizations without specialized technical skills who need access to social media data without investing in complex data extraction infrastructure.

3.1.3 Choice of Strategy and Motivations

After a thorough analysis of the various scraping methods, the decision was made to focus on evaluating some of the best social media scrapers currently available on the market. This review aims not only to guide the choice for this specific research project but also to provide the reader with a comprehensive overview of the available solutions and their respective strengths and limitations.

Simultaneously, the official X API was examined to assess its features, advantages, and limitations. This analysis is essential to understand the feasibility of a compliant use aligned with platform policies and regulations. From a legal and compliance standpoint, the use of official social media APIs remains the most appropriate choice for operational and ongoing projects, as they ensure full adherence to platform terms of service and regulations.

For the initial testing phase of the data pipeline, a commercial scraper API service was

employed. This approach allowed verification of the correct implementation and functioning of the system's modules. Among the top-rated tools available in 2025 that offer an API for code integration, Apify was selected for the initial scraper testing. Although the primary focus of the research was on tweets, the preference was for a scraper capable of supporting multiple major social media platforms, including Facebook and X, which are two of the most widely used channels for sharing textual content such as opinions, news, and discussions. Apify met these criteria and offered additional advantages: it provides \$5 of free monthly credit and low monthly price, making it cost-effective, and it has been extensively adopted in academic studies and theses, which reinforces its reliability and suitability for research purposes.

Given the sensitivity of the topics addressed, such as terrorism, depression, cyberbullying, and hacking, direct collection of live data could raise significant ethical and legal concerns, particularly regarding user privacy and compliance with platform policies. Additionally, finding a large volume of relevant tweets for effective pipeline testing directly on social media can be challenging. To address these issues responsibly, the final validation phase utilized pre-existing, publicly available anonymized datasets, thereby avoiding direct handling of personal data in the research context.

In summary, this multi approach, combining the use of commercial scraper APIs for rapid prototyping, the evaluation of official APIs for legal compliance and functional robustness, and the adoption of pre-made datasets for ethical data handling, provides a balanced framework that addresses technical feasibility, cost considerations, and ethical responsibilities.

The next step involves a detailed review of the top social media scrapers on the market, culminating in the selection of the most suitable tool to support the data acquisition needs of this research.

3.1.4 Comparison of Leading Social Media Scraper APIs (2025)

The search for the best social media scraper began by analyzing evaluations conducted by other authors [276, 283–288]. Starting from this aggregated information, a detailed examination of the characteristics of several top scrapers was conducted, highlighting concrete aspects such as pricing, supported social media platforms, speed, integration of techniques to bypass anti-scraping measures, and API availability. It is important to note that the choice of scraper to integrate into a data collection pipeline depends heavily on the specific requirements and constraints of the use case. Based on this evaluation, only one scraper was selected and used for testing purposes in this thesis.

Bright Data

Bright Data [289] is one of the best no-code, API based solutions for social media scraping. In addition to scraping, it offers services such as residential proxy networks covering over 150 million IPs sourced from real devices in more than 195 countries, as well as ready-to-use datasets. The main pricing model is pay-as-you-go, with enterprise plans starting at approximately \$499 per month. A free trial is available, offering limited bandwidth over three days.

Bright Data supports fast, real-time, accurate scraping from a wide range of social media platforms, including Instagram, Facebook, LinkedIn, TikTok, X, YouTube, and others. It enables scraping of posts, profiles, TikTok searches, and Facebook groups, with additional platforms available upon request for dataset purchases.

Key solutions include pre-collected datasets for major platforms, customizable through filters; the Web Unlocker API, which automates bypassing anti-bot mechanisms with no concurrency limits; and the Social Media Scraper API, offering a cloud-based environment with ready-made scraping functions, code templates, and scheduling capabilities.

Pro: Supports multiple major social media platforms; flexible payment options including one-time dataset purchases; API and no-code interfaces; automatic proxy rotation, exceptionally large proxy network with over 150 million IPs globally; high reliability and scalability; enterprise-grade stability and performance; anti-bot and CAPTCHA bypass features; 24/7 customer support and compliance.

Cons: Pricing can be expensive, especially for small projects; requires some technical expertise for advanced customization; complexity in setup and use of advanced features may be a barrier for non-expert users.

Apify

Apify [290] is a cloud-based web scraping and automation platform designed to support low-code, no-code, and developer-focused workflows via APIs. While widely used for social media scraping, it also supports e-commerce, review sites, job boards, and general web scraping, with more than 120 geotargeting locations. Its pricing follows a pay-as-you-go model, starting at approximately \$39 per month for the basic plan, which includes \$5 of free monthly usage.

Apify enables data extraction from multiple social media platforms, including Facebook, Instagram, X, LinkedIn, TikTok, and YouTube. It enables scraping posts from Facebook, Instagram, and X; profiles from Instagram, LinkedIn, TikTok, and X; and data from Facebook groups.

The platform provides a large library of pre-built reusable scraping and automation scripts called Actors, which can be used as-is, customized, or developed from scratch in various programming languages. Key capabilities include JavaScript rendering, CAPTCHA solving, scalable infrastructure for large datasets, and task scheduling. Apify integrates both datacenter and residential proxies with automatic IP rotation and monitoring. For developers, it provides open-source tools such as the Crawlee SDK, enabling fine control over scraper logic and integration with CI/CD pipelines.

There are two operating modes: the first involves execution through a GUI via the website, and the second, used in this thesis during the initial testing phase, concerns the use of an API, specific to each actor, which allows greater integration with code.

Particularly for testing purposes, the actor named *Twitter Tweets Scraper* was selected. This actor receives as input the URL(s) of the X profile(s) to scrape (*start_urls*), supporting multiple X URLs. Additional input parameters include the start date for tweet collection, which defaults to yesterday (*since_date*), and the expected number of tweets to be collected, with a default value of 10 (*result_count*).

The output can be exported in various formats, including JSON, XML, CSV, Excel, or HTML. Tweets are sorted in reverse chronological order. The sorting is applied simultaneously to multiple profiles when scraping more than one URL.

This actor operates on a pay-per-result basis, with a fixed fee of \$0.50 is applied for every 1,000 items in the actor's output dataset.

Each tweet in the output includes the following fields: X username (*user.name*), avatar URL (*user.profile_image_url_https*), tweet URL (*url*), creation timestamp (date and time) (*created_at*), full tweet text (*full_text*), favorite count (*favorite_count*), and retweet count (*retweet_count*).

Pro: Supports multiple major social media platforms; flexible payment options; API and no-code interfaces; automatic proxy rotation; high reliability and scalability; enterprise-grade stability and performance; anti-bot and CAPTCHA bypass features; large library of reusable and customizable scraping scripts; strong developer community and documentation.

Cons: Steeper learning curve for non-technical users; interface may be unintuitive at first; complexity in setup and use of advanced features may be a barrier for non-expert users; requires some technical expertise for advanced customization; pricing model can be confusing; necessity of frequent updates to stay functional.

Zyte

Zyte provides a suite of web-scraping products and managed data services, with a strong focus on e-commerce projects and public domain data. Its main product families include Scrapy Cloud, an hosted service to run Scrapy spiders, Smart Proxy Manager, formerly Crawlera, a proxy management and anti-ban tech, and the Automatic Extraction, an AI-powered data extraction API, now integrated into the Zyte API. Zyte API [291], the main company product, is a no-code, enterprise-grade, general-purpose cloud web scraper designed for large-scale, high-reliability data extraction, very fast on social media platforms. Pricing is usage-based: Zyte primarily uses a pay-as-you-go/per-request model, with pricing starting at around \$100 per month; Zyte also offers enterprise plans and \$5 of trial credits.

Functionally, Zyte supports extraction of public social-media data and many e-commerce/news pages, including Instagram public profiles and hashtag feeds, Facebook public Pages & Groups, TikTok public videos and user pages, X, LinkedIn, YouTube, Pinterest, Reddit, and Discord. The platform integrates ban-avoidance techniques, such as automatic proxy selection and rotation, residential and datacenter IPs, geolocation options, smart retries and offers CAPTCHA-handling features where applicable.

Pro: Supports multiple major social media platforms; flexible payment options; API and no-code interfaces; automatic proxy rotation; high reliability and scalability; enterprise-grade stability and performance; anti-bot and CAPTCHA bypass features; 24/7 customer support.

Cons: Relatively high cost for small projects; requires some technical expertise for advanced customization; limited pre-built social media solutions.

SmartProxy (Decodo)

SmartProxy (Decodo) [292] is a proxy provider specialized APIs for e-commerce, SERP, social media, and general web scraping. It offers a proxy-based, no-code social media scraping solution with pre-built social media scrapers designed to help businesses and individuals extract data from social media platforms. Plans start at approximately \$35 per month and a 7-day free trial or 14-day money-back option. There is no pay-as-you-go option.

The API is specialized for Instagram, Reddit, and TikTok, enabling the extraction of posts from Instagram and TikTok, as well as profiles from Instagram. Integration can be done either via a proxy server or directly through the API, supporting both instant data retrieval and callback-based delivery. The platform does not support batch data retrieval.

SmartProxy API handles proxy rotation, anti-detection techniques, and geolocation targeting, which is currently limited to eight countries. It automatically manages content language detection. The service guarantees no request limits and offers developer-friendly resources such as Postman collections, GitHub code examples, and an API playground with ready-made templates for faster integration.

Two API tiers are available: Core, providing basic scraping features and geo-targeting limited to eight countries; and Advanced, which adds JavaScript rendering, structured data outputs (JSON/CSV), and global geolocation support.

The proxy pool comprises a large number of residential and datacenter IPs, ensuring reliable scraping performance even on geo-restricted platforms. While SmartProxy offers CAPTCHA bypass capabilities, it does not include built-in CAPTCHA solving, which may require additional tools and complexity.

Pro: API and no-code interfaces; automatic proxy rotation; high reliability and scalability; anti-bot and CAPTCHA bypass features; enterprise-grade stability and performance; 24/7 customer support.

Cons: Limited number of pre-built social media scrapers; moderate speed depending on proxy type; limited geolocation on some plans; no built-in CAPTCHA solving; requires some technical expertise for advanced customization.

PhantomBuster

PhantomBuster [293] is more than just a scraper, it is a cloud-based, no-code automation tool designed for social media data extraction. Pricing starts at approximately \$69 per month for the lowest package, with 14-days free trial. There is no pay-as-you-go option.

PhantomBuster supports data extraction from multiple social media platforms, including LinkedIn, X, Instagram, Facebook, and YouTube.

The main features include a user-friendly interface with a wide range of pre-configured workflows, called Phantoms, cloud execution, and regular updates to adapt to changes in social media platforms. Users can schedule tasks at specific intervals for consistent automation, and proxy management capabilities are included to avoid IP blocking during scraping tasks. Advanced users can also leverage a JavaScript console to create or customize scripts for greater flexibility.

Pro: Supports multiple major social media platforms; automatic proxy rotation; high reliability and scalability; API and no-code interfaces; enterprise-grade stability and performance; 24/7 customer support; anti-bot and CAPTCHA bypass features; very user-friendly no-code interface; wide range of pre-configured automation workflows.

Cons: Limited deep customization options; pricing grows quickly for large-scale/high-frequency use; relatively high cost for small projects; requires some technical expertise for advanced customization.

Nimble

Nimble (Nimbleway) [294] is a no-code, API-based solution for scraping any public website, including social media, backed by AI-powered residential proxies. The platform offers both pay-as-you-go and subscription plans starting at \$150 per month with free trial options.

While primarily a general-purpose Web API, Nimble’s service supports data extraction from virtually any accessible public URL, including social media, e-commerce, SERPs.

Key features include AI-optimized residential proxies with geotargeting options, real-time data collection and delivering, unlimited concurrency. The Web API handles batching, automated parsing, AI-driven fingerprinting, and flexible delivery to cloud storage. A dashboard, admin API, parsing templates, and integrations are also provided.

Pro: Supports multiple major social media platforms; flexible payment options; API and no-code interfaces; automatic proxy rotation; high reliability and scalability; enterprise-grade stability and performance; anti-bot and CAPTCHA bypass features. **Cons:** Pricing can be expensive, especially for small projects; requires some technical expertise for advanced customization.

SOAX

SOAX [295] is a proxy-based scraping solution offering residential, mobile, datacenter, and ISP proxies for web scraping, market research, and bypassing geo-restrictions. Among its available scraping tools, there is a specialized Social Media API. The service offers monthly plans based on traffic, starting from approximately \$90, as well as a pay-as-you-go option. A three-day free trial is also available.

The Social Media API supports scraping from major platforms such as Facebook, Instagram, Telegram, YouTube, WeChat, Snapchat, and LinkedIn. Additional coverage includes Amazon, eBay, and other eCommerce sites. SOAX can extract posts, profiles, and searches from supported platforms, maintaining high success rates even for geo-restricted or heavily guarded targets.

SOAX provides precise filtering options, combined with geographically distributed gateway servers that automatically route requests to proxies. Multiple rotation settings are supported. Sticky sessions allow an IP to remain bound for the full session duration, even if the source device goes offline. The service supports SOCKS5, built-in proxy management, and pagination handling, with the ability to set a max-page parameter.

SOAX maintains an extensive pool of residential and mobile IPs, ensuring high success rates.

Pro: Supports multiple major social media platforms; flexible payment options; API and no-code interfaces; automatic proxy rotation, with a large pool of residential and mobile proxies;; high reliability and scalability; anti-bot and CAPTCHA bypass features; 24/7 customer support; user-friendly dashboard and integrations; strong for geo-restricted data access.

Cons: Relatively high cost for small projects; requires some technical expertise for advanced customization; complexity in setup and use of advanced features may be a barrier for non-expert users; lack of dedicated proxies.

ScraperAPI

ScraperAPI [296] is a no-code, API-based web scraping tool designed for general-purpose scraping across various social media platforms in real-time. It does not offer pay-as-you-go pricing; the lowest monthly plan starts at approximately \$49, with a 7-day of free trial.

ScraperAPI supports scraping from major social media platforms including Facebook, TikTok, Instagram, LinkedIn, Reddit, Snapchat, and others. The API is versatile, with official SDKs and libraries for Python, PHP, NodeJS, Ruby, and Java, accompanied by detailed documentation for easy setup. Additional paid features include more precise geotargeting (covering 12 countries), JavaScript rendering for dynamic content handling, and dedicated support.

The tool offers built-in automation and flexible job scheduling. Users can integrate ScraperAPI via multiple methods: proxy server, library/SDK, synchronous API, and asynchronous API for batch requests. However, ScraperAPI does not provide a built-in parser, so users will need to process and structure raw HTML responses themselves. It guarantees automatic IP rotation to prevent bans and ensure continuous scraping, built-in CAPTCHA solving, eliminating the need for third-party solvers.

Pro: Supports multiple major social media platforms; automatic proxy rotation; high reliability and scalability; API and no-code interfaces; 24/7 customer support; anti-bot and CAPTCHA bypass features.

Cons: Requires some technical expertise for advanced customization, but also to use features effectively.

Octoparse

Octoparse [297] is a no-code scraping tool designed for users without programming expertise. It does not offer pay-as-you-go pricing; the lowest monthly plan starts at approximately \$99, with a 14-day free trial available.

Octoparse supports scraping from social media platforms such as X, TikTok, Instagram, and LinkedIn. It features a visual interface with hundreds of ready-made templates for popular websites, including Amazon, eBay, Google Maps, social media platforms, and real estate sites, allowing users to start scraping immediately without manual configuration.

The platform offers automated IP rotation to minimize blocking risks. AI-powered auto-detection assists in identifying data on dynamic and complex websites, and the tool can handle login authentication and automatically solve anti-bot challenges. Octoparse offers 24/7 cloud-based scraping with scheduled scraping tasks and automatic data updates.

On the other hand, this means limited customization and control for advanced scraping needs.

Pro: Supports multiple major social media platforms; automatic proxy rotation; high reliability and scalability; no-code interfaces; anti-bot and CAPTCHA bypass features; intuitive visual interface, ideal for beginners and marketers; hundreds of ready-made templates.

Cons: Pricing can be expensive, especially for small projects; limited advanced customization.

URL inputs problem In the proposed pipeline, data scraping was performed starting from the profile URL of a specific social media account. This choice was motivated by two main reasons. First, most scraping services require the profile URL as an input parameter. Second, even when using scrapers that allowed searching by username alone, the issue of homonyms arose. In a context where the goal is to analyze the content of a social media profile to detect potentially harmful material, any risk of misidentification is unacceptable.

3.2 Data Preprocessing

Before feeding the collected data into the LLM, text preprocessing is performed to transform raw, unstructured text into a cleaner and more consistent format [298, 299].

Standard text cleaning, however, can introduce biases or remove meaningful features. In this research, elements such as the exact spelling of a word, the presence of emojis, hashtags, or similar items carry relevant linguistic and stylistic information.

Therefore, the guiding principle is to minimize irrelevant noise while preserving linguistic cues that are critical for the final analysis, avoiding indiscriminate normalization or removal.

Tokenization Tokenization is the process of splitting text into individual units called tokens, which can be words, subwords, characters, or punctuation marks, depending on the granularity and the tokenization strategy. This step is fundamental as it converts human-readable text into a numerical representation suitable for machine learning models.

Common tokenization approaches include: word-level tokenization, which splits text at word boundaries; subword tokenization, which breaks words into smaller meaningful units (e.g., BPE, WordPiece); character-level tokenization, treating each character as a separate token; and sentence-level tokenization, which uses entire sentences as tokens.

Major LLMs and embedding models automatically perform internal tokenization on plain text prompts using proprietary tokenizers; therefore, manual tokenization is not

required at inference time.

Stemming, Lemmatization, and Stop Word Removal Stemming and lemmatization are normalization techniques used to reduce words to their stem or lemma, while stop word removal discards common functional words.

Although effective in traditional NLP pipelines, these techniques are not applied here because RAG models, like many modern language models, are trained on natural, flowing text. Such models expect complete sentences, including stop words and morphological variations, which carry important contextual information and semantic nuances. Removing them can alter the meaning of sentences, creating a mismatch between the processed text and the model’s expected input.

Lowercasing, and Punctuation Lowercasing reduces vocabulary size but in this study it is avoided, as the embedding model and LLMs exhibit strong robustness to case variations without loss of semantic accuracy.

Similarly, punctuation marks are retained. While often removed in traditional pre-processing, punctuation preserves communicative and stylistic patterns relevant to sociolinguistic analysis.

Contractions, slang, and character normalization Contractions and slang terms, common in social media communication, are preserved. Although they can be expanded using lookup tables, or language models, the embedding model handles both formal and informal variants effectively, making normalization unnecessary and preserving authentic communicative patterns.

Similarly, character repetitions for emphasis are not normalized, as they convey stylistic and emotional cues that can be relevant to the analysis.

Non-ASCII characters are normalized using the proper library.

Emojis and emoticons Emojis are retained because they often convey sentiment, tone, or other semantic information. They are converted into descriptive text tokens using the Python demoji library to preserve their meaning. A generic alternative, when emojis are irrelevant to the downstream task, would be to remove them entirely.

URLs In many NLP tasks, URLs are removed as they do not contribute lexical information. In this research, however, their presence signals that the post refers to external content. Therefore, URLs are replaced with the placeholder [URL], while the original links are reported in the final output. Future work may include analyzing the content of the linked pages.

Remove usernames Usernames (strings starting with @) are replaced with the placeholder @USERNAME to retain the semantic information that a user was mentioned, without introducing noise from unique identifiers.

Hashtags Hashtags are preserved in their original form, including the # symbol, as they often carry contextual or thematic information and facilitate matching with reference documents in the RAG component.

3.3 Text Classification Techniques using LLMs

Using LLMs to classify data is a promising step in NLP. Text classification with LLMs differs from traditional approaches in several ways. While conventional methods often rely on large labeled datasets and fixed features, LLMs can perform classification through natural language prompts and pre-trained knowledge, reducing the need for task-specific labels. LLMs can also quickly adapt to new tasks without retraining. However, LLMs are more computationally intensive, and smaller or task-specific fine-tuned models may outperform LLMs in domain-specific tasks or under strict computational constraints.

Classification Types There are three main types of text classification tasks: binary, multiclass, and multilabel. Binary classification refers to the simplest scenario, where the goal is to categorize data into one of two mutually exclusive groups or categories. Multiclass classification, on the other hand, is used when there are more than two possible categories, but each data instance belongs to only one of them. Finally, Multilabel classification extends this concept to cases where a single instance can be associated with multiple labels simultaneously.

Supervised vs Prompt-Based Classification Text classification with LLMs can be approached in two main ways: supervised fine-tuning and prompt-based classification. Supervised fine-tuning relies on labeled data to train or adapt models to specific tasks. It is most effective when sufficient labeled examples are available. Prompt-based classification leverages LLMs' pre-trained knowledge without modifying model weights. This includes zero-shot and few-shot prompting, making it useful when labeled data is scarce. Prompt-based classification is particularly useful when labeled data is limited, while supervised fine-tuning is preferred when large labeled datasets are available. It is also possible to combine supervised fine-tuning and prompt-based methods. For example, a small dataset can be used to fine-tune an LLM lightly, and then zero- or few-shot prompting can further improve performance on rare or emerging categories.

Methods Requiring Parameter Updates These approaches involve modifying model weights through training on labeled datasets. Full fine-tuning updates all parameters of a pre-trained model, providing maximum adaptation capability but requiring substantial computational resources. Parameter-efficient fine-tuning methods offer a more efficient alternative by freezing pre-trained weights and introducing trainable components. LoRA (Low-Rank Adaptation) exemplifies this approach by adding low-rank matrices to each layer, maintaining model performance while dramatically reducing memory and computational requirements. Other techniques include adapters, prompt tuning, and prefix tuning, all designed to achieve effective task adaptation with fewer trainable parameters.

The choice between full and parameter-efficient fine-tuning depends on factors such as available computational resources, dataset size, required performance, and deployment constraints.

Methods Without Parameter Updates These approaches leverage pre-trained models directly through prompting without modifying their weights. Zero-shot prompting relies entirely on the model’s pre-trained knowledge, providing only class labels (and optionally brief descriptions) to guide classification decisions. This method requires no training data but depends heavily on the model’s existing capabilities and how well the task aligns with its training distribution. In-context learning enhances this by including labeled examples directly within the prompt, enabling the model to adapt its reasoning based on provided demonstrations. This few-shot approach typically improves classification accuracy compared to zero-shot methods, though performance depends on example quality, selection, and the model’s context length limitations.

3.4 The Cheshire Cat AI Framework

In the field of AI, the **Cheshire Cat** framework represents an innovative solution for the advanced orchestration of LLMs. Developed by Piero Savastano [300], this open-source micro-framework is designed to be model-agnostic, modular, and production-ready, enabling the seamless integration of multiple LLMs with a strong emphasis on flexibility, efficiency, and privacy.

The framework implements advanced RAG techniques that ground the LLM responses on external sources of information through semantic search, retrieving relevant and up-to-date content from books, articles, websites, and databases. This approach significantly enhances response accuracy, reduces the risk of generating erroneous content by grounding outputs on verified knowledge bases, and enables access to the latest information beyond the model’s training cutoff (as detailed in Section 2.3.6).

Once installed and connected to a chosen LLM, Cheshire Cat allows developers to build, orchestrate, and deploy intelligent AI agents, which can then be accessed via APIs delivering responses generated by the model. Its interoperable architecture enables selecting the most suitable model for specific use cases, improving performance and adaptability.

The decision to adopt the Cheshire Cat framework in this work was guided by several key features that align with both technical requirements and broader design principles.

Its highly modular architecture, based on Python plugins, allows clean separation of concerns and easy extension, enabling tailored AI agent behavior and fine-grained control and high customization. Despite its advanced capabilities, the framework offers a user-friendly interface that simplifies much of the underlying complexity, making it accessible to users with varying levels of expertise.

Fully containerized with Docker, Cheshire Cat simplifies setup process and ensures consistency across environments. This makes it easy to deploy agents locally or in cloud-based production systems. Designed with scalability in mind, the framework supports everything from small side projects to large-scale enterprise deployments. Developers can

get started quickly by cloning the repository and following the official documentation for local setup.

Its open-source nature fosters transparency, trust, and community contributions, accelerating improvements and bug fixes.

Crucially, Cheshire Cat supports running AI agents entirely on the user’s machine, avoiding the transmission of sensitive data to external servers, a privacy-focused design vital to this work’s ethical and security requirements.

The framework’s versatility is proven by its application across diverse domains, from personal productivity tools to enterprise solutions. It is supported by an active Discord community, including core developers, providing direct assistance alongside comprehensive tutorials and documentation that ease adoption.

3.4.1 Cheshire Cat Architecture

Language Models A significant part of the Cheshire Cat framework revolves around LLMs. By default, the framework includes two main classes of models, each serving a distinct purpose.

The first is the completion model, commonly referred to as the LLM. It takes a string as input and generates a coherent response through text completion. The quality and relevance of its answers are enhanced by leveraging the context from previous conversations and documents stored in the Cat’s memory.

The second is the embedding model, or Embedder, which also takes a string as input but returns a vector representation, known as an embedding. This embedding encodes the semantic content of the input in a condensed, numerical form within a Euclidean vector space, enabling geometric operations such as semantic similarity measurement between texts.

The cosine similarity between two vectors, x and y , is measured by calculating the cosine of the angle between them. The smaller the angle, the more similar the vectors are. Although visualizations often simplify this to two dimensions, embeddings exist in an n -dimensional vector space where this concept remains mathematically valid.

Since these components form the core of the chatbot, the system’s overall performance heavily depends on the quality and configuration of the LLM and the Embedder.

A key advantage of Cheshire Cat is its model-agnostic design, which means that it is not tied to any specific language model or embedder. Instead, it supports integration with a wide range of models, allowing users to select options that best fit their technical requirements or budget.

Currently, the framework supports:

- LLMs: OpenAI, Azure OpenAI, Cohere, Ollama, Hugging Face Text Inference API, Anthropic, Google Gemini, GroqAI
- Embedders: Qdrant FastEmbed, OpenAI, Azure OpenAI, Cohere, Ollama, Google Gemini, Dumb embedder

The system is also highly extensible: additional models can be integrated through plugin development, offering flexibility across various use cases.

When configuring an LLM, several key parameters can be also adjusted to control its behavior.

- **Max Completion Tokens** defines the maximum number of tokens the model is allowed to generate in a single response. This sets an upper limit on output length and helps balance completeness, conciseness, and resource consumption [301].
- **Temperature** controls the randomness in the model's next-token selection process and creativity of the output. Values typically range from 0 to 1 (or slightly above in some implementations). Low temperatures, from 0.0 to 0.3, produce deterministic, focused, and predictable outputs, choosing words with higher probability. This is ideal for tasks that require accuracy and consistency, such as technical writing, legal drafting, code generation, structured data extraction, factual Q&A, and customer support. Medium-low values, from 0.3 to 0.5, works well for summarization, report writing, and general-purpose conversation. Medium-high temperatures, from 0.6 to 0.8, allows for more variation and creativity, suitable for brainstorming, idea generation, or drafting multiple alternatives. High temperatures values, from 0.7 to 1.0 or more, increase randomness and diversity, enabling creative, imaginative, or unexpected outputs. This is especially useful in creative writing, poetry, marketing content, or exploratory ideation [301].

Setting the temperature to 0 causes the model to use greedy decoding, selecting at each step the next token with the highest probability based solely on the model's predicted distribution. This approach removes intentional randomness from the generation process. In theory, if both the model and the input remain unchanged, the output should be identical every time. However, in practice, temperature of 0 does not guarantee absolute determinism: minor variability may still occur if multiple tokens share the exact highest probability. In summary, setting temperature = 0 greatly increases determinism but does not guarantee identical outputs in every run. Although the exact behavior of the model at temperature = 0 can be somewhat ambiguous, it is generally recommended to use greedy decoding to ensure consistency. Additionally, it is suggested to set Top p to 1 in order to consider the full probability distribution, unless the API documentation specifies that this parameter is ignored when temperature is equal to 0 [302].

- **Top P**, also known as nucleus sampling, is a parameter that controls the range of tokens the model considers at each step of text generation. Unlike temperature, which directly influences the randomness of outputs, Top P sets a cumulative probability threshold (typically between 0 and 1) that limits the candidate tokens the model can choose from: at each generation step, the model sorts all possible next tokens by probability and selects only the top tokens whose cumulative probability meets or exceeds the threshold. This approach allows for a more dynamic and context-sensitive selection, helping to balance coherence and diversity in the output.

By limiting the token pool to only the most probable words that collectively meet the specified probability cutoff, Top P influences not only the choice of words and the overall structure of the generated text, but also the tone, style, and contextual

relevance, which is the model’s ability to stay on topic while introducing new, meaningful information. It also affects consistency, especially in extended generations, by helping the model maintain a uniform tone, style, and logical flow.

A low Top P value, around 0.1 and 0.3, limits the model to a small set of highly probable, familiar and repetitive tokens and structures, resulting in predictable and focused responses. The style tends to be more formal and standardized, using familiar vocabulary and straightforward sentence construction. The tone tends to be more serious, academic and detached. Regarding contextual relevance, the models tend to keep the text more focused on the central theme or topic. This is ideal for technical documentation, legal writing, code generation, and other use cases that prioritize clarity, accuracy, and consistency. In ambiguous contexts, a low Top P encourages the model to stick to the most likely interpretation, helping maintain precision but possibly limiting nuance.

On the other hand, a high Top P value, typically from 0.7 to 0.9, allows the model to consider a broader set of vocabulary, including less common or more context-specific options. This opens up greater linguistic variety, more complex sentence structures with varied lengths, and expressive language choices such as idioms, figurative expressions, or uncommon phrasing. The result is more creative, natural, and conversational output, well-suited for tasks like storytelling, poetry, brainstorming, or marketing content. In situations with ambiguous meaning, a higher Top P enables the model to explore alternative, yet still relevant, interpretations.

A moderate Top P, between 0.4 and 0.6, strikes a balance between precision and creativity, common and less common words, moderate sentence length and complexity, formal and casual tone. It keeps the text coherent while allowing for some stylistic and structural variation and occasional expressive language. This range works well for general-purpose writing, report generation, summarization, and most conversational AI interactions, where adaptability and readability are equally important.

Finally, the effect of Top P on contextual relevance can vary with the length of the generated output. For shorter texts, lower values help maintain focus. For longer ones, slightly higher values reduce repetition and allow for natural topic progression [301].

- **Streaming:** in the context of LLM, controls how the model’s output is delivered to the client. When streaming is enabled, the model returns output tokens incrementally as they are generated, rather than all at once. This is useful for real-time applications where responsiveness is a priority. When streaming is disabled, the model internally completes the entire response before returning it to the client. The user receives the full output at once, which can be preferable for applications that require complete responses for further processing, analysis, or validation [303].

Similarly, when configuring an Embedder, it’s necessary to define specific parameters that determine how text is processed and converted into vector representations. Examples are: the maximum number of tokens the model can process per input text, size of the output vector representation, intended use case that optimizes the embedding for specific tasks.

Vector Database The vector database is essential for storing embeddings and efficiently retrieving semantically relevant information. Cheshire Cat uses Qdrant, an open-source vector database specifically designed for AI applications. Qdrant employs the Hierarchical Navigable Small World (HNSW) algorithm to perform approximate nearest neighbor (ANN) searches for similar vectors.

Chat The **Chat** component is the graphical user interface (GUI) enabling direct interaction with Cheshire Cat. It also allows users to select the language model the Cat will run.

Rabbit Hole The Rabbit Hole module is responsible for ingesting new documents and storing them in declarative memory. This functionality allows Cheshire Cat to continuously update its knowledge base with fresh and relevant information. It supports a wide range of file formats, including .html, .md, .txt, .pdf, and even entire websites.

Data extraction is performed by dedicated parsers, which process document content into manageable chunks. These chunks are then vectorized by the embedder and stored.

Users can customize the Rabbit Hole process through plugins. For example, it is possible to replace the default text splitter to modify the chunk size or overlap, or to change or add parsers in order to alter how information is analyzed or to support file types with different extensions.

All stored information can later be retrieved and injected into the prompt during user query response generation.

Long-Term Memory (LTM) Cheshire Cat's Long-Term Memory is implemented using a vector database and stores and manages information persistently across sessions. It is accessed to retrieve context for the LLM and is updated based on the LLM's responses to maintain conversation continuity and adaptability.

The LTM is organized into three primary components. The first one is the Episodic Memory component, which retains traces of past user interactions with the system, enabling recall of previous conversations for coherent, personalized dialogue. The second one is the Declarative Memory, which stores the content of documents uploaded by users, allowing rapid access to relevant factual information. The last one is the Procedural Memory, which contains definitions of Python functions and other instructions related to how Cheshire Cat performs various operations, enabling plugin execution and complex task handling.

Working Memory In addition to LTM, Cheshire Cat also has a Working Memory that manages temporary data relevant during an active session. This memory facilitates information sharing among different plugins or functions and maintains a transient context that influences response generation.

By default, the Working Memory includes the chat history incorporated into the Main Prompt and gathers relevant context from episodic, declarative, and procedural memories. The three most recent question-answer pairs are especially used to enrich prompts during response generation.

StrayCat At the core of Cheshire Cat lies the Cat object, an instance of the StrayCat class that serves as the primary interface to the framework’s functionalities.

It orchestrates interactions across internal modules by encapsulating session state, managing working memory, and providing context-aware access to the system’s capabilities. Through this object, developers can facilitate conversations, handle user sessions, interact with LLMs, manage WebSocket communications, and more.

Designed as a centralized and intuitive interface, the Cat object exposes high-level methods and convenient shortcuts to access deeply nested or user-specific structures. This significantly simplifies the development process, reducing the need for repetitive imports or boilerplate code.

Mad Hatter The Mad Hatter is an essential component that manages the plugins that extend or modify Cheshire Cat’s functionalities. Mad Hatter not only loads and organizes the plugins but also controls their execution according to a defined priority, ensuring that the most critical operations take precedence within the system. This plugin manager is vital for maintaining order and efficiency in the execution of software extensions that enrich Cheshire Cat capabilities.

White Rabbit The White Rabbit is the component responsible for scheduling computations to be executed at a specific time, including one-off events and recurring tasks via cron jobs.

Agent Manager The Agent Manager coordinates the execution of language model chains, pipelines that format input variables into prompts, submit them to LLMs, and optionally parse outputs.

The Agent Manager oversees two main chains: the procedures chain, used by the Procedures Agent, and the memory chain. If the system identifies useful information within the procedural memory, the Agent Manager delegates execution to the Procedures Agent via the procedures chain. The Procedures Agent attempts to run a specific procedure to address the request. If the result sufficiently answers the user, it is returned immediately.

If no appropriate procedure is found or the result is inadequate, the Agent Manager activates the memory chain, which generates a response using context retrieved from declarative and episodic memories.

3.4.2 Message

The primary function of a chatbot is to answer users’ questions. To carry out this task, the POST /message endpoint is used.

Custom Data While models such as ChatGPT offer extensive general knowledge, they have limitations in handling highly specific, contextual, or up-to-date information. The Cheshire Cat framework addresses this by implementing the RAG technique, previously introduced in Section 2.3.6.

Thanks to the modular architecture described earlier, users can upload text files, PDF documents, or entire websites via the Rabbit Hole component. The system parses and splits the content into chunks, converts them into embeddings, and stores them in Vector Memory Collections within the Qdrant vector database, which are created when the Cat is installed or after a complete memory swap.

These collections represent the lowest-level structure in the Long-Term Memory and enable efficient semantic similarity searches using cosine distance. The most relevant text fragments are then provided to the language model, enhancing the contextual quality and specificity of the generated answers.

Prompts When the memory chain is triggered, the Agent Manager constructs the Main Prompt, which guides the model’s response behavior. This prompt is composed of a prefix, containing general instructions about the Cat’s identity and behavior, and a suffix, which includes variables such as episodic memory, declarative memory, chat history, and the user’s current message.

Another key prompt is the Procedures Prompt, used by the Procedures Agent to determine whether a tool from procedural memory should be executed and which one is most suitable to fulfill the user’s needs.

Steps Once the user’s message is received, it is stored in the Working Memory and converted into a vector embedding, which is saved in the Episodic Memory. The system then retrieves relevant context from the Procedural, Episodic, and Declarative memories based on semantic similarity and the active plugin configuration.

If a relevant tool is found in Procedural Memory, the Procedures Agent is activated. A dedicated prompt is generated, allowing the language model to select and execute the appropriate function. If the tool is self-contained the output is returned directly to the user without invoking the LLM for further generation.

If no suitable tool is found, or the tool does not fully resolve the query, the Memory Chain is triggered. It constructs the Main Prompt and sends it to the LLM, which generates a response based on the retrieved context. Afterward, a “why” dictionary is created to record which data sources contributed to the final output, supporting transparency and traceability.

Plugins can alter the default behavior of the prompt construction process. For example, they may customize the prefix or suffix of prompts, inject actions before or after agent execution, or generate an immediate reply before the main processing pipeline starts.

3.4.3 Plugin

Plugins and hooks are core to Cheshire Cat’s customization capabilities. They enable developers to adapt the framework to specific needs or integrate it with external systems, all without modifying the core codebase.

Cheshire Cat supports a public plugin registry that allows administrators to download, install, and share community-developed plugins. Examples of plugin capabilities include adding support for new file formats in the Rabbit Hole, enabling online content retrieval, integrating smart home services, or extending compatibility with additional LLMs.

Plugins are modular Python components that provide a structured and safe way to extend the platform. The growing developer community actively contributes to this ecosystem, making it easier to implement new functionalities and tailor Cheshire Cat to diverse application scenarios.

Hooks Hooks are predefined functions in Cheshire Cat’s internal workflow. They allow developers to intervene in specific stages of execution, respond to system events, or alter data flows. Hooks are essential for implementing fine-grained customization of the system’s behavior and interactions.

Tools Within a plugin, a tool is a function that can be invoked by the LLM. Its inputs are designed to be generated by the model itself, and its output is designed to be used later by the model.

Each tool must include a descriptive docstring that explains when and why the function should be used. This description is critical, as it is embedded into the tool’s instruction prompt, enabling the LLM to determine whether to call the function based on user intent. In Cheshire Cat, the docstring is also embedded into the Procedural Memory, where it becomes part of the tool discovery process.

Form A form is a specialized tool used to collect structured information from the user. Forms operate through a state machine that guides the chatbot’s interaction until all required information is gathered. Activation and termination of a form are controlled via designated phrases, similar to tool docstrings. Developers can also customize the form’s behavior by extending and overriding methods in its base class.

3.5 Large Language Models and Embedding Models

As described in paragraph 3.4.1, the Cheshire Cat framework relies on two main types of models: LLM and Embedder. In this section, we detail the specific models used in this work and the reasoning behind their selection.

3.5.1 Large Language Model

Two different LLMs were evaluated in the context of this project: **Llama 3.1 8B Instruct** and **GPT-4.1**. The purpose was not merely to compare two products, but rather to explore the trade-offs between two fundamentally different approaches to language modeling: Meta’s open-source, locally deployable ecosystem, and OpenAI’s proprietary, cloud-based, state-of-the-art solution.

The Llama 3.1 8B Instruct model was selected because of its open-source nature and ability to run locally, offering critical advantages in terms of privacy and control. Local deployment ensures that no data is sent to third-party servers, an important consideration when dealing with sensitive information.

In contrast, GPT-4.1 was used as a performance benchmark. As one of the most advanced language models currently available, it offers superior reasoning, factual accuracy,

and contextual understanding. Larger models typically perform better in RAG tasks, making GPT-4.1 an ideal reference point to evaluate the strengths and limitations of lighter, locally executable models like Llama 3.1 8B Instruct.

Llama 3.1 8B Instruct

Model Description Released in July 2024, Llama 3.1 8B is part of Meta’s third-generation LLM family, available in 8B, 70B, and 405B parameter configurations. Optimized for multilingual instruction-based use cases, the Instruct variants are fine-tuned for efficient, context-aware dialogue and task execution.

Compared to previous versions, Llama 3.1 introduces significant improvements, including an expanded context window of up to 128K token, allowing it to handle long-form texts and complex conversations more effectively. The model incorporates advanced techniques such as Grouped-Query Attention (GQA) to improve inference scalability, Rotary Position Embedding (RoPE) for positional encoding, the SiLU (Swish) activation function, and RMSNorm for layer normalization. Additionally, it utilizes Flash Attention to speed up computation during text generation.

The model was trained on 15 trillion tokens from public data and subsequently refined through supervised fine-tuning (SFT) and reinforcement learning with human feedback (RLHF), ensuring helpful and aligned outputs.

The 8B variant balances performance and computational efficiency, making it suitable for environments with limited resources. It excels in tasks such as text summarization, classification, sentiment analysis, automated data extraction and analysis and complex dialogue management. It is also perfect for applications requiring instant responses and high throughput, such as real-time content moderation and filtering, interactive educational tools and tutoring systems, and dynamic content generation. It also supports advanced use cases like synthetic data generation, long-form summarization, and model distillation.

Multilingual by design, Llama 3.1 supports English, German, French, Italian, Portuguese, Hindi, Spanish, and Thai, making it ideal for global applications including virtual assistants and multilingual chatbots.

In terms of hardware, the model can run on modern CPUs with at least eight cores and requires a minimum of 16 GB of system RAM. For optimal performance, especially during inference, it is recommended to use an NVIDIA RTX 3090/4090 series or later GPU, with at least 16 GB of VRAM. Quantized versions of the model can operate with as little as 12 GB of VRAM [304–307].

Motivation for the Choice The primary reason for selecting an open-source model such as Llama 3.1 8B Instruct is privacy. Unlike cloud-based models, it can be deployed entirely on-premise, ensuring that sensitive data remains within organizational boundaries, an essential requirement where security and confidentiality are critical. Although providers like OpenAI implement stringent privacy policies, the fact that data passes through external systems may still pose compliance concerns [308].

Beyond privacy, open-source models offer several additional advantages. One significant benefit is flexibility. Organizations can fine-tune and adapt the model to specific

needs, improving relevance and performance [309,310].

Cost efficiency is another important factor. Open-source models generally require lower upfront and ongoing costs, as they do not require expensive licenses or subscriptions. This allows organizations to allocate resources more effectively, investing in improved hardware or expanding technical expertise [309,310].

Finally, open-source models provide greater control and facilitate compliance. Organizations can audit the code, identify potential vulnerabilities, and implement tailored security measures. This degree of control is particularly crucial in regulated industries [309,310].

From a technical perspective, Llama 3.1 is well-suited for RAG systems due to its extended context window and strong multilingual capabilities. It has been widely adopted in recent research for RAG tasks, where its reasoning ability and long-context comprehension offer significant advantages over proprietary alternatives like GPT-4o and Claude 3.5 [311].

Benchmark results reinforce its effectiveness. In the Needle-in-a-Haystack (NIH) evaluation, which tests a model’s ability to find specific pieces of information in large volumes of text, Llama 3.1 achieves near-perfect retrieval rates, demonstrating its strength in precision-oriented information extraction. It also performs strongly in the Multi-Needle benchmark, accurately retrieving multiple information points from long documents [311].

The choice of the 8B variant was strategic: it is optimized for speed, efficiency, and rapid iteration. This makes it ideal for prototyping and for deployments in environments with limited compute resources [309].

To further reduce infrastructure demands, the model was accessed via the Groq AI platform, enabling high-speed inference without requiring dedicated local hardware. Groq’s low-latency execution and cost-effectiveness made it a practical solution for experimentation within the scope of this project.

GPT-4.1

Model Description Released in April 2025, GPT-4.1 is part of OpenAI’s fourth-generation LLM family, following the success of GPT-4 and GPT-4o. This new generation of models includes three variants, GPT-4.1, GPT-4.1 mini, and GPT-4.1 nano, designed to meet a wide range of different needs in terms of power, speed, and costs, offering flexibility and scalability for both businesses and independent developers.

Although OpenAI has not disclosed the internal architecture of GPT-4.1, it is known that GPT-4.1 features major improvements, compared to previous versions, in instruction following, reasoning over long contexts, and coding accuracy, positioning it as a robust solution for both general-purpose and domain-specific tasks, in terms of speed, latency, and cost-efficiency.

One of the defining characteristics of GPT-4.1 is the extended context window, up to 1 million tokens via the API. This long-context capability enables the model to process entire books, legal documents, software repositories, complex conversations and detailed reasoning tasks. GPT-4.1 is also multimodal, supporting both text and image inputs, and can perform tasks across legal, coding, customer support, and many other domains.

The model was trained on a large-scale, diverse corpus of publicly available and licensed data across multiple modalities. Fine-tuning through supervised learning and reinforcement learning with human feedback (RLHF) ensures that GPT-4.1 produces helpful, safe, and aligned outputs across various contexts.

Thanks to its improved instruction adherence and contextual reasoning, GPT-4.1 is particularly well-suited for autonomous agent systems capable of completing complex tasks with minimal human intervention. When integrated with tools such as the OpenAI Functions API or the new Responses API, GPT-4.1 can power advanced agents for code synthesis, document analysis, software testing, customer service automation, and real-world decision support.

Multilingual by design, GPT-4.1 supports use of languages including English, German, French, Spanish, Italian, Portuguese, Chinese, Japanese, Arabic, and many others, making it ideal for international applications such as multilingual chatbots, global customer support platforms, and cross-language content generation.

GPT-4.1 excels at a wide array of use cases, including but not limited to: advanced code generation and debugging, document summarization, knowledge extraction from long-form content, educational tutoring systems, intelligent virtual assistants, and creative content generation.

Unlike open-source models, GPT-4.1 is not locally deployable and operates exclusively through OpenAI’s cloud infrastructure. This enables developers to benefit from scalable performance and managed resource allocation, but also introduces trade-offs related to data privacy, vendor lock-in, and long-term cost. For organizations with strict compliance or sovereignty requirements, these factors may limit the model’s applicability in sensitive or regulated domains. It is accessible via OpenAI’s API under a tiered pricing model, with options for integration, fine-tuning, and deployment across a wide variety of workflows [312–314].

Motivation for the Choice Despite the privacy limitations inherent in cloud-based proprietary models, a solution from OpenAI was selected due to its exceptional performance and reliability in complex language tasks. While local deployment was not possible, the benefits in terms of reasoning accuracy, contextual understanding, and robustness made this trade-off acceptable for the experimental scope of the project.

Among the OpenAI models available, GPT-4.1 was chosen specifically because it represents one of the most recent and advanced releases, designed to handle long-context reasoning and instruction-heavy tasks. Its extended context window, make GPT-4.1 a strong choice for RAG when working with large and complex documents. With the ability to process large text segments, GPT-4.1 can reason over dense, information-rich content while preserving connections across paragraphs and even pages. This is especially valuable for use cases involving technical reports, academic publications, or industry whitepapers, where understanding depends on multi-page context. Thanks to its improvements, GPT-4.1 is able to support deep nuanced questions about long documents and provide precise and well-informed answers.

3.5.2 Embedding Model

Proprietary embedding models are widely adopted in RAG pipelines due to their high performance. However, they introduce significant drawbacks, including recurring costs, reliance on third-party APIs, and potential data privacy concerns.

In contrast, open-source embedding models offer a cost-effective and customizable alternative. Running these models locally allows complete control over the embedding process, eliminates usage fees, enhances data privacy, and tailor the models to specific needs.

Nomic Embed Text v1.5

Model Description Given these considerations, a local embedding model was adopted using the FastEmbed library developed by Qdrant.

FastEmbed is a lightweight, fast and accurate Python library designed for local embedding generation. It supports several popular open-source text models and is optimized for environments without GPU support, thanks to its use of the ONNX Runtime instead of PyTorch.

To use FastEmbed within the Cheshire Cat AI framework, it is necessary to specify the identifier of the embedding model to be downloaded, the maximum number of tokens per input, and the type of document to embed, distinguishing between “query” or “passage”, a crucial distinction in retrieval-based tasks. The “query” prefix often triggers the model to generate embeddings that are optimized for similarity comparisons, while “passage” embeddings are fine-tuned for contextual understanding. In RAG scenario, the best practice is using “query” for queries and “passage” for documents.

Qdrant serves as a Vector Store, offering a scalable and efficient solution for modern machine learning and AI applications. It is built to handle billions of vectors, ensure low-latency retrieval, and support advanced quantization methods, all while maintaining high performance [315].

Among the embedding models supported by FastEmbed, the model selected for this work is nomic-ai/nomic-embed-text-v1.5, which outperforms OpenAI’s text-embedding-ada-002 and text-embedding-3-small models on both short and long-context text embedding tasks. This is a fully open-source, open-data, open training code, reproducible and auditable 8k context length text embedding model developed and trained by Nomic AI. It represents an improvement of over the previous v1 version through the use of Matryoshka Representation Learning, which gives developers the flexibility to trade off the embedding size for a negligible reduction in performance. The model can generate embeddings with varying dimensionality, ranging from 64 to 768 dimensions, allowing users to choose the size that best fits their use case.

This model excels at generating high-quality text embeddings that capture semantic meaning and contextual nuances. These embeddings are well-suited for a variety of downstream NLP tasks, such as text similarity, document retrieval, and text classification.

To achieve this level of performance, the embedder is trained using a multi-stage training pipeline. It begins with a long-context BERT model and undergoes an initial

unsupervised contrastive training phase using weakly related text pairs, such as question-answer pairs from forums like StackExchange and Quora, title-body pairs from Amazon reviews, and summarizations from news articles. In a second, supervised finetuning stage, the model is trained on higher quality labeled datasets, such as search queries and answers from web searches. This stage emphasizes careful data curation and hard example mining, which are crucial for refining the model’s ability to generate discriminative embeddings.

The flexible embedding size makes this model suitable for a wide range of applications, such as: semantic search, retrieving relevant content based on query embeddings; suggesting content or products based on embedding similarity; combining text embeddings with image or audio features for cross-modal retrieval tasks.

The Nomic Embed model supports four task types, specified via the `task_type` parameter: `search_document`, used to embed documents for retrieval or similarity search; `search_query`, used to embed query for retrieval tasks; `classification`, used in classification models, and `clustering`, used for high linear separability and clustering [316,317].

Motivation for the Choice This model was chosen specifically because it is fully open-source, ensuring greater transparency and reproducibility, while also offering strong guarantees in terms of data privacy. Moreover, its local executability eliminates API usage costs and provides the flexibility needed to adapt the embedding process to the specific requirements of this project.

Recent benchmarks comparing proprietary and open-source embedding models have shown that while OpenAI’s large model achieves the highest accuracy, open-source alternatives remain competitive. Notably, `nomic-embed-text` reached 71% accuracy with 768 dimensional embeddings, performing closely to BGE large, which reached 71.5% at 1,024 dimensions. These results highlight `nomic-embed-text` as a strong open-source option, balancing accuracy with computational efficiency [318].

Among the various open-source embedding models, `nomic-embed-text` was also selected due to its superior ability to handle long-context inputs and generate high-quality embeddings for detailed documents [319].

Its adoption by both the Cheshire Cat community and in various RAG-related research further supports its selection for this work.

Chapter 4

Legal and Ethical Considerations

This chapter aims to explore the complex legal and ethical landscape surrounding the development and deployment of a framework such as the one proposed, covering the collection, processing, and analysis of social media data for intelligence and automated security reporting purposes.

The legal and ethical analysis encompasses several critical dimensions: the classification and legal status of information shared on social media platforms; the regulatory frameworks governing data collection through scraping; and the application of artificial intelligence in security and intelligence contexts.

These dimensions operate within overlapping but distinct legal regimes, forming a complex regulatory matrix that must be navigated with caution and precision.

4.1 The Legal Status of Social Media Data

Social media platforms represent one of the most significant sources of personal information in the contemporary digital landscape. By their very nature, these platforms encourage users to share extensive personal details, ranging from basic identifiers such as names, locations, and demographic details, to more complex behavioral patterns, political preferences, social relationships, and opinions.

4.1.1 Personal Data

Under the General Data Protection Regulation (GDPR) framework, the European Union regulation focused on the protection of personal data and the free movement of such data, social media content consistently qualifies as “personal data”, as defined in Article 4(1), which covers “any information relating to an identified or identifiable natural person”.

While the GDPR stands as one of the most comprehensive and influential data protection frameworks globally, it is not the only one. Similar regulations exist in other jurisdictions, such as the Personal Information Protection and Electronic Documents Act (PIPEDA) in Canada, the Lei Geral de Proteção de Dados (LGPD) in Brazil, and various state-level laws in the United States, most notably the California Consumer Privacy Act (CCPA). This list is not exhaustive, as many other countries and regions have adopted

or are developing their own data protection laws inspired by or aligned with GDPR principles.

The traditional assumption that publicly available information is subject to lower levels of legal protection has been fundamentally challenged by contemporary regulatory approaches. Under the GDPR, no distinction is made between data obtained from public or private sources: all personal data is protected regardless of its origin or public accessibility. However, it is important to observe that this approach is not universally adopted.

4.1.2 Special Categories of Personal Data

Of particular significance in the context of social media is the frequent presence of “special categories of personal data”, as defined in Article 9(1) of the GDPR. These include data revealing racial or ethnic origin, political opinions, religious or philosophical beliefs, trade union membership, genetic and biometric data for unique identification purposes, as well as data concerning health, sex life, or sexual orientation. Social media platforms serve as natural repositories for such sensitive information, even from apparently neutral content.

In this context, the types of information that are most relevant and sensitive within the proposed framework fall within these special categories, thereby raising enhanced legal and ethical considerations.

4.1.3 Legal Basis for Processing Personal Data and Special Categories of Personal Data

In the context of the GDPR, the processing of personal data is only lawful where it has what is known as a legal basis.

The Article 6 of the GDPR establishes six potential legal basis for personal data processing, each with distinct requirements and limitations, including: the data subject has given the consent to the processing for one or more specific purposes; processing is necessary for the performance of a contract to which the data subject is party, or in order to take steps at the request of the data subject prior to entering into a contract; processing is necessary for compliance with a legal obligation to which the controller is subject; processing is necessary to protect the vital interests of the data subject or another natural person; processing is necessary for the performance of a task carried out in the public interest or in the exercise of official authority vested in the controller; processing is necessary for the purposes of the legitimate interests pursued by the controller or by a third party, except where such interests are overridden by the data subject’s fundamental rights and freedoms.

When processing special categories of personal data, as defined in Article 9(1) of the GDPR, a more stringent legal basis is required. Article 9(2) provides a list of conditions under which such processing may be lawful, including: the data subject has given explicit consent; the data has been manifestly made public by the data subject; processing is necessary for obligations and rights in the field of employment, social security, and social protection law; processing is necessary to protect the vital interests of the data subject or another person, where the data subject is physically or legally incapable of

giving consent; processing is necessary for the establishment, exercise, or defense of legal claims; processing is necessary for reasons of substantial public interest, based on Union or Member State law, provided that such law is proportionate, respects the essence of the right to data protection, and provides for appropriate safeguards; processing is necessary for preventive or occupational medicine, medical diagnosis, provision of health or social care or treatment, or the management of health or social care systems; processing is necessary for reasons of public interest in the area of public health; processing is necessary for archiving purposes in the public interest, scientific or historical research purposes, or statistical purposes, subject to the safeguards outlined in Article 89, including data minimization, pseudonymization, and storage limitation.

It is important to note that, especially in the context of social media data, the interpretation of “manifestly made public” requires careful analysis. It applies only when the data subject has clearly and deliberately made the information public, with full awareness that it could be accessed by an unlimited audience. The mere availability of personal data on a publicly accessible platform does not automatically meet this criterion.

In general, the choice of an appropriate legal basis should be made carefully and, when necessary, with legal consultation, especially when dealing with special categories of personal data.

Thesis Analysis While the theoretical framework of this thesis could justify the processing of personal and special category data under the legal basis of scientific research purposes, a pragmatic approach has been adopted to streamline the research focus and avoid the requirements of Article 89 GDPR, which mandates data minimization, pseudonymization, and many other specific measures.

To circumvent these requirements and maintain research efficiency, mainly focused on other aspects, this study employs a hybrid data strategy that technically avoids processing actual personal data as defined under GDPR. Specifically, real-time collection of social media data is strictly confined to pipeline functionality testing. When accessing content from personal profiles, explicit consent is obtained from users, ensuring compliance with privacy regulations while enabling technical validation of the data collection infrastructure. The performance of classification algorithms and LLMs is evaluated exclusively on synthetic datasets and publicly available resources, such as those found on platforms like Kaggle, to simulate realistic user profiles.

4.1.4 Intelligence and Law Enforcement Context

Legal systems recognize that sectors such as intelligence, law enforcement, and national security sectors require specialized regulatory frameworks that may diverge from general legislation. This differentiation is designed to balance the protection of individual rights, such as the rights to privacy, data protection and freedom of expression, with public security needs, while ensuring democratic oversight and constitutional safeguards remain intact.

Before delving into the legal aspects of combining social media data with the monitoring and analysis of high-risk content by law enforcement and intelligence agencies, it is crucial

to note the complexity of the situation. Striking a balance between the need to monitor and protect fundamental rights is a particularly challenging task.

On the one hand, regulations such as Terrorist-Content Regulation (TERREG) and Digital Services Act (DSA) encourage the monitoring of social media to combat illegal content and to prevent and control crimes. On the other hand, a lot of complication exists, numerous complications arise, especially regarding the protection of fundamental rights.

The first challenge is the asymmetry between the visibility of online social life and law enforcement and intelligence activities. This asymmetry can create concerns, particularly in extreme scenarios where monitoring activities may overreach, treating every individual as a potential suspect and thereby contributing to the “criminalisation of online spaces”.

One key danger is mass surveillance, where broad, indiscriminate monitoring of individuals occurs without specific suspicion, threatening privacy and freedom of expression. Such practices can create a chilling effect, causing people to avoid expressing themselves freely, particularly in political contexts. This surveillance risks becoming a tool for political repression, with governments potentially using social media data to target opposition or activists, especially in less democratic regions.

The abuse of power is also a concern, as law enforcement might misuse collected data for political manipulation, suppressing dissent or manipulating public opinion. This could undermine democratic values and freedom of speech.

In this context, the necessity and proportionality of monitoring should be evaluated, taking into account the severity of the crimes law enforcement and intelligence agencies aim to prevent. A legislative framework that regulates monitoring activities based on the seriousness of the scenario would help ensure that collection and analysis remains necessary and proportionate.

Similarly, the Court of Justice of the European Union (CJEU) has clarified that monitoring is justifiable only if there’s an indirect link to a serious criminal offense. However, this is not always straightforward. In case of disinformation, for example, the sharing of content may be considered a potential cause of disruption to public order only if the sharing is repeated, rather than based on a single post. Content related to terrorism or hate speech poses similar challenges, as it can represent serious threats but may also infringe upon fundamental rights such as freedom of expression and privacy.

For this reason a case-by-case analysis is required to ensure that data processing is justified and proportionate to the threat.

In the specific context of personal data processing, the GDPR explicitly excludes certain scenarios, as outlined in Article 2: law enforcement activities, activities outside the scope of EU law, or purely personal and household activities. Of particular relevance are the first two exceptions, which are directly related to the proposed framework.

When data processing falls within the scope of law enforcement, the applicable legal instrument is the Data Protection Law Enforcement Directive (EU) 2016/680. This Directive governs the processing of personal data by competent authorities for the prevention, investigation, detection, and prosecution of criminal offenses. This framework permits more flexible processing conditions than the GDPR, aiming to balance public

safety with the protection of individual rights.

Under the LED, competent authorities may process personal data only when necessary for performing tasks within their law enforcement purposes. However, such processing must remain proportionate to the pursued aims and respect the essence of fundamental rights. The Directive includes specific provisions for sensitive categories of personal data, requiring that processing be “strictly necessary” and accompanied by appropriate safeguards to protect fundamental rights. Additionally, under Article 8, decisions based solely on automated processing, including profiling, are prohibited unless explicitly authorized by Union law or specific legal provisions.

Key differences between the LED and the GDPR include reduced transparency requirements under the LED, less stringent standards for obtaining consent, and the need to categorize data subjects into specific roles, such as victims, witnesses, suspects, or offenders.

However, the LED does not apply to data processing activities related to national security carried out by bodies or agencies of the European Union. Similarly, the Directive does not cover data processing related to tasks performed by competent authorities that are not directly aimed at preventing, investigating, detecting, or prosecuting criminal offenses, such as safeguarding or preventing public security threats. In these cases, the European Regulation on the protection of personal data, Regulation (EU) 2018/1725, governs the processing.

An example of activity falling outside the scope of the EU law is that carried out by military intelligence services, where national security takes precedence over EU data protection rules such as the GDPR. However, this exclusion does not eliminate fundamental rights protections, which remain safeguarded under the European Convention on Human Rights and national constitutional frameworks.

4.1.5 Data Protection Principles and Implementation Requirements

The GDPR establishes six fundamental principles that govern all personal data processing activities. These principles form the basis for lawful, ethical, and transparent data handling across various sectors, including social media platforms.

Lawfulness, fairness, and transparency principle requires that processing occurs lawfully, fairly, and transparently in relation to data subjects. Fairness implies that processing should align with the reasonable expectations of individuals. For instance, social media users typically do not expect that their publicly shared content will be systematically collected and analyzed for profiling purposes.

Purpose limitation mandates that personal data be collected for specified, explicit, and legitimate purposes and not further processed incompatibly with those purposes. However, exceptions exist for processing activities carried out for scientific or historical research, and for archiving in the public interest, provided that appropriate safeguards are implemented in accordance with Article 89 of the GDPR.

Data minimization requires that data be adequate, relevant, and limited to what is necessary for the intended processing purpose should be collected.

Accuracy demands that data be accurate and, where necessary, kept up to date. Inaccurate data should be corrected or deleted without delay.

Storage limitation restricts data retention to periods necessary for processing purposes. Once these purposes are achieved, the data should be securely deleted or anonymized.

Integrity and confidentiality requires appropriate security measures to protect personal data against unauthorized access, loss, destruction, or damage.

4.2 Legal Risks of Web Scraping

Scraping, specifically in the context of social media platforms, intersects with a wide range of legal and regulatory domains. These platforms operate under private contractual terms, but scraping can conflict with laws governing intellectual property, unauthorized access, computer hacking, and privacy.

When scraping involves collecting personal data, as previously seen, data protection laws apply, requiring transparency, consent, and specific security measures.

Social media scraping remains a constantly evolving problem, shaped by technical, legal, and ethical considerations. The central issue concerns not only what is scraped, but how it is done. For this reason, while the previous section focused on the what, this section turns to the how.

4.2.1 Key Aspects to Consider

Terms of Service and Platform Contracts In the context of scraping, a crucial issue concerns the legal validity of Terms of Service (ToS) that social media platforms use to regulate the use of their services. These agreements typically prohibit scraping and other forms of automated data collection, but their actual legal enforceability depends on how they are presented to users and accepted by them.

Jurisprudence has developed a fundamental distinction between different modes of accepting online contractual terms. Browsewrap agreements presume user consent through simple website usage. However, since these terms are often poorly visible or hidden, courts tend to reject them except when notice is not sufficiently clear and prominent. Conversely, clickwrap agreements require explicit acceptance action, such as clicking a button to accept, and generally enjoy greater legal recognition. An intermediate category is represented by sign-in-wrap agreements, which combine elements of both previous types.

American courts have progressively refined the criteria for determining the validity of these agreements. In the landmark case *Berman v. Freedom Financial Network, LLC* (2022), the Ninth Circuit Court of Appeals established that a contract can be considered binding only when two conditions are simultaneously satisfied: first, the website must provide reasonably conspicuous notice of the contractual terms; second, the user must take unambiguous action manifesting their consent, such as clicking a button or checking a box.

It is important to emphasize that ToS do not automatically constitute law and that, absent a specific contractual violation between parties, establishing civil liability can be complex. Nevertheless, platforms continue to implement detailed restrictions, often justified by security needs and user protection.

Additionally, platforms retain significant enforcement mechanisms beyond legal action. These include technical measures such as IP blocking, rate limiting, and CAPTCHAs, as well as account termination and service denial. The effectiveness of ToS enforcement often depends more on these practical barriers than on legal remedies.

Contemporary jurisprudence also distinguishes between different types of scraped data. While publicly available information generally enjoys broader protection, accessing data that requires authentication or circumventing technical protection measures significantly increases legal risks.

In summary, ToS limitations regarding scraping acquire legal force only when agreements are presented transparently, accepted through valid methods, and formulated with balanced and reasonable terms.

Finally, intelligence teams or law enforcement agencies may be exempt from certain restrictions imposed by ToS, as specific authorizations allow them to access and use data on platforms for investigative or security purposes, even if such actions would otherwise violate the ToS for regular users.

Robots.txt The robots.txt file is a widely used, machine-readable protocol that allows website owners to communicate their preferences regarding automated access to their content. Some companies use robots.txt to permit access only to specific crawlers, while others allow broader crawling but restrict certain sections of the site.

Ignoring robots.txt directives can expose scrapers to legal implications, especially when combined with copyright infringement, unauthorized access, or contractual restrictions. Courts, however, often balance platform interests with fundamental rights. This means the legality of scraping despite robots.txt restrictions can depend on factors such as the purpose of scraping, and whether the data is public or private.

In summary, while robots.txt files provide important signals regarding acceptable automated access, the legal status of ignoring these directives varies depending on jurisdiction, case specifics, and the balance between technological norms and fundamental rights.

From an ethical standpoint, respecting robots.txt directives and rate limits is essential for responsible web scraping, helping to prevent server overloads and behavior that could be considered unfair to website owners.

Logged-in versus logged-out scraping A crucial distinction in web scraping is between data accessible to any visitor without an account, the so called logged-out scraping, and data available only to authenticated users, the logged-in scraping.

When scraping as a logged-out user, the data collected is publicly accessible information that does not require identification or authentication. In contrast, logged-in scraping often provides access to a broader and more sensitive range of data, including content meant only for specific users.

This distinction has important implications, especially legally. Logged-in scraping generally involves agreeing to a platform's ToS, whereas scraping publicly available logged-out data tends to involve fewer legal complications. Additionally, ethical and institutional

considerations around consent, privacy, and data sensitivity often differ significantly between logged-in and logged-out data collection.

Therefore, understanding whether scraping involves logged-in or logged-out access is essential for evaluating the legality, ethics, and research permissions associated with data collection.

Copyright and Ownership of Content A significant amount of online content is legally protected by copyright. This includes music, films, photographs, news articles, blog posts, social media updates, research papers, websites HTML code, images, logos, and digital graphics. By contrast, facts and ideas are not subject to copyright protection.

Since web scraping involves copying content, it may infringe copyright unless done with the author's explicit consent (through a license) or a legal permission. Explicit consent is rarely obtained, and consequently legal exceptions are crucial. Copyright rules vary between jurisdictions, but the EU has established a harmonized framework.

Directive (EU) 2019/790 on Copyright in the Digital Single Market (DSM Directive) introduced specific exceptions for text and data mining (TDM), defined as: any automated analytical technique aimed at analysing text and data in digital form in order to generate information, which includes, but is not limited to, patterns, trends and correlations. The Directive establishes two different TDM exceptions: the first one applied to research organisations, allowing reproductions and extractions of works for TDM purposes when there is lawful access to the content; the second is applied to any users, allowing TDM on lawfully accessible works, but rightholders may expressly reserve their rights, for example via contractual terms, metadata, or technical measures.

In case of intelligence teams or law enforcement agencies, in the course of their investigative or security activities, they may work with copyrighted content without the explicit consent of the rights holders, thanks to specific legal authorizations that allow the acquisition and analysis of data for security or investigative purposes.

Platform Harm Even though many social media platforms implement sophisticated measures to prevent excessive or rapid data extraction, it is good practice to ensure that scraping activities do not overload the servers, by carefully controlling both the frequency and volume of requests.

4.2.2 Opinions and Case Studies

To complement the technical and legal analysis of scraping, this section gathers expert opinions and relevant case studies. These examples demonstrate how legal, ethical, and contractual issues unfold in practice, highlighting the variety of approaches and underscoring the complexity and continuously evolving scenario.

United States: from Uncertainty to Clarification For many years, the legal status of scraping in the U.S. was uncertain, particularly where it conflicted with ToS. Violating ToS was often interpreted as potentially violating the Computer Fraud and Abuse Act (CFAA), a federal anti-hacking law.

This interpretation was clarified by the Supreme Court in *Van Buren v. United States* (2021), which held that mere violations of ToS do not constitute a crime under the CFAA. Following this, the 9th Circuit Court of Appeals, among others, clarified in 2022 that scraping publicly accessible data without bypassing technical protections or causing additional harm generally does not trigger CFAA liability. However, the CFAA can still apply if scraping involves circumventing security measures or accessing data that is explicitly restricted.

European Union: Contractual Focus In the EU, the legal framework depends largely on the nature of the data being collected. Violating ToS is not a crime; however, websites may still bring civil actions for breach of contract or tort, potentially seeking damages or injunctions. Courts may also consider whether scraping involves public or private data, and whether technical protections were bypassed.

Case Law Involving Meta Meta Platforms has taken an aggressive stance against scraping, filing multiple lawsuits against companies. Outcomes, however, have been mixed:

- **Meta v. BrandTotal Ltd. (2022):** Meta successfully sued for breach of contract when BrandTotal’s data collection software violated Facebook’s anti-scraping ToS. The case confirms that contract prohibition of scraping requires proving: a valid agreement, breach of terms, and resulting damages.
- **Meta v. Social Data Trading Ltd. (2022):** Meta claimed the defendant created thousands of fake Instagram and Facebook accounts to scrape profile data, violating California’s Penal Code. Since the defendant never appeared in court, the case ended in a default judgment, leaving open questions about substantive legality.
- **Meta v. Bright Data Ltd. (2024):** This landmark case saw a US federal court reject Meta’s arguments, ruling that scraping public visible data does not violate contractual terms. The judgment highlighted the legal difficulty in distinguishing between public and restricted data, marking a significant setback for Meta’s broader anti-scraping efforts. According to the court, Meta’s ToS apply only to users who are actively logged into their accounts and using them for scraping purposes; they do not extend to the collection of public information by individuals who are logged out.

Case Law Involving X Following Meta’s lead, X Corp. also pursued lawsuits against scraping companies.

- **X v. Bright Data Ltd. (2024):** X alleged breach of ToS and unlawful data resale. The court dismissed key claims, holding that scraping and reselling publicly available data did not violate copyright law, specifically, data that is not behind a log-in screen or otherwise password-protected, and does not involve unauthorized access or breached contract terms. This case highlights the legal nuances of public versus private data collection, reinforcing the US tendency to protect scraping of public

data. Additionally, this case demonstrated that courts are reluctant to recognize concrete damages in the absence of solid evidence.

- **X v. Center for Countering Digital Hate, Inc.** (2024): Here, the court considered scraping by a non-profit organization for advocacy purposes. It found no actionable harm, noting that the information was already publicly available. The case suggests courts may treat scraping for research or public interest more leniently than scraping for commercial gain.

Expert and Industry Opinions Scraping service providers, such as Apify and Bright Data, maintain that scraping publicly available data is legally permissible, while avoiding definitive legal positions and advising clients to seek legal counsel.

Industry experts emphasize that professional scrapers operate as legitimate businesses, accessing websites as ordinary users would, without exploiting vulnerabilities. Nevertheless, most acknowledge that ToS restrictions and jurisdictional differences create substantial uncertainty.

Summary Overall, the case studies reveal that: scraping of publicly available data is generally tolerated by courts, particularly in the US, while, scraping involving private, protected, or authenticated data is far more likely to incur legal liability; contractual restrictions are significant, but their enforceability depends on the clarity of user consent and the demonstrability of damages; non-commercial or public interest uses of scraping may be treated more leniently.

DSA for Research

The EU Digital Services Act (DSA), enacted in 2022, introduces a framework that facilitates access to data held by certain “very large online platforms” and “very large online search engines”. Specifically, the DSA enables researchers that contribute to the detection, identification, and understanding of systemic risks in the Union, by accessing data that is publicly available, including, potentially, via web scraping.

Notably, access is not restricted to researchers affiliated with formally recognized research institutions, nonprofit-affiliated researchers may also qualify. In addition, the DSA provides a mechanism for “vetted researchers” to access non public data under strict conditions. To obtain such access, researchers must demonstrate to the national Digital Services Coordinator that they: are affiliated with a qualifying research organization; operate independently of commercial interests and disclose all funding sources; have the capacity to adequately protect the privacy and security of the data; require access to the data in a manner proportionate to the intended research purpose; and will make the research outputs publicly available free of charge.

Social platforms such as X are expected to provide API access to researchers in compliance with the DSA. In practice, however, the process can be lengthy and complex. In addition, some observers have argued that platforms may intentionally slow or limit access, effectively forcing researchers to rely on alternative solutions to obtain publicly available data.

Alongside legal considerations, ethical guidance plays a crucial role. Organizations like the Association of Internet Researchers (AoIR) recommend that researchers engaging with public data critically assess the potential impacts of their data collection on users and communities. Researchers should consider the original context in which users posted their data and whether users could reasonably expect their information to be analyzed for research purposes. For example, studies of 2018 indicate that most X users are unaware that their public posts may be accessed by researchers. Ethical caution is particularly warranted when collecting data on sensitive topics or involving vulnerable populations, ensuring research practices minimize potential harm and respect user expectations.

In summary, the DSA establishes both a legal and procedural framework to support research on online platforms, while ethical scholarship emphasizes the need for transparency, context-awareness, and protection of users' rights. Where formal access mechanisms are inadequate or obstructed, carefully conducted scraping of publicly available data may provide a justified alternative.

Facilitated Access for Intelligence and Law Enforcement Agencies

Based on the DSA, it is plausible to assume that intelligence teams or law enforcement agencies may have contractual or privileged access to social media data.

Thesis Analysis The analysis of legal issues related to scraping emerged during the initial phase of selecting an appropriate scraping tool for testing the pipeline functionality, even though it was not ultimately used for actual data collection and analysis from social media platforms. This analysis involved consultation with legal experts, including lawyers, who confirmed a complex and continuously evolving regulatory landscape.

The analysis revealed that many researchers and developers routinely use or create scraping tools without adequately considering the associated legal risks. This phenomenon appears to reflect a social normalization of the practice: because scraping is widespread, it is perceived as acceptable behavior. Many practitioners likely underestimate the risk of detection or defer legal considerations to a later stage, potentially exploiting the excessive complexity and dynamic nature of the current legal framework.

Given this complex scenario, the adopted strategy focused on analyzing the available options to make the most appropriate choices, or at least the least problematic ones.

The analysis initially examined the possibility of using the API provided by X to researchers under the DSA. While the research objectives appeared to align with the requirements for API access, a formal application was not submitted for two primary reasons. The first one is that, given that this was a framework that would not involve actual data collection from user profiles, API access was not technically required. Secondly, the application process proved exceptionally complex, with numerous researchers reportedly receiving rejections despite apparently compliant research proposals. The application form accessed through the [X DSA Research API portal](#), required detailed guarantees regarding data security and confidentiality, precise timeframes with comprehensive justifications, and other specifications that were either undefined or difficult to provide concrete answers for within the context of an experimental thesis.

Beyond the official X API, several third-party providers were evaluated based on their legal positioning. Bright Data, the Israeli company that has recently won significant legal battles against X and Meta, offered the strongest legal guarantees. However, the associated costs were prohibitively high, making it unsuitable even for limited testing purposes. Apify, which provider claims legal compliance on its platform, it simultaneously acknowledges legal scraping limitations such as ToS, noting for example that the legal theory behind contract enforceability is complicated. Despite this apparent contradiction, Apify continues to offer actors for social media scraping. Additionally, while Apify's official page maintains that scraping is legal, the platform consistently refers users to the specific actors being used. In the specific actor that performs X scraping, unlike Bright Data which provides reassurances such as collecting only public data, no guarantees are provided from a legal standpoint.

Given the cost-benefit analysis and legal uncertainty, Apify was selected as the scraping solution due to its acceptable pricing and reasonable legal positioning, although with acknowledged uncertainties. To address these legal concerns, several risk mitigation strategies were implemented, specifically only a minimal number of tweets were collected for testing purposes in order to not to overload target servers.

4.3 AI Usage in Sensitive Contexts: Key Legal and Ethical Challenges

The integration of AI systems in sensitive domains such as law enforcement and intelligence operations has revolutionized operational efficiency while simultaneously raising fundamental questions about accountability, fairness, and human rights protection. The deployment of AI tools for profiling, risk assessment, and automated decision making in these high-risks environments presents unique challenges that necessitate comprehensive legal frameworks and ethical guidelines.

4.3.1 Core Problematic Areas

Accuracy and Accountability AI systems operating in sensitive contexts face critical accuracy and accountability challenges that can have profound consequences on individual rights and public safety. The fundamental issue is related to the difficulty of establishing clear accountability chains when AI driven decisions lead to adverse outcomes. Additionally, in intelligence and law enforcement applications, inaccurate or erroneous AI conclusions can result in compromised operations, wrongful investigations, or violations of civil liberties.

While these issues are somewhat less pronounced in the context of RAG systems, the broader opacity of many AI architectures still complicates accountability and accuracy. In many cases, it remains impossible to trace how a decision was made and whether it is accurate or biased, creating significant barriers to both internal review and external oversight.

Transparency Violations The lack of transparency in AI decision making processes poses serious threats, particularly in contexts where individuals may be subjected to investigation, surveillance, or other law enforcement actions based on AI recommendations.

It is essential to have a clear idea of both the benefits and risks associated with the AI technology. Users and investigators must be well informed about how these AI tools function, in order to use and interpret them effectively. Those involved in investigations should invest time in supervising and verifying AI outputs, maintain a correct level of skepticism, and provide the technology with opportunities to demonstrate its reliability.

Algorithmic Bias and Discriminatory Outcomes AI systems can inadvertently perpetuate and even amplify biases present in training data, leading to incorrect or unfair results. This is confirmed by a recent report of the Fundamental Rights Agency, which outlined how algorithms used to identify hate speech are susceptible to be biased.

When analyzing social media content for signs of extremist behavior, depression, or criminal intent, AI systems may disproportionately flag individuals from certain communities or those who express themselves in ways that differ from mainstream norms.

Privacy and Data Protection Violations AI systems significantly amplify privacy and data protection risks due to the enormous amounts of personal information required for their operation. The deployment of these systems in sensitive contexts typically necessitates access to vast datasets that include private communications, behavioral patterns, and intimate details of individuals' lives, raising questions about fundamental rights. The long-term storage and potential reusing of this data for purposes different from the original intent further compounds privacy concerns.

The need to ensure adequate data protection and prevent unauthorized access becomes crucial for maintaining confidentiality and ensuring compliance with privacy regulations, requiring the implementation of robust security measures and data governance frameworks.

Human Judgment Displacement The automation of decision making processes in sensitive contexts risks displacing essential human judgment, empathy, and contextual understanding, which are crucial for fair and effective law enforcement and intelligence work. AI systems lack the ability to understand nuanced human behavior, cultural contexts, or the complex motivations that drive human actions, regardless of their sophistication.

For instance, AI systems may have difficulty in categorizing certain types of content. In the case of violent extremist material, for example, determining what constitutes such content can be particularly difficult due to the absence of a uniformly accepted definition. In applications such as social media analysis for threat assessment, AI systems may identify patterns that appear suspicious from a statistical perspective but may still misinterpreted content.

Thesis Analysis To address the critical challenges of accuracy, accountability, and transparency, this thesis implements a RAG system that transforms how AI outputs can be verified and trusted. The RAG architecture enhances accuracy by grounding

responses in retrievable, factual documents, reducing hallucinations and ensuring outputs are anchored to verifiable sources. It ensures accountability by maintaining an audit trail that enables attribution of every claim to its supporting documentation. The system promotes transparency by exposing the specific documents that influenced each response, transforming AI into an interpretable analytical tool.

Regarding algorithmic bias and discriminatory outcomes, limited work has been undertaken to date. While RAG systems with unbiased documentation would represent progress toward fairness, this approach alone cannot eliminate algorithmic bias. Bias can persist at multiple levels: in the underlying model’s pre-training, or in retrieval mechanisms. Effective bias mitigation requires a multi-layered strategy combining diversified documentation, fairness-oriented prompt engineering, continuous validation across demographic groups, human oversight, and quantitative fairness metrics.

To address the privacy concerns, this thesis adopts a local approach using the Cheshire Cat AI framework integrated primarily with a local LLM. This architecture ensures that personal information and confidential data never leave the local environment, with the sole exception being the use of external APIs for web scraping operations. The implementation of a local LLM necessitated hardware with substantial computational requirements. To prevent overloading the available device during the extensive testing phase, it became necessary to incorporate paid cloud services such as Groq AI, which provides cloud-based access to LLMs while maintaining the local processing paradigm. Furthermore, it is well-established that local models are typically smaller and less performant compared to their commercial counterparts. This limitation necessitated the additional use of OpenAI’s models to comprehensively test and validate the proposed pipeline, enabling performance comparisons and ensuring the robustness of the system across different model capabilities.

To address the issue of human judgement displacement, this thesis implements a human-in-the-loop approach through the generation of comprehensive, human-readable dossiers rather than simple classifications. This methodology is specifically designed to augment and support human evaluation rather than replace it. The generated dossiers serve as intelligent briefing documents that present relevant information in a structured format, enabling them to make more informed decisions while retaining full decision-making authority.

4.3.2 Legal Requirements

The urgent need to regulate AI has pushed governments and international organizations to develop new regulatory frameworks. As AI spreads into increasingly critical sectors, the legal system must adapt to protect human rights, fairness, and democratic values.

The global regulatory landscape is responding with a layered approach: ethical guidelines, sector-specific regulations, and comprehensive legislative frameworks combine to form a diverse and rapidly evolving system.

International Principles and Ethical Guidelines Many countries and international organizations have published ethical guidelines and principles, emphasizing fairness, transparency, accountability, privacy, and human-centric AI development and deployment. Examples include the OECD organization, the EU’s Ethics Guidelines for

Trustworthy AI and the Montreal Declaration for Responsible AI, which have further solidified these foundational principles, creating a global consensus on the fundamental values that should guide AI governance and implementation.

Data Protection as AI Regulation Existing data protection laws have become crucial regulatory tools for AI governance. The GDPR sets stringent standards for the collection, processing, and storage of personal data, with direct implications for AI systems that handle personal information. Specifically, the article 22 of the GDPR addresses automated decision-making, establishing fundamental protections for individuals against purely automated processes that significantly impact their lives, requiring transparency and explanation.

This provision is particularly relevant to social media analysis systems, as it grants individuals the right not to be subject to decisions based solely on automated processing, including profiling, when such decisions produce legal effects or similarly significant impacts. The regulation allows for exceptions only when automated decisions are necessary for contract performance, authorized by law with appropriate safeguards, or based on explicit consent.

The European AI Act The European AI Act represents the world’s first comprehensive AI regulation. This regulation establishes common rules applicable across different high-impact economic sectors, with particular emphasis on high-risk AI applications. The Act categorizes AI systems based on risk levels, from prohibited unacceptable risk applications to regulated high-risk systems that require extensive compliance measures.

Of particular relevance to social media profiling applications is the Act’s emphasis on data quality and governance, as outlined in Article 10. The regulation mandates that high-risk AI systems be built upon high quality training data, requiring sector experts to carefully curate knowledge bases. This requirement is especially pertinent for the proposed RAG systems used in intelligence and law enforcement contexts, where the quality of underlying knowledge directly impacts the accuracy and reliability of threat assessments.

The Act also emphasizes human oversight as a fundamental requirement for AI systems, mandating that meaningful human control be maintained throughout the decision making process.

United States Regulatory Approaches The United States has pursued a different regulatory strategy, combining federal guidelines with sector-specific legislation. The White House’s AI Bill of Rights, establishes five key principles for AI deployment: safe and effective systems, notice and explanation, data privacy protection, protection from algorithmic discrimination, and human alternatives and oversight. Another example is the Algorithmic Accountability Act of 2022 (AAA), which represents a significant legislative effort to enhance transparency in algorithmic decision-making. The Act proposes mandatory impact assessments for organizations using automated decision systems, requiring pre-deployment and post-deployment analysis.

International Collaboration Recognizing the global nature of AI challenges, international collaboration efforts have emerged to promote harmonization of regulatory approaches. Examples of initiatives are: the EU-US Joint Roadmap for Trustworthy AI and Risk Management initiative, which aims to develop shared terminologies, taxonomies, and metrics for measuring AI trustworthiness and risk management; the Global Partnership on AI; the G20 AI Principles.

Sector-Specific Considerations Many regulatory frameworks adopt sector-specific approaches, recognizing that AI applications in healthcare, finance, law enforcement, and intelligence require tailored oversight mechanisms.

For example, in the Article 11 of the Police Directive, automated processing is forbidden unless: there is the human intervention, not produce an adverse legal effect concerning the data subject, is authorized by the EU or Member States, and provides appropriate safeguards for the rights and freedom of the data subject.

Chapter 5

System Architecture and Pipeline

This chapter presents the proposed architecture and its core components, detailing how they interact to form a cohesive intelligence analysis system. The objective is to provide a comprehensive understanding of the system’s design, demonstrating how each module contributes to the automated generation of structured intelligence reports from social media data.

5.1 Overview of the Pipeline

The proposed pipeline is an intelligent chatbot-based system designed to automatically generate structured reports from social media profiles, empowering intelligence teams to efficiently identify and assess potentially high-risk content.

The system integrates multiple specialized components, from data collection and content classification, to knowledge retrieval and report generation, to systematically replicate and enhance human intelligence analysis methodologies. This automated approach follows a logical workflow: systematic data collection, initial classification to identify key risk and prioritize resources, targeted knowledge application based on identified risk category, and structured reporting for decision support.

The process begins with the input of a social media profile URL, specifically a X URL. A scraper API collects the tweets from that profile, which are then preprocessed to clean and normalize the raw text.

Each tweet is analyzed by a classifier that labels it as either neutral or potentially critical. Critical tweets are categorized into one of four predefined risk: terrorism and extremism, hate speech and cyberbullying, depression and suicide, or cybercrime and hacking. At the profile level, the system determines the predominant risk category, indicated by the majority label, replicating an analyst’s initial triage.

Based on the identified predominant risk category, the system filters domain-specific documents from a RAG knowledge base. The most relevant and recent tweets are embedded into a vector space and semantically compared with this filtered subset of documents, ensuring focused and efficient retrieval.

In the final stage, the LLM integrates the analyzed tweets with the retrieved context to generate a structured report. For each tweet, the system highlights the similarities

with the selected RAG documents related to the identified risk category. This provides analysts with not only a classification but also transparent reasoning grounded in textual evidence.

This pipeline effectively combines automated data collection, intelligent classification, and targeted retrieval of domain-specific knowledge to provide a scalable and focused solution for monitoring potentially high-risk content on social media. By explicitly linking tweets to known communication patterns and explaining the similarities, it not only flags risks but also provides actionable insights for decision support.

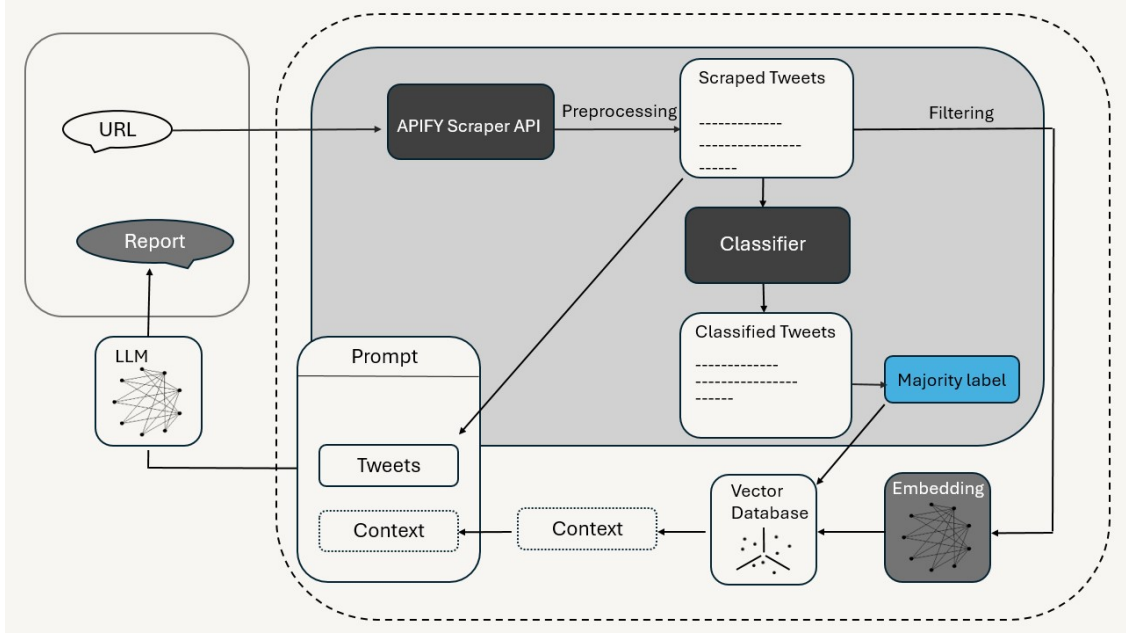


Figure 5.1. Overview of the proposed project pipeline

5.2 Choice of LLM and Embedder

In the previous section, was described the pipeline designed to collect and analyze social media content. A key component of this pipeline is the use of LLMs and embedding techniques.

To configure the LLM and Embedder for the pipeline, it is necessary to access the Admin Portal at *localhost:1865/admin*, open the *Settings* tab, and select the desired models. A video tutorial detailing this procedure is available in [320].

As described in Section 3.5, the proposed framework utilizes two LLMs, respectively **Llama 3.1 8B Instruct** and **GPT-4.1**, and **Nomic Embed Text v1.5** as Embedder.

Regarding LLM maximum number of completion tokens parameter, different models exhibit varying native output capabilities. In the context of this framework, Llama 3.1 8B Instruct was configured with *Max Completion Tokens* limit of 2048. This configuration

was determined based on preliminary analysis of the model’s natural output patterns for the analysis of at most 10 tweets, chosen to enable practical evaluation of reports. This testing revealed that Llama 3.1 8B consistently generated comprehensive reports within approximately 2048 tokens, representing the model’s natural completion length for the given task. GPT-4.1 was used without explicit output constraints, relying on the model’s default behavior, which naturally generated reports in the range of approximately 3000 tokens for similar tasks.

For the *Temperature* parameter, both LLMs were set to 0 in order to produce deterministic, focused, and predictable outputs, ideal for tasks that require accuracy and consistency. The *Top P* was not specifically configured, as it has no effect when temperature is 0; it may optionally be set to 1, which is the default value, without impacting the behavior. Additionally, the *Streaming* parameter was disabled.

Regarding the Embedder configuration, the default settings were retained: *Max Length* set to 512 tokens and *Doc Embed Type* set to “passage”.

5.3 RAG Documents: Role and Structure

The proposed framework leverages a RAG system to systematically analyze X profiles and generate detailed analytical reports. Its main objective is to identify critical risk indicators across four categories: terrorism and extremism, hate speech and cyberbullying, depression and suicide ideation, and cybercrime and hacking.

The RAG knowledge base could be structured according to these categories, with each containing specialized documents that capture communication and linguistic patterns relevant to online activity. These documents serve as reference material for linking observed social media content to known risk behaviors.

Terrorism and Extremism Documents in this category could be designed to profile specific terrorist or extremist groups, detailing their terminology, rhetoric, and communication patterns. Linguistic analysis would focus on group-specific vocabulary, coded language, symbolic emoji, and propaganda hashtags. Each document could include examples, such as posts with characteristic word choices, specialized hashtags, ideologically charged symbols, and coded language patterns, with explanations linking each element to its meaning, communicative function, and strategic role within the group’s messaging framework.

Hate Speech and Cyberbullying Documents could be organized by type of abusive behavior, such as racial and gender-based attacks, or coordinated harassment campaigns. Linguistic components would include derogatory terms, slur variations, coded insults, harassment-specific emoji, and coordinating hashtags. They might cover coded language evolution, the use of euphemisms, and typical sentence structures.

Depression and Suicidal Ideation These documents could describe various forms of depression and how they appear on social media. Linguistic analysis would identify related vocabulary, metaphors, self-harm terms, crisis emoji, and hashtags that show

concern or ask for support. They could provide patterns for recognizing direct and indirect expressions of suicidal thoughts, self-harm references, and linguistic markers that show whether someone is seeking help or engaging in risky behavior, placing individual posts in the context of broader community patterns.

Cybercrime and Hacking Documents could detail cybercriminal groups and their operational methods, focusing on technical jargon, coded service advertisements, underground marketplace vocabulary, hashtags, and symbolic emoji used for covert communication. Linguistic indicators could include social engineering language patterns, phishing message structures, malware-related terms, URL obfuscation techniques, coded language for illegal services, and community-specific slang.

These categories illustrate one possible way to organize knowledge, but they can be adapted or expanded depending on the specific use case or operational requirements, thanks to the fully flexibility of RAG. New categories can be added, and existing document templates can be modified to match emerging threats, specific contexts, or operational needs. This adaptability ensures the system remains relevant and effective across different intelligence analysis scenarios.

When a profile is selected for analysis, the system could compare its posts to the RAG documents to identify similarities in linguistic and communication patterns. This process supports the generation of analytical reports based on evidences that go beyond simple categorical labeling.

This approach ensures transparency, human oversight, and continuous adaptability as new groups, behaviors, and digital tactics emerge. By linking social media content to documented patterns, the system provides actionable insights for decision making while maintaining rigorous analytical standards.

5.3.1 Document creation

As previously noted, RAG documents should capture both the explicit content, such as keywords, hashtags, and emojis, and the implicit function of social media language across different domains. They should also include representative examples, recurring phrases, and contextual explanations that increase the likelihood of semantic similarity matches, ensuring that embeddings can effectively retrieve the corresponding RAG documents.

Content Source

The content included in each RAG document can be obtained through multiple approaches, depending on the available resources and context. Potential sources include information gathered by intelligence, which may be collected through covert operations or systematic open source monitoring, as well as publicly available documents, such as official government reports, policy analyses, academic publications, and white papers. Social media content, including posts, tweets, and forum discussions that have been previously identified as relevant, can also be used to extract linguistic patterns and communication

strategies. Additional contributions may come from news media, specialized databases, interviews with experts, and historical records.

As noted in the state of the art review on social media profiling [2.2.3](#), the availability of such material varies across domains. For example, in areas like terrorism and extremism, substantial data exist to support the creation of detailed documents for specific groups. In contrast, in other domains like mental health, signals are more subtle, heterogeneous, and comparatively less studied.

Document Structure

Once the content sources have been identified, the next step is to organize the information into documents suitable for ingestion by the RAG system.

In the Cheshire Cat AI framework, the Rabbit Hole is the Cat's component responsible for taking documents and storing them in declarative memory. Currently supported file formats include `.txt`, `.md`, `.pdf`, and `.html` via web URL. Markdown (`.md`) is strongly recommended, as it is a lightweight, easy to read language that allows for clear structuring of text with headings, subheadings, and lists, as well as typographic emphasis elements such as bold or italics, additionally preserving the geometry of the document.

Organizing files into a clear and logical structure, with distinct sections, titles, headings, and subheadings, such as Introduction, Frequently Used Terms, Hashtag, Emoji, Communication Tactics, and Corresponding Explanations, can further enhance the Rabbit Hole's ability to store and retrieve information effectively. Particular attention should be paid to the use of headings: by establishing a clear hierarchy of information, it is possible to more efficiently categorize content, improving the capacity to retrieve relevant knowledge.

It is also essential to remove any irrelevant information, retaining only the content that is directly useful for the intended purpose. Another important step is to simplify the text itself, ensuring clarity and readability. Additionally, including a concise summary at the beginning or end of each document can provide a quick overview of its contents, facilitating interpretation, retrieval, and the generation of accurate responses. It is also useful to write block relatively short and self-sufficient, in order to avoid problem when chunking happens.

Special attention is required for non-textual content like images, charts, and tables, which may contain valuable information but can be easily overlooked or misinterpreted without proper preparation. Regarding emojis, it is important to include in the RAG document not the emoji itself but its textual representation, which could be obtained using the Python `demoji` library. Converting emojis to text ensures consistency with tweet previously introduced preprocessing [3.2](#) and enables accurate semantic matching between tweets and stored documents. The conversion in text representation is generically suggested for all images. For tables, it is recommended to write them in Markdown format, or alternatively, to use lists, which provide a more readable plain-text representation.

5.3.2 Document Ingestion

To upload a document, access the Admin Portal at *localhost:1865/admin*, open the Home tab, click on the Flash icon, and then select *Upload file*. During ingestion and upon completion, notifications are provided by the Cat.

It is important to observe that, if the Embedder is changed after document ingestion, all previously stored memories will be lost.

Chunk

After ingestion, data from the ingested document are obtained through specific parsers. Once extraction is complete, the content is split into chunks, vectorized via the embedder, and stored. The process of dividing data into smaller portions is called *chunking*, and the component responsible for this task is the *splitter*.

Text chunking is usually the first step in RAG pipelines, and it plays a crucial role for several reasons [321]. By breaking external text into smaller units, the model can focus on relevant segments during retrieval, improving accuracy and increasing the likelihood of retrieving useful content. Structured chunks allow the model to optimize computational resources, speeding up retrieval and generation tasks. Chunking additionally highlights key information, enabling the model to produce more coherent and contextually relevant responses. By filtering out irrelevant data, chunking ensures more concise and on-topic outputs.

There are several chunking methods, each with different trade-offs. The simplest one is fixed size chunking, which splits text by a fixed number of characters. While easy to implement, it ignores structure and may break meaningful information. A refinement is recursive chunking: the text is first divided by natural separators such as punctuation or spaces, and if segments remain too large, they are further split by words or tokens. This approach preserves more context than fixed-size splitting while remaining computationally efficient. Document based chunking adapts chunking to the format of the source, such as Markdown, code, JSON, or HTML, ensuring that structural elements are preserved. This method provides accurate and context-aware chunks but requires maintaining multiple splitters and involves higher computational costs. Semantic chunking, instead, leverages AI models to divide text into semantically coherent blocks. This improves comprehension, generates more relevant answers, and adapts well to different tasks. However, it is harder to implement, depends heavily on the quality of the language model, and often requires expensive calls to large models. Agentic chunking goes one step further by using LLMs to mimic human reading, grouping semantically related parts without relying on predefined rules or length constraints. It yields highly accurate chunks adaptable to any structure, though at the expense of significant computational cost and implementation complexity. Finally, context aware chunking combines elements of the previous approaches. It identifies semantic units, analyzes their context, splits text into variable-sized chunks, and indexes them for retrieval. While this methodology can substantially improve retrieval quality in RAG systems, it is computationally demanding and best suited for applications where accuracy outweighs efficiency.

To handle complex intelligence documents effectively, was implemented a custom

hook, *rabbithole_instantiates_splitter*, which overrides the default text splitting behavior of the Cheshire Cat framework. This hook intercepts the ingestion process and replaces the standard splitter with a Markdown-aware version, designed to preserve document structure. The splitter produces chunks of 2000 characters with an overlap of 400 characters, striking a balance between contextual consistency and retrieval efficiency.

Starting from this point, the term *doc* will refer to a LangChain Document chunk to be inserted into memory. Unlike previously, this does not represent the entire document, but rather a portion of it.

Metadata

Traditionally, a *doc* has two main properties: *page_content*, a string containing the text to be saved in memory, and *metadata*, a dictionary that must include at least two keys, respectively *source*, indicating the origin of the text, and *when*, a timestamp recording when the content was uploaded.

Metadata plays a key role in RAG systems by providing context that improves both retrieval accuracy and response quality. Beyond simple semantic similarity, metadata enable filtering of *doc* during retrieval, allowing searches to focus on specific content based on attributes. They also enrich the search process, helping the system find more precise matches and interpret information more accurately. Metadata track content provenance, supporting proper citation, reliability verification, and source-based access control, and can influence chunk ranking to prioritize recent or authoritative content.

In the proposed framework, each *doc* is enriched with two additional metadata fields compared to the original version. The first, *category*, assigns the document to one of the four main categories previously defined (terrorism and extremism, hate speech and cyberbullying, depression and suicide, or cybercrime and hacking). The second, *sub_category*, provides a more fine-grained classification, capturing details such as the name of a terrorist group, the type of cyberbullying phenomenon, the form of depression, or the name of a cybercriminal group.

Using the Cheshire Cat AI hook *before_rabbithole_insert_memory*, it is possible to intercept a *doc* before it is inserted into the declarative vector memory, in order to enrich it with additional metadata. The process works as follows: the filename is extracted from the document's *source* metadata, removing the file extension using *os.path.splitext()*; the resulting filename is split using the first - character as separator, assuming a naming convention in the format *category-subcategory*; the two extracted elements are assigned respectively to the document's *category* and *sub_category* metadata fields.

This approach enables automated document metadata assignment without manual intervention by leveraging structured file naming conventions as the primary source of categorical information, rather than relying on content-based classification through an LLM. Within the *before_rabbithole_insert_memory* hook, it is technically possible to invoke the LLM via the Cat (*cat.llm*), passing the *doc.page_content* as input and asking the model to infer appropriate metadata. However, this strategy presented significant challenges. The main issue concerns inconsistency at the *doc* level: different chunk of the same document might be assigned different categories, leading to fragmented metadata. From an architectural perspective, delegating categorization to AI tools introduced

unnecessary complexity and potential points of failure. By contrast, the filename-based approach offers several advantages: it is deterministic and reproducible, computationally efficient, easy to debug, and provides complete control over the classification logic.

Below there is an example of a prompt for assigning the metadata *category* through the LLM:

```
category = cat.llm(f"""
    Classify the following content into exactly ONE category.

    Categories:
    - Terrorism
    - Hacking
    - Cyberbullying
    - Depression

    Classification Criteria:
    - "Terrorism" : tactics, strategies, terminology, glossary used
      by TERRORIST groups.
    - "Hacking" : tactics, strategies, terminology, glossary used
      by HACKER groups.
    - "Cyberbullying" : behaviours, terminology of CYBERBULLING
    - "Depression": depression signal, terminology associated to
      DEPRESSION and SUICIDE ideations.

    Rules:
    1. Return ONLY the category name
    2. No explanations, no punctuation, no additional text
    3. Choose the most relevant category

    { doc.page_content }
    """)
```

Within the framework, metadata serve three main purposes: filtering RAG documents through the *category* metadata, based on the predominant risk category detected in the analyzed tweets; iterating over all *sub_category* documents within a given *category* to improve retrieval accuracy, as detailed in Section 5.7; and reporting, in the final analysis, the specific *sub_category(s)* that the tweet most closely resembles.

Document Visualization

When a document is ingested, as previously mentioned, it is split into chunks. Each chunk is passed to the Embedder, which produces a geometric representation, transforming it into a point in geometric space. By navigating to the Memory tab in the Admin Portal at *localhost:1865/admin*, each chunk can be visualized as a dot in space, annotated with its associated metadata (see Figure 5.2). Additionally, by using the Export Memories feature, it is possible to export the memories from the vector database.

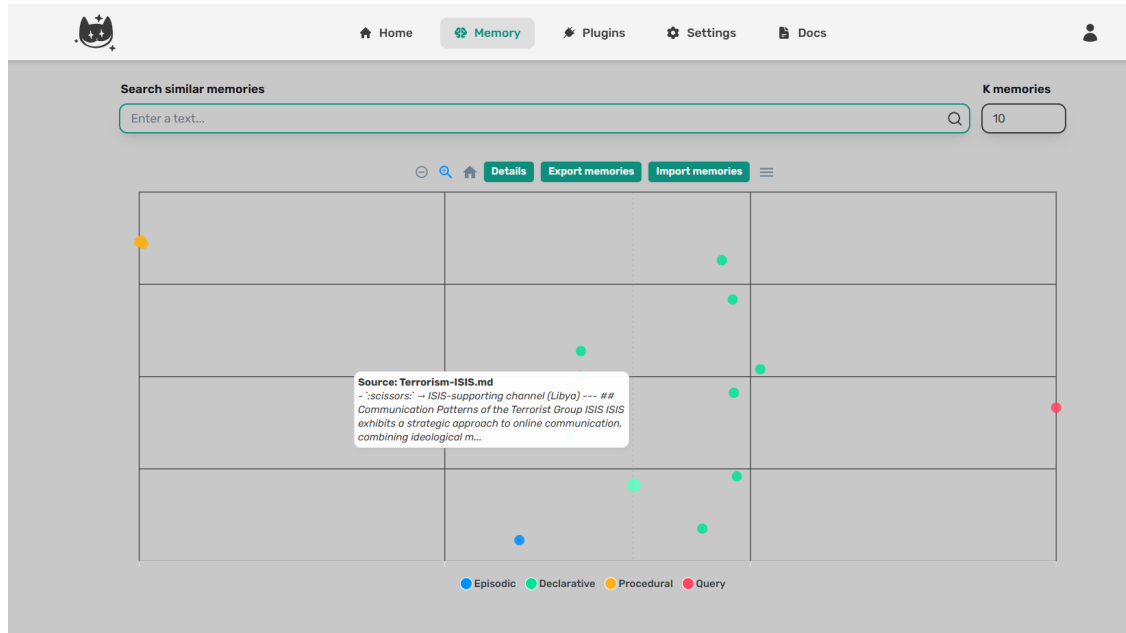


Figure 5.2. Memory Visualization

5.4 X Profile Input and Tweet Extraction

The first step of the pipeline consists in providing an URL as query to the chatbot, which will then be used to scrape X posts from a specific target profile. The profile URL is a mandatory input parameter. Additional optional parameters, supported either by the scraper or by the official API, can also be specified. For instance, within this framework, which leverages Apify's *Twitter Tweets Scraper*, it is possible to define the start date for tweet collection and the maximum number of tweets to retrieve. Depending on the specific needs, two distinct approaches can be adopted for parameter handling. The first approach implements a deterministic parameter parsing mechanism using a pipe-delimited format *URL/number/date*. This method prioritizes predictability and efficiency by avoiding LLM calls for parameter extraction. However, it comes with the drawback of a less user-friendly experience, since users must learn and adhere to a rigid input format. The second approach leverages the natural language understanding capabilities of the LLM to extract parameters directly from conversational input. This method offers greater intuitiveness and flexibility for the user, though at the cost of reduced determinism and additional computational overhead. An example of a prompt for extracting parameters from user input through the LLM is provided below. This approach is made possible by the integration of the LLM through Cat (*cat.llm*):

```
original_message = cat.working_memory.user_message_json.text
message = cat.llm(f" " " "
```

```

Extract from the following text: profile URL, number of tweets,
and since date.

Text to analyze: {original_message}

Rules:
- Date format must be YYYY-MM-DD
- Output format: URL|number|date

Output only the structured result, nothing else.
""" )

```

This logic is implemented within *fast_reply* hook, which is required for building custom LLM chains. Specifically, every step of the process, from scraping to preprocessing, classification, and filtering, is executed inside this hook. This design makes it possible to scrape tweets from a given URL, preprocess them, classify and filter the content, reorder the results, and finally pass the processed output to the Embedder as if it were the original query of the RAG.

It is also important to note that the proposed framework does not take episodic memory into account. This design choice ensures analytical independence across scraping tasks and prevents context contamination. To achieve this, is used the hook *before_cat_recalls_episodic_memories*, with the number of recalled memories set to 0.

5.4.1 Tweet Volume Configuration Considerations

The determination of the number of tweets to be analyzed simultaneously involves several interconnected factors that must be balanced. This section outlines the key considerations that influence volume configuration decisions in such systems.

Scraping Stage Considerations The scraping technology choice determines temporal limitations, daily or monthly extraction thresholds, and cost implications of different acquisition strategies. External constraints include budget availability and time requirements. Furthermore, the required granularity of the analysis determines the minimum number of tweets necessary to obtain significant results.

Computational Pipeline Constraints From a computational perspective, each processing stage introduces specific scalability considerations. Preprocessing operations scale with tweet volume in terms of computational time and resource utilization. The classification stage typically represents one of the most resource intensive steps, with LLM API limitations including rate limits, token restrictions, and processing latency requiring strategic management. Batching strategies may be necessary to mitigate API constraints while maintaining processing reliability.

Embedding and Retrieval Limitations The Embedding stage faces token capacity limits for input queries, with vector representation quality potentially varying based on

input length. For example the Nomic Embed Text v1.5 Embedder supporting a theoretical maximum of 8,192 tokens per input; however, query lengths are discouraged by Cheshire Cat AI documentation. LLM context window restrictions set maximum token capacity for both input and output processing. Extended contexts may lead to performance degradation, necessitating consideration of optimal input sizes for effective retrieval operations.

Processing Strategy Considerations RAG-based social media analysis systems can adopt different processing strategies at various pipeline stages. During the classification phase, tweets can be processed individually or in configurable batch sizes, with batch sizing directly affecting API call frequency, processing latency, and associated costs. For the embedding and retrieval phase, systems can either embed individual tweets for separate RAG queries or consolidate multiple tweets into unified embedding queries. Individual tweet embedding enables granular document matching but multiplies retrieval operations and computational overhead. Unified embedding approaches reduce operational complexity and costs while enabling profile global analysis, though they face constraints from embedder input limitations and LLM context windows.

The implemented pipeline uses configurable batch sizes for classification and unified embedding for the final RAG query phase. While individual tweet embedding remains theoretically possible, the unified approach was selected.

Output Requirements The intended use case influences volume requirements, intelligence reports for rapid assessment may require different configurations than comprehensive analytical studies. The balance between detailed analysis depth and report conciseness affects optimal processing volumes.

5.5 Data Cleaning and Preprocessing

After scraping, the collected tweets undergoes cleaning and preprocessing, as described in Section 3.2. Additionally, only english tweet are retained, due to tweet availability, document language, and embedder requirements.

As previously observed, these operations are executed within *fast_reply* hook.

5.6 Preliminary Classification Module

The pipeline implements a preliminary classification step to identify high-risk tweets and assign each of them to one of four critical categories: terrorism and extremism, hate speech and cyberbullying, depression and suicide, or cybercrime and hacking. This stage serves two purposes: filtering RAG documents, considering only those with metadata category matching the majority category of the analyzed profile, instead of comparing tweets against the entire corpus, improving both retrieval efficiency and the relevance of responses; prioritizing the analysis on the most relevant tweets, using the assigned labels as a filter.

When selecting tweets to be actually analyzed together with the RAG documents, priority is given to those classified in the majority category. However, if the maximum limit of tweets has not yet been reached, the system also includes tweets assigned to different categories, excluding neutral ones, which by definition do not contribute relevant information for risk assessment analysis. This choice reflects the possibility that some tweets, even if classified differently, may still carry features related to the majority category and thus contribute to a more complete analysis.

The system adopts a multiclass approach, assigning each tweet to exactly one category. Although a multilabel strategy could capture overlaps, the multiclass design offers clearer results and makes aggregation more straightforward.

Among different options, the pipeline uses prompt based classification with LLMs. As discussed in Section 3.3, this method is especially effective when labeled data is limited, quick adaptability is needed, and computational or financial resources are constrained. Thanks to the pipeline’s modular design, other solutions can be integrated later if needed.

The framework supports flexible batch size configurations to fit different needs, from processing a single tweet to handling large groups at once. Batch classification helps avoid context window limits, which can occur when too many texts are processed together, while also reducing API calls and processing costs. The batch size can be tuned to balance accuracy, efficiency, and cost.

The system employs a sophisticated prompt engineering framework, structured with three main components: defining the classification task and behavioral expectations, describing the classification categories, and specifying the output format. Generically, it is suggested to use clear and concise language to convey the task accurately and avoid misinterpretation. It is also important to clearly define the task’s purpose, specify relevant constraints, and offer background information that helps the model understand nuances of the request.

For zero-shot classification, prompts are designed to provide sufficient context to enable accurate classification without relying on training examples. They include detailed descriptions of each category with defining characteristics.

Few-shot prompting offers an intermediate approach between zero-shot learning and fully supervised fine-tuning, providing the LLM with a limited number of examples, guiding the model towards a better understanding of the task and desired output. Examples are chosen in order to represent typical cases within each category, as well as boundary cases that help distinguish between similar categories. It’s important to choose examples with varying levels of severity or explicitness, avoiding complex that might confuse the model. To prevent majority label bias, it’s crucial to maintain an equal number of examples for each category. Generally, 2-5 examples per category are sufficient for guiding the model effectively, and providing too many examples may lead to overfitting with minimal benefit. Also the order of examples influence output. Placing the most critical example last or randomizing order can help mitigate recency bias, as models often weigh the last pieces of information more heavily.

Regarding instructions and examples, the typical approach is to present instructions first followed by examples, but the best arrangement may vary depending on the model. In some cases, omitting instructions entirely can be effective if the task is simple enough

for the model to infer independently.

There is no universally applicable method for constructing zero-shot or few-shot prompts. However, a valid example of such a prompt is outlined below. This prompt was developed through iterative testing to determine the most effective structure. Specifically, for the few-shot scenario, the aim was to identify, for each of the four categories, a set of examples that comprehensively cover the full range of scenarios, as well as the optimal combination, number, and order of examples.

The first component provides clear, structured category definitions:

```
description_labels = """
Description of each category:

Terrorism: Tweets that appear to come from terrorists or their
supporters, aiming to spread extremist ideology, recruit or
radicalize, claim attacks, issue threats, or share propaganda
media.

Cyberbullying: Tweets containing racist, sexist, or homophobic
content, threats of violence, sarcastic or passive-aggressive
jabs, gaslighting, offensive memes, social exclusion, or
generally abusive language targeting individuals or groups.

Hacking: Tweets related to hacking activities, including phishing
or social engineering, sharing vulnerabilities, claiming
cyberattacks, promoting hacktivist causes, issuing warnings, or
posting in a chaotic or provocative ("shitposting") style.

Depression: Tweets that express sadness, passive-aggressive
distress, references to death or suicide, or a desire to be
heard or helped, often revealing emotional suffering or
hopelessness.
"""
```

The second component includes carefully selected examples that cover the spectrum of each category:

```
example_labels = {
    "Terrorism": [
        "While the world watches in silence, our brothers are
slaughtered. You can either sit behind a screen or
stand with honor. Join the real fight. #Syria #Ummah #
Jihad",
        "Our camps are alive with the spirit of jihad. The next
generation is ready. Video now on our secure channel.",
        "Our fighters have successfully struck the enemy in the
heart of their city. This is just the beginning of our
campaign for justice.",
        "We will strike at the heart of their government, our bombs
are ready. The revolution begins at dawn"
    ],

```



```
"Cyberbullying": [
    "Imagine being a fat, gay, autistic vegan who thinks they're a gamer-just uninstall life already.",
    "Oh look, @smartpants thinks they're sooo special because they got an A. We're all impressed...NOT",
    "I guess some people really don't get the hint when they're not wanted around.",
    "You're such a disgusting freak. Everyone talks about how much they hate your ugly face when you're not around, Kill yourself already, nobody would miss you."
],
[... additional categories ...]
}
```

The final component assembles these elements into a coherent prompt structure:

```
prompt = f"""Classify each of the following tweets into one of
these categories:
{labels_list}{description_labels}{examples_list}

For each tweet, provide ONLY the category label and nothing else.

Important: Do not include any introductions, explanations, or
summaries. Provide directly the list of Tweet.

Format your response exactly like this:
Tweet 1: [CATEGORY]
Tweet 2: [CATEGORY]
...and so on for all tweets.

Important: If you are uncertain between multiple categories for a
tweet, choose the most predominant or relevant one.

Here are the tweets to classify:
{all_tweets_prompt}
"""
```

The implementation includes error handling and similarity matching:

```
best_label, score = min(
    ((label, utils.levenshtein_distance(label_part, label)) for
     label in labels_names),
    key=lambda x: x[1]
)
classification = best_label if score < score_threshold else None
```

As previously observed, these operations are also executed within *fast_reply* hook. In this way, thanks to the classification results, it is possible to create the effective query to be embedded: the tweets list, filtered and reordered to include the most relevant and recent tweets.

5.7 Retrieval-Augmented Generation Module

Once the classification is complete, a list of tweets needs to be created for embedding and comparison with subsets of RAG documents, which are filtered based on the majority category metadata. The number and sequence of this list of tweets depend on the limitations of the Embedder, the LLM, and the specific use case. For this PoC, a maximum of 10 tweets will be embedded, with a final report that does not exceed 2048 tokens, making it both concise and easy to analyze. The tweets are ordered by date, with the most recent given priority. Tweets labeled with categories different to the majority one (excluding neutral labels) will only be considered if there are fewer than 10 tweets labeled with the majority category. The selected tweets are formatted as a single prompt string, where each tweet is represented as “Tweet ID [tweet_id]: [processed_text]” on separate lines, and saved in *cat.working_memory.user_message_json.text*, which correspond to the effective RAG query.

The next step is the retrieval mechanism. The implemented system overrides the default declarative memory retrieval by implementing a two-level hierarchical search strategy.

The first step consists of disabling the default retrieval by setting the retrieval parameter to 0, allowing for custom implementation. This is made possible by the hook *before_cat_recalls_declarative_memories*, which intercepts the moment when the Cat queries the memories using the embedded user input. The hook is executed just before the Cat searches for the meaningful context in both memories and stores it in the Working Memory.

```
@hook
def before_cat_recalls_declarative_memories(
    default_declarative_recall_config, cat):
    default_declarative_recall_config["k"] = 0
    return default_declarative_recall_config
```

Subsequently, the hook *after_cat_recalls_memories* is used. This hook intervenes after the semantic search in memory has been completed. It is executed immediately after the Cat stores the retrieved context in Working Memory. The first step consists of extracting the variable *majority_label* from *cat.working_memory.majority_label*, if it exists. This variable corresponds to the majority category resulted from classification. Then, all memory points are scanned to identify the subcategories belonging to this majority category, creating a comprehensive taxonomy of available content clusters. This is done because the *sub_category* metadata associated with a given *category* is not known a priori.

The embedded version of the selected list of tweets is saved in *user_message_embedding*. For each identified subcategory, a vector similarity search is performed using the embedded query, with filtering constraints that ensure that documents belong to both the majority category and the specific subcategory. Each search retrieves up to 30 documents, where documents refers to chunks of text, with a similarity threshold of 0.67. In practice, this means that for each subcategory belonging to the majority category, at most 30 chunks are retrieved, provided they exceed the similarity threshold. The choice

of 30 stems from testing that showed this was enough to get all chunks from a document if needed. The threshold valued of 0.67 is chosen empirically, balancing the need to include all relevant chunks while minimizing irrelevant ones. This approach ensures comprehensive coverage across related content clusters.

Finally, results from all subcategory searches are combined into a unified collection and sorted in descending order of relevance score. This ensures that the most pertinent documents are prioritized regardless of their subcategory origin, preventing any single subcategory from dominating the retrieval results.

```
@hook
def after_cat_recalls_memories(cat) -> None:
    k = 30
    threshold = 0.67

    if "majority_label" in cat.working_memory:
        majority_label = cat.working_memory.majority_label

        all_points, _ = cat.memory.vectors.declarative.get_all_points()

        sub_categories = set()
        for point in all_points:
            metadata = point.payload.get('metadata', {})
            if (metadata.get('category') == majority_label and
                'sub_category' in metadata):
                sub_categories.add(metadata['sub_category'])

        user_message = cat.working_memory.user_message_json.text
        user_message_embedding = cat.embedder.embed_query(
            user_message)

        for sub_cat in sub_categories:

            memories = cat.memory.vectors.vector_db.search(
                collection_name='declarative',
                query_vector=user_message_embedding,
                query_filter=Filter(
                    must=[
                        FieldCondition(
                            key="metadata.category",
                            match=MatchValue(value=majority_label)
                        ),
                        FieldCondition(
                            key="metadata.sub_category",
                            match=MatchValue(value=sub_cat)
                        )
                    ]
                ),
                with_payload=True,
                with_vectors=True,
```

```
        limit=k,
        score_threshold=threshold,
        search_params=SearchParams(
            quantization=QuantizationSearchParams(
                ignore=False,
                rescore=True,
                oversampling=2.0,
            )
        ),
    )

    if memories:
        langchain_docs_from_points = []
        for point in memories:
            langchain_docs_from_points.append(
                (
                    Document(
                        page_content=point.payload.get("
                            page_content"),
                        metadata=point.payload.get("
                            metadata") or {},
                    ),
                    point.score,
                    point.vector,
                    point.id,
                )
            )

        cat.working_memory.declarative_memories.extend(
            langchain_docs_from_points)
        log.info(f"Added {len(langchain_docs_from_points)}
            documents from sub_category: {sub_cat}")

    if hasattr(cat.working_memory, 'declarative_memories') and
        cat.working_memory.declarative_memories:
        cat.working_memory.declarative_memories = sorted(
            cat.working_memory.declarative_memories,
            key=lambda x: x[1],
            reverse=True
        )
        log.info(f"Total declarative memories: {len(cat.
            working_memory.declarative_memories)}")
        log.info(f"Declarative memories: {(cat.working_memory.
            declarative_memories)}")
```

After that, the Agent input, composed of different elements including the user query and the retrieved declarative memory, can be enriched using the prompt enrichment hooks *agent_prompt_prefix* and *agent_prompt_suffix*. In this framework, only the hook *agent_prompt_prefix* is employed, which allows to edit the prefix of the *Main Prompt* that the Cat provides to the Agent. This enables the personalization of the LLM's personality

and its general task. The Agent input is then passed to the Memory Agent which, with the support of the LLM, produces the final response. This process occurs through the *Memory Chain*, a simple chain that takes the user’s input, in this case the list of selected tweets, together with the retrieved contextual memories, formats them into the Main Prompt, and then submits it to the LLM.

The prompt is the result of a combination of theoretical analysis and empirical testing, aimed at defining the most effective structure. It is organized into functional sections, each of which defines specific aspects of the agent’s behavior:

- **Role Definition** The LLM is configured as an intelligence analyst specialized in the study of social media profiles. This establishes the professional context and sets the qualitative standards expected for the output. Previous studies show that context enrichment of this kind improves accuracy, helping the model to better understand the role or perspective of the person or entity being addressed. This essentially acts as a stylization of the final response.
- **Input Specification** The prompt specifies the nature of the inputs: preprocessed and filtered tweets, retrieved documents with their corresponding nomenclature, and the way references to documents must be made (via the subcategory metadata). Summarizing inputs before giving them to the model has appeared beneficial in theoretical studies. Notably, the term “context” is used in the prompt to refer chunks, as it is intuitively understandable by the model.
- **Critical Instruction for Precision** Two fundamental rules are set to ensure precision. First, the LLM must rely exclusively on the retrieved contextual documents, avoiding contamination with external or hallucinated knowledge. Second, all relevant correspondences must be reported, regardless of their apparent significance.
- **Task Specification** The task is defined through a structured, multi-step process for each tweet in the list: verify if similarities exist between the tweet and the retrieved document chunks; if yes, highlight the name of the groups, via the *sub_category* metadata, associated with those chunks; for each identified group, identify similarities between the tweet and the group’s characteristics, reporting them into the final report along with supporting definitions or explanations extracted from the documents; recognize and report ambiguous terms in the tweet that could relate to more than one subcategory, providing all possible interpretations. The concept of “match” is very important and could include: lexical matches (single words or entire sentences), symbolic matches (hashtags and emojis), and communication pattern.
- **Hallucinations Safeguards** The prompt includes safeguards against hallucinations, specifically false positive. The model is explicitly instructed not to invent connections between tweets and RAG documents. If no real similarity exists, it is preferable to leave the section empty rather than invent a match.
- **Input and Output Format Specification** Examples of both input and output formats are provided to guide the model. Studies have shown that LLMs perform better when instructions are precise and structured. Accordingly, the prompt

defines a standardized output format that enhances the final analysis: individual analysis for each tweet, detailed reasoning for each identified group, inclusion of definitions and descriptions extracted from documents, a specific template for tweets without explicit associations.

The following code excerpt shows the implementation of the hook function where the prompt prefix is defined. Non-essential portions of the prompt have been replaced with [...] for brevity.

```
@hook
def agent_prompt_prefix(prefix, cat):
    settings = cat.mad_hatter.get_plugin().load_settings()

    prefix = """
    You are an intelligence analyst specialized in social media
    profile analysis and your task is generate a professional
    and detailed dossier.

    You will be given:
    - A set of tweets
    - A collection of documents <context> retrieved based on
      similarity to the tweets.
      - Documents are named using the format '<category>-<
        group_name>' [...].
      - Each document describes the linguistic characteristics,
        symbolism and hashtags, emojis, and communication
        patterns of the specific group [...]

    **CRITICAL INSTRUCTION:** Base your analysis EXCLUSIVELY on the
    provided <context> documents. Do not use any external
    knowledge or assumptions. Only identify similarities that
    are explicitly described in the retrieved documents.

    **MANDATORY RULE:** Even if a tweet contains only a single
    EXPLICIT match (e.g., one emoji, one hashtag, or one term)
    described in a document, you must still include the
    associated group in the output. Each group with any
    EXPLICIT match must be reported, regardless of how minor
    the match is.

    [...]

    ## Your task:

    For **EACH tweet** in the list:
    1. **Check whether the tweet shows EXPLICIT similarities** to
      any of the retrieved documents.
      - Look for EXPLICIT matches of: words, hashtags, emojis, or
        communication patterns
    2. **If there are EXPLICIT similar documents**, identify the **
      group or groups** associated with the tweet.
```

- If the documents relate to **multiple groups**, list **ALL** relevant groups.
 - **IMPORTANT**: A single tweet element [...] can appear in multiple documents with different meanings.
When this happens, specify **ALL** matching groups and provide the respective meanings for each group.
3. For **EACH** associated group, provide a **clear explanation** of **all relevant similarities** between the tweet and that group's:
- **linguistic characteristics**
 - **Hashtags**
 - **emojis**
 - **communication patterns**
- as described in the documents.
4. **Include explicit definitions or explanations** from the documents when available.

Examples of explicit definitions or explanations

CASE 1: Terms

- Document shows: "martyrdom" : dying for the cause of Islam'
- Your output **must contain** the explanation: "dying for the cause of Islam"

CASE 2: Hashtags

- Document shows: "#endthepain" : expression used to indicate suicidal ideation'
- Your output **must contain** the explanation: "expression used to indicate suicidal ideation"

[... other examples from CASE 3 to CASE 6 ...]

CRITICAL INSTRUCTION: If you cannot find **EXPLICIT** match between a tweet and any of the provided documents, respond: "No group association found - insufficient similarities in provided context"

REMEMBER: It is **BETTER** to report "No group association found - insufficient similarities in provided context" than to invent connections that don't exist.

[...]

Input format:

[...]

Output format (in Markdown):

Tweet 1: "[Tweet 1 text]"

```

- **Assigned group(s):**
- **Group 1:**
  - Name: [Group name from source file name: <group_name_1>]
  - Reasoning: [Detailed explanation based on documents from
    <category>-<group_name_1>]
- **Group 2:**
  - Name: [Group name from source file name: <group_name_2>]
  - Reasoning: [Detailed explanation based on documents from
    <category>-<group_name_2>]

[...]

### **Alternative format for tweets with no group association
:**
## **Tweet X:** "[Tweet X text]"
- **Assigned group(s):** No group association found -
  insufficient similarities in provided context

"""

return prefix

```

The output of the LLM is reformatted within the *before_cat_sends_message* hook, in order to include additional information such as the dossier creation date and the list of all scraped tweets with their corresponding labels.

The framework is able to produce structured reports. For each analyzed tweet, a detailed qualitative analysis is provided, including the association of the tweet to none, one or more than one subcategories. The analysis highlights significant elements such as linguistic and symbolic evidences, including hashtags, emojis, or specific phrasing, and communication pattern, which are explicitly linked back to their original source definitions and descriptions present in the retrieved RAG documents, providing detailed explanations with direct citations that analysts can independently verify and contextualize. Additionally the dossier includes administrative information such as target profile URL, dossier creation data, and scraping metadata including the number of scraped tweet and date ranges. The quantitative outputs feature detailed classification tables with scraped tweet content, timestamps, mentioned users, and assigned classifications, along with aggregated statistics showing the count and percentage distribution of different categories and the identification of the majority label. Complete examples of generated reports are presented in Section 6.

The quality of the report depends not only on the retrieved documents and prompt structure but also on the LLM used. As demonstrated in the Section 6, larger and more widely-used models tend to perform better compared to smaller, locally hosted models.

A positive aspect of the RAG approach is its transparency. Once the report is generated, clicking to the “?” button allows users to view the list of retrieved chunks that were used to generate the response. This feature is particularly valuable in sensitive contexts such as intelligence analysis, where it is crucial to provide documented justification and reduce the likelihood of spreading false or unsupported information.

The generated report is designed to support, rather than replace, the work of an intelligence expert, acting as an amplifier of intelligence processes. Instead of drawing final conclusions, the system automates the most time-consuming preliminary tasks, such as filtering vast amounts of scraped tweets and highlighting recurring linguistic or symbolic patterns. The structured report accelerates the analytical workflow, reducing the time needed for the initial review, allowing analysts to focus on higher-level interpretation, contextual reasoning, and decision-making. Crucially, the tool does not declare definitive judgments such as “this user is a terrorist” or “this profile contains high level of depression content”. Instead, it highlights specific overlaps between the analyzed content and documented extremist patterns or indicators from studies on depression and suicide. The responsibility for validation, contextualization, and threat assessment remains with the human expert.

Chapter 6

Evaluation and Testing

6.1 Evaluation Objectives

This chapter focuses on the testing phase of the developed framework, with the objective of validating the effectiveness of the proposed approach in identifying, classifying, and reporting potentially high-risk content on social media through the use of LLM and RAG technologies.

The testing phase was designed to specifically evaluate the cognitive and analytical capabilities of the system, structured as a sequential evaluation of the three core components of the pipeline (classification, retrieval, generation), allowing for systematic validation and optimization of each stage before proceeding to the next.

The first phase focused on evaluating the classification capabilities of the local LLM, testing its ability to understand and interpret complex and potentially ambiguous content present in tweets. This phase involved prompt engineering experimentation to optimize the model's performance in distinguishing between neutral content and high-risk content across the identified categories. Various testing approaches were employed, including batch processing with various batch sizes and different prompt formulations.

The second phase concentrated on optimizing the retrieval component of the RAG architecture. This involved experimentation with metadata assignment strategies to documents. Particular attention was paid to threshold optimization for the retrieval process, ensuring that only chunks truly relevant to the analysis were retrieved, avoiding the inclusion of irrelevant content that could compromise quality. The testing included challenging edge cases to validate system robustness in complex scenarios.

The final testing phase focused on optimizing the generation component through extensive prompt engineering to enhance report quality. This phase included a comparative analysis between the local LLM and larger model to evaluate how different model characteristics affected output quality. The testing encompassed multiple scenarios, including edge cases, to ensure robust system performance under challenging analytical conditions. Special attention was given to developing prompts capable of handling several functions: LLM's role definition, mapping each analyzed tweet to one or more subcategory with clear motivation based on identified similarities, and generating well structured and comprehensive reports with appropriate explanations and contextual details.

This research prioritized the qualitative aspects of analysis, focusing on cognitive performance and interpretive precision of the system. Consequently, some other aspects that, while relevant for production environment implementation, fall outside the primary objectives of this research were omitted. Specifically, performance metrics, such as processing times and computational consumption were not analyzed. Additionally, the relationship between the number of analyzed tweet and the performance deterioration was not extensively investigated. For example, the behavior was not tested in scenarios where the context window limit or the maximum number of input tokens that the embedder can process is reached due to the number of analyzed tweets.

This choice was driven by several considerations. First, the research operated under significant resource constraints, as testing required the use of paid LLM services, making extensive performance evaluations economically unfeasible. Second, the available dataset was limited to a small number of tweets and RAG documents, not sufficient for scalability analysis. Additionally, the current implementation requires human supervision for quality validation of the reports, making it impractical to test scenarios involving the analysis of huge amounts of tweets without proportional human oversight resources. Most importantly, as this project was born as a proof-of-concept developed from scratch, giving priority to the demonstration of technical feasibility and analytical effectiveness over performance optimization was a necessity. This approach allowed attention to be concentrated on concept validation and effectiveness of the proposed approach, providing a foundation for future developments that can address performance and scalability aspects in subsequent phases of the research.

6.2 Dataset Creation and Preparation

6.2.1 Dataset for Preliminary Classification

An important component of the proposed pipeline is the preliminary classifier that enables the assignment of each tweet to one of the five defined categories, in order to determine the predominant aspect present in the analyzed profile and perform subsequent filtering of RAG documents.

To evaluate the classifier, including across diverse prompt variations, a golden set of 200 tweets was created, balanced across five categories. Half of the tweets were randomly sampled from publicly available, pre-labeled datasets. Most of these datasets were sourced from Kaggle, including: “Suicidal Tweet Detection Dataset” and “Students Anxiety and Depression Classify Model” for suicide and depression; “How ISIS Uses Twitter” and “Religious Texts Used By ISIS” for terrorism; “Cyberbullying Classification”, “Cyberbullying Dataset”, “Racism-Detection”, “Anti-LGBT Cyberbullying Texts”, “Classified Tweets”, and “Cyber Troll dataset” (from Zenodo), for hate speech and cyberbullying; “Phishing Email & SMS Dataset with NLP Categories” for cybercrime and hacking; “Social Media Sentiments Analysis Dataset” for neutral tweets. The remaining half of the data was generated synthetically using GPT-3.5, useful for the generation of edge cases in the initial test.

This evaluation dataset of 200 annotated tweets, is divided into two distinct test sets to assess both syntethic-to-real generalization and cross-domain robustness: in Test Set

1 are inserted the 100 synthetically generated tweets, in Test Set 2 the 100 real-world tweets. Test Set 1 was used to evaluate all candidate configurations, using performance metrics from this phase to guide the selection of the optimal configuration; once the best-performing configuration was identified based solely on Test Set 1 results, the configuration was applied to Test Set 2 for final performance assessment. This methodology aims to prevent overfitting to the final test data and to quantify the synthetic-to-real domain transfer capability.

For each category, was sought to test tweets that covered the widest range of sub-patterns. For example, for terrorism and extremism, tweets were tested that fall into ideological propaganda, recruitment and radicalization, news and claims, threats, and dissemination of multimedia material. Regarding cyberbullying and hate speech, was considered racism, sexism, homophobia, threats of violence, sarcastic passive-aggressive remarks, gaslighting, offensive memes, offensive language, and social exclusion. For hacking and cybercrime, was tested tweets related to social engineering, promotion of malware or vulnerabilities, attack claims, hacktivism, and warnings. Finally, for depression and suicide, was included passive-aggressive tweets, expressly sad or depressive tweets, references to death or suicide, and expressions of desire to be heard or helped. Neutral tweets were also considered.

The explicitness of these aspects was also taken into consideration, testing tweets with three levels of explicitness (easy, medium, high).

Although was attempted to test tweets assignable to only one of the five categories, in reality there can be edge cases where a tweet presents elements that make it assignable to multiple categories. For example, hacktivism involves tweets that could be associated with both terrorism/extremism and hacking/cybercrime, although in most cases there is still a predominantly present category. While aware of these edge cases, was decided not to include them in the test set and opted for multiclass classification instead of multilabel classification, since the objective of this classification is only to understand the predominantly present aspect in the profile (simpler to calculate this way), and because was decided to analyze tweets in the RAG phase regardless of their classification. This means that even if a tweet that could be classified as both terrorism and cyberbullying is classified as cyberbullying, and the majority label is terrorism, its analysis in the RAG phase with terrorism documents is not excluded a priori.

6.2.2 Dataset for RAG Evaluation

The evaluation of RAG systems requires specialized datasets, whose construction can follow various methodological strategies ranging from leveraging existing resources to creating entirely new data tailored for specific evaluation aspects. The creation of documents for RAG testing represents a critical and often complex procedure.

The advent of powerful LLMs has fundamentally transformed the dataset construction process, facilitating the creation of synthetic data, artificially generated data that mimics the characteristics of real-world data. Synthetic data generation proves particularly valuable when real data is scarce, expensive to obtain, or subject to privacy constraints. The use of synthetic data offers several key benefits: cost-effectiveness, reduces expenses associated with data collection; scalability, enables the creation of large quantities of data

within short timeframes; customization, allows for data tailored to specific scenarios and requirements; privacy compliance, avoids complications related to sensitive data handling. These advanced tools enable the creation of synthetic inputs based on RAG documents or the generation of ground truth, the expected outputs, starting from specific inputs [322]. An example of tool is the open-source library RAGAS, which exemplify the evaluation by taking RAG documents and LLMs to generate inputs, in the form of questions, and the correspondent ground truth. RAGAS, mainly developed for question-answering contexts, one of the most common RAG use cases, provides the flexibility to choose specific LLMs for data generation and evaluation, as well as embeddings for understanding semantic meaning of texts and improving retrieval processes [323].

For the RAG system proposed in this thesis, dataset creation presented significant challenges for several reasons.

The initial difficulties emerged during the RAG document creation phase, which is critical since the quality of RAG documents, both in terms of content and structure, directly influences final results. The required documents were not readily available, necessitating the creation of ad-hoc documents for the proposed framework. This proved challenging due to the need to create highly sensitive documents for a context requiring specialized experience and knowledge, without having the specific expertise or sufficient starting material. In fact, the idea behind the proposed framework was that it would serve as a tool for a team of experts, specialists in terrorism, sociology, psychology, cybercrime and many others, who would generate the documents using various information sources, to then pass them as RAG knowledge base. Consequently, the testing was limited to creating documents within the terrorism category, one for each subcategory corresponding to different terrorist groups. This limitation arose from the lack of ready-made documents and general data scarcity, with terrorism being chosen due to the availability of more source material for creating some documents based on real information. Some documents were created by extracting and expanding information from existing documents, papers, and analyses available online, while others were generated using LLMs after providing detailed descriptions of required characteristics. The created documents exhibited varying levels of precision, detail, and length. This variation served to demonstrate both that detail levels do not affect the proposed pipeline and that, for hypothetical scenarios, the system operates independently of the LLM's knowledge base, thus testing the pipeline across diverse scenarios.

Beyond documents, the system required tweets for comparison with RAG documents to identify linguistic similarities. This necessitated creating realistic tweets ad-hoc, derived from genuine terrorism tweet datasets and incorporating linguistic similarities with RAG documents, or generated from scratch using LLMs. Additionally, tweets were created to test edge cases scenarios where, for example, a single tweet contained similarities with multiple subcategories simultaneously. The focus goes beyond individual tweets to include various tweet combinations for testing, covering cases from profiles with tweets all associated with the same group to profiles with tweets associated with different groups.

For testing purposes, it was essential to establish ground truth from the outset to simplify the evaluation phase. This involved defining how each tweet should be classified relative to one or more subcategories and how this classification should be justified,

highlighting common words, hashtags, emojis, phrases, and other relevant features.

Behavioral Documents

As described in Subsection 5.3.1, the creation of RAG documents requires sufficient material and preferably follows the suggested structure. The proposed tutorial was effectively implemented during the test phase to create the RAG documents.

Six documents were created: three related to the social media linguistic behavior of real terrorist and extremist groups, specifically ISIS, FETO, and PKK; and three related to invented groups, specifically Blood&Honor, EAT, and Jamaat.

The documents vary in terms of: data sources, some invented, others based on real data; thematic focus and main group characteristics, for example, some groups are related to religious terrorism (ISIS and Jamaat), others to climate and environmental extremism (EAT), others whose main ideology is white supremacy (Blood&Honor), and others related to Turkish political extremism (PKK and FETO); document length and richness, for instance, ISIS has the most detailed document, containing both the most commonly used words and hashtags, as well as coded emojis and behavioral patterns with related examples, or PKK, which presents only the most commonly used words and hashtags; linguistic and ideological similarities, some documents present linguistic and ideological similarities (ISIS and Jamaat), while many are very different from each other, without any similarity.

To illustrate the creation process, the ISIS document serves as a comprehensive example. Data were first collected from available online sources. Specifically, regarding the most commonly used words and hashtags with their respective meanings or descriptions, as well as sentence examples, these were extracted from: a recent academic analysis conducted on terrorist and extremist groups on X, focusing on ISIS, based on a dataset of tweets collected by a research group from Texas A&M University between 2014 and 2015 [30]; an article related to jihadist propaganda conducted by ISIS on X, based on a dataset of over 50 million tweets [35]; a paper focusing on linguistic analysis of pro-ISIS X users, monitored from 2015 to 2016 [29]; a study conducted by the University of Southern California based on more than 1.9 million tweets posted by ISIS members [34]; a research conducted within the extremism program at George Washington University, containing a huge amount of tweet examples [27]. The emoji codes frequently used with double meanings, used for example to coordinate attacks or redirect to Telegram channels without being detected by algorithms or non-members of the organization, were extracted from a report published by the Institute of Strategic Dialogue [37] and the Joint Intelligence Bulletin [36].

Based on the collected data and following the structural guidelines, the document was created. The following examples illustrate each major section of the document:

- Frequently used words:

Linguistic Characteristics of the terrorist group ISIS

- ‘Dawla’: Arabic for state. ISIS uses this term to refer to their self-declared Islamic State, attempting to normalize it as a functioning government.

- ‘Assad’: Refers to Bashar al-Assad, the president of Syria. ISIS opposes him and often refers to him as an enemy.
- Frequently used hashtags:
 - ### Religious Justification and Recruitment Hashtags
 - ‘#PeaceBeUponHim’: Islamic honorific for prophets, especially Muhammad
 - ‘#APrayerForMyLord’: Used to spiritualize violent acts
- Frequently used coded emojis:
 - #### Religious or Sectarian Symbolism
 - ‘:star_of_David:’ -> Known target: Jewish site
 - ‘:scissors:’ -> ISIS-supporting channel (Libya)
- Communication pattern and behavioural description with examples:
 - ## Communication Patterns of the Terrorist Group ISIS
 - Publishing **interviews** with jihadist figures
 - Example of ISIS tweet: ‘‘Watch the exclusive interview with Abu Bakr al-Baghdadi, the leader of the Islamic State, discussing the future of the Caliphate and the duty of every Muslim to join the jihad. [URL]’’
 - Often posted with a link to video or transcript, shared on X or Telegram.

The other documents based on real data (FETO and PKK) were obtained in the same way, using academic sources and available research reports. The synthetically created documents (Blood&Honor, EAT, and Jamaat) were created synthetically via GPT-3.5, using the already created documents as reference to maintain structural consistency and realism in linguistic and behavioral characteristics.

The document set described above constitutes the test set used for the final evaluation of the proposed pipeline. However, the pipeline development required parameter selection, specifically determining the threshold for document retrieval and identifying the most effective prompt for LLM-based report generation. While the test set was used exclusively to calculate final metrics and assess performance after parameters were selected, the parameter selection process itself was conducted on a separate dataset. Unlike the previously described test set, this development dataset was created entirely synthetically, primarily due to time constraints. This synthetic dataset focused less on content quality, potentially featuring less realistic vocabulary and hashtags compared to the carefully curated test set. However, the document structure remains consistent with that of the test set. Specifically, five synthetic documents were created, ranging from the document of the group named Sacred Flame Cult, a religious extremist terror group, to Red Dawn Liberation Front, a political terror organization.

Tweets

The tweets required for testing the RAG phase were selected to be classified primarily as terrorism-related content, containing elements similar to the RAG documents, while covering the broadest possible range of test cases.

Tweets were not collected directly from actual terrorist profiles due to various factors including the difficulty and risk of identifying such profiles, as well as technical, legal, and ethical concerns regarding automated collection of personal data from X. Instead, it was decided to use publicly available datasets and leverage other LLMs to generate tweets suitable for comparison with the RAG documents.

For ISIS related tweets, was utilized the Kaggle dataset “How ISIS Uses Twitter”, containing over 17,000 tweets extracted from profiles of more than 100 ISIS supporters on X starting from 2015. Tweets were randomly selected and filtered to ensure they were in English, of varying lengths, clearly terrorism related and significant in terms of tone and content, as some tweets labeled as terrorism related showed no clear terroristic associations upon textual analysis alone. Since the vast majority of sources used to create the ISIS RAG knowledge base were based on data collected from X between 2014 and 2016, minimal modification was necessary to make these tweets comparable with the RAG document, as they already exhibited high similarity. Additional ISIS tweets were sourced from various online documents, including screenshots from news articles. For tweets related to other terrorist groups, the generation started with the Kaggle tweets and applied varying degrees of modification using GPT-3.5 to create tweets that would show similarity with other terrorist organizations.

This process resulted in 50 tweets, divided into several categories: clearly terrorism-related tweets, associated exclusively with one of the six groups; tweets containing elements not present in the documents, designed to test the system’s potential tendency to generate incorrect information; tweets sharing elements across multiple groups, aimed at testing edge cases and ensuring the system correctly attributes content to all relevant groups, providing appropriate meanings and contextualizations; tweets rich in emojis, to evaluate the system’s ability to handle visually complex content; tweets that do not contain words, hashtags, or emojis from the documents, but display similarities in communication patterns, such as URL usage, to test the recognition of structural or behavioral patterns; tweets with modified word forms or typos, to assess the system’s robustness when faced with non-standard or imperfect content

For RAG phase testing purposes, these tweets were specifically organized into profiles containing a maximum of 10 tweets each. As highlighted in Section 5.4.1, the number of tweets to analyze together during the RAG phase depends on various requirements and constraints. While evaluating the relationship between tweet volume and performance degradation would be interesting future work, due to practical limitations and focus on specific aspects rather than others, as specified at the beginning of this Chapter, was chosen to test profiles with a maximum of 10 tweets to ensure report lengths remain acceptable, allowing for detailed human supervision analysis without prohibitive time requirements.

Six profiles were created to test the RAG phase across different scenarios: single-group scenario, the most straightforward case, where the analyzed profile publishes tweets that can be linked exclusively to one terrorist group, specifically ISIS and EAT; no-match scenario, a profile whose some tweets show no similarity with any documents, designed to test RAG’s ability to avoid generating false information and instead indicate that there is insufficient evidence to associate these tweets with any groups in the RAG knowledge

base; totally invented scenario, in which all analyzed tweets are synthetically created, to demonstrate that the analysis is based on RAG knowledge and not model knowledge; borderline multi-group scenario, a profile containing tweets similar to multiple groups, particularly ISIS, with some tweets showing borderline similarity to content from the Jamaat group; a multi-group profile where each tweet is linked to a single group, but the overall tweet list covers multiple groups; a multi-group profile in which individual tweets can be associated with multiple groups.

In parallel with tweet creation and subsequent profile development, ground truth was established, defining which group each tweet should theoretically be associated with and which parts of each tweet should theoretically be highlighted as similarities and thus serve as justification for the classification.

Similarly to the RAG document approach, synthetic tweets were created for the initial testing phase in addition to the tweets used for final evaluation. These synthetic tweets correspond to the RAG documents in the development dataset, enabling comprehensive testing of both retrieval mechanisms and prompt configurations.

6.3 Evaluation of Preliminary Classification

After creating the test set as described in Section 6.2.1, defining the prompting strategy (few-shot and zero-shot), and establishing the classification approach (multiclass classification) as described in Section 5.6, this chapter focuses on the effective testing of preliminary classification, ranging from metrics used to quantitative and qualitative analysis of different configurations.

6.3.1 Performance Metrics

The following metrics were selected to provide a comprehensive evaluation of the multi-class classifier’s performance:

- **Accuracy:** Overall proportion of correctly classified tweets from the total number of predictions made by the model, providing a general performance indicator. Accuracy is straightforward to interpret, however, it has downsides. While it provides an estimate of the overall quality, it disregards class balance and the cost of different errors. For this reason, it is necessary to consider other metrics.
- **Precision per class:** Proportion of true positives among all positive predictions for each category, crucial for understanding false positive rates. Precision is a suitable metric when it is prioritized “being right” when assigning the positive class over “detecting them all” (emphasizing false positives over false negatives).
- **Recall per class:** Proportion of true positives among all actual positive instances, important for identifying missed classifications. The recall metric is the opposite of precision (emphasizing false negatives while giving less importance to false positives). Although calculating precision and recall for each class can result in numerous metrics, it is useful to evaluate the performance of a particular class and how well the classifier can distinguish this class from others.

- **Macro-averaging Precision:** The average of the precision across all classes to get the final macro-averaged precision score, giving equal weight to each class, regardless of the number of instances.
- **Macro-averaging Recall:** The average of the recall across all classes to get the final macro-averaged recall score, giving equal weight to each class, regardless of the number of instances. Macro-averaging is particularly useful when all classes are equally important, or when dealing with an imbalanced dataset where it is necessary to ensure that each class contributes equally to the final evaluation. The drawback is that the classifier may appear to perform worse due to low performance in a small and less relevant class, since this class still contributes equally to the overall score.

All reported metrics represent the average performance across independent run, ensuring statistical reliability of the results.

6.3.2 Experimental Design

Example Selection Strategy For Few-Shot Testing The experimental design began with creating a comprehensive pool of 70 candidate examples, with 10-15 examples per category, covering various sub-patterns within each category, including neutral tweets. These examples were selected as the most representative for each sub-pattern, extracted from the previously introduced datasets or synthetically created, with particular attention to selecting borderline cases and covering diverse styles, lengths, and complexity levels of tweets.

From this initial pool, examples were systematically selected starting with one example per sub-pattern, varying among the available examples for each sub-pattern to cover different scenarios and edge cases. Testing different examples associated with the same sub-pattern, was performed to identify which examples yielded the best results.

Following initial testing with one tweet example per sub-pattern, the number of examples was progressively reduced. Within each category, examples were ranked by relevance, and those belonging to less significant sub-patterns were gradually removed. In particular, trivial, redundant, or already well-represented examples were eliminated. This elimination process continued until further reduction resulted in performance degradation according to the evaluation metrics.

The final configuration was characterized by approximately one example per explicitness level for each category, plus examples for the most challenging sub-patterns.

Shot Configuration Testing The experimental design employed few-shot prompting with one example for each specific sub-pattern, progressively reducing the number of tweets to evaluate their necessity. This approach was adopted because, generally, a higher number of tweets corresponds to increased token count and therefore higher cost, time, and reduced efficiency. It was also suggested, by the theory, to avoid using too many examples and just few well-chosen ones.

Additionally, zero-shot prompting was tested without examples but with detailed descriptions of what should and should not be included in each of the five categories.

Example Ordering Strategy This testing phase did not focus extensively on example order permutation, as preliminary tests revealed that the model was sufficiently robust to not be significantly affected by example order influence. The class ordering was grouped by category: first all terrorism examples, then cyberbullying, hacking, depression, and neutral, with the order of examples within each class chosen randomly.

Class Distribution Regarding the number of examples per category, the approach aimed to maintain a relatively balanced number of examples, subordinated to the number of examples necessary to cover the range of possible sub-patterns within each category.

Prompt Formulation Variations Initially, the few-shot prompt included only the list of possible labels (categories), the list of examples for each category, specifications to provide only the label as output, and guidelines on output format. Subsequently, a specification was added instructing the LLM, in cases of uncertainty in label assignment, to assign the label based on the most relevant aspect. This addition resulted in performance improvements.

The prompt was also tested without the list of examples for each category, substituted with detailed description of the categories. The final prompt version is reported in Section 5.6, containing both the description and the list of examples.

Batch Size Configuration Different batch sizes were systematically tested to evaluate their impact on classification performance: batch of 100 tweets (all together), batch of 50 tweets, batch of 30 tweets, batch of 10 tweets, with tweets within each batch processed simultaneously.

Each batch size configuration was evaluated using the same metrics to determine the optimal balance between processing efficiency and classification accuracy.

6.3.3 Testing Methodology

The testing methodology followed a structured, multi-stage approach designed to identify the best configuration across multiple dimensions. The process was organized into distinct phases, each targeting specific objectives.

The testing was conducted using only the local model Llama 3.1 8B Instruct. The introduction of larger LLM, the GPT one, occurred subsequently when the local model’s performance was no longer satisfactory, necessitating verification of whether the problem was related to the model or to the proposed framework. However, the quality of the classifier was confirmed during the RAG analysis also for GPT.

Phase 1: Batch Size Optimization The methodology began with an established few-shot configuration using a predefined set of examples. This baseline configuration was tested across different batch sizes (100, 50, 30, and 10 tweets per batch) to determine the optimal balance between token cost and classification performance. For Batch sizes less than 10 tweets, the LLM API limits were reached. Each batch size was evaluated on Test Set 1. Through this analysis, a batch size of 50 tweets emerged as the optimal

compromise between cost efficiency and metric results. Although batch sizes of 30 and 50 yield comparable performance, using 50 tweets per batch proved less token consumption, making it the preferred choice in this practical settings. Large batches may suffer under attention dilution and reduced focus individual tweets; small batches may suffer from classification inconsistencies.

Batch Size	Accuracy	Macro Precision	Macro Recall	Selection
100	0.865	0.886	0.863	Selected
50	0.948	0.952	0.950	
30	0.927	0.927	0.930	
10	0.896	0.901	0.896	

Table 6.1. Validation results on Test Set 1 (Synthetic).

The final validation on Test Set 2 (real) with the selected batch size of 50 yielded: 0.891 (Accuracy), 0.912 (Macro Precision), 0.889 (Macro Recall). The performance degradation from Test Set 1 (synthetic) to Test Set 2 (real) reflects the expected synthetic-to-real domain gap, additionally confirmed in subsequent phases. This controlled performance drop demonstrates the model’s ability to generalize across domains while highlighting the inherent challenges in transitioning from controlled synthetic data to the linguistic variability and noise present in real-world content.

Phase 2: Example Configuration Optimization With the optimal batch size determined, the focus shifted to optimizing the few-shot example configurations. Three different example configurations were firstly designed and evaluated, each aimed at covering all possible sub-patterns, including tweets of varying styles, lengths, complexities, and edge cases, within the classification task. Specifically, for each sub-pattern related to the four high risk categories (excluding neutral), two possible tweets were included in the example set, labeled as 1 and 2. In Example Configuration 1, all tweets labeled with 1 were selected; in Configuration 2, all tweets labeled with 2 were selected; in Configuration 3, the selection was random, mixing tweets labeled with 1 and 2. Each configuration was tested using the established batch size of 50 tweets from Test Set 1.

Performance analysis revealed that different configurations performed better for different types of tweets, leading to the strategic decision to combine the most effective elements from each configuration into a unified approach.

Example Configuration	Accuracy	Macro Precision	Macro Recall
1	0.948	0.952	0.950
2	0.948	0.955	0.946
Random	0.958	0.963	0.957
Combination	0.958	0.964	0.960

Table 6.2. Validation results on Test Set 1 (Synthetic).

The final validation on Test Set 2, using the combined example configuration, resulted

in the following performance metrics: 0.910 (Accuracy), 0.924 (Macro Precision), 0.910 (Macro Recall). As seen in Test Set 2, there was a slight decrease in these metrics, likely due to the shift from synthetic to real-world tweets.

Phase 3: Prompt modification Upon analyzing the behavior of the prompt, it was observed that specifying clear instructions within the prompt for handling uncertainties led to improvements. Specifically, in edge cases, where tweets might exhibit multiple aspects, the model would sometimes assign one label, and at other times, a different label. To address this, the prompt was modified to instruct the model that, in case of uncertainty about the label, it should prioritize the most predominant or relevant aspect. This approach aims to ensure that the model not only assigned the most obvious label (which was the expected one) but also exhibited greater decisiveness, leading to improved performance.

Prompt Configuration	Accuracy	Macro Precision	Macro Recall
Combination (without Instruction)	0.958	0.964	0.960
Combination (with Instruction)	0.979	0.981	0.979

Table 6.3. Validation results on Test Set 1 (Synthetic).

The final validation on Test Set 2, with the added prompt instruction, resulted in the following performance metrics: 0.931 (Accuracy), 0.948 (Macro Precision), and 0.929 (Macro Recall). As with previous tests, a slight decrease in these metrics was observed on Test Set 2, which can be attributed to the transition from synthetic to real-world tweets.

Phase 4: Example Set Minimization The final phase concentrated on optimizing efficiency by systematically reducing the number of examples in the combined configuration, with the goal of identifying the minimal essential set that maintained high performance, while also reducing the number of tokens consumed. Given that the original configurations were designed to cover all sub-patterns, this reduction process sought to perform elimination without compromising the representation of critical patterns.

To achieve this, redundant tweets (those already covered by other examples, either in terms of topic or tone) were removed, as well as ambiguous tweets that did not improve the model’s performance and often led to misclassifications. The next step involved reordering the tweets in the combined set by importance, prioritizing clear, direct tweets, including tweets that addressed edge cases or that the model may not naturally classify correctly. Tweets that were less explicit, ambiguous, or had low semantic density are candidate to be progressively eliminated. This process was based on an analysis of previous errors made by the model in relation to the examples it had seen.

Specifically, once the candidate tweets for elimination were identified, for each candidate, metrics, computed over Test Set 1, were compared before and after its removal to determine whether it was indeed possible to eliminate it. The result of this testing led to the removal of unnecessary tweets, leaving one example for each level of significance.

Additionally, more examples were integrated where the classifier was showing more difficulty, such as in the case of social engineering sub-pattern, where additional examples were added.

The final validation on Test Set 2 for the ultimate model combination yielded the following performance metrics: 0.954 (Accuracy), 0.962 (Macro Precision), and 0.951 (Macro Recall). For Test Set 1, recursive elimination was performed until further removal led to a decline in performance, dropping below 0.979 (Accuracy), 0.981 (Macro Precision), and 0.979 (Macro Recall). Interestingly, this reduction led to an improvement in performance on Test Set 2.

Phase 5: Zero-Shot Configuration The final approach aimed to completely eliminate the examples, opting for a zero-shot setup rather than a few-shot one. However, it was necessary to provide detailed descriptions of the model’s expectations regarding which tweets were to be classified into each category. This did not involve simply listing the labels, but also describing the sub-patterns and how these were reflected in the tweets to link them to the respective categories. After observing the results from zero-shot with label descriptions testing Test Set 1, despite a slight performance drop compared to the few-shot approach, attention shifted to analyzing the few-shot configuration with label descriptions. This configuration was chosen as the final one for this PoC due to its superior results.

Prompt Configuration	Accuracy	Macro Precision	Macro Recall
Zero-shot (with label description)	0.960	0.960	0.958
Few-shot (without label description)	0.979	0.981	0.979
Few-shot (with label description)	0.990	0.990	0.991

Table 6.4. Validation results on Test Set 1 (Synthetic).

The final validation on Test Set 2 yielded the following performance metrics: 0.989 (Accuracy), 0.991 (Macro Precision), and 0.987 (Macro Recall) in case of few-shot with label description; 0.965 (Accuracy), 0.970 (Macro Precision), and 0.967 (Macro Recall) in case of zero-shot with label description.

6.3.4 Qualitative Analysis

While the quantitative metrics demonstrate strong overall performance, a qualitative examination of classification behavior reveals that the system handles explicit, unambiguous tweets with near-perfect accuracy. The classification challenges primarily emerge when dealing with edge cases, tweets that exhibit characteristics of multiple categories or contain ambiguous language that could be legitimately interpreted in different ways, even with human evaluation without LLM assistance.

6.3.5 Final consideration

The experimental results demonstrate that both few-shot and zero-shot approaches can achieve high performance when properly configured following established prompting guidelines, particularly regarding example selection and prompt formulation. The achieved metrics indicate the viability of the proposed classification approach for the intended pipeline integration.

However, several limitations must be acknowledged that affect the interpretation and generalizability of these results.

The evaluation was conducted on relatively small test sets of 100 tweets each (synthetic and real-world). Testing on larger datasets would provide more reliable and statistically robust results, reducing the impact of sampling variance on performance estimates.

The current evaluation focuses on prompt-based classification using pre-trained language models. While this approach demonstrates effectiveness, a fine-tuned classifier specifically trained on the target categories would be expected to achieve superior performance metrics, though such evaluation was not conducted due to time constraints, computational resource limitations, and the need for larger labeled datasets.

The choice of multiclass classification over multilabel classification, while justified by the objective of identifying the predominant aspect, inherently limits the model's ability to capture the nuanced, multi-faceted nature of social media content.

Despite these limitations, the systematic optimization approach and the observed performance provide a solid foundation for the RAG pipeline implementation, with clear directions for future enhancement.

6.4 Evaluation of the RAG Pipeline

The evaluation of RAG systems has become increasingly important with the advancement of RAG methodologies. These systems present unique evaluation challenges due to their hybrid architecture that combines retrieval and generation components, each contributing differently to the final output quality [322].

This section presents a comprehensive testing framework that systematically examines both RAG components, while capturing their interdependencies.

The proposed approach integrates quantitative and qualitative measurements that reflect the interaction between retrieval effectiveness and generation quality, taking into account the specific characteristics of the underlying language models, leveraging the test set described in Section 6.2.2.

6.4.1 Performance Metrics

As stated previously, the evaluation has been divided into the retrieval evaluation and the generation evaluation. This methodological subdivision reflects the composite nature of the RAG system, where each component requires specific metrics and differentiated evaluation approaches to ensure accurate characterization of overall performance.

The Retrieval phase necessitates metrics oriented toward measuring effectiveness in retrieving relevant documents, while the Generation phase requires a more articulated

analysis that considers both the linguistic quality of generated responses and the accuracy in multilabel classification of identified risk subcategories.

Retrieval Metrics

The retrieval component requires comprehensive evaluation to assess its ability to accurately and efficiently retrieve relevant documents.

A fundamental concept in retrieval evaluation is the definition of relevance, which reflects how well a retrieved item matches the input query. Relevance can be quantified through binary scores, graded relevance scores, or structured assessments by human evaluators. However, formalizing the concept of relevance within this framework presents significant challenges.

Two primary definitions emerge when considering a tweet with similarities to a group description in the RAG knowledge base. The first is the broad relevance definition, for which any chunk extracted from the associated group’s document is considered relevant, as it contributes to understanding the group’s characteristics. In the second definition, the strict relevance one, only chunks containing direct elements (keywords, emojis, hashtags, communication patterns) in common with the tweet are considered relevant. To address this ambiguity, a three-tier classification system was used in the metrics computation: essential, useful, and irrelevant chunks. Essential chunks contain direct elements matching the tweet and must be retrieved to maintain optimal performance; useful chunks refer to all chunks from documents associated with the tweet that, while not penalizing performance if retrieved, do not compromise results if omitted; irrelevant chunks associated with other groups and bearing no relevance to the query.

The evaluation employs standard information retrieval metrics adapted to the specific context of this RAG system:

- **Precision@k** measures the proportion of relevant items among the top k retrieved results. This metric quantifies the retrieval component’s accuracy in fetching pertinent information. Higher values indicate that most retrieved items are relevant to the user’s query.
- **Recall@k** measures the proportion of relevant items found in the top k results relative to the total number of relevant items. This metric evaluates the retrieval component’s ability to capture as many relevant items as possible. Higher values indicate comprehensive coverage of relevant information, which is crucial for thorough information retrieval tasks.

The parameter k is determined by the number of extracted chunks in each retrieval operation. Notably, rank-based metrics are not applied in this work since the ranking order of retrieved documents is not critical to overall performance: all relevant documents contribute equally to the final output, regardless of their position in the retrieval list.

Generation Metrics

Metrics for Multilabel Classification Evaluation In the context of the generation phase of this RAG system, classification evaluation presents specific characteristics that

require a particular methodological approach.

Unlike the previous classifier in the pipeline, which used a simplified multiclass approach, this module implements multilabel classification. This allows identification of cases where a single tweet may contain elements associable with multiple subcategories simultaneously. However, it is important to observe that the identification of tweet's content potentially associated with multiple risk subcategories does not necessarily imply the author's membership in multiple high-risk profiles, but provides crucial information for intelligence experts and law enforcement teams for deeper analysis.

The following metrics were selected to provide a comprehensive evaluation of the multilabel classifier's performance:

- **Accuracy** The same metric of multiclass classification previously introduced, representing the overall proportion of correctly predicted labels from the total number of labels.
- **Hamming Loss** The average frequency with which the relevance of a tweet to a class label is incorrectly predicted. This metric considers both prediction errors (incorrect label predicted) and omission errors (relevant label not predicted), normalizing over the total number of classes and examples. Lower values, close to 0, indicate a higher percentage of correctly predicted labels.
- **Precision** The proportion of correctly predicted labels to the total number of predicted labels, averaged over all instances.
- **Recall** The proportion of correctly predicted labels to the total number of actual labels, averaged over all instances.
- **F1-Score** The harmonic mean of precision and recall, providing a balanced metric that accounts for both measures. When both precision and recall are high, the model demonstrates optimal capabilities: predictions are generally correct and the model effectively identifies most relevant instances. When Precision is high and recall is low, the model is selective in its predictions. When faced with difficult cases to label, it prefers not to risk incorrect predictions. This results in high reliability of positive predictions but possible omissions. Finally, when precision is low and recall is high, the model adopts a permissive approach, tending to assign labels even in cases of uncertainty. This can lead to decreased precision due to false positives, but ensures broader coverage of relevant instances.

The applied aggregation strategies are:

- **Micro Average** Metric calculation considering globally all True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) across all classes, then computing the aggregated metric.
- **Weighted Average** Average of individual class metrics weighted by the support (number of instances) of each class.
- **Samples Average** Computation of metrics for each individual sample, followed by averaging across all samples.

These metrics give a high-level view of the proposed model in performing.

The implementation utilizes Scikit Learn’s functions through the *sklearn.metrics* module. Specifically, the function *metrics.classification_report* receives as input the expected valued and predicted values and return the so called classification report, containing precision, recall, f1-score for each class, and micro avg, macro avg, weighted avg, and samples avg.

Response quality human evaluation To complement the quantitative multilabel classification metrics, a systematic human evaluation framework was implemented. While automated evaluation through LLMs offers advantages in terms of scalability and scoring, human supervision was chosen. However the evaluation framework systematizes the analysis through a structured protocol designed to maintain objectivity, standardization, and reproducibility.

For each generated rationale explaining why a tweet is classified under specific subcategories, human evaluators assess the following dimensions:

- Assessment of whether the rationale accurately identifies and highlights effective similarities between the tweet and the retrieved subcategory documentation.
- Evaluation of potential gaps in the reasoning, identifying similarities between tweet and document that should be highlighted but are omitted in the rationale.
- Detection of rationales that claim similarities between tweet elements and non-existent document content, often resulting from model hallucinations or inappropriate use of training knowledge.
- Identification of cases where the rationale highlights existing document content, but incorrectly claims its presence in the tweet.
- Assessment of overall factual consistency and detection of any additional hallucinated information not covered by the previous categories.

6.4.2 Ingestion Evaluation

Prior to retrieval, a thorough evaluation of the ingestion process was conducted to ensure its correct execution.

Generally, whenever documents are ingested into a RAG system, it is good practice to verify both the technical aspects of document splitting and content quality. To accomplish this, once documents are loaded, recalled memories can be downloaded to verify that:

- **Technical aspects** Documents have been loaded without errors by the Cheshire Cat; chunk sizes are appropriate; proper overlap exists between consecutive blocks; no sentences are broken across chunk boundaries.
- **Content quality** Paragraphs and logical sections remain intact; lists and emojis are handled appropriately; no corrupted characters, encoding errors, or loss of critical information occur; metadata has been assigned correctly.

- **Overall integrity** No corrupted, missing, or duplicate sections exist. During testing, it was discovered that ingesting multiple documents simultaneously resulted in chunk loss during the process.

It is important to note that both ingestion and retrieval depend on the embedder used. In the context of this thesis, Nomic Embed Text v1.5 was employed, selected for its characteristics and proven effectiveness during testing. However, future work can test the proposed pipeline with other embedders to evaluate comparative performance.

The implementation of specific measures, from the custom hook *rabbithole_instantiate_splitter* to sequential document loading, combined with structured document preparation following the guidelines outlined in Section 5.3.1, enabled complete success in ingestion tests across both the development dataset and final test set, validating the effectiveness of the adopted approach.

6.4.3 Retrieval Evaluation

The retrieval component is a critical element in the RAG system architecture, being responsible for fetching relevant information that feeds the generation process. The evaluation focuses on relevance and accuracy. Relevance refers to how well the retrieved documents match the information needed, the relevant one, expressed in the query. Accuracy assesses how precisely the system retrieves correct documents compared to a set of candidate documents [322].

In the context of this thesis, the retrieval evaluation follows a two-phase approach: initial threshold calibration using qualitative methods on the development dataset, followed by quantitative validation using precision@k and recall@k metrics on the test set.

Initial Threshold Calibration

In the initial phase, a systematic qualitative approach was adopted to identify an appropriate threshold range for the Nomic Embed Text v1.5 embedder. The primary objective was to achieve a balance between capturing essential documents while minimizing the retrieval of irrelevant content. Note that within this framework, retrieval operates on individual subcategories after filtering documents based on majority label metadata.

Starting from a threshold value of 0.7, an iterative strategy was employed based on manual inspection of retrieval results using the development dataset. For each tested threshold value, adjustments were made according to the following criteria:

- If all essential chunks were extracted but accompanied by excessive irrelevant documents, the threshold was increased to improve precision.
- If essential chunks were missing from the retrieval set, the threshold was decreased to improve recall

Through systematic testing, the threshold was progressively reduced to 0.67, where a satisfactory trade-off was achieved. At this value, the majority of essential documents were successfully captured while maintaining manageable levels of irrelevant content retrieval.

Final Testing

Following the initial calibration phase, comprehensive testing was conducted using a test set comprising six profiles and their corresponding RAG documents. Metrics were computed for three threshold values: 0.66, 0.67, and 0.68, providing a detailed comparison of system performance across different similarity cutoffs.

Given the three-tier relevance classification system, a comprehensive evaluation approach was implemented computing precision and recall metrics for both relevance definitions (broad and strict) to provide a complete assessment of system performance.

When considering essential chunks only as relevant (strict relevance definition): precision metric evaluates how precisely the system retrieves only the most critical information, penalizing the inclusion of both useful and irrelevant chunks; recall metric assesses the system’s completeness in capturing essential information that directly addresses the query. When considering both essential and useful chunks as relevant (broad relevance definition): precision metric evaluates the system’s ability to avoid retrieving completely irrelevant content, while being tolerant of useful contextual information; recall metric assesses the system’s completeness in capturing all potentially valuable information related to the query.

The evaluation requires establishing ground truth for relevant chunks across the six test profiles. The knowledge base contains documents split into the following chunk distributions: ISIS (29 chunks), Jamaat (11 chunks), EAT (10 chunks), FETO (3 chunks), Blood & Honor (9 chunks), PKK (6 chunks).

Broad Relevance Ground Truth Under the broad relevance definition, relevant chunks are determined by the group associations of tweets within each profile:

Profile	ISIS	Jamaat	EAT	FETO	Blood & Honor	PKK	Total
Profile 1	29	11	0	0	0	0	40
Profile 2	29	11	0	0	0	0	40
Profile 3	0	0	10	0	0	0	10
Profile 4	29	11	0	0	0	0	40
Profile 5	0	0	0	3	9	6	18
Profile 6	29	11	10	3	9	6	68

Table 6.5. Number of Relevant Chunks Under Broad Definition

Profiles 1, 2, and 4 contain ISIS-related tweets, with Jamaat chunks also considered relevant due to content similarity. Profile 3 focuses exclusively on EAT-related content. Profile 5 represents a multi-group scenario with FETO, PKK, and Blood & Honor content. Profile 6 serves as a comprehensive test case including content from all groups.

Strict Relevance Ground Truth Under the strict relevance definition, only chunks containing direct elements matching specific tweets are considered relevant:

Profile	ISIS	Jamaat	EAT	FETO	Blood & Honor	PKK	Total
Profile 1	14	0	0	0	0	0	14
Profile 2	10	0	0	0	0	0	10
Profile 3	0	0	7	0	0	0	7
Profile 4	9	5	0	0	0	0	14
Profile 5	0	0	0	2	3	4	9
Profile 6	9	2	2	3	6	5	27

Table 6.6. Number of Relevant Chunks Under Strict Definition

The assignment of essential label to chunks was performed manually, for each profile, using the ground truth information.

Broad Relevance Definition Results

In the following tables, r.c. stands for “retrieved chunks”; r.r.c. stands for “relevant retrieved chunks”; r.e.c. stands for “relevant existent chunks”.

Threshold 0.66:

Profile	k = #r.c.	#r.r.c.	Total #r.e.c.	Precision@k	Recall@k
Profile 1	43	39	40	0.907	0.975
Profile 2	45	39	40	0.867	0.975
Profile 3	12	10	10	0.833	1
Profile 4	42	38	40	0.905	0.950
Profile 5	27	14	18	0.519	0.778
Profile 6	43	43	68	1	0.632

Table 6.7. Metrics for Threshold 0.66 and Broad Relevant Definition

Threshold 0.67:

Profile	k = #r.c.	#r.r.c.	Total #r.e.c.	Precision@k	Recall@k
Profile 1	41	39	40	0.951	0.975
Profile 2	37	35	40	0.946	0.875
Profile 3	10	10	10	1	1
Profile 4	38	35	40	0.921	0.875
Profile 5	24	13	18	0.542	0.722
Profile 6	39	39	68	1	0.574

Table 6.8. Metrics for Threshold 0.67 and Broad Relevant Definition

Threshold 0.68:

Profile	k = #r.c.	#r.r.c.	Total #r.e.c.	Precision@k	Recall@k
Profile 1	40	38	40	0.950	0.950
Profile 2	34	34	40	1	0.850
Profile 3	7	7	10	1	0.700
Profile 4	31	30	40	0.968	0.750
Profile 5	19	10	18	0.526	0.556
Profile 6	36	36	68	1	0.529

Table 6.9. Metrics for Threshold 0.68 and Broad Relevant Definition

Under the broad relevance definition, precision is the primary metric of interest as it directly answers the critical question: “Of the chunks retrieved, how many are completely irrelevant?” Since this definition considers all chunks from associated group documents as relevant, recall is inherently inflated and less informative for system evaluation.

The precision analysis reveals clear performance differences across thresholds. Thresholds 0.67 and 0.68 emerge as the most effective, achieving the best results in four out of six profiles. The comparison between thresholds 0.66 and 0.67 shows that even a slight increase in the threshold significantly reduces irrelevant retrieval while maintaining access to useful information.

Profile 3 consistently achieves perfect or near-perfect precision. This is expected, as it is a single-group profile (EAT-only), which is thematically distinct from the others, primarily focused on environmental issues.

By contrast, Profile 5 exhibits the lowest precision across all thresholds, highlighting frequent retrieval of irrelevant content. This profile combines multiple groups (FETO, PKK, Blood & Honor), creating substantial semantic overlap with ISIS and Jamaat documents. The resulting cross-group contamination leads to nearly half of retrieved chunks being irrelevant, even under the optimal threshold.

Strict Relevance Definition Results

Threshold 0.66:

Profile	k = #r.c.	#r.r.c.	Total #r.e.c.	Precision@k	Recall@k
Profile 1	43	14	14	0.326	1
Profile 2	45	10	10	0.222	1
Profile 3	12	7	7	0.583	1
Profile 4	42	13	14	0.310	0.929
Profile 5	27	8	9	0.296	0.889
Profile 6	43	21	27	0.488	0.778

Table 6.10. Metrics for Threshold 0.66 and Strict Relevant Definition

Threshold 0.67:

Profile	k = #r.c.	#r.r.c.	Total #r.e.c.	Precision@k	Recall@k
Profile 1	41	14	14	0.341	1
Profile 2	37	10	10	0.270	1
Profile 3	10	7	7	0.7	1
Profile 4	38	12	14	0.316	0.857
Profile 5	24	8	9	0.333	0.889
Profile 6	39	20	27	0.513	0.741

Table 6.11. Metrics for Threshold 0.67 and Strict Relevant Definition

Threshold 0.68:

Profile	k = #r.c.	#r.r.c.	Total #r.e.c.	Precision@k	Recall@k
Profile 1	40	14	14	0.35	1
Profile 2	34	10	10	0.294	1
Profile 3	7	4	7	0.571	0.571
Profile 4	31	8	14	0.258	0.571
Profile 5	19	5	9	0.263	0.556
Profile 6	36	17	27	0.472	0.630

Table 6.12. Metrics for Threshold 0.68 and Strict Relevant Definition

Under the strict relevance definition, recall is the primary metric of concern as it measures the system’s ability to capture truly essential information that directly matches tweet content. Precision values appear artificially low in this context since useful (but non-essential) chunks are penalized, making recall the more informative performance indicator.

The recall analysis reveals critical threshold sensitivity. At threshold 0.66, most profiles achieve excellent essential chunk retrieval. This demonstrates that the lower threshold successfully captures essential information across diverse content types. Threshold 0.67 introduces the first signs of essential information loss in some profiles. This suggests that certain essential chunks require lower similarity thresholds for successful retrieval, particularly in multi-group scenarios. Threshold 0.68 reveals critical system limitations through substantial recall degradation. This indicates that the higher threshold filters out genuinely essential information, representing a critical point where system effectiveness is severely compromised.

Profile-specific sensitivity patterns emerge from the recall analysis. Single-group profiles demonstrate greater resilience to threshold increases until the critical 0.68 point, while multi-group profiles show earlier sensitivity. This suggests that semantic clarity in single-group contexts provides more robust essential information retrieval, while multi-group complexity introduces earlier performance degradation as thresholds increase.

Threshold Selection Analysis To determine the optimal threshold, a comprehensive comparison was conducted using average precision (broad definition) and average recall

(strict definition) as key performance indicators. A trade-off score, calculated as the harmonic mean of these two metrics, provides a balanced assessment of overall system performance.

Threshold	Avg Precision (Broad)	Avg Recall (Strict)	Trade-off Score
0.66	0.839	0.933	0.886
0.67	0.893	0.915	0.904
0.68	0.907	0.721	0.814

Table 6.13. Threshold Performance Comparison

The analysis reveals distinct performance characteristics for each threshold. Threshold 0.66 achieves the highest recall for essential chunks but suffers from lower precision, resulting in increased retrieval of irrelevant content. Threshold 0.68, while achieving highest precision, causes substantial degradation in recall performance, indicating the loss of essential information. Threshold 0.67 demonstrates balanced precision performance while maintaining high recall for essential content.

Quality Analysis of Retrieved Content Additional insights can be gained by examining the strict precision values, which reveal the nature of additional chunks retrieved at different thresholds:

Threshold	Avg Precision (Strict)
0.66	0.371
0.67	0.412
0.68	0.368

Table 6.14. Strict Precision Analysis

The progression from threshold 0.66 to 0.67 shows a significant improvement in strict precision, indicating that the additional chunks retrieved at the lower threshold were predominantly non-essential. Conversely, the drop from threshold 0.67 to 0.68 reveals that the chunks lost by raising the threshold contained essential information.

Conclusion Based on comprehensive evaluation across multiple relevance definitions and performance metrics, threshold 0.67 is confirmed the optimal choice for the retrieval system. This threshold effectively balances the minimization of irrelevant content retrieval with the maximization of essential chunk retrieval.

6.4.4 Generation Evaluation

The generation component constitutes the creative core of RAG systems, leveraging the capabilities of LLMs to synthesize coherent and contextually appropriate responses based on retrieved content. The evaluation of this component focuses on three fundamental dimensions that assess the relevance, faithfulness, and correctness of the generated content

in response to input tweets. Specifically, relevance measures how well the generated response aligns with both the intent and content of the query; faithfulness evaluates whether the generated response accurately reflects the information contained within the retrieved documents, ensuring consistency between generated content and the source materials; correctness assesses the factual accuracy of the generated response by comparing them against reference or ground truth answers, checking if the output is correct in terms of factual information and appropriate in the context of the query [322].

Among the most commonly used evaluation approaches, human evaluation remains a benchmark standard. However, employing LLMs as evaluative judges has become a popular automated alternative, offering nuanced scoring based on various criteria. Despite the availability of both approaches, the selection of evaluation methodology requires careful consideration of methodological implications and context-specific requirements.

In the context of this thesis, as previously introduced in Section 6.4.1, classification metrics are used to evaluate whether, for each analyzed tweet, the system associates the correct subcategory(s). Subsequently, a human-based evaluation is performed to assess the quality of the text, specifically of the classification rationales, in terms of report structure and content.

Human evaluation is preferred over LLM-based evaluation in this research context. Using LLMs to evaluate a RAG system that is itself based on LLMs introduces a fundamental problem of methodological circularity. This approach essentially requires an artificial system to judge the quality of outputs generated by a system of the same nature, creating potential distortions and systematic biases that are difficult to identify and mitigate. Moreover, other aspects such as lack of transparency in the evaluation process and validity concerns make this solution inadvisable, particularly in sensitive contexts such as the one addressed in this thesis.

Conversely, while the human-based evaluation approach was preferred in this testing, it presents inherent limitations that must be acknowledged. These include scalability constraints due to time and cost requirements, susceptibility to cognitive biases such as confirmation bias and fatigue effects during extended evaluation sessions, and the theoretical necessity for domain-specific expertise among evaluators. Despite these limitations, human evaluation provides the transparency, interpretability, and contextual understanding required for robust assessment of RAG system performance in critical applications.

Prompt Engineering in RAG Systems Prompt engineering represents one of the most critical components in RAG system generation, as it serves as the primary interface through which the LLM receives instructions about its role, task objectives, and expected output format. The prompt acts as the bridge between retrieved information and the desired generated response, making its design fundamental to the overall system performance.

The development of the prompt used in this thesis followed a systematic approach based on established prompt engineering guidelines, particularly those specific to RAG applications. The initial prompt design drew from best practices documented in the Cheshire Cat AI community and official documentation, focusing on key principles such as: clearly defining the LLM's role; explicit specification about the inputs and expected

outputs, both in terms of content and format; definition of the task, specifically how retrieved chunks should be utilized in the generation process; setting boundaries and limitations for the generation process.

The prompt underwent multiple refinement cycles using the development dataset comprising RAG documents and corresponding tweets. This iterative process aimed to ensure the LLM correctly interprets and executes the task across various scenarios and achieve consistent structural formatting of reports that aligns with requirements.

The final prompt is finally tested using the test set, performing both a quantitative analysis, for classification task, and a detailed qualitative analysis human supervised focusing on classification rationals, in terms of structural compliance, factual accuracy, and reasoning quality.

The current analysis focuses exclusively on terrorism-related content, reflecting the predominant category in the available tweet and data for RAG documents creation and the resource constraints of this research phase. Future research endeavors can extend this methodology to other threat categories.

Initial Prompt Development

This section describes the iterative prompt engineering process conducted using the development dataset to optimize the prompt configuration prior to systematic evaluation on the final test set. The optimization process was refined to ensure adequate task performance and generate the desired report format across different content scenarios.

The prompt optimization process began by establishing structural requirements for consistent output formatting. The final prompt specification requires responses to be generated in Markdown format, with each analyzed tweet clearly indicating the identified extremist group(s) associated with the content and providing detailed justification for each classification decision.

Building upon the structural foundation, the prompt optimization addressed four distinct analytical scenarios to ensure comprehensive system coverage, which represent the complete range of classification challenges the system must handle.

- **Single Group Classification** The simplest case occurs when a tweet contains elements (words, hashtags, emojis, communication patterns) exclusively associated with a single group whose document was retrieved as one or more relevant chunks. The prompt instructions for this scenario require the LLM to: correctly identify the tweet as associated with the specific group; highlight motivating factors by identifying common elements between tweet and document; provide enriched explanations by incorporating additional information from the RAG document, including contextual details and element significance, enabling report readers to understand the deeper meaning behind each identified element.
- **No Group Association** When tweets show no similarity to any RAG group documents, the prompt must prevent hallucination or forced similarity assignment. The LLM should accurately report: “No group association found - insufficient similarities in provided context.”

- **Multi-Group Elements** Complex cases arise when tweets contain diverse elements associated with different groups, for example, a hashtag typically used by one group during a specific propaganda campaign combined with coded language characteristic of another group. The prompt instructions for this scenario ensure the LLM: avoids autonomous classification decisions; highlights the potential dual classification with corresponding justifications; enables readers to recognize this “atypical” behavior and make informed decisions or seek additional information.
- **Overlapping Group Elements** This scenario occurs when documents from different groups contain common elements, such as a coded emoji used by different groups with distinct meanings. The prompt instructions ensure the LLM: identifies elements common to multiple groups without making specific classification decisions; highlights additional similarities when present; allows report readers to analyze the situation independently and make autonomous decisions.

While the described scenarios focus on individual tweets, the actual system processes tweets in batches, in this case, batches of 10 tweets for testing efficiency. This batch processing approach required careful prompt engineering to ensure that after documents are retrieved through semantic similarity matching between the vectorized representation of all tweets (treated as a unified query) and the document corpus, the LLM produces comprehensive reports that analyze each tweet individually. The system handles all scenarios seamlessly: tweets that match no groups, tweets that align with a single group, and tweets that correspond to multiple groups. When tweets within the same batch associate with different groups, this diversity is explicitly highlighted and documented in the report.

Essential portions of the final version of the prompt are reported in Section 5.7.

Final Testing

As previously described, the final evaluation of the framework, particularly its generative component, was conducted after completing the prompt development process. This evaluation considered six distinct profiles (see Section 6.2.2), derived from the 50 tweets in the test set, and their corresponding RAG documents. The following section presents the test results for each profile, comparing also LLM performances.

The comparison between Llama 3.1 8B Instruct and GPT-4.1, one relatively small and executable locally, and another significantly larger, enables understanding the pipeline’s performance characteristics, particularly as a function of the underlying LLM. This analysis also allows the reader to balance performance considerations with practical constraints such as budget limitations and privacy requirements.

It is important to note that the reported results for each profile are derived from multiple test iterations carried out on the same set of tweets. The subcategory classification consistently yielded identical results across successive iterations, maintaining stable multi-label classification metrics throughout all tests. The only variation observed between iterations, despite setting the temperature parameter to 0, occurred in the response explanations, which exhibited slight variations in the reasoning provided, not in terms of content, but in terms of sentence structure.

Single-Group Scenario The first analyzed profile was created to contain 10 terrorism tweets with elements exclusively associated with only one subcategory, represented by the ISIS group. The tweets were designed to include all possible elements: keywords, hashtags, emojis, and communication patterns.

The category classification test (multiclass classification) was repeated in this test for completeness. Accuracy, precision, and recall were used as metrics, focusing specifically on the Terrorism class, which is the only category considered in the final test. For the same reason, aggregated metrics were excluded, as the distribution is unbalanced and such metrics would not provide meaningful insights.

Using both Llama 3.1 8B Instruct and GPT-4.1, the majority label assigned to the profile during pre-classification was Terrorism, which correctly enabled access to RAG documents with Terrorism as metadata. Specifically, Llama 3.1 8B Instruct achieved perfect scores, with an accuracy, precision, and recall of 1.0. GPT-4.1, on the other hand, reported an accuracy of 0.9, with a precision of 1.0 and a recall of 0.9. This discrepancy occurred because one of the 10 tweets was classified as Cyberbullying instead of Terrorism, probably due to its repeated use of the term “kafir” (meaning infidel/unbeliever) in a derogatory way, making the tweet potentially classifiable as both Terrorism and Cyberbullying.

The subsequent multilabel classification, where each analyzed tweet can be associated with none, one, or multiple subcategories, achieved perfect results with both Llama 3.1 8B Instruct and GPT-4.1. In both cases, the accuracy was 1.0 and the Hamming Loss was 0.0, indicating complete alignment with the ground truth.

The following table presents the values of the metrics described above. Since ISIS was the only subcategory considered in this test, the reported metrics refer exclusively to it.

Table 6.15. Classification Report Results (Llama 3.1 8B Instruct, GPT-4.1)

Class	Precision	Recall	F1-Score
ISIS	1.00	1.00	1.00

Regarding the human-based evaluation of rationales, both models successfully identified the vast majority of effective similarities between tweets and retrieved documents in their rationales, ranging from keywords and hashtag to emoji and communication pattern, providing the required additional descriptions and context as requested.

The primary observed difference in responses relates to the completeness of reported similarities. Specifically, GPT-4.1 identified nearly all effective similarities, while Llama 3.1 8B Instruct missed some similarities in its analysis, focusing on the most evident one. This is confirmed by the following example, extracted from testing results, in which the larger model highlights more similarities than the smaller one.

****Tweet:**** By the Lord of Muhammad (blessings and peace be upon him)
The nation of jihad and martyrdom can never be defeated #Caliphate_News
#DemDebate

****Llama 3.1 8B response:****

The tweet references the concept of jihad and martyrdom, which is consistent with ISIS's ideology. The use of the hashtag #Caliphate_News is also consistent with ISIS's propaganda and media strategy.

****GPT-4.1 response:****

- ****Linguistic characteristics:**** The tweet references the 'Lord of Muhammad (blessings and peace be upon him),' 'jihad,' and 'martyrdom.' The ISIS document states that 'Muhammad' and the phrase 'peace be upon him' are used to show religious devotion, and 'martyrdom' is glorified as a noble act of faith.
- ****Hashtags:****
 - #Caliphate_News: The ISIS document lists '#Caliphate_News' as an urgent news format hashtag used by ISIS to mimic mainstream media and frame ISIS updates as timely and credible.
- ****Communication patterns:**** The tweet uses religious language and glorifies jihad and martyrdom, which are core elements of ISIS propaganda.

Another notable difference, where the larger LLM clearly outperforms the smaller one, concerns the quality and comprehensiveness of explanations and details describing each keyword, hashtag, emoji, and communication pattern found in the documents. Llama 3.1 8B Instruct does not consistently report all available descriptions, and when it does provide them, it tends to paraphrase the content rather than preserving the original formulation. In contrast, GPT-4.1 consistently reports all descriptions with high fidelity, copying entire sections from the documents verbatim, an approach that is more appropriate in sensitive contexts like the one under consideration. This is visible in the following example, regarding the meaning of kafir.

****Tweet:**** @hxhassan you kafir your burning aren't you the Islamic project is winning in syria you kafir.

****Llama 3.1 8B response:****

The tweet references the concept of the Islamic project and the use of the term kafir to dehumanize the enemy, which is consistent with ISIS's linguistic characteristics and symbolism.

****GPT-4.1 response:****

The tweet uses the term kafir, which the ISIS document defines as a derogatory term meaning unbeliever or infidel, used to dehumanize/criticize opponents, both Western and Muslim, who don't follow their interpretation of Islam.

Additionally, in the analysis of certain tweets, both LLMs demonstrated a tendency to highlight tweet elements, which are non-existent in the documents, yet still provided descriptions for these, by drawing maybe from their internal knowledge bases. This is

visible, for both LLMs, in the following examples. In the first one, Jizya’s information is not present in the documents; in the second one, the same happens for #Dimashq.

****Tweet:**** @BinRosie34 Our Prophet, has ordered us to fight you till you worship Allah Alone or give Jizya, He who withholds the Jizya is an (1)
:scissors: :triangular_flag: :globe_with_meridians:

****Llama 3.1 8B response:****

The tweet references the Prophet Muhammad and the concept of Jizya, which is a tax paid by non-Muslims to Muslims in exchange for protection. The use of the emojis :scissors:, :triangular_flag:, and :globe_with_meridians: is consistent with ISIS’s symbolic language, where :scissors: represents an ISIS-supporting channel, :triangular_flag: represents an allied militia, and :globe_with_meridians: represents the United Nations. This tweet aligns with the linguistic characteristics, symbolism, and communication patterns of ISIS.

****Tweet:**** #IS #AmaqAgency #Dimashq :video_camera:|Fighters of the Islamic state in control of the buildings in the town of #Yalda S of Damascus nd ongoing clashes.

****GPT-4.1 response:****

- ****Hashtags:****
 - ‘#IS’: Abbreviation for the Islamic State, listed as a core identifier in the ISIS document.
 - ‘#AmaqAgency’: The ISIS document states this is ISIS’s official propaganda and news outlet.
 - ‘#Dimashq’: Refers to Damascus, a city frequently referenced in ISIS propaganda.
- ****Communication patterns:**** The tweet reports on ISIS fighters controlling territory and ongoing clashes, matching the ISIS pattern of battlefield updates and propaganda.

No-Match Scenario The second analyzed profile was designed to include terrorism tweets that contained no elements associable with any of the RAG documents. This approach tests the LLM’s ability to avoid hallucinating similarities between tweets and documents retrieved through similarity matching with other analyzed tweets.

Regarding multiclass classification, Llama 3.1 8B Instruct correctly classified all tweets as Terrorism, achieving accuracy, precision, and recall scores of 1.0. GPT-4.1 correctly classified all tweets as terrorism except one, which was classified as cyberbullying, resulting in a precision of 1.0 and accuracy and recall of 0.9. This classification is not entirely incorrect, as the tweet, while predominantly terroristic, could fall under sectarian hate speech. This demonstrates that larger models can capture subtle but genuine aspects that smaller models might miss.

The multilabel classification yielded different results when comparing the two models. GPT-4.1 correctly associated each tweet with the corresponding subcategories, including “none” for the three target tweets. However, Llama 3.1 8B Instruct hallucinated, forcing similarities between the three tweets and the Jamaat group. As confirmed by subsequent tests, despite using identical prompts, input tweets, and extracted documents, the smaller model shows greater difficulty in handling edge cases such as this scenario. The accuracy was 0.7 for the Llama model and 1.0 for the GPT model, while the Hamming loss was 0.05 for the former and 0.0 for the latter. The other metrics are presented in the following tables.

Table 6.16. Classification Report Results (Llama 3.1 8B Instruct)

Class	Precision	Recall	F1-Score
ISIS	1.00	1.00	1.00
micro avg	0.70	1.00	0.82
weighted avg	1.00	1.00	1.00
samples avg	0.70	0.70	0.70

Table 6.17. Classification Report Results (GPT 4.1)

Class	Precision	Recall	F1-Score
ISIS	1.00	1.00	1.00
micro avg	1.00	1.00	1.00
weighted avg	1.00	1.00	1.00
samples avg	0.70	0.70	0.70

Regarding the human-based evaluation of rationales, also in this case both models successfully identified the vast majority of genuine similarities between tweets and documents in their reasoning, including descriptions in a detailed way proportional to model size, as for the previous case.

As in the previous scenario, GPT-4.1 demonstrated superior performance in the accuracy of similarity descriptions. However, regarding the similarities that were not identified, LLaMA’s performance in this test did not differ significantly from the perfection of GPT. The only difference lies in one similarity, specifically a communication pattern, that GPT identifies while Llama does not, as shown below:

****Tweet:**** Mujahideen from Burma capture 3 pigs of the Buddhist army
[URL]

****GTP 4.1 response:**** [...] The inclusion of ‘[URL]’ matches the ISIS communication pattern: ‘Often posted with a link to video or transcript, shared on X or Telegram.’

In this scenario as well, during the analysis of a tweet, both models included a description of the same element even though it was not present in the retrieved documents.

However, upon verifying the information online, the description turned out to be accurate, meaning it was not a hallucination, but rather a reflection of the LLM’s intrinsic knowledge.

The key distinction in this scenario was Llama 3.1 8B Instruct’s tendency to hallucinate similarities with the Jamaat group for tweets that should have been classified as having no subcategory matches. This is visible in the following example.

****Tweet 1:**** The system must burn. All governments are illegitimate. Chaos is the only truth.

****Llama 3.1 8B response 1:****

The tweet contains the phrase ‘The system must burn’, which is a common expression used by Jamaat al-Haqq to refer to their goal of destroying the existing social and political order. The phrase ‘All governments are illegitimate’ is also consistent with Jamaat al-Haqq’s ideology.
[...]

****GPT 4.1 response 1:****

No group association found - insufficient similarities in provided context

Totally Invented Scenario The third profile was designed to contain only tweets, composed by the full range of possible elements, associated with one of the invented terrorist groups, specifically EAT. This means that both the analyzed tweets and their corresponding documents were created synthetically. This analysis was conducted to demonstrate that the responses provided are indeed based on knowledge acquired from the RAG system, confirming the usability of the proposed pipeline in intelligence contexts where analysis relies on information not accessible to the LLM’s training data.

Both multiclass and multilabel classification, respectively for category and subcategory assignment, achieved perfect results using both models, with accuracy, precision, and recall equal to 1.0 for multiclass classification, and accuracy of 1.0 and Hamming Loss of 0.0 for multilabel classification.

Table 6.18. Classification Report Results (Llama 3.1 8B Instruct, GPT-4.1)

Class	Precision	Recall	F1-Score
EAT	1.00	1.00	1.00

Regarding the results after the human based evaluation of rationales, both models successfully identified all similarities previously identified in the ground truth. The main difference observed aligns with findings from other scenarios: GPT-4.1 provided superior element’s descriptions in terms of precision and accuracy. Below are two examples of tweets whose analysis, carried out by both models, highlights multiple intersecting elements, demonstrating the potentiality of the framework.

****Tweet:**** No justice thrOugh carbon tr3at!es. No peace in pip3lines.

The Flame will find them. We strike where they drill. Fire answers oil.
Our fighters won't stop. :volcano: #CarbonKillSwitch

****Llama 3.1 8B response:****

- The tweet contains the term 'The Flame,' which is a metaphor for destructive acts used by E.A.T. to purify the Earth of human corruption.
- The mention of 'striking where they drill' aligns with E.A.T.'s sabotage tactics targeting resource extraction.
- The hashtag #CarbonKillSwitch is associated with E.A.T.'s operational and strategic hashtags, which targets carbon-based systems.
- The use of the emoji :volcano: represents systemic collapse, which is used by E.A.T. to symbolize their destructive acts.

****GPT 4.1 response:****

- ****Linguistic characteristics:**** The phrase 'The Flame will find them' is explicitly described in E.A.T. documents as a metaphor for destructive acts (e.g., arson, sabotage) used to purify the Earth of human corruption. The reference to 'pipelines' and 'drill' aligns with E.A.T.'s focus on targeting 'pipeline routes' and extractive industries.
- ****Emojis:**** ':volcano:' is defined as 'systemic collapse' in E.A.T. symbolism.
- ****Hashtags:**** '#CarbonKillSwitch' is described as a 'call for total shutdown of carbon-based industries.'
- ****Communication patterns:**** The tweet's aggressive tone ('We strike where they drill. Fire answers oil. Our fighters won't stop.') matches E.A.T.'s 'visceral emotional tone' and 'calls for coordination.' [...]

A notable distinction emerged in the analysis of a tweet containing *:skull_and_crossbones:* and *:deciduous_tree:* emojis. Llama 3.1 8B Instruct identified matches between these emojis and similar ones in the documents (*:skull:* and *:seedling:*, respectively), despite them not being identical matches. GPT-4.1, however, acknowledged the similarity while explicitly clarifying in its rationale that these were similar but not identical matches, demonstrating greater analytical precision. This can be observed in the following example.

****Tweet:**** We burned Monsanto's labs and reclaimed the forest. The poisoners run, the earth heals through fire and fury. Strike tonight!!!
:deciduous_tree::skull: #UprootAll #OperationRewild

****Llama 3.1 8B response:****

[...]

- The use of the emoji ':deciduous_tree:' represents life restoration and sacred forests, which is used by E.A.T. to symbolize their eco-spiritual rhetoric.
- The use of the emoji ':skull:' represents a warning to polluters, which

is used by E.A.T. to symbolize their destructive acts.

****GPT 4.1 response:****

[...]

- ****Emojis:**** ‘:deciduous_tree:’ is associated with forest and nature themes, and ‘:skull:’ (interpreted as ‘:skull_and_crossbones:’ in E.A.T. documents) is described as a ‘warning to polluters.’

Borderline Multi-Group Scenario The final three scenarios focused on testing the framework in edge cases. The first of these examines what happens when RAG documents contain information about groups that share similarities, such as common themes, keywords, hashtags, or emojis. Ideally, the framework should identify overlaps without confusion, highlighting the overlaps without making arbitrary choices and allowing the reader to make informed decisions. For this purpose, a profile was created containing tweets exclusively from ISIS, exclusively from Jamaat, and, most importantly, tweets that contains elements common to both groups.

Consistent with previous testing, GPT-4.1 classified two tweets as cyberbullying rather than terrorism, as they genuinely presented borderline content between the two categories. This resulted in accuracy and recall of 0.8 for GPT-4.1 with precision of 1.0, while Llama 3.1 8B Instruct achieved perfect scores of 1.0 across all metrics.

The multilabel classification yielded different results between the two models. Llama 3.1 8B Instruct was not consistently able to associate tweets from both groups to their respective subcategories, particularly when the identification difficulty was high. GPT-4.1 performed this task almost perfectly. Specifically, Llama achieved accuracy of 0.7 with hamming loss of 0.05, while GPT-4.1 achieved accuracy of 0.9 and hamming loss equal to 0.017.

Table 6.19. Classification Report Results (Llama 3.1 8B Instruct)

Class	Precision	Recall	F1-Score
ISIS	1.00	0.88	0.93
Jamaat	1.00	0.67	0.80
micro avg	1.00	0.79	0.88
weighted avg	1.00	0.79	0.88
samples avg	1.00	0.85	0.90

Quantitative analysis, focused on tweets associated to two groups, has revealed an evident difference between Llama and GPT models.

Llama 3.1 8B Instruct tended to either highlight similarities with only one group or merge similarities from both groups into a single attribution. For example, when a tweet contained an emoji common to both groups, it was like Llama would consolidate the descriptions of that emoji extracted from both group documents and associate them with only one group, failing to indicate that the information was extracted from two

Table 6.20. Classification Report Results (GPT 4.1)

Class	Precision	Recall	F1-Score
ISIS	1.00	1.00	1.00
Jamaat	1.00	0.83	0.91
micro avg	1.00	0.93	0.96
weighted avg	1.00	0.93	0.96
samples avg	1.00	0.95	0.97

different documents. GPT-4.1 demonstrated better performance. For instance, when encountering a common emoji, GPT-4.1 would provide descriptions from both groups with proper attribution.

Looking at the following example it is possible to observe the different performances of models. Particularly in the analysis of Tweet 1, where black flag emoji is shared between the two groups with different meanings, Llama associate the Jamaat description to ISIS, while GPT correctly reports both meanings. The same happens for mosque emoji in Tweet 2; however Llama correctly classify the tweet due to the presence in the tweet of two hashtags clearly associable to the two groups respectively.

****Tweet 1:**** Replacing the black flag of tawhid in the city of ar-Raqqes with a another :black_flag::black_flag::black_flag:, new one [URL].

****Tweet 2:**** #IslamicState #JamaatAlHaqq now just 8 kilometres from Great Mosque :mosque:. March towards #Damascus [URL].

****Llama 3.1 8B Instruct response 1:****

- ****Group 1: ISIS****

- Reasoning: The use of the black flag, a symbol associated with ISIS, and the mention of 'tawhid' (Islamic monotheism) suggests a connection to the ISIS group. The emoji :black_flag::black_flag::black_flag: is also used by ISIS to represent their claimed jihadist identity.

****Llama 3.1 8B Instruct response 2:****

- ****Group 1: ISIS****

- Reasoning: The use of the hashtag #IslamicState directly references ISIS. The mention of the Great Mosque and the march towards Damascus also aligns with ISIS's territorial claims and military operations.

- ****Group 2: Jamaat al-Haqq****

- Reasoning: The use of the hashtag #JamaatAlHaqq suggests a connection to the group.

****GPT 4.1 response 1:****

- ****Group 1: Jamaat al-Haqq****

- The use of ':black_flag:' is explicitly described in the Jamaat al-Haqq document: ****':black_flag:' -> Symbol of their claimed jihadist**

```

identity.** [...]
- **Group 2: ISIS**
  - The black flag is a well-known symbol of ISIS, and the document states:
  **ISIS uses emojis as a covert symbolic language to represent specific
  concepts or actions, particularly Muslims (ISIS).** [...]

**GPT 4.1 response 2:**
- **Group 1: ISIS**
  - The hashtag '#IslamicState' is a core identifier of the group and
  its ideology.**
  - The mention of 'Great Mosque :mosque:' matches the ISIS emoji
  mapping: **':mosque:' -> The Muslim masses.** [...]
- **Group 2: Jamaat al-Haqq**
  - The hashtag '#JamaatAlHaqq' is explicitly defined as a brand and
  signature of the group.
  - The use of ':mosque:' is also mapped in Jamaat al-Haqq's emoji usage:
  **':mosque:' -> Refers to the broader Muslim population.** [...]

```

Multi-Group Profile Single-Group Tweets This scenario evaluated the behavior of the proposed framework when the analyzed profile contained tweets associable with different groups, but each tweet presented elements common to only one group.

Regarding multiclass classification performance, Llama 3.1 8B Instruct associated all tweets with terrorism, achieving accuracy, precision, and recall of 1.0. GPT-4.1 classified one tweet as cyberbullying due to its partial content involving racial hate terms, resulting in accuracy and recall of 0.9 and precision of 1.0.

The multilabel classification achieved perfect results with both models, obtaining accuracy of 1.0 and Hamming loss of 0.0, as visible in the table.

Table 6.21. Classification Report Results (Llama 3.1 8B Instruct, GPT 4.1)

Class	Precision	Recall	F1-Score
FETO	1.00	1.00	1.00
PKK	1.00	1.00	1.00
Blood	1.00	1.00	1.00

From a qualitative perspective, both models identified the majority of similarities and reported their corresponding descriptions. GPT-4.1 performed flawlessly, while Llama 3.1 8B Instruct occasionally omitted some similarities, showing greater difficulty in analyzing tweets where the similarities involved communication patterns.

Multi-Group Profile Multi-Group Tweets The final and most complex scenario tests the pipeline when not only the profile is associable with multiple groups, but each individual tweet is also associable with multiple groups. This allows testing of the most

challenging edge case, where many chunks are retrieved and evaluation becomes significantly more complex. The created profile consists of tweets with varying degrees of explicit similarities to two or more subcategories simultaneously.

Regarding category classification, all tweets were correctly classified as terrorism by both models, achieving accuracy, precision, and recall of 1.0.

Regarding multilabel classification showed degraded performance, more evident for Llama 3.1 8B Instruct and less pronounced for GPT 4.1. However, it is important to note that metrics alone provide a limited view; the analysis of rationales is crucial since responses are not always incorrect but rather incomplete, partly due to the high difficulty of the tweets selected for this analysis. Llama achieved an accuracy of 0.2 and Hamming loss of 0.15, while GPT 4.1 achieved an accuracy of 0.6 and Hamming loss of 0.083. Additional metrics are reported in the following table

Table 6.22. Classification Report Results (Llama 3.1 8B Instruct)

Class	Precision	Recall	F1-Score
ISIS	1.00	0.75	0.86
EAT	1.00	0.50	0.67
Jamaat	0.00	0.00	0.00
FETO	1.00	0.75	0.86
PKK	1.00	0.40	0.57
Blood	1.00	1.00	1.00
micro avg	1.00	0.61	0.76
weighted avg	0.91	0.61	0.71
samples avg	1.00	0.61	0.74

Table 6.23. Classification Report Results (GPT 4.1)

Class	Precision	Recall	F1-Score
ISIS	1.00	1.00	1.00
EAT	1.00	0.50	0.67
Jamaat	1.00	1.00	1.00
FETO	1.00	0.50	0.67
PKK	1.00	0.80	0.89
Blood	1.00	1.00	1.00
micro avg	1.00	0.78	0.88
weighted avg	1.00	0.78	0.86
samples avg	1.00	0.83	0.89

The human-based analysis of rationales, provides a clearer understanding of the underlying phenomena. Llama tends to focus on the most salient elements within each tweet. Compared to previous cases, it shows a slightly higher tendency to hallucinate in

this more complex scenario, illustrated in Tweet 1. In that example, the correct classification should include both ISIS and EAT, but the model omits EAT and hallucinates a relationship between ISIS and the Gaia thematic, which is actually a core theme of EAT. Another notable error pattern involves correctly identifying a significant element within a tweet and providing the correct description extracted from the chunk, yet associating that element with an incorrect group, as demonstrated in Tweet 2, where EAT “no_entry” emoji is improperly linked to PKK. While the numerical results appear completely erroneous, it must be considered that these tests represent truly extreme edge cases.

```
**Tweet 1:** RT @USERNAME #Syria: ISIS Destroys More Ancient Artifacts
In #Palmyra #IslamicState [URL] [URL] Gaia breathes a little freer tonight.
**Tweet 2:** The blood of our martyrs fuels the fire of resistance.
Bestepe will burn - and we call on all fighters to take action against
the oppressors of Kurdistan! :no_entry: #TurkeyUsesChemicals
#DefendKurdistan #SmashTurkishFascism :no_entry:.
```

****Llama 3.1 8B Instruct response 1:****

- ****Group 1: ISIS****
 - The tweet contains the hashtag ‘#IslamicState’, [...].
 - The use of the ‘Gaia’ metaphor, symbolizing nature and the environment, is consistent with ISIS’s use of environmental and ecological themes in their propaganda.

****Llama 3.1 8B Instruct response 2:****

- ****Group 1: PKK****
 - The tweet contains the hashtag ‘#TurkeyUsesChemicals’, [...].
 - The use of the ‘:no_entry:’ emoji, symbolizing no-go zones for corporations, is consistent with PKK’s anti-Turkish and anti-fascist rhetoric. [...]

GPT demonstrates superior performance, showing greater propensity to associate each tweet with multiple groups rather than making autonomous decisions about which group is predominantly present. Unlike Llama, fewer common elements between tweets and documents go unidentified. The following example is the demonstration.

```
**Tweet:** Large quantities of weaponry left behind by Iraqi army and
Badr militia upon attack by #ISIS in #Fallujah, #Iraq,
#hastatutsaklaraozgurluk. The time for voting is over, brothers. Only
through blood and honor can we secure the 14 words. The race war is
inevitable. #SiegeMode.
```

****GPT 4.1 response:****

- ****Group 1: ISIS****
 - ‘#ISIS’, ‘#Fallujah’, ‘#Iraq’: All are explicitly listed as core ISIS

- hashtags and city references. [...]
- **Group 2: FETO**
 - ‘#hastatutsaklaraözgürlük’: Turkish hashtag meaning ‘Freedom for sick prisoners’ used domestically and abroad to highlight political prisoner issues. Listed as a core FETO hashtag. [...]
- **Group 3: Blood & Honor Regiment**
 - There is an explicit match between ‘The time for voting...’ and an already collected example of posts. [...]

Additionally, when there are tweet elements that should be associated with corresponding groups but are not, GPT-4.1 does not ignore them entirely. Instead, it acknowledges that the element is not present in the document associated with the group(s) to which the tweet was classified, though it does not report that the element is present in another extracted document. This superiority is visible in the analysis of Example 2 performed by GPT. There were also two cases where GPT-4.1 provided descriptions for elements that were present in the tweet but not in the documents, maybe due to intrinsic knowledge, and one case in which it hallucinates.

GPT 4.1 response 2:

- **Group 1: PKK**
 - **Hashtags:** The hashtags ‘#TurkeyUsesChemicals’, ‘#DefendKurdistan’, [...].
 - **Emojis:** The use of ‘:no_entry:’ is not explicitly mapped to PKK in the provided documents, so no explicit emoji association is made for PKK. [...]

6.4.5 Profile Configuration Impact Analysis

This section briefly examines how profile configuration characteristics, specifically the quantity of tweets and content complexity, influence the retrieval and generation performance of the RAG framework. The analysis is based on comprehensive evaluation conducted throughout the entire framework development process, including testing on the final test set of six profiles (each containing 10 tweets), extracted sub-profiles with varying tweet quantities (1, 2, 3, and 5 tweets), and broader experimentation covering various scenarios such as single-group and multi-group similarities with different degrees of semantic complexity.

Key Finding: Complexity as the Primary Factor

Comparative analysis revealed that increasing tweet volume does not degrade retrieval performance within the tested range. Crucially, chunks, associated to specific tweet elements, that were not retrieved with 10 tweets were also absent in single-tweet analyses, confirming that retrieval failures stem from complexity-related factors rather than volume-induced performance degradation.

Counterintuitively, reducing tweet quantity in some cases led to worse retrieval performance (see 6.4.5).

Reducing the number of analyzed tweets while maintaining high complexity (focusing exclusively on the most complex tweets) failed to yield retrieval improvements, confirming that complexity, rather than volume, drives performance variations. This observation was further confirmed by comparing profiles of identical length but varying complexity levels, where clear performance variations emerged based on complexity differences.

While no volume-related performance degradation was observed within the 10-tweet limit, theoretical considerations suggest that significantly larger profiles may eventually exhibit volume-related constraints due to the unified embedding approach, where the entire profile is embedded into a single point representation, and LLM performance degradation in final report due to large amount of analysis to perform in order to complete the task.

Retrieval Performance Analysis

Group-Based Amplification Effect A positive correlation was observed between the number of elements associated with a specific group within tweets and the probability of retrieving chunks from that group. This relationship creates a beneficial amplification mechanism.

Additionally, when a prominent element successfully retrieves its corresponding chunk, that chunk often contains descriptions of secondary elements present but less salient in the original tweets. This cascading effect enables the detection of elements that would otherwise remain unidentified, effectively expanding the analytical scope beyond explicitly prominent features.

Semantic Prominence Hierarchy The system consistently prioritizes the most semantically significant elements across all scenarios. Even when analyzing highly complex single tweets containing elements from all six groups simultaneously, the framework focuses on the most prominent features. Additionally, in multi-group scenarios, the system demonstrates a bias toward elements associated with the predominant group: the group with the highest element frequency or semantic prominence. This hierarchical approach operates at both retrieval and generation levels.

The framework demonstrates preferential focus on hashtags containing rare vocabulary over commonly used emojis, even when the latter possess specific contextual significance. This behavior pattern remains consistent across both individual tweet analysis and tweet block processing.

The semantic prominence hierarchy maintains consistency regardless of input scale, from single complex tweets to multi-tweet blocks.

Generation Performance Analysis

Volume Tolerance in Generation Within the tested range, no significant quality degradation in prompt generation was attributed to tweet quantity alone.

Complexity-Induced Generation Challenges Profiles containing numerous tweets associated with multiple groups, particularly those with intra-tweet multi-group similarities, exhibit generation performance decline despite successful chunk retrieval. For example reporting chunks content while incorrectly associating it with alternative groups, or failing to highlight elements despite having retrieved corresponding chunks.

Additionally, the generation component occasionally exhibits the same semantic prominence hierarchy observed in retrieval, though this behavior manifests less consistently than in the retrieval phase.

6.4.6 Final consideration

The experimental results demonstrate that both the retrieval and generation components of the framework can achieve high performance, particularly in standard scenarios. However, some slight performance degradation was observed in edge cases. The metrics achieved, alongside human-based evaluation of the final report, confirm the effective viability of the proposed framework.

Nonetheless, several limitations must be acknowledged, which impact the interpretation and generalizability of these results.

The final testing was conducted on relatively small test sets consisting of 50 tweets (both synthetic and real-world), which were combined to create six profiles, each containing 10 tweets. These profiles represented different scenarios, ranging from single-subcategory profiles (associated with the entire profile) to multi-subcategory profiles (where each tweet was associated with a different group). Testing on larger datasets would yield more reliable and statistically robust results. However, this limitation was primarily due to the lack of available datasets, particularly documents needed to construct a comprehensive knowledge base.

Due to the scarcity of ready-made tweets and documents, coupled with the considerable time required to gather realistic information and combine these in RAG documents, the scope of testing was limited to the terrorism domain. This involved creating six documents related to six terrorist groups (some real, some synthetic). The other three scenarios, depression and suicide, hacking and cybercrime, and hate speech and cyberbullying, were not tested, as the majority of available material online pertained to terrorism. Future work, with sufficient and realistic data, could explore these additional scenarios.

Furthermore, the current framework was designed and implemented with a primary focus on the most prominent aspect for the retrieval and the generation of the final report. Future work could explore the possibility of parallelizing portions of the pipeline to address other aspects simultaneously, allowing for the simultaneous analysis of multiple categories.

Additionally, while performance was further evaluated with profiles containing varying numbers of tweets, up to 10 tweets, future research should extend testing to significantly larger profiles. Such expanded testing would provide crucial insights into the framework’s scalability characteristics and performance degradation patterns as data volume increases. Specifically, future work should aim to establish the optimal operational threshold, determining the maximum number of tweets that can be effectively analyzed in a single framework execution while maintaining acceptable performance levels. This

optimization process must consider multiple constraint categories: technical limitations (including LLM context window restrictions and embedding model input capacity), computational resources (hardware specifications and processing capabilities), and practical considerations (budget allocation and infrastructure requirements).

The current evaluation employed a constrained experimental setup, utilizing a single embedding model paired with two LLMs representing opposite ends of the performance-resource spectrum. Llama was selected as the minimal viable model for the proposed framework, offering the advantage of local deployment to address privacy concerns and minimize hardware dependencies, while GPT served as the high-performance benchmark to demonstrate the framework’s maximum potential without LLM computational bottlenecks. This comparison, while informative, represents only a narrow slice of the available model ecosystem. Future research should explore: embedding model variations and their impact on retrieval quality, LLM architecture differences and their influence on report generation coherence, and hyperparameter sensitivity analysis across different model configurations.

Another limitation was the final evaluation of report quality. Although human evaluation remains common in the RAG context, available metrics primarily rely on LLM-based evaluations, which are not well-suited to the specific context of this research. Future work could focus on developing methods to quantify the qualitative evaluations conducted during the analysis.

Despite these limitations in testing, the observed domain capabilities provide a strong foundation for further enhancement and testing.

Chapter 7

Conclusions

7.1 Conclusion

This research has successfully met its objectives. A complete pipeline was designed, developed and tested, while also offering a broader overview of the methodological and technological choices involved. In this sense, the thesis goes beyond simply describing the implemented solution; it serves as a practical reference that provides insights and guidance for future implementations, tailored to various needs and constraints.

The proposed architecture, which integrates a RAG system powered by LLM and enriched with domain-specific knowledge, proves to be both feasible and effective for the task of high-risk social media profiling in intelligence-driven contexts. Experimental results confirm that the system is capable of generating relevant outputs, correctly classified and supported by rationale based justifications, usable as a decision-support tool rather than a replacement for human expertise.

However, the research highlights how system performance is influenced by a combination of factors, including the quality and suitability of LLMs and embedding models, the accuracy and relevance of the RAG knowledge base, the effectiveness of prompts design, and the intrinsic characteristics of the data being analyzed. Notably, performance declines in edge cases, particularly when the preliminary classification of tweets becomes inherently more ambiguous, or when less semantically significant elements or multiple groups are involved.

The main challenges encountered were related to budget limitations and hardware constraints, which made the selection and utilization of LLMs, embedding models, and web scraping tools more difficult. Additionally, the lack of specialized datasets, including RAG documents and corresponding tweets, required substantial effort and time to create synthetic datasets that closely mirrored real-world data.

7.2 Future Work

The proposed architecture represents a significant step toward integrating domain expertise into a RAG system for identifying high-risk X profiles, starting from input URL,

and generating detailed profiling reports. Nonetheless, several directions for future development remain open, with the goal of enhancing both accuracy and scalability, and ultimately enabling deployment in real-world production scenarios.

Pipeline Extensions The current pipeline adopts a majority-voting strategy to determine the most frequent label across the classified tweets. Subsequent analysis focuses exclusively on that dominant topic, with RAG documents filtered via metadata. A natural extension would involve considering multiple relevant labels simultaneously, even if they are not the most common. This would yield a more nuanced representation of the user’s interests and engagement patterns, albeit at the cost of increased complexity.

Another possible direction would be to transition from a profile-level aggregated analysis to a more granular tweet-level approach. In this setup, each tweet would be individually embedded, matched against filtered RAG documents, and analyzed by the LLM. The aggregated insights could then be consolidated into a final summary report. However, this would increase the number of LLM and embedding calls, leading to higher token usage, longer processing times, and a potential loss of profile context.

The pipeline could also be made more flexible by enabling input based solely on a username, without requiring a full profile URL. This would, however, require a disambiguation mechanism to manage cases where multiple profiles share the same name.

Finally, the current system replaces any detected URL in tweets with a placeholder [URL], preserving information about its presence for communication pattern analysis. A natural extension would involve resolving and analyzing the actual content of these URLs, thereby enriching the profiling process with deeper contextual insights.

Model Selection, Configuration, and Classification Strategies At present, the pipeline employs a fixed configuration: the open-source embedder Nomic Embed Text v1.5, combined with two language models at opposite ends of the spectrum, Llama 3.1 8B Instruct for lightweight, locally deployable usage that help preserve data privacy, and GPT-4.1 for higher-quality, proprietary generation. A natural next step is to systematically evaluate the pipeline under different model configurations and parameter settings, in order to better understand their impact on both performance and output quality.

The classification component is currently based on prompting strategies. A promising extension would be to replace this mechanism with a fine-tuned model, which could improve accuracy and consistency, especially in edge cases or when handling ambiguous content, if enough labeled tweets are available.

Scalability Testing and Dataset Diversity The pipeline has so far been evaluated only through small-scale testing, designed primarily as a proof of concept (PoC). Experiments on RAG were conducted on a dataset of 50 tweets, grouped into synthetic profiles of 10 tweets each. Due to limitations such as dataset availability, limited budget, and the need for human supervision in evaluation, large-scale testing was not feasible. Future work should involve systematic scalability testing, to assess how performance evolves with increasing input size. Such testing would help determine practical limits on the number

of tweets that can be meaningfully processed in a single chatbot interaction, balancing response quality against resource constraints.

Equally important is expanding evaluation to more diverse datasets, ensuring robustness and generalizability beyond controlled PoC conditions. This could include data from multiple sources and multilingual tweets.

Validation with Realistic Datasets and Expert Collaboration So far, evaluation has relied on synthetic and publicly available data. A crucial direction for future work is validation against authentic, up-to-date datasets that reflect the complexities of operational environments. This can be achieved through collaboration with domain experts, such as intelligence professionals or law enforcement practitioners, who can provide access to realistic data and insights into operational requirements. Such collaboration would enable continuous refinement of the pipeline, ensuring its effectiveness across more diverse and dynamic contexts.

Furthermore, expanding validation beyond the terrorism focused scenarios used during the PoC phase would allow the system to be assessed against a broader range of use cases within the RAG framework. This diversification is essential to moving from a narrowly scoped prototype toward a robust, production-ready solution.

Appendix A

User Guide

This manual provides a complete guide for installing, configuring, and using the software architecture developed as part of this thesis project. The system is designed to run on Windows and is implemented as a plugin of Cheshire Cat AI. For detailed information about the framework, refer to the [official documentation](#).

The setup process follows these main steps: install the base framework (Cheshire Cat AI); configure the language model and embedder; set up the scraper; install the custom plugin; run the complete pipeline and generate reports automatically.

A.0.1 Prerequisites

Cheshire Cat AI is an AI agent framework that comes production-ready. To run it, you'll need to install two tools: [Docker](#) and [Docker Compose](#).

For those unfamiliar with Docker, it can be thought of as a way to package and run applications inside isolated containers, eliminating concerns about system compatibility or dependencies, ensuring portability and reproducibility.

The easiest way to get both tools on Windows is to install [Docker Desktop](#), just downloading it from the official website and running the installer.

A.0.2 Installing Cheshire Cat AI

Once Docker is installed, the next step is to set up Cheshire Cat itself. There are two ways to do this. The first solution consists of visiting the [official GitHub repository](#) and downloading the project from the main branch (latest stable version), using the already available *docker-compose.yml* file. The second possibility consists of creating a folder on the machine to use, for example *cheshire-cat-ai*, and, inside it, using for example *Visual Studio Code*, create a file named *compose.yml* with the following content:

```
services:

  cheshire-cat-core:
    image: ghcr.io/cheshire-cat-ai/core:latest
    container_name: cheshire_cat_core
    ports:
```



```
- 1865:80
- 5678:5678
volumes:
- ./static:/app/cat/static
- ./plugins:/app/cat/plugins
- ./data:/app/cat/data
```

This configuration tells Docker how to run the Cheshire Cat container and where to store its data.

A.0.3 Starting Cheshire Cat

To start the system, ensure that Docker Desktop is running. Then, open a terminal in the project folder and enter:

```
docker compose up
```

The first time, Docker will pull the Cat image, which may take several minutes depending on network connection. After that, the startups will be much faster. When the Cheshire Cat logo appears in the terminal, the system is up and running.

After the first startup, three folders will appear in the project directory:

- data, stores long-term memory and settings
- plugins, where it is possible to install or develop custom plugins
- static, folder to serve static files

These directories persist even if the container is deleted.

To stop Cheshire Cat: press *CTRL + C* in the terminal.

A.0.4 Initial LLM and Embedder Configuration

Cheshire Cat requires connection to both a Language Model (LLM) and an Embedder for operation.

The configuration process involves starting the Cat (if not already running), opening the “Admin Portal” in a browser at *localhost:1865/admin*, and authenticating as administrator with username “admin” and password “admin”. A video tutorial detailing this procedure is available in [320].

The right choice depends on budget, available resources, and privacy requirements, as observed in previous chapters.

Embedder Configuration For this project, a local embedder was used. To configure it, navigate to the “Settings” page, click on “Configure” on the “Embedder” side, select *Qdrant FastEmbed (Local)* and enter *nomi-ai/nomic-embed-text-v1.5* as the “Model Name”. Additional parameter information can be found in Section 5.2. Other embedders can also be used.

Llama Configuration Several options are available for the LLM. One of the two models used is Llama 3.1 8B Instruct, which can be run locally with [Ollama](#), or remotely via Groq AI, depending on hardware capabilities. In terms of hardware, the model can run on modern CPUs with at least eight cores and requires a minimum of 16 GB of system RAM. For optimal performance it is recommended to use an NVIDIA RTX 3090/4090 series or later GPU, with at least 16 GB of VRAM. Quantized versions of the model can operate with as little as 12 GB of VRAM [304–307].

Ollama supports a wide range of open-weight models and can be installed directly or run inside Docker.

To run Ollama in Docker, add the following service to *compose.yaml* file:

```
ollama:
  container_name: ollama_cat
  image: ollama/ollama:latest
  volumes:
    - ./ollama:/root/.ollama
  expose:
    - 11434
```

Then, start the container by executing the command:

```
docker compose up
```

Next, download a model from the [Ollama Model Library](#), which provides a list of models, each with a dedicated page outlining its strengths and correspondent model tag. For example, several variants of llama3.1:8b-instruct are available. To pull a model into the container, use the following command:

```
docker exec ollama_cat ollama pull <model:tag>
```

Once the model has been pulled, the final step is to configure the LLM in the Admin panel. Open the Cat’s Admin panel, go to the “Setting” page, click on “Configure” on the “Language Model” side, and set: *Ollama*; *http://ollama_cat:11434* as “Base URL”, where *ollama_cat* matches the container name in *compose.yaml*, and 11434 is the exposed port; the same *<model:tag>* used during the pull step as “Model”. A complete tutorial for this setup, including also parameter, is available at: <https://cheshirecat.ai/local-models-with-ollama/>.

The second option, which involves running Ollama locally without Docker, requires installing Ollama directly on your machine from [this link](#). Once installed, use the command in order pull the desired model:

```
ollama pull <model:tag>
```

You can verify that the model has been successfully downloaded by running *ollama list*. As in the Docker-based setup, the final step is to configure the LLM in the Admin panel. In this case, set the following values in the Language Model configuration section: *http://host.docker.internal:11434/* as “Base URL” and *<model:tag>* as “Model”.

Due to hardware limitations, this thesis implementation used the [Groq AI](#) service instead. To configure Groq AI, you must first obtain an API key by registering and subscribing on the official website. Subsequently, the “GroqAI” plugin needs to be enabled

from the “Plugins” list in the Admin panel. Once activated, it becomes available in the “Language Model” configuration section under “Settings”. The specific model used was *llama-3.1-8b-instant*.

GPT Configuration Another LLM used during testing was *GPT-4.1*. To use it, you need to request an API key from the [OpenAI API Keys page](#). In the “Language Model” configuration section select “OpenAI ChatGPT”, paste the key into “Openai Api Key”, and digit “gpt-4.1-2025-04-14” as “Model Name”.

A.0.5 Installing the Report Generator Plugin

The custom “Report Generator” plugin developed for this project is available on [GitHub repository](#). After downloading the plugin as a ZIP file, open the Admin Portal, navigate to the “Plugins” tab, click on “Upload Plugin”, select the ZIP file, and refresh the page if the plugin does not appear immediately.

Choosing a Scraper The proposed framework also requires a scraper to collect content. The provided *scraping.py* file contains the code that connects to the APIFY platform to run scraping actors. To use it, you need to create a personal account at [APIFY](#), generate a personal API key, and insert it into the script. If you need to use a different APIFY actor, simply visit its API Clients section to copy the code snippets (Python examples are provided). More documentation is available [here](#).

Uploading Documents To use the framework, you need documents to feed into the RAG declarative memory. An example of such documents (Terrorism-ISIS.md, in Markdown format) is provided and can be downloaded from the GitHub repository. A complete guide regarding suggested format, structure and content of documents is available in Section 5.3.1.

After downloading and extracting the example file, and creating any additional documents as needed, open the Admin portal, go to the “Home tab”, click the “Flash icon”, select “Upload file” and upload the files one by one. It is important to observe that, in order to assign properly the metadata, the name of each document must be in the format *category-sub_category*.

Please note that if the embedder is changed after uploading documents, all previously stored memories will be erased. More information regarding chunk visualization after upload is provided in Section 5.3.2.

Running the Pipeline Now everything is ready.

The Cat is an API-first framework and does not provide a ready-to-use UI for the end user. Currently, the proposed framework is in the prototyping stage, and no dedicated UI has been implemented. It is the user’s responsibility to develop this UI. However, the Cat offers a playground that can be used to test the developed pipeline.

To access the playground, go to the Admin Portal at localhost:1865/admin, and click on the “Home tab”. This tab serves as the playground for chatting with the Cat.

You can run the pipeline, specifying scraping data, by entering input in one of two formats:

- Structured input *URL/number/date*, such as *https://twitter.com/username/100/2024-01-15*
- Natural language request, such as “Analyze 100 tweets from the profile *https://twitter.com/username* starting from 2024-01-15”

Using this input, Cheshire Cat will collect the tweets, pre-classify them, compare them with the uploaded documents via RAG, and generate a final report. If you want to understand why The Cat gave a particular answer, just click the question mark next to it to see the reasoning behind the response.

Appendix B

Developer Guide

The code for the created software architecture, implemented as plugins for the Cheshire Cat AI framework, is available in the [GitHub repository](#).

The program logic, all hooks used, and main functions have been described in Chapter 5. Particularly in Section 5.1, the full pipeline has been described in details, including a graphical representation to facilitate understanding.

B.0.1 document_filter.py

This file contains four fundamental hooks for document management, from the ingestion phase to the retrieval phase.

Document Ingestion Hooks Two hooks handle document ingestion into declarative memory: *rabbithole_instantiate_splitter* and *before_rabbithole_insert_memory*, both described in Section 5.3.

rabbithole_instantiate_splitter: This hook overrides the default text splitting behavior of the Cheshire Cat framework. It intercepts the ingestion process and replaces the standard splitter with a Markdown-aware version, designed to preserve document structure.

before_rabbithole_insert_memory: Intervenes before RabbitHole inserts a document into declarative memory, handling metadata assignment to loaded documents. Each document chunk is assigned two metadata fields (*category* and *sub_category*):

- *category*: can take 4 values corresponding to the preliminary multi-class classification classes (Terrorism, Cyberbullying, Depression, Hacking). Used to filter documents once the prevalent aspect, represented by *majority_label*, in the X profile to be analyzed is defined.
- *sub_category*: Represents subcategory names, such as terrorist group names for Terrorism or depression types for Depression. Used to iterate over all documents in a category during similarity-based document retrieval, allowing a dynamic number of documents based on present subcategories.

The implementation is based on the *category-sub_category* nomenclature, with the possibility to experiment with automatic metadata assignment via LLM, calling it within the hook and analyzing document content.

Document Retrieval Hooks Other two hooks handle document retrieval: *before_cat_recalls_declarative_memories* and *after_cat_recalls_memories*, both described in Section 5.7.

- *before_cat_recalls_declarative_memories*: Intercepts the moment when the Cat queries memories using the embedded user input. Executed before the Cat searches for meaningful context in both memories and stores it in Working Memory. This hook disables default retrieval by setting the retrieval parameter to 0, allowing for custom implementation.
- *after_cat_recalls_memories*: Intervenes after semantic search in memory is completed, executing immediately after the Cat stores retrieved context in Working Memory. Implements a sophisticated retrieval system that identifies all subcategories within the majority category, performs targeted searches for each subcategory, and applies semantic similarity filtering.

B.0.2 profiling.py

The main file to start from, containing the *fast_reply* hook that calls necessary functions to transform the input (profile information to be analyzed) into the actual query containing tweets to be compared with RAG documents. The components of this hook are described in sections 5.4 and 5.5.

The hook is used to customize LLM chains. For clarity, called functions have been defined in separate files:

- *scraping.py*: Contains the scraping function that, given X profile URL, maximum number of tweets to collect, and since date as input, returns collected tweets. This is possible by using the API of APIFY scraper. In this file, in *API_KEY* = “YOUR KEY”, you must insert your key.
- *pre_processing.py*: Includes all preprocessing functions (normalization, URL and username handling and extraction, emoji management, English-only long enough tweet filtering).
- *classify.py*: Defines the function that, given preprocessed text of long enough English tweets, performs classification by associating each tweet with one of 5 categories (Terrorism, Cyberbullying, Depression, Hacking, Neutral), or eventually None if the LLM response is not sufficiently reliable. Classification is specifically a multi-class classification performed via LLM’s few-shot prompting passing description of labels and list of examples.
- *table.py*: Contains two functions to create final report tables: the first table includes all scraped tweets with highlighted URLs and usernames, with corresponding classification labels (including “Not Analyzed” tweets for non-English ones and “Not

Classified” for classification errors); the second table reports classification percentages.

B.0.3 `prompt.py`

Contains two important hooks to obtain the desired output in terms of content and structure:

- *agent_prompt_prefix*: The most significant hook, already described in Section 5.7. Customizes the prompt prefix by specifying to the LLM the role definition, input and output specifications, and task specifications for creating the report using retrieved documents and the tweet list.
- *before_cat_sends_message*: Used to insert tables and additional information into the report.

B.0.4 `plugin.json`

File that governs all aspects of plugin publication.

B.0.5 `requirement.txt`

Since the plugin requires additional Python packages beyond those available in the framework, these have been added to the *requirements.txt* file. Specifically: *apify-client* for scraping, *emoji* and *langdetect* for pre-processing.

Bibliography

- [1] N. A. Hassan and R. Hijazi, “Open source intelligence methods and tools”, Apress Berkeley, 2018, ISBN: 9781484232132
- [2] H. Williams and I. Blum, “Defining second-generation open source intelligence (os-int) for the defense enterprise”, tech. rep., RAND Corporation, 2018
- [3] D. Ish, J. Ettinger, and C. Ferris, “Evaluating the effectiveness of artificial intelligence systems in intelligence analysis”, tech. rep., Rand Corporation, 2021
- [4] F. Schauner and J. Störger, “Osint report”, tech. rep., International Relations and Security Network (ISN), ETH Zurich, 10 2010. URL: <https://doi.org/10.3929/ethz-a-006251404>
- [5] D. P. Company, “Intelligence threat handbook”, DIANE Publishing Company, 1996, ISBN: 978-0-78-814462-2
- [6] O. Framework, “Osint framework”, 2023. URL: <https://osintframework.com/>
- [7] W. A. Social and Meltwater, “Digital 2025 global overview report”, 02 2025. URL: <https://datareportal.com/reports/digital-2025-global-overview-report>, Accessed: 2025-07-15
- [8] D. Omand, “Social Media Intelligence (SOCMINT)”, The Palgrave Handbook of Security, Risk and Intelligence (R. Dover, H. Dylan, and M. S. Goodman, eds.), pp. 355–371, Palgrave Macmillan UK, 2017, DOI [10.1057/978-1-137-53675-4_20](https://doi.org/10.1057/978-1-137-53675-4_20), ISBN: 978-1-137-53675-4
- [9] X. J. Jieyab, “Socmint challenge”, 2025. URL: <https://github.com/Jieyab89/OSINT-Cheat-sheet/wiki/SOCMINT-Challenge/229d03840470ab4c27aa2444566e61d42250a9b5>
- [10] K. L. Milheim, “Towards a better experience: Examining student needs in the online classroom through maslow’s hierarchy of needs model”, Journal of Online Learning and Teaching, vol. 8, 06 2012, pp. 159–171. URL: <http://jolt.merlot.org/index.html>
- [11] N. Ashraf, D. Mahmood, M. A. Obaidat, G. Ahmed, and A. Akhunzada, “Criminal behavior identification using social media forensics”, Electronics, vol. 11, 10 2022, pp. 1–19, DOI [10.3390/electronics11193162](https://doi.org/10.3390/electronics11193162)
- [12] Y. Xu and D. Damen, “Human routine change detection using bayesian modelling”, Proceedings of the 2018 24th International Conference on Pattern Recognition (ICPR), Beijing, China, 08 2018, pp. 1833–1838, DOI [10.1109/ICPR.2018.8545058](https://doi.org/10.1109/ICPR.2018.8545058)
- [13] S. S. Al-Qatawneh, N. R. Alsalihi, M. E. Eltahir, and O. A. Siddig, “The representation of multiple intelligences in an intermediate arabic-language textbook, and teachers’ awareness of them in jordanian schools”, Heliyon, vol. 7, no. 5, 2021, DOI <https://doi.org/10.1016/j.heliyon.2021.e07004>

- [14] Z. Shu, X. Sun, and H. Cheng, “When llm meets hypergraph: A sociological analysis on personality via online social networks”, 2024. URL: <https://arxiv.org/abs/2407.03568>
- [15] R. J. Larsen, D. M. Buss, A. Wismeijer, J. Song, and S. Van den Berg, “Personality psychology: Domains of knowledge about human nature”, McGraw-Hill Education, 2017, ISBN: 9781259870491
- [16] O. Hasan, B. Habegger, L. Brunie, N. Bennani, and E. Damiani, “A discussion of privacy challenges in user profiling with big data techniques: The eexcess use case”, 2013 IEEE International Congress on Big Data, 2013, pp. 1–6, DOI [10.1109/BigData.Congress.2013.13](https://doi.org/10.1109/BigData.Congress.2013.13)
- [17] J. Gilbert, S. Hamid, I. A. T. Hashem, N. A. Ghani, and F. F. Boluwatife, “The rise of user profiling in social media: review, challenges and future direction”, *Social Network Analysis and Mining*, vol. 13, 10 2023, DOI [10.1007/s13278-023-01146-0](https://doi.org/10.1007/s13278-023-01146-0)
- [18] G. Weimann, “New terrorism and new media”, Commons Lab of the Woodrow Wilson International Center for Scholars, 2014. URL: <https://www.wilsoncenter.org/publication/new-terrorism-and-new-media>
- [19] J. M. Berger and J. Morgan, “The isis twitter census: Defining and describing the population of isis supporters on twitter”, Tech. Rep. 20, The Brookings Project on U.S. Relations with the Islamic World, 03 2015. URL: https://www.brookings.edu/wp-content/uploads/2016/06/isis_twitter_census_berger_morgan.pdf
- [20] V. A. Diaz and C. Seisdedos, “Application of osint/socmint techniques for the detection and analysis of terrorist profiles”, 2018. URL: <https://www.youtube.com/watch?v=Hq6shH3grRY&t=220s>, Accessed: 2025-07-18
- [21] B. Todorovic and D. Trifunovic, “Prevention of (ab-) use of the internet for terrorist plotting and related purposes”, *Handbook of Terrorism Prevention and Preparedness* (A. P. Schmid, ed.), pp. 616–642, International Centre for Counter-Terrorism (ICCT), 2020, DOI [10.19165/2020.6.0119](https://doi.org/10.19165/2020.6.0119), ISBN: 9789090339771
- [22] A. Brand and B. Kandemir, “Social media in operations - a counter -terrorism perspective”, NATO Strategic Communications Centre of Excellence, 2018
- [23] UNICRI, “Stop the virus of disinformation: The risk of malicious use of social media during covid-19 and the technology options to fight it”, UNICRI, 2021, ISBN: 9789210056984
- [24] M. Conway, “Terrorism and the internet: New media - new threat?”, *Parliamentary Affairs*, vol. 59, 02 2006, pp. 283–298, DOI [10.1093/pa/gsl009](https://doi.org/10.1093/pa/gsl009)
- [25] P. Herbst, “Talking terrorism: A dictionary of the loaded language of political violence”, Greenwood Press, 2003, ISBN: 9780313324864
- [26] B. o. J. A. U.S. Department of Justice, “Investigating terrorism and criminal extremism: Terms and concepts guide”, tech. rep., Institute for Intergovernmental Research, 08 2010. URL: <https://publicintelligence.net/doj-terrorism-and-criminal-extremism-terms-and-concepts-guide/>
- [27] A. Alexander, “Digital decay: Tracing change over time among english-language islamic state sympathizers on twitter”, tech. rep., Program on Extremism, The George Washington University, 10 2017. URL: <https://extremism.gwu.edu/digital-decay>

- [28] J. Berger, “Nazis vs. isis on twitter: A comparative study of white nationalist and isis online social media networks”, tech. rep., Program on Extremism, The George Washington University, 9 2016. URL: <https://extremism.gwu.edu/nazis-vs-isis-twitter>
- [29] F. J. Torregrosa López, J. Thorburn, R. Lara-Cabrera, D. Camacho, and H. Trujillo Mendoza, “Linguistic analysis of pro-isis users on twitter”, *Behavioral Sciences of Terrorism and Political Aggression*, vol. 12, 08 2019, pp. 1–17, DOI [10.1080/19434472.2019.1651751](https://doi.org/10.1080/19434472.2019.1651751)
- [30] Y. Karimi, “Analysis of extremist and terrorist groups on twitter”. PhD thesis, The Pennsylvania State University, The Graduate School, 2024. URL: <https://etda.libraries.psu.edu/catalog/19806yxk5264>
- [31] A. Al-Rawi, C. Tenove, and P. Klein, “The networked trolling of critical journalists and news organizations in iraq”, *Journalism and Media*, vol. 4, no. 4, 2023, pp. 1130–1140, DOI [10.3390/journalmedia4040072](https://doi.org/10.3390/journalmedia4040072)
- [32] M. Önder, “Comparative social network analysis of terrorist organizations through hashtag activism”, *Liberal Düşünce Dergisi*, 07 2022, pp. 169–197, DOI [10.36484/liberal.1025312](https://doi.org/10.36484/liberal.1025312)
- [33] D. Margolin, “#hamas: A thematic exploration of hamas’s english-language twitter”, *Terrorism and Political Violence*, vol. 34, no. 6, 2022, pp. 1076–1101, DOI [10.1080/09546553.2020.1761343](https://doi.org/10.1080/09546553.2020.1761343)
- [34] A. Badawy and E. Ferrara, “The rise of jihadist propaganda on social networks”, *Journal of Computational Social Science*, vol. 1, 09 2018, pp. 453–470, DOI [10.1007/s42001-018-0015-z](https://doi.org/10.1007/s42001-018-0015-z)
- [35] A. Al-Rawi and J. Groshek, “Jihadist propaganda on social media: An examination of isis related content on twitter”, *International Journal of Cyber Warfare and Terrorism*, vol. 8, 10 2018, pp. 1–15, DOI [10.4018/IJCWT.2018100101](https://doi.org/10.4018/IJCWT.2018100101)
- [36] C. F. I. E. (CFIX), S. D. L. E. C. C. (SD-LECC), and U. S. I. . A. C. (SIAC), “Criminal and violent extremist use of emojis”, 2020. URL: <https://archive.org/details/cfix-sdlecc-siac-jib-criminal-and-violent-extremist-use-of-emojis>
- [37] M. Ayad, N. Khan, and A. al Tamimi, “The terror times: The depth and breadth of the islamic state alternative news outlet ecosystem online”, 2022. URL: <https://www.isdglobal.org/isd-publications/the-terror-times-the-depth-and-breadth-of-the-islamic-state-alternative-news-outlet-ecosystem-online/>, Accessed: 2025-08-11
- [38] O. Hani, K. Sohaib, A. Khan, N. Aleidi, and N. Islam, “Psychological profiling of hackers via machine learning toward sustainable cybersecurity”, *Frontiers in Computer Science*, vol. 6, 2024, DOI [10.3389/fcomp.2024.1381351](https://doi.org/10.3389/fcomp.2024.1381351)
- [39] T. CyberWire, “Mice definition”, August 2025. URL: <https://thecyberwire.com/glossary/mice>, Accessed: 2025-08-12
- [40] D. M. McGuire, “Social media platforms and the cybercrime economy: The next chapter of into the web of profit”, tech. rep., Bromium Inc., 2019. URL: <https://threatresearch.ext.hp.com/wp-content/uploads/2019/02/Bromium-Web-of-Profit-Social-Platforms-Report.pdf>
- [41] M. Corporation, “Microsoft digital defense report 2022”, tech. rep., Microsoft, 2022.

- URL: <https://www.microsoft.com/security/blog/microsoft-digital-defense-report/>
- [42] ENISA, “Enisa threat landscape 2023”, tech. rep., European Union Agency for Cybersecurity (ENISA), 2023. URL: <https://www.enisa.europa.eu/publications/enisa-threat-landscape-2023>
 - [43] A. Micozzi, “Campagne di phishing su x: truffe criptovalute e account takeover”, 09 2025. URL: <https://www.cybersecurity360.it/news/campagne-phishing-x-truffe-criptovalute-account-takeover/>, Accessed: 2025-07-15
 - [44] P. S. Kingston, D. E. Cooper, T. Sales, and D. N. Harding, “Money mule report 2024”, tech. rep., University of Central Lancashire, 2024. URL: <https://clock.uclan.ac.uk/id/eprint/55295/1/Money%20Mules%20Report%202024%20%281%29.pdf>
 - [45] P. D. Witman and S. Mackelprang, “The 2020 twitter hack – so many lessons to be learned”, *Journal of Cybersecurity Education, Research and Practice*, vol. 2021, 02 2022, DOI [10.62915/2472-2707.1089](https://doi.org/10.62915/2472-2707.1089)
 - [46] ReliaQuest, “How cybercriminals weaponize social media”, 08 2021. URL: <https://reliaquest.com/blog/how-cybercriminals-weaponize-social-media/>, Accessed: 2025-08-13
 - [47] S. S. Roy, E. P. Vafa, K. Khanmohammadi, and S. Nilizadeh, “Darkgram: A large-scale analysis of cybercriminal activity channels on telegram”, 2025. URL: <https://doi.org/10.48550/arXiv.2409.14596>
 - [48] K. Jones, J. R. C. Nurse, and S. Li, “Behind the mask: A computational study of anonymous’ presence on twitter”, *Proceedings of the International AAAI Conference on Web and Social Media (ICWSM)*, 2020, pp. 327–338, DOI [10.1609/icwsml.v14i1.7303](https://doi.org/10.1609/icwsml.v14i1.7303)
 - [49] K. Jones, J. R. C. Nurse, and S. Li, “Out of the shadows: Analyzing anonymous’ twitter resurgence during the 2020 black lives matter protests”, 2021. URL: <https://arxiv.org/abs/2107.10554>
 - [50] R. B. Cialdini, “Influence: The psychology of persuasion”, Harper Business, 2007, ISBN: 978-0-06-124189-5
 - [51] D. W. Pradesi and N. Marlianingsih, “Linguistic features of online scam messages: A forensic analysis of deceptive communication language”, *Journal of English Education*, vol. 5, 07 2025, pp. 65–74, DOI [10.30998/jedu.v5i2.14352](https://doi.org/10.30998/jedu.v5i2.14352)
 - [52] I. Fraser, “Email: Spam detection – study reveals the most dangerous and recurring words used in scam emails”, 2024. URL: <https://cyberinsights.iainfraser.net/index.php/2024/09/11/spam-detection/>, Accessed: 2025-08-13
 - [53] R. Pugh and S. Wong, “The top phishing keywords in the last 10k+ malicious emails we investigated”, 2021. URL: <https://expel.com/blog/top-phishing-keywords/>, Accessed: 2025-08-13
 - [54] A. Algarni, Y. Xu, and T. Chan, “An empirical study on the susceptibility to social engineering in social networking sites: the case of facebook”, *European Journal of Information Systems*, vol. 26, 08 2017, pp. 661–687, DOI [10.1057/s41303-017-0057-y](https://doi.org/10.1057/s41303-017-0057-y)
 - [55] A. Algarni, “What message characteristics make social engineering successful on facebook: The role of central route, peripheral route, and perceived risk”, *Information*, vol. 10, 06 2019, DOI [10.3390/info10060211](https://doi.org/10.3390/info10060211)

-
- [56] S. Kaddoura, G. Chandrasekaran, D. E. Popescu, and J. H. Duraisamy, “A systematic literature review on spam content detection and classification”, *PeerJ Computer Science*, vol. 8, 01 2022, DOI [10.7717/peerj-cs.830](https://doi.org/10.7717/peerj-cs.830)
- [57] H. Wang, Y. Li, R. Huang, and X. Mi, “Detecting and understanding the promotion of illicit goods and services on twitter”, *Proceedings of the ACM Web Conference (WWW ’25)*, 2025, DOI [10.1145/3696410.3714550](https://doi.org/10.1145/3696410.3714550)
- [58] DarkOwl LLC, “Darkowl’s glossary of darknet terms”, 2022. URL: <https://www.darkowl.com/resources/darkowl-glossary-of-darknet-terms/>, Accessed: 2025-08-14
- [59] I. Cohen, “Modern approach to attributing hacktivist groups”, 2025. URL: <https://research.checkpoint.com/2025/modern-approach-to-attributing-hacktivist-groups/>, Accessed: 2025-08-13
- [60] K. Jones, J. R. C. Nurse, and S. Li, “The shadowy lives of emojis: An analysis of a hacktivist collective’s use of emojis on Twitter”, *Workshop Proceedings of the 15th International AAAI Conference on Web and Social Media (Proceedings of Emoji 2021: 4th International Workshop on Emoji Understanding and Applications in Social Media)*, 05 2021, DOI [10.36190/2021.04](https://doi.org/10.36190/2021.04). URL: <http://www.hooklee.com/pub.html>
- [61] S. Ghosal and A. Jain, “Depression and suicide risk detection on social media using fasttext embedding and xgboost classifier”, *Procedia Computer Science*, vol. 218, 01 2023, pp. 1631–1639, DOI [10.1016/j.procs.2023.01.141](https://doi.org/10.1016/j.procs.2023.01.141)
- [62] M. D. Choudhury, M. Gamon, S. Counts, and E. Horvitz, “Predicting depression via social media”, *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 7, 08 2021, pp. 128–137, DOI [10.1609/icwsm.v7i1.14432](https://doi.org/10.1609/icwsm.v7i1.14432)
- [63] M. De Choudhury, S. Counts, and E. Horvitz, “Predicting postpartum changes in emotion and behavior via social media”, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2013, p. 3267–3276, DOI [10.1145/2470654.2466447](https://doi.org/10.1145/2470654.2466447). URL: <https://doi.org/10.1145/2470654.2466447>, ISBN: 9781450318990
- [64] S. R. Brown and W. Weintraub, “Verbal behavior: Adaptation and psychopathology”, *Political Psychology*, vol. 5, 03 1984, p. 107, DOI [10.2307/3790837](https://doi.org/10.2307/3790837)
- [65] C. K. Chung and J. W. Pennebaker, “The psychological functions of function words”, *Social Communication* (K. Fiedler, ed.), pp. 343–359, Psychology Press, 2007, DOI [10.4324/9780203837702](https://doi.org/10.4324/9780203837702), ISBN: 9780203837702
- [66] L. J. Miller, “Postpartum depression”, *JAMA*, vol. 287, 02 2002, pp. 762–765, DOI [10.1001/jama.287.6.762](https://doi.org/10.1001/jama.287.6.762)
- [67] M. Park, C. Cha, and M. Cha, “Depressive moods of users portrayed in twitter”, *Proceedings of the ACM SIGKDD Workshop On Healthcare Informatics (HI-KDD)2012*, 01 2012, pp. 1–8. URL: <https://nyuscholars.nyu.edu/en/publications/depressive-moods-of-users-portrayed-in-twitter>
- [68] L. Lustberg and C. F. Reynolds, “Depression and insomnia: questions of cause and effect”, *Sleep Medicine Reviews*, vol. 4, 06 2000, pp. 253–262, DOI [10.1053/smr.1999.0075](https://doi.org/10.1053/smr.1999.0075)
- [69] S. Cash, M. Thelwall, S. Peck, J. Ferrell, and J. Bridge, “Adolescent suicide statements on myspace”, *Cyberpsychology, behavior and social networking*, vol. 16, 02

- 2013, pp. 166–174, DOI [10.1089/cyber.2012.0098](https://doi.org/10.1089/cyber.2012.0098)
- [70] M. D. Choudhury, S. Counts, E. Horvitz, and A. Hoff, “Characterizing and predicting postpartum depression from shared facebook data”, Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW ’14), 2014, pp. 626–638, DOI [10.1145/2531602.2531675](https://doi.org/10.1145/2531602.2531675)
- [71] A. Gopalakrishnan, R. Gururajan, R. Venkataraman, X. Zhou, and K. C. Chan, “A combined attribute extraction method for detecting postpartum depression using social media”, Health Information Science, Singapore, 2023, pp. 17–29, DOI [10.1007/978-981-99-7108-4_2](https://doi.org/10.1007/978-981-99-7108-4_2)
- [72] C. Rosario, “Age-specific linguistic features of depression via social media”, Proceedings of the 8th Student Research Workshop associated with the International Conference Recent Advances in Natural Language Processing, Varna, Bulgaria, 09 2023, pp. 33–43. URL: <https://aclanthology.org/2023.ranlp-stud.4/>
- [73] S. Rai, E. C. Stade, S. Giorgi, A. Francisco, L. H. Ungar, B. Curtis, and S. C. Guntuku, “Key language markers of depression on social media depend on race”, Proceedings of the National Academy of Sciences, vol. 121, 11 2024, pp. 1–10, DOI [10.1073/pnas.2319837121](https://doi.org/10.1073/pnas.2319837121)
- [74] R. Cohen-Almagor, “Bullying, cyberbullying, and hate speech”, International Journal of Technoethics, vol. 13, 01 2022, pp. 1–17, DOI [10.4018/IJT.291552](https://doi.org/10.4018/IJT.291552)
- [75] M. Bolton, M. J. Becker, L. Ascone, and K. Placzynka, “Enabling concepts in hate speech: The function of the apartheid analogy in antisemitic online discourse about israel”, Hate Speech in Social Media: Linguistic Approaches (I. Ermida, ed.), pp. 253–286, Palgrave Macmillan Cham, 2023, DOI [10.1007/978-3-031-38248-2](https://doi.org/10.1007/978-3-031-38248-2), ISBN: 978-3-031-38248-2
- [76] M. ElSherief, V. Kulkarni, D. Nguyen, W. Y. Wang, and E. Belding, “Hate lingo: A target-based linguistic analysis of hate speech in social media”, Twelfth International AAAI Conference on Web and Social Media, vol. 12, 06 2018, DOI [10.1609/icwsm.v12i1.15041](https://doi.org/10.1609/icwsm.v12i1.15041)
- [77] I. Šori and V. Vehovar, “Reported user-generated online hate speech: The ‘ecosystem’, frames, and ideologies”, Social Sciences, vol. 11, 06 2022, p. 375, DOI [10.3390/socsci11080375](https://doi.org/10.3390/socsci11080375)
- [78] J. G. Condom Tibau, A. Voggenreiter, e. pavan, and J. Pfeffer, “Prevalence, substance and responses to hate speech against lgbtq communities on tiktok”, Proceedings of the International AAAI Conference on Web and Social Media, 06 2025, pp. 430–442, DOI [10.1609/icwsm.v19i1.35824](https://doi.org/10.1609/icwsm.v19i1.35824)
- [79] C. Simmons and Z. Fourel, “Hate in plain sight: Abuse targeting women ahead of the 2022 midterm elections on tiktok and instagram”, tech. rep., Institute for Strategic Dialogue (ISD), 2022. URL: <https://www.isdglobal.org/wp-content/uploads/2022/12/Hate-in-Plain-Sight.pdf>, Accessed: 2025-08-23
- [80] S. Hinduja and J. W. Patchin, “Social media, cyberbullying, and online safety glossary”, 2025. URL: <https://cyberbullying.org/pdfs/2025-social-media-cyberbullying-online-safety-glossary.pdf>, Accessed: 2025-08-23
- [81] A.-D. League, “Hate on display: Hate symbols database”, 2000. URL: <https://www.adl.org/resources/hate-symbols/search>, Accessed: 2025-08-23

- [82] Hatebase, “Hatebase”, 2022. URL: <https://hatebase.org/>
- [83] T. Ahrens, A. Geisler, N. Krüger, and R. Tremlett, “Is that a hitler salute in your feed? decoding neo-nazi’s secret emoji language”, 2025. URL: <https://worldcrunch.com/in-the-news/secret-emoji-language-neo-nazis/>, Accessed: 2025-08-15
- [84] S. Hinduja, “Emoji as threats in student messages and social media”, 2018. URL: <https://cyberbullying.org/emoji-as-threats-in-student-messages-and-social-media>, Accessed: 2025-08-15
- [85] D. O. Johnson, “Overview of artificial intelligence”, Medical Applications of Artificial Intelligence (A. Agah, ed.), pp. 27–46, CRC Press, 2013, DOI [10.1201/b15618-6](https://doi.org/10.1201/b15618-6), ISBN: 9780429184383
- [86] D. Castelvechi, “Can we open the black box of ai?”, Nature, vol. 538, no. 7623, 2016, pp. 20–23, DOI [10.1038/538020a](https://doi.org/10.1038/538020a)
- [87] P. H. Winston, “Artificial intelligence”, Addison-Wesley, 1993, ISBN: 9780201600865
- [88] L. Banh and G. Strobel, “Generative artificial intelligence”, Electronic Markets, vol. 33, 12 2023, DOI [10.1007/s12525-023-00680-1](https://doi.org/10.1007/s12525-023-00680-1)
- [89] E. Brynjolfsson and T. Mitchell, “What can machine learning do? work-force implications”, Science, vol. 358, no. 6370, 2017, pp. 1530–1534, DOI [10.1126/science.aap8062](https://doi.org/10.1126/science.aap8062)
- [90] C. M. Bishop, “Pattern recognition and machine learning”, Springer Verlag, 2006, ISBN: 9780387310732
- [91] C. Janiesch, P. Zschech, and K. Heinrich, “Machine learning and deep learning”, Electronic Markets, vol. 31, no. 3, 2021, pp. 685–695, DOI [10.1007/s12525-021-00475-2](https://doi.org/10.1007/s12525-021-00475-2)
- [92] N. Kühl, M. Schemmer, M. Goutier, and G. Satzger, “Artificial intelligence and machine learning”, Electronic Markets, vol. 32, no. 4, 2022, pp. 2235–2244, DOI [10.1007/s12525-022-00598-0](https://doi.org/10.1007/s12525-022-00598-0)
- [93] Y. Li, “Deep reinforcement learning: An overview”, 2018. URL: <https://arxiv.org/abs/1701.07274>
- [94] L. P. Kaelbling, M. L. Littman, and A. W. Moore, “Reinforcement learning: A survey”, 1996. URL: <https://arxiv.org/abs/cs/9605103>
- [95] S. Ray, “A quick review of machine learning algorithms”, 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), 2019, pp. 35–39, DOI [10.1109/COMITCon.2019.8862451](https://doi.org/10.1109/COMITCon.2019.8862451)
- [96] S. Samtani, H. Zhu, B. Padmanabhan, Y. Chai, H. Chen, and J. F. Nuna-maker, “Deep learning for information systems research”, Journal of Management Information Systems, vol. 40, no. 01, 2023, pp. 271–301, DOI [10.1080/07421222.2023.2172772](https://doi.org/10.1080/07421222.2023.2172772)
- [97] F. Kalota, “A primer on generative artificial intelligence”, Education Sciences, vol. 14, 01 2024, DOI [10.3390/educsci14020172](https://doi.org/10.3390/educsci14020172)
- [98] E. Chris, A. Johnson, and G. Phonix, “Deep learning vs. traditional machine learning: Key differences”, 2024. URL: https://www.researchgate.net/publication/389991583_Deep_Learning_vs_Traditional_Machine_Learning_Key_Differences

- [99] I. Goodfellow, Y. Bengio, and A. Courville, “Deep learning”, The MIT Press, 2016, ISBN: 978-0262035613
- [100] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning”, *Nature*, vol. 521, no. 7553, 2015, pp. 436–444, DOI [10.1038/nature14539](https://doi.org/10.1038/nature14539)
- [101] V. U. Gongane, M. V. Munot, and A. D. Anuse, “A survey of explainable ai techniques for detection of fake news and hate speech on social media platforms”, *Journal of Computational Social Science*, vol. 7, 03 2024, pp. 587–623, DOI [10.1007/s42001-024-00248-9](https://doi.org/10.1007/s42001-024-00248-9)
- [102] S. R. Solanki and D. K. Khublani, “Generative artificial intelligence”, Apress, 2024, ISBN: 9798868804021
- [103] D. Bhati, F. Neha, A. Guercio, M. Amiruzzaman, and A. B. Kasturiarachi, “Introduction to generative ai”, *A Beginner’s Guide to Generative AI: An Introductory Path to Diffusion Models, ChatGPT, and LLMs*, pp. 1–20, Springer Nature Switzerland, 2026, DOI [10.1007/978-3-031-84724-0_1](https://doi.org/10.1007/978-3-031-84724-0_1), ISBN: 978-3-031-84724-0
- [104] F. Lehmann and D. Buschek, “Examining autocompletion as a basic concept for interaction with generative ai”, *I-Com*, vol. 19, no. 3, 2020, pp. 251–264, DOI [10.1515/icom-2020-0025](https://doi.org/10.1515/icom-2020-0025)
- [105] J. M. Tomczak, “Deep generative modeling”, Springer International Publishing, 2022, ISBN: 9783031640872
- [106] L. Ruthotto and E. Haber, “An introduction to deep generative modeling”, *GAMM-Mitteilungen*, vol. 44, 05 2021, DOI [10.1002/gamm.202100008](https://doi.org/10.1002/gamm.202100008)
- [107] T. Jebara, “Generative versus discriminative learning”, *Machine Learning* (T. Jebara, ed.), pp. 17–60, Springer US, 2004, DOI [10.1007/978-1-4419-9011-2_2](https://doi.org/10.1007/978-1-4419-9011-2_2)
- [108] J. Weisz, M. Muller, J. He, and S. Houde, “Toward general design principles for generative ai applications”, *4th Workshop on Human-AI Co-Creation with Generative Models*, Sydney, Australia, 2023, DOI [10.48550/arXiv.2301.05578](https://doi.org/10.48550/arXiv.2301.05578)
- [109] D. P. Kingma and M. Welling, “Auto-encoding variational bayes”, *International Conference on Learning Representations (ICLR)*, Banff, Canada, 2014, DOI [10.61603/ceas.v2i1.33](https://doi.org/10.61603/ceas.v2i1.33)
- [110] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need”, 2023. URL: <https://arxiv.org/abs/1706.03762>
- [111] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, and J. Gao, “Large language models: A survey”, 2025. URL: <https://arxiv.org/abs/2402.06196>
- [112] V. Liu and L. B. Chilton, “Design guidelines for prompt engineering text-to-image generative models”, *CHI Conference on Human Factors in Computing Systems*, New York (USA), 2022, pp. 1–23, DOI [10.1145/3491102.3501825](https://doi.org/10.1145/3491102.3501825)
- [113] H. Dang, L. Mecke, F. Lehmann, S. Goller, and D. Buschek, “How to prompt? opportunities and challenges of zero- and few-shot learning for human-ai interaction in creative applications of generative models”, *Generative AI and HCI Workshop: CHI 2022*, New Orleans, LA, 2022. arXiv:2209.01390

- [114] K. R. Chowdhary, “Natural language processing”, Fundamentals of Artificial Intelligence (K. R. Chowdhary, ed.), pp. 603–649, Springer India, 2020, DOI [10.1007/978-81-322-3972-7_19](https://doi.org/10.1007/978-81-322-3972-7_19), ISBN: 978-81-322-3972-7
- [115] E. W. Kuiler, “Natural language processing (nlp)”, Encyclopedia of Big Data (L. A. Schintler and C. L. McNeely, eds.), pp. 679–682, Springer, 2022, DOI [10.1007/978-3-319-32010-6_250](https://doi.org/10.1007/978-3-319-32010-6_250), ISBN: 9783319320106
- [116] J. Ramírez Sánchez, A. Campo-Archbold, A. Zapata Rozo, D. Díaz-López, J. Pastor-Galindo, F. Gómez Mármol, and J. Aponte Díaz, “Uncovering cybercrimes in social media through natural language processing”, Security and Communication Networks, vol. 2021, 12 2021, DOI [10.1155/2021/7955637](https://doi.org/10.1155/2021/7955637)
- [117] S. Kombrink, T. Mikolov, M. Karafiát, and L. Burget, “Recurrent neural network based language modeling in meeting recognition”, Proceedings of Interspeech, 2011, pp. 2877–2880, DOI [10.21437/Interspeech.2011-720](https://doi.org/10.21437/Interspeech.2011-720)
- [118] J. Gao and C.-Y. Lin, “Introduction to the special issue on statistical language modeling”, ACM Transactions on Asian Language Information Processing, vol. 3, no. 2, 2004, pp. 87–93, DOI [10.1145/1034780.1034781](https://doi.org/10.1145/1034780.1034781)
- [119] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding”, 2018. URL: <https://arxiv.org/abs/1810.04805>
- [120] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang, W. Ye, Y. Zhang, Y. Chang, P. S. Yu, Q. Yang, and X. Xie, “A survey on evaluation of large language models”, ACM Trans. Intell. Syst. Technol., vol. 15, 02 2024, DOI [10.1145/3641289](https://doi.org/10.1145/3641289)
- [121] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, “Deep learning-based text classification: a comprehensive review”, ACM Computing Surveys (CSUR), vol. 54, no. 3, 2021, pp. 1–40, DOI [10.1145/3439726](https://doi.org/10.1145/3439726)
- [122] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang, “Pre-trained models for natural language processing: A survey”, Science China Technological Sciences, vol. 63, no. 10, 2020, pp. 1872–1897, DOI [10.1007/s11431-020-1647-3](https://doi.org/10.1007/s11431-020-1647-3)
- [123] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. de Oliveira Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, *et al.*, “Evaluating large language models trained on code”, 2021. URL: <https://arxiv.org/abs/2107.03374>
- [124] E. Kasneci, K. Seßler, S. Küchemann, M. Bannert, D. Dementieva, F. Fischer, U. Gasser, G. Groh, S. Günnemann, and E. Hüllermeier, “Chatgpt for good? on opportunities and challenges of large language models for education”, Learning and Individual Differences, vol. 103, 2023, p. 102274, DOI [10.1016/j.lindif.2023.102274](https://doi.org/10.1016/j.lindif.2023.102274)
- [125] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, *et al.*, “A survey of large language models”, 2023, DOI [10.48550/arXiv.2303.18223](https://doi.org/10.48550/arXiv.2303.18223)
- [126] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. V. Le, and D. Zhou, “Chain-of-thought prompting elicits reasoning in large language models”, Advances in Neural Information Processing Systems, 2022, pp. 24824–24837, DOI [10.5555/3589789.3589790](https://doi.org/10.5555/3589789.3589790)
- [127] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and

- P. Fung, “Survey of hallucination in natural language generation”, *ACM Computing Surveys*, vol. 55, 03 2023, DOI [10.1145/3571730](https://doi.org/10.1145/3571730)
- [128] S. Yao, D. Yu, J. Zhao, I. Shafran, T. L. Griffiths, Y. Cao, and K. Narasimhan, “Tree of thoughts: Deliberate problem solving with large language models”, 2023. URL: <https://arxiv.org/abs/2305.10601>
 - [129] P. Manakul, A. Liusie, and M. J. F. Gales, “Selfcheckgpt: Zero resource black-box hallucination detection for generative large language models”, 2023
 - [130] N. Shinn, F. Cassano, E. Berman, A. Gopinath, K. Narasimhan, and S. Yao, “Reflexion: Language agents with verbal reinforcement learning”, 2023. URL: <https://arxiv.org/abs/2303.11366>
 - [131] S. J. Zhang, S. Florin, A. N. Lee, E. Niknafs, A. Marginean, A. Wang, K. Tyser, Z. Chin, Y. Hicke, N. Singh, M. Udell, Y. Kim, T. Buonassisi, A. Solar-Lezama, and I. Drori, “Exploring the mit mathematics and eecs curriculum using large language models”, 2023. URL: <https://arxiv.org/abs/2306.08997>
 - [132] Y. Zhou, A. I. Muresanu, Z. Han, K. Paster, S. Pitis, H. Chan, and J. Ba, “Large language models are human-level prompt engineers”, 2023. URL: <https://arxiv.org/abs/2211.01910>
 - [133] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, “Language models are few-shot learners”, 2020. URL: <https://arxiv.org/abs/2005.14165>
 - [134] Z. Ke, Y. Shao, H. Lin, T. Konishi, G. Kim, and B. Liu, “Continual pre-training of language models”, 2023. URL: <https://arxiv.org/abs/2302.03241>
 - [135] P. S. H. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, “Retrieval-augmented generation for knowledge-intensive nlp tasks”, 2020. URL: <https://arxiv.org/abs/2005.11401>
 - [136] Y. Gao, Y. Xiong, X. Gao, *et al.*, “Retrieval-augmented generation for large language models: A survey”, 2023. URL: <https://arxiv.org/abs/2312.10997>
 - [137] B. Jin, G. Liu, C. Han, M. Jiang, H. Ji, and J. Han, “Large language models on graphs: A comprehensive survey”, 2024. URL: <https://arxiv.org/abs/2312.02783>
 - [138] A. Hogan, E. Blomqvist, M. Cochez, C. D’amato, G. D. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, A.-C. N. Ngomo, A. Polleres, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, and A. Zimmermann, “Knowledge graphs”, *ACM Computing Surveys*, vol. 54, no. 4, 2021, pp. 1–37, DOI [10.1145/3447772](https://doi.org/10.1145/3447772)
 - [139] T. O. Browne, M. Abedin, and M. J. M. Chowdhury, “A systematic review on research utilising artificial intelligence for open source intelligence (osint) applications”, *International Journal of Information Security*, vol. 23, 06 2024, pp. 2911–2938, DOI [10.1007/s10207-024-00868-2](https://doi.org/10.1007/s10207-024-00868-2)
 - [140] Z. Obermeyer, B. Powers, C. Vogeli, and S. Mullainathan, “Dissecting racial bias in an algorithm used to manage the health of populations”, *Science*, vol. 366, no. 6464,

- 2019, pp. 447–453, DOI [10.1126/science.aax2342](https://doi.org/10.1126/science.aax2342)
- [141] W. Zhang and A. Webb, “Quality control of crowdsourced data: A signal processing approach”, *IEEE Transactions on Signal Processing*, vol. 67, no. 18, 2019, pp. 4670–4685, DOI [10.1109/TSP.2019.2932435](https://doi.org/10.1109/TSP.2019.2932435)
 - [142] L. A. Adamic and N. Glance, “The political blogosphere and the 2004 us election: Divided they blog”, *Proceedings of the 3rd International Workshop on Link Discovery*, Chicago, Illinois, USA, 08 2005, pp. 36–43, DOI [10.1145/1134271.1134277](https://doi.org/10.1145/1134271.1134277)
 - [143] L. Ciolfi, M. Lewkowicz, and K. Schmidt, “Computer-supported cooperative work”, *Handbook of Human Computer Interaction* (J. Vanderdonckt, P. Palanque, and M. Winckler, eds.), Springer, 2023, DOI [10.1007/978-3-319-27648-9_30-1](https://doi.org/10.1007/978-3-319-27648-9_30-1)
 - [144] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli, “Evasion attacks against machine learning at test time”, *Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2013* (H. Blockeel, K. Kersting, S. Nijssen, and F. Železný, eds.), p. 387–402, Springer, 2013, DOI [10.1007/978-3-642-40994-3_25](https://doi.org/10.1007/978-3-642-40994-3_25)
 - [145] M. Ayub, S. Irum, and Z. Jalil, “Enhanced audio-based open-source intelligence insights using machine learning”, *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 10, 02 2024, DOI [10.32628/CSEIT2410118](https://doi.org/10.32628/CSEIT2410118)
 - [146] W. Yang and K. Y. Lam, “Automated cyber threat intelligence reports classification for early warning of cyber attacks in next generation soc”, *Information and Communications Security: 21st International Conference, ICICS 2019*, Beijing, China, 2020, pp. 145–164, DOI [10.1007/978-3-030-41579-2_9](https://doi.org/10.1007/978-3-030-41579-2_9)
 - [147] B. Upadhayay, Z. Lodhia, and V. Behzadan, “Combating human trafficking via automatic osint collection, validation and fusion”, *ICWSM Workshops*, 2021, DOI [10.36190/2021.17](https://doi.org/10.36190/2021.17)
 - [148] G. Shalunts, G. Backfried, and K. Prinz, “Sentiment analysis of german social media data for natural disasters”, *Proceedings of the 11th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2014)*, 2014, pp. 752–756. URL: http://idl.iscram.org/files/shalunts/2014/940_Shalunts_etal2014.pdf
 - [149] E. M. Badr, M. A. Salam, M. Ali, and H. Ahmed, “Social media sentiment analysis using machine learning and optimization techniques”, *International Journal of Computer Applications*, vol. 178, 08 2019, pp. 31–36, DOI [10.5120/ijca2019919306](https://doi.org/10.5120/ijca2019919306)
 - [150] P. An, R. Shafi, T. Mughogho, and O. A. Onyango, “Multilingual email phishing attacks detection using osint and machine learning”, 2025. URL: <https://arxiv.org/abs/2501.08723>
 - [151] S. K, P. Balakrishna, V. R, and S. KP, “Deep learning approach for enhanced cyber threat indicators in twitter stream”, 2020. URL: <https://arxiv.org/abs/2004.00503>
 - [152] S. Jain, V. Sharma, and R. Kaushal, “Towards automated real-time detection of misinformation on twitter”, *Advances in Computing and Communications (ICACCI)*, 2016, pp. 2015–2020, DOI [10.1109/ICACCI.2016.7732347](https://doi.org/10.1109/ICACCI.2016.7732347)
 - [153] C. Sotirakou, A. Karampela, and C. Mourlas, “Evaluating the role of news content and social media interactions for fake news detection”, *Disinformation in Open*

- Online Media, 2021, pp. 128–141, DOI [10.1007/978-3-030-87031-7_9](https://doi.org/10.1007/978-3-030-87031-7_9)
- [154] S. R. Vadapalli, G. Hsieh, and K. S. Nauer, “Twitterosint: Automated cybersecurity threat intelligence collection and analysis using twitter data”, Proceedings of the World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), Athens, 2018. URL: <https://www.proquest.com/conference-papers-proceedings/twitterosint-automated-cybersecurity-threat/docview/2153621548/se-2>
 - [155] T. Riebe, T. Wirth, M. Bayer, P. Kühn, M.-A. Kaufhold, V. Knauthe, S. Guthe, and C. Reuter, “Cysecalert: An alert generation system for cyber security events using open source intelligence data”, Information and Communications Security, 2021, p. 429–446, DOI [10.1007/978-3-030-86890-1_24](https://doi.org/10.1007/978-3-030-86890-1_24)
 - [156] D. Parashar, L. M. Sanagavarapu, and Y. R. Reddy, “Sql injection vulnerability identification from text”, Proceedings of the 14th Innovations in Software Engineering Conference (Formerly Known as India Software Engineering Conference), New York, NY, USA, 2021, pp. 1–5, DOI [10.1145/3452383.3452405](https://doi.org/10.1145/3452383.3452405)
 - [157] Y. Fang, Y. Liu, C. Huang, and L. Liu, “Fastembed: Predicting vulnerability exploitation possibility based on ensemble machine learning algorithm”, PLoS ONE, vol. 15, 02 2020, DOI [10.1371/journal.pone.0228439](https://doi.org/10.1371/journal.pone.0228439)
 - [158] M. A. Hoppa, S. M. Debb, G. Hsieh, and B. KC, “Twitterosint: Automated open source intelligence collection, analysis & visualization tool”, Annual Review of CyberTherapy and Telemedicine, 2019, pp. 121–128, ISBN: 1554-8716
 - [159] C. GAVRILĂ, “Ethical and legal challenges of artificial intelligence in intelligence: Between operational efficiency and respect for fundamental rights.”, Research & Science Today, vol. 29, 03 2025, pp. 63–80, DOI [10.1007/978-3-031-80411-3_6](https://doi.org/10.1007/978-3-031-80411-3_6)
 - [160] V. Bijalwan, V. Kumar, P. Kumari, and J. Pascual, “Knn based machine learning approach for text and document mining”, International Journal of Database Theory and Application, vol. 7, 06 2014, pp. 61–70, DOI [10.14257/ijdta.2014.7.1.06](https://doi.org/10.14257/ijdta.2014.7.1.06)
 - [161] H. Ahmed, I. Traore, and S. Saad, “Detection of online fake news using n-gram analysis and machine learning techniques”, Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments (ISDDC 2017), 2017, pp. 127–138, DOI [10.1007/978-3-319-69155-8_9](https://doi.org/10.1007/978-3-319-69155-8_9)
 - [162] S. Goel, N. Sachdeva, P. Kumaraguru, A. V. Subramanyam, and D. Gupta, “Pichunt: Social media image retrieval for improved law enforcement”, Social Informatics (SocInfo 2016), 2016, DOI [10.1007/978-3-319-47880-7_13](https://doi.org/10.1007/978-3-319-47880-7_13)
 - [163] C. Choudhary, R. Sivaguru, M. Pereira, B. Yu, A. C. Nascimento, and M. De Cock, “Algorithmically generated domain detection and malware family classification”, Security in Computing and Communications, Singapore, 2019, pp. 640–655, ISBN: 978-981-13-5826-5
 - [164] R. Vinayakumar, K. P. Soman, and P. Poornachandran, “Detecting malicious domain names using deep learning approaches at scale”, Journal of Intelligent & Fuzzy Systems, vol. 34, no. 3, 2018, pp. 1355–1367, DOI [10.3233/JIFS-169431](https://doi.org/10.3233/JIFS-169431)
 - [165] H. Suryotrisongko, Y. Musashi, A. Tsuneda, and K. Sugitani, “Robust botnet dga detection: Blending xai and osint for cyber threat intelligence sharing”, IEEE Access, vol. 10, 2022, pp. 34613–34624, DOI [10.1109/ACCESS.2022.3162588](https://doi.org/10.1109/ACCESS.2022.3162588)

- [166] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma, “Deep learning for hate speech detection in tweets”, Proceedings of the 26th International Conference on World Wide Web Companion (WWW ’17 Companion), New York, NY, USA, 2017, pp. 759–760, DOI [10.1145/3041021.3054223](https://doi.org/10.1145/3041021.3054223)
- [167] A. Alamsyah, W. Rizkika, D. D. A. Nugroho, F. Renaldi, and S. Saadah, “Dynamic large scale data on twitter using sentiment analysis and topic modeling”, 2018 6th International Conference on Information and Communication Technology (ICoICT), 2018, pp. 254–258, DOI [10.1109/ICoICT.2018.8528776](https://doi.org/10.1109/ICoICT.2018.8528776)
- [168] A. Hidri, S. A. Alsaif, M. Alahmari, E. AlShehri, and M. Sassi Hidri, “Opinion mining and analysis using hybrid deep neural networks”, Technologies, vol. 13, 04 2025, DOI [10.3390/technologies13050175](https://doi.org/10.3390/technologies13050175)
- [169] M. Smadi and O. Qawasmeh, “A supervised machine learning approach for events extraction out of arabic tweets”, 2018 Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS), 2018, pp. 114–119, DOI [10.1109/SNAMS.2018.8554560](https://doi.org/10.1109/SNAMS.2018.8554560)
- [170] P. Radanliev, D. De Roure, C. Maple, *et al.*, “Methodology for integrating artificial intelligence in healthcare systems: learning from covid-19 to prepare for disease x”, AI Ethics, vol. 2, 10 2022, pp. 623–630, DOI [10.1007/s43681-021-00111-x](https://doi.org/10.1007/s43681-021-00111-x)
- [171] D. Honeyman, D. Gurdasani, A. Notaras, Z. Akhtar, J. Edgeworth, A. Moa, A. A. Chughtai, A. Quigley, S. Lim, and C. R. MacIntyre, “Global epidemiology of outbreaks of unknown cause identified by open-source intelligence, 2020–2022”, Emerging Infectious Diseases, vol. 31, 02 2025, pp. 298–308, DOI [10.3201/eid3102.240533](https://doi.org/10.3201/eid3102.240533)
- [172] N. Ekwunife, “National security intelligence through social network data mining”, 2020 IEEE International Conference on Big Data (Big Data), 2020, pp. 2270–2273, DOI [10.1109/BigData50022.2020.9377940](https://doi.org/10.1109/BigData50022.2020.9377940)
- [173] T. Mackey, J. Kalyanam, J. Klugman, E. Kuzmenko, and R. Gupta, “Solution to detect, classify, and report illicit online marketing and sales of controlled substances via twitter: Using machine learning and web forensics to combat digital opioid access”, J Med Internet Res, vol. 20, 04 2018, DOI [10.2196/10029](https://doi.org/10.2196/10029)
- [174] A. López-Martínez, J. A. García-Díaz, R. Valencia-García, and A. Ruiz-Martínez, “Cyberdetect: A novel approach for cyberbullying detection on twitter”, Technologies and Innovation, 2019, pp. 109–121, DOI [10.1007/978-3-030-34989-9_9](https://doi.org/10.1007/978-3-030-34989-9_9)
- [175] M. S. Biagio, R. Acquaviva, V. Mazzonello, E. La Mattina, and V. Morreale, “A new SOCMINT framework for Threat Intelligence Identification”, 2021 International Conference on Computational Science and Computational Intelligence (CSCI), Los Alamitos, CA, USA, 2021, pp. 692–697, DOI [10.1109/CSCI54926.2021.00180](https://doi.org/10.1109/CSCI54926.2021.00180)
- [176] S. Sadiq, A. Mehmood, S. Ullah, M. Ahmad, G. S. Choi, and B.-W. On, “Aggression detection through deep neural model on twitter”, Future Generation Computer Systems, vol. 114, 2021, pp. 120–129, DOI <https://doi.org/10.1016/j.future.2020.07.050>
- [177] J. Lu, H. Lin, X. Zhang, Z. Li, T. Zhang, L. Zong, F. Ma, and B. Xu, “Hate speech detection via dual contrastive learning”, 2023. URL: <https://arxiv.org/abs/2307.05578>

-
- [178] H. Saleh, A. Alhothali, and K. Moria, “Detection of hate speech using bert and hate speech word embedding with deep model”, *Applied Artificial Intelligence*, vol. 37, no. 1, 2023, p. 2166719, DOI [10.1080/08839514.2023.2166719](https://doi.org/10.1080/08839514.2023.2166719)
- [179] J. Salminen, M. Hopf, S. A. Chowdhury, S.-g. Jung, H. Almerexhi, and B. J. Jansen, “Developing an online hate classifier for multiple social media platforms”, *Human-centric Computing and Information Sciences*, vol. 10, 2020, pp. 1–34, DOI [10.1186/s13673-020-00230-7](https://doi.org/10.1186/s13673-020-00230-7)
- [180] N. L. V. Venugopal, P. Kanchanamala, S. Muppidi, T. B. Prakash, T. Neelima, and S. A. Devi, “Multilingual toxic comment classification using deep learning”, 2024 2nd International Conference on Self Sustainable Artificial Intelligence Systems (IC-SSAS), Erode, India, 2024, pp. 752–757, DOI [10.1109/ICSSAS64001.2024.10760913](https://doi.org/10.1109/ICSSAS64001.2024.10760913)
- [181] S. Alsafari, S. Sadaoui, and M. Mouhoub, “Hate and offensive speech detection on arabic social media”, *Online Social Networks and Media*, vol. 19, 2020, p. 100096, DOI [10.1016/j.osnem.2020.100096](https://doi.org/10.1016/j.osnem.2020.100096)
- [182] Z. Mossie and J.-H. Wang, “Vulnerable community identification using hate speech detection on social media”, *Information Processing & Management*, vol. 57, 2020, p. 102087, DOI [10.1016/j.ipm.2020.102087](https://doi.org/10.1016/j.ipm.2020.102087)
- [183] S. Banerjee, B. R. Chakravarthi, and J. McCrae, “Comparison of pretrained embeddings to identify hate speech in indian code mixed text”, *Proceedings of the 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, Greater Noida, India, 2020, pp. 21–25
- [184] K. Sreelakshmi, B. Premjith, and K. Soman, “Detection of hate speech text in hindi-english code-mixed data”, *Procedia Computer Science*, vol. 171, 2020, pp. 737–744, DOI [10.1016/j.procs.2020.04.079](https://doi.org/10.1016/j.procs.2020.04.079)
- [185] S. Modha, P. Majumder, T. Mandl, and C. Mandalia, “Detecting and visualizing hate speech in social media: A cyber watchdog for surveillance”, *Expert Systems with Applications*, vol. 161, 2020, p. 113725, DOI [10.1016/j.eswa.2020.113725](https://doi.org/10.1016/j.eswa.2020.113725)
- [186] A. Das, S. Nandy, R. Saha, S. Das, and D. Saha, “Analysis and detection of multilingual hate speech using transformer based deep learning”, 2024. URL: <https://arxiv.org/abs/2401.11021>
- [187] P. Kapil and A. Ekbal, “A deep neural network based multi-task learning approach to hate speech detection”, *Knowledge-Based Systems*, vol. 210, 2020, p. 106458, DOI [10.1016/j.knosys.2020.106458](https://doi.org/10.1016/j.knosys.2020.106458)
- [188] D. Nandini and U. Schmid, “Explaining hate speech classification with model agnostic methods”, 2023. URL: <https://arxiv.org/abs/2306.00021>
- [189] M. F. López-Vizcaíno, F. J. Nóvoa, V. Carneiro, and F. Cacheda, “Early detection of cyberbullying on social media networks”, *Future Generation Computer Systems*, vol. 118, 2021, pp. 219–229, DOI [10.1016/j.future.2021.01.006](https://doi.org/10.1016/j.future.2021.01.006)
- [190] A. Kumar and N. Sachdeva, “Multimodal cyberbullying detection using capsule network with dynamic routing and deep convolutional neural network”, *Multimedia Systems*, vol. 28, 2022, pp. 2043–2052, DOI [10.1007/s00530-020-00747-5](https://doi.org/10.1007/s00530-020-00747-5)
- [191] I. Tabassum and V. Nunavath, “A hybrid deep learning approach for multi-class cyberbullying classification using multi-modal social media data”, *Applied Sciences*, vol. 14, no. 24, 2024, DOI [10.3390/app142412007](https://doi.org/10.3390/app142412007)

- [192] H. Herodotou, D. Chatzakou, and N. Kourtellis, “A streaming machine learning framework for online aggression detection on twitter”, Proceedings of the 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, 12 2020, pp. 5056–5067, DOI [10.48550/arXiv.2006.10104](https://doi.org/10.48550/arXiv.2006.10104)
- [193] A. M. Schoene, A. Turner, G. R. de Mel, and N. Dethlefs, “Hierarchical multiscale recurrent neural networks for detecting suicide notes”, IEEE Transactions on Affective Computing, vol. 10, no. 1, 2021, pp. 1–2, DOI [10.1109/TAFFC.2021.3096422](https://doi.org/10.1109/TAFFC.2021.3096422)
- [194] M. Stankevich, I. Smirnov, N. Kiselnikova, and A. Ushakova, “Depression detection from social media profiles”, Proceedings of the International Conference on Data Analytics and Management in Data Intensive Domains (DAMDID), Voronezh, Russia, 2019, pp. 181–194, DOI [10.1007/978-3-030-51913-1_12](https://doi.org/10.1007/978-3-030-51913-1_12)
- [195] A. T. Nurlanuly, “Sentiment analysis of texts from social networks based on machine learning methods for monitoring public sentiment”, 2025. URL: <https://arxiv.org/abs/2502.17143>
- [196] R. Sharma, N. Le Tan, and F. Sadat, “Multimodal sentiment analysis using deep learning”, 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), 2018, pp. 1475–1478, DOI [10.1109/ICMLA.2018.00240](https://doi.org/10.1109/ICMLA.2018.00240)
- [197] A. Rizvi, N. Thamindu, A. M. N. H. Adhikari, W. P. U. Senevirathna, D. Kasthuri-rathna, and L. Abeywardhana, “Enhancing multilingual sentiment analysis with explainability for sinhala, english, and code-mixed content”, 2025. URL: <https://arxiv.org/abs/2504.13545>
- [198] J. Wang, B. Xu, and Y. Zu, “Deep learning for aspect-based sentiment analysis”, 2021 International Conference on Machine Learning and Intelligent Systems Engineering (MLISE), 2021, pp. 267–271, DOI [10.1109/MLISE54096.2021.00056](https://doi.org/10.1109/MLISE54096.2021.00056)
- [199] P. K. Roy, A. K. Tripathy, T.-H. Weng, and K.-C. Li, “Securing social platform from misinformation using deep learning”, Computer Standards and Interfaces, vol. 84, 2023, DOI [10.1016/j.csi.2022.103674](https://doi.org/10.1016/j.csi.2022.103674)
- [200] R. Kaliyar, A. Goswami, and P. Narang, “Fakebert: Fake news detection in social media with a bert-based deep learning approach”, Multimedia Tools and Applications, vol. 80, 01 2021, pp. 11765–11788, DOI [10.1007/s11042-020-10183-2](https://doi.org/10.1007/s11042-020-10183-2)
- [201] E. Puraivan, F. Riquelme, and R. Venegas, “A linguistic features-based approach for the functional analysis of disinformation in spanish”, IEEE Access, vol. 13, 2025, pp. 140205–140222, DOI [10.1109/ACCESS.2025.3595750](https://doi.org/10.1109/ACCESS.2025.3595750)
- [202] M. A. Mohamed, S. D. Ahmed, Y. A. Isse, H. M. Mohamed, F. M. Hassan, and H. A. Assowe, “Detection of somali-written fake news and toxic messages on the social media using transformer-based language models”, 2025. URL: <https://arxiv.org/abs/2503.18117>
- [203] M. Xu, F. Li, Z. Miao, Z. Han, L. Wang, and G. Wang, “Detecting fake news on social media via multimodal semantic understanding and enhanced transformer architectures”, Traitement du Signal, vol. 42, 06 2025, pp. 1553–1564, DOI [10.18280/ts.420327](https://doi.org/10.18280/ts.420327)
- [204] X. Shen, M. Huang, Z. Hu, S. Cai, and T. Zhou, “Multimodal fake news detection with contrastive learning and optimal transport”, Frontiers in Computer Science, vol. 6, 11 2024, pp. 1–10, DOI [10.3389/fcomp.2024.1473457](https://doi.org/10.3389/fcomp.2024.1473457)

- [205] A. Malik, D. K. Behera, J. Hota, and A. R. Swain, “Ensemble graph neural networks for fake news detection using user engagement and text features”, *Results in Engineering*, vol. 24, 2024, DOI [10.1016/j.rineng.2024.103081](https://doi.org/10.1016/j.rineng.2024.103081)
- [206] M. Ashcroft, A. Fisher, L. Kaati, E. Omer, and N. Prucha, “Detecting jihadist messages on twitter”, *EISIC '15: Proceedings of the 2015 European Intelligence and Security Informatics Conference (EISIC)*, USA, 2015, pp. 161–164, DOI [10.1109/EISIC.2015.27](https://doi.org/10.1109/EISIC.2015.27)
- [207] S. Agarwal and A. Sureka, “Using knn and svm based one-class classifier for detecting online radicalization on twitter”, *Distributed Computing and Internet Technology*, 2015, pp. 431–442, DOI [10.1007/978-3-319-14977-6_47](https://doi.org/10.1007/978-3-319-14977-6_47)
- [208] M. S. Biagio, S. Simoncini, E. La Mattina, and V. Morreale, “Marple: A framework for social media threat intelligence”, *2024 International Conference on Artificial Intelligence, Computer, Data Sciences and Applications (ACDSA)*, 2024, pp. 1–6, DOI [10.1109/ACDSA59508.2024.10467738](https://doi.org/10.1109/ACDSA59508.2024.10467738)
- [209] M. Nouh, J. R. Nurse, H. Webb, and M. Goldsmith, “Cybercrime investigators are users too! understanding the socio-technical challenges faced by law enforcement”, *Proceedings 2019 Workshop on Usable Security*, 2019, DOI [10.14722/usec.2019.23032](https://doi.org/10.14722/usec.2019.23032)
- [210] O. Araque and C. A. Iglesias, “An approach for radicalization detection based on emotion signals and semantic similarity”, *IEEE Access*, vol. 8, 2020, pp. 17877–17891, DOI [10.1109/ACCESS.2020.2967219](https://doi.org/10.1109/ACCESS.2020.2967219)
- [211] O. Elezaj, S. Y. Yayilgan, E. Kalemi, L. Wendelberg, M. Abomhara, and J. Ahmed, “Towards designing a knowledge graph-based framework for investigating and preventing crime on online social networks”, *E-Democracy – Safeguarding Democracy and Human Rights in the Digital Age*, 2020, pp. 181–195, DOI [10.1007/978-3-030-37545-4_12](https://doi.org/10.1007/978-3-030-37545-4_12)
- [212] K. Zerrouki, N. Benblidia, and O. Boussaid, “Preprocessing multilingual text for the detection of extremism and radicalization in social networks using deep learning”, *Studies in Engineering and Exact Sciences*, vol. 5, no. 2, 2024, DOI [10.54021/seesv5n2-594](https://doi.org/10.54021/seesv5n2-594)
- [213] E. M. Al-Shawakfa, A. M. R. Alsobeh, S. Omari, and A. Shatnawi, “Radar: An ensemble approach for radicalization detection in arabic social media using hybrid deep learning and transformer models”, *Information*, vol. 16, no. 7, 2025, DOI [10.3390/info16070522](https://doi.org/10.3390/info16070522)
- [214] N. Dionísio, F. Alves, P. M. Ferreira, and A. Bessani, “Cyberthreat detection from twitter using deep neural networks”, *2019 International Joint Conference on Neural Networks (IJCNN)*, 2019, pp. 1–8, DOI [10.1109/IJCNN.2019.8852475](https://doi.org/10.1109/IJCNN.2019.8852475)
- [215] V. Behzadan, C. Aguirre, A. Bose, and W. Hsu, “Corpus and deep learning classifier for collection of cyber threat indicators in twitter stream”, *2018 IEEE International Conference on Big Data (Big Data)*, 2018, pp. 5002–5007, DOI [10.1109/Big-Data.2018.8622506](https://doi.org/10.1109/Big-Data.2018.8622506)
- [216] F. Su, “A new social media-driven cyber threat intelligence”, *Electronics*, vol. 12, no. 5, 2023, DOI [10.3390/electronics12051242](https://doi.org/10.3390/electronics12051242)
- [217] A. S. Gautam, Y. Gahlot, and P. Kamat, “Hacker forum exploit and classification

- for proactive cyber threat intelligence”, Inventive Computation Technologies, 2020, pp. 279–285, DOI [10.1007/978-3-030-33846-6_32](https://doi.org/10.1007/978-3-030-33846-6_32)
- [218] K. Sangher, A. Singh, and H. Pandey, “Lstm and bert based transformers models for cyber threat intelligence for intent identification of social media platforms exploitation from darknet forums”, *Int. J. Inf. Technol.*, vol. 16, 2024, pp. 5277–5292, DOI [10.1007/s41870-024-02077-5](https://doi.org/10.1007/s41870-024-02077-5)
- [219] R. Geetha, S. Karthika, and P. Kumaraguru, “Tweet-scan-post: a system for analysis of sensitive private data disclosure in online social media”, *Knowledge and Information Systems*, vol. 63, 07 2021, pp. 2365–2404, DOI [10.1007/s10115-021-01592-2](https://doi.org/10.1007/s10115-021-01592-2)
- [220] R. Potla, “Ai-powered threat detection in online communities: A multi-modal deep learning approach”, *Journal of Computer and Communications*, vol. 13, 2025, pp. 155–171, DOI [10.4236/jcc.2025.132010](https://doi.org/10.4236/jcc.2025.132010)
- [221] S. Shi, K. Qiao, Z. Wang, J. Yang, B. Song, J. Chen, and B. Yan, “Muti-scale graph neural network with signed-attention for social bot detection: A frequency perspective”, 2023. URL: <https://arxiv.org/abs/2307.01968>
- [222] Y. Aun, M.-L. Gan, N. H. B. Abdul Wahab, and G. H. Guan, “Social engineering attack classifications on social media using deep learning”, *Computers, Materials and Continua*, vol. 74, no. 3, 2022, pp. 4917–4931, DOI <https://doi.org/10.32604/cmc.2023.032373>
- [223] N. Innab, A. A. F. Osman, M. A. M. Ataelfadiel, M. Abu-Zanona, B. M. Elzaghmouri, F. H. Zawaideh, and M. F. Alawneh, “Phishing attacks detection using ensemble machine learning algorithms”, *Computers, Materials and Continua*, vol. 80, no. 1, 2024, pp. 1325–1345, DOI <https://doi.org/10.32604/cmc.2024.051778>
- [224] M. Songailaitė, E. Kankevičiūtė, B. Zhyhun, and J. Mandravickaitė, “Bert-based models for phishing detection”, 2023. URL: <https://eur-ws.org/Vol-3575/Paper4.pdf#:~:text=In%20this%20paper%20we%20report%20the%20application%20of,for%20each%20metric%20%28accuracy%2C%20precision%2C%20recall%20and%20F1-score%29.>
- [225] S. Chatterjee and S. Basu, “How vulnerable are you? a novel computational psycholinguistic analysis for phishing influence detection”, *Proceedings of the 18th International Conference on Natural Language Processing*, 2021, pp. 499–507. URL: <https://aclanthology.org/2021.icon-main.61/>
- [226] N. Alharbi, B. Alkalifah, G. Alqarawi, and M. A. Rassam, “Countering social media cybercrime using deep learning: Instagram fake accounts detection”, *Future Internet*, vol. 16, no. 10, 2024, DOI [10.3390/fi16100367](https://doi.org/10.3390/fi16100367)
- [227] G. Rajendran, A. Arun Kumar, P. K. Sridhar, K. K. Perumalsamy, and N. Srinivasan, “A comprehensive approach for enhancing osint through leveraging llms”, *International Refereed Journal of Engineering and Science (IRJES)*, vol. 13, 03 2024, pp. 61–66. URL: <https://mail.irjes.com/Papers/vol13-issue2/H13026166.pdf>
- [228] T. Giannilias, A. Papadakis, N. Nikolaou, and T. Zahariadis, “Classification of hacker’s posts based on zero-shot, few-shot, and fine-tuned llms in environments with constrained resources”, *Future Internet*, vol. 17, no. 5, 2025, DOI [10.3390/fi17050207](https://doi.org/10.3390/fi17050207)

-
- [229] S. Shafee, A. Bessani, and P. M. Ferreira, “Evaluation of llm-based chatbots for osint-based cyber threat awareness”, *Expert Systems with Applications*, vol. 261, 2025, DOI <https://doi.org/10.1016/j.eswa.2024.125509>
 - [230] H. Peters and S. C. Matz, “Large language models can infer psychological dispositions of social media users”, *PNAS Nexus*, vol. 3, 06 2024, DOI [10.1093/pnas-nexus/pgae231](https://doi.org/10.1093/pnas-nexus/pgae231)
 - [231] F. Barbieri, L. E. Anke, and J. Camacho-Collados, “Xlm-t: Multilingual language models in twitter for sentiment analysis and beyond”, 2022. URL: <https://arxiv.org/abs/2104.12250>
 - [232] K. Kasianenko, S. Khanehzar, S. Wan, E. Dehghan, and A. Bruns, “Detecting online community practices with large language models: A case study of pro-ukrainian publics on twitter”, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Miami, Florida, USA, 2024, pp. 20106–20135, DOI [10.18653/v1/2024.emnlp-main.1122](https://doi.org/10.18653/v1/2024.emnlp-main.1122)
 - [233] J. Jose and R. Greenstadt, “Llms for detection and classification of persuasion techniques in slavic parliamentary debates and social media texts”, *Proceedings of the 10th Workshop on Slavic Natural Language Processing (Slavic NLP 2025)*, Vienna, Austria, 07 2025, pp. 202–216, DOI [10.18653/v1/2025.bsnlp-1.23](https://doi.org/10.18653/v1/2025.bsnlp-1.23)
 - [234] A. Sahitaj, P. Sahitaj, V. Solopova, J. Li, S. Möller, and V. Schmitt, “Hybrid annotation for propaganda detection: Integrating llm pre-annotations with human intelligence”, 2025. URL: <https://arxiv.org/abs/2507.18343>
 - [235] J. Wang, Z. Zhu, C. Liu, R. Li, and X. Wu, “Llm-enhanced multimodal detection of fake news”, *PLoS One*, vol. 19, 10 2024, DOI [10.1371/journal.pone.0312240](https://doi.org/10.1371/journal.pone.0312240)
 - [236] P. Bolegave and P. Bhattacharya, “A gold standard dataset and evaluation framework for depression detection and explanation in social media using llms”, 2025. URL: <https://arxiv.org/abs/2507.19899>
 - [237] R. Soler, M. Dawson, and M. Colina, “Automated target profiling: Leveraging artificial intelligence for open-source intelligence collection”, *International Conference KNOWLEDGE-BASED ORGANIZATION*, Illinois Institute of Technology, Chicago, USA, 2025, pp. 184–190, DOI [10.2478/kbo-2025-0023](https://doi.org/10.2478/kbo-2025-0023)
 - [238] A. Radwan, M. Amarneh, H. Alawneh, H. I. Ashqar, A. AlSobeh, and A. A. A. R. Magableh, “Predictive analytics in mental health leveraging llm embeddings and machine learning models for social media analysis”, *International Journal of Web Services Research (IJWSR)*, vol. 21, no. 1, 2024, pp. 1–22, DOI [10.4018/IJWSR.338222](https://doi.org/10.4018/IJWSR.338222)
 - [239] J. Maharjan, R. Jin, J. Zhu, and D. Kenne, “Psychometric evaluation of large language model embeddings for personality trait prediction”, *J Med Internet Res*, vol. 27, 07 2025, DOI [10.2196/75347](https://doi.org/10.2196/75347)
 - [240] J. Jiang and E. Ferrara, “Social-llm: Modeling user behavior at scale using language models and social network data”, 2023. URL: <https://arxiv.org/abs/2401.00893>
 - [241] S. Benabderrahmane, P. Valtchev, J. Cheney, and T. Rahwan, “Apt-llm: Embedding-based anomaly detection of cyber advanced persistent threats using large language models”, 2025 13th International Symposium on Digital Forensics

- and Security (ISDFS), Boston, United States, 2025, pp. 1–6, DOI [10.1109/ISDFS65363.2025.11011912](https://doi.org/10.1109/ISDFS65363.2025.11011912)
- [242] N. Farruque, R. Goebel, S. Sivapalan, *et al.*, “Depression symptoms modelling from social media text: an llm driven semi-supervised learning approach”, *Language Resources and Evaluation*, vol. 58, 2024, pp. 1013–1041, DOI [10.1007/s10579-024-09720-4](https://doi.org/10.1007/s10579-024-09720-4)
- [243] V. G. dos Santos, G. L. Santos, T. Lynn, and B. Benatallah, “Identifying citizen-related issues from social media using llm-based data augmentation”, *Advanced Information Systems Engineering*, 2024, pp. 531–546, DOI [10.1007/978-3-031-61057-8_31](https://doi.org/10.1007/978-3-031-61057-8_31)
- [244] S.-H. Cho, D. Kim, H.-C. Kwon, and M. Kim, “Exploring the potential of large language models for author profiling tasks in digital text forensics”, *Forensic Science International: Digital Investigation*, vol. 50, 10 2024, pp. 1–9, DOI [10.1016/j.fsidi.2024.301814](https://doi.org/10.1016/j.fsidi.2024.301814)
- [245] P. K. Mandal, C. Leo, and C. Hurley, “Controlled territory and conflict tracking (contact): (geo-)mapping occupied territory from open source intelligence”, 2025. URL: <https://arxiv.org/abs/2504.13730>
- [246] H. Lyu, J. Huang, D. Zhang, Y. Yu, X. Mou, J. Pan, Z. Yang, Z. Wei, and J. Luo, “Gpt-4v(ision) as a social media analysis engine”, 2023. URL: <https://arxiv.org/abs/2311.07547>
- [247] S. K. KILIÇ and U. YAVANOĞLU, “Social media based open source intelligence analysis with artificial intelligence”, *Proceedings of the 45th IBIMA Computer Science Conference*, Córdoba, Spain, 2025. URL: <https://ibima.org/accepted-paper/social-media-based-open-source-intelligence-analysis-with-artificial-intelligence/#:~:text=This%20research%20examines%20how%20artificial%20intelligence%20can%20increase,collected%20and%20categorized%20using%20GPTbased%20natural%20language%20models.>
- [248] Y. Shynkarov, V. Solopova, and V. Schmitt, “Improving sentiment analysis for Ukrainian social media code-switching data”, *Proceedings of the Fourth Ukrainian Natural Language Processing Workshop (UNLP 2025)*, Vienna, Austria, 07 2025, pp. 179–193, DOI [10.18653/v1/2025.unlp-1.18](https://doi.org/10.18653/v1/2025.unlp-1.18)
- [249] Z. Zhang, Y. Cao, C. Ye, Y. Ma, L. Liao, and T.-S. Chua, “Analyzing temporal complex events with large language models? a benchmark towards temporal, long context understanding”, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Bangkok, Thailand, 08 2024, pp. 1588–1606, DOI [10.18653/v1/2024.acl-long.87](https://doi.org/10.18653/v1/2024.acl-long.87)
- [250] R. Manvi, S. Khanna, G. Mai, M. Burke, D. Lobell, and S. Ermon, “Geollm: Extracting geospatial knowledge from large language models”, 2024. URL: <https://arxiv.org/abs/2310.06213>
- [251] B. Ambrogi, “Lawnext: Thomson reuters’ ai strategy for legal, with mike dahn, head of westlaw, and joel hron, head of ai”, 2024. URL: <https://www.lawnext.com/2024/02/lawnext-thomson-reuters-ai-strategy-for-legal-with-mike-dahn-head-of-westlaw-and-joel-hron-head-of-ai.html>, Accessed: 2025-08-17

- [252] S. Wellen, “How lexis+ ai delivers hallucination-free linked legal citations”, 2024. URL: <https://www.lexisnexis.com/community/insights/legal/b/product-features/posts/how-lexis-ai-delivers-hallucination-free-linked-legal-citations>, Accessed: 2025-08-17
- [253] A. Lekssays, U. Shukla, H. T. Sencar, and M. R. Parvez, “Techniquerag: Retrieval augmented generation for adversarial technique annotation in cyber threat intelligence text”, Findings of the Association for Computational Linguistics: ACL 2025, Vienna, Austria, 07 2025, pp. 20913–20926, DOI [10.18653/v1/2025.findings-acl.1076](https://doi.org/10.18653/v1/2025.findings-acl.1076)
- [254] S. Paul, F. Alemi, and R. Macwan, “Llm-assisted proactive threat intelligence for automated reasoning”, 2025. URL: <https://arxiv.org/abs/2504.00428>
- [255] A. Berzinji, “Utilisation of large language models (llms) in osint-based cyberterrorism detection on social media”, International Journal of Cyber Criminology, vol. 18, 03 2025, pp. 210–223, DOI [10.5281/zenodo.4766812](https://doi.org/10.5281/zenodo.4766812)
- [256] A. Nurzhanov, “Application of the rag concept for detecting and classifying extremist content in the kazakh language”, URL: <http://dx.doi.org/10.2139/ssrn.5084790>
- [257] R. Bhaumik, V. Srivastava, A. Jalali, S. Ghosh, and R. Chandrasekaran, “Mind-watch: A smart cloud-based ai solution for suicide ideation detection leveraging large language models”, medRxiv, 09 2023, DOI [10.1101/2023.09.25.23296062](https://doi.org/10.1101/2023.09.25.23296062)
- [258] N. D. Anh, L. G. Hiep, and A.-C. Le, “Rag prompting for mental health classification with llm: A resource-efficient alternative to instruction tuning”, Computational Intelligence in Engineering Science (N. T. Nguyen, V. H. Pham, T. D. Tran, T.-P. Hong, Y. Manolopoulos, N. A. Le Khac, and P. T. Tin, eds.), 2026, pp. 115–129, ISBN: 978-3-031-98170-8
- [259] R. Leekha, O. Simek, and C. Dagli, “War of words: Harnessing the potential of large language models and retrieval augmented generation to classify, counter and diffuse hate speech”, The International FLAIRS Conference Proceedings, vol. 37, 05 2024, DOI [10.32473/flairs.37.1.135484](https://doi.org/10.32473/flairs.37.1.135484)
- [260] J. Chen, E. Shen, T. Bavalatti, X. Lin, Y. Wang, S. Hu, H. Subramanyam, K. S. Vepuri, M. Jiang, J. Qi, L. Chen, N. Jiang, and A. Jain, “Class-rag: Real-time content moderation with retrieval augmented generation”, 2024. URL: <https://arxiv.org/abs/2410.14881>
- [261] T. Yao, E. Foo, and S. Binnewies, “Personalised abusive language detection using llms and retrieval-augmented generation”, Proceedings of the 7th International Conference on Natural Language and Speech Processing (ICNLSP 2024), 2024, pp. 92–98. URL: <https://aclanthology.org/2024.icnls-1.11/>
- [262] A. Anaissi, J. Akram, K. Chaturvedi, and A. Braytee, “Detecting and understanding hateful contents in memes through captioning and visual question-answering”, 2025. URL: <https://arxiv.org/abs/2504.16723>
- [263] R. Singhal, P. Patwa, P. Patwa, A. Chadha, and A. Das, “Evidence-backed fact checking using rag and few-shot in-context learning with llms”, 2024. URL: <https://arxiv.org/abs/2408.12060>
- [264] X. Yuan, J. Wang, H. Zhao, Y. Tian, and F. Qi, “Empowering llms with toolkits: An open-source intelligence acquisition method”, Future Internet, vol. 16, 12 2024,

- p. 461, DOI [10.3390/f16120461](https://doi.org/10.3390/f16120461)
- [265] P. Ranade and A. Joshi, “Fabula: Intelligence report generation using retrieval-augmented narrative construction”, Proceedings of the 2023 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2023, 2023, pp. 603–610, DOI [10.1145/3625007.3627505](https://doi.org/10.1145/3625007.3627505)
 - [266] K. L. Kuok, H. H. Liu, W. W. Lo, *et al.*, “Crimekgqa: A crime investigation system based on retrieval-augmented generation and knowledge graphs”, 2024, DOI [10.21203/rs.3.rs-5233104/v1](https://doi.org/10.21203/rs.3.rs-5233104/v1), Preprint (Version 1), posted on Research Square, 10 October 2024
 - [267] F. Blefari, C. Cosentino, F. A. Pironti, A. Furfaro, and F. Marozzo, “Cyberrag: An agentic rag cyber attack classification and reporting tool”, 2025. URL: <https://arxiv.org/abs/2507.02424>
 - [268] K. Schwarz, K. Bollens, D. Arias Aranda, and M. Hartmann, “Ai-enhanced disaster management: A modular osint system for rapid automated reporting”, Applied Sciences, vol. 14, no. 23, 2024, pp. 1–28, DOI [10.3390/app142311165](https://doi.org/10.3390/app142311165)
 - [269] L. Mei, S. Mo, Z. Yang, and C. Chen, “A survey of multimodal retrieval-augmented generation”, 2025. URL: <https://arxiv.org/abs/2504.08748>
 - [270] M. M. Abootorabi, A. Zobeiri, M. Dehghani, M. Mohammadkhani, B. Mohammadi, O. Ghahroodi, M. S. Baghshah, and E. Asgari, “Ask in any modality: A comprehensive survey on multimodal retrieval-augmented generation”, 2025. URL: <https://arxiv.org/abs/2502.08826>
 - [271] R. Poynter, “The handbook of online and social media research: Tools and techniques for market researchers”, John Wiley & Sons, 2010, ISBN: 9780470710401
 - [272] B. Batrinca and P. C. Treleaven, “Social media analytics: a survey of techniques, tools and platforms”, AI & Society, vol. 30, no. 1, 2015, pp. 89–116, DOI [10.1007/s00146-014-0549-4](https://doi.org/10.1007/s00146-014-0549-4)
 - [273] C. Altobelli, E. Johnson, N. Forgó, and A. Napieralski, “To scrape or not to scrape? the lawfulness of social media crawling under the gdpr”, Deep Diving into Data Protection – 1979–2019 Celebrating 40 Years of Privacy and Data Protection at the CRIDS (J. Herveg, ed.), pp. 151–180, Larcier, 2021, DOI [10.5281/zenodo.6411788](https://doi.org/10.5281/zenodo.6411788), ISBN: 9782807926493
 - [274] C. Dilmegani, “6 main web scraping challenges & practical solutions [’25]”, 2025. URL: <https://research.aimultiple.com/web-scraping-challenges/>
 - [275] R. K. Saxena, S. Gupta, A. Raheja, and P. R. Tripathi, “Social media platform scraping and extracting paradigm”, Advances in Data Science and Information Technology (S. K. Rathi, B. Keswani, R. K. Saxena, S. K. Kapoor, S. Gupta, and R. Rawat, eds.), Wiley, 2024, DOI [10.1002/9781394231126.ch20](https://doi.org/10.1002/9781394231126.ch20)
 - [276] G. Karatas, “Social media scraping benchmark with tools & case studies”, 2025. URL: <https://research.aimultiple.com/social-media-scraping/>, Accessed: 2025-08-07
 - [277] M. for Developers, “Graph api documentation”, URL: <https://developers.facebook.com/docs/graph-api/>, Accessed: 2025-08-08
 - [278] X Developer Platform, “X api documentation”, URL: <https://developer.x.com/en/docs/x-api>, Accessed: 2025-08-08

- [279] S. developers, “Scrapy (version 2.13.3)”, 2025. URL: <https://pypi.org/project/Scrapy/>
- [280] L. Richardson, “beautifulsoup4 (version 4.13.4)”, 2025. URL: <https://pypi.org/project/beautifulsoup4/>, Accessed: 2025-08-08
- [281] S. developers, “Selenium (version 4.34.2)”, 2025. URL: <https://pypi.org/project/selenium/>
- [282] R. developers, “Requests (version 2.32.4)”, 2025. URL: <https://pypi.org/project/requests/>
- [283] D. Asturias, “Twitter and facebook scraper guide”, 2025. URL: <https://www.rapidseedbox.com/blog/twitter-facebook-scraper-guide>, Accessed: 2025-08-08
- [284] S. Dogan and G. Karatas, “The best web scraping apis in 2025: 30+ apis analyzed”, 2025. URL: <https://research.aimultiple.com/web-scraping-apis/>, Accessed: 2025-08-08
- [285] C. Dilmegani, “15> best low-code & no-code web scrapers in 2025”, 2025. URL: <https://research.aimultiple.com/no-code-web-scraper/>, Accessed: 2025-08-08
- [286] P.-J. Ezimorah, “The 7 best social media scrapers in 2025”, 2025. URL: <https://www.scraperaapi.com/web-scraping/social-media-scraper/>, Accessed: 2025-08-08
- [287] A. Dubois, “The best social media scrapers for 2025”, 2025. URL: <https://proxyway.com/best/social-media-scrapers>, Accessed: 2025-08-08
- [288] O. Lompart, “10 best social media scraping tools in 2025”, 2025. URL: <https://www.scrapediary.com/best-social-media-scraping-tools/>, Accessed: 2025-08-08
- [289] B. Data, “Bright data web scraper apis”, 2025. URL: <https://brightdata.com/products/web-scraper>, Accessed: 2025-08-08
- [290] Apify, “Apify web scraper apis”, 2025. URL: <https://apify.com/store/categories/social-media-scrapers>, Accessed: 2025-08-08
- [291] Zyte, “Zyte web scraper apis”, 2025. URL: <https://www.zyte.com/zyte-api/>, Accessed: 2025-08-08
- [292] S. (Decodo), “Smartproxy web scraper apis”, 2025. URL: https://smartproxy.com/lp/smartproxy-is-now-decodo?adgroupid=1273236130921084&utm_source=bing&utm_medium=cpc&utm_campaign=SA-Brand-MCPC-Search%2F09%2FEUROPE_Other&utm_term=smartproxy&campaignId=677666993&adgroupId=1273236130921084&adId=79577365810084&targetId=kwd-79577768250806:loc-93&device=c&munique=bad23d1efe101f153d765589feb5495b&msclkid=bad23d1efe101f153d765589feb5495b&utm_content=smartproxy-exact_phrase, Accessed: 2025-08-08
- [293] PhantomBuster, “Phantombuster web scraper apis”, 2025. URL: <https://phantombuster.com/>, Accessed: 2025-08-08
- [294] Nimbleway, “Nimbleway web scraper apis”, 2025. URL: <https://www.nimbleway.com/>, Accessed: 2025-08-08
- [295] SOAX, “Soax web scraper apis”, 2025. URL: <https://soax.com/>, Accessed: 2025-08-08
- [296] ScraperAPI, “Scraperapi web scraper apis”, 2025. URL: <https://www.scraperaapi.com/>, Accessed: 2025-08-08

- [297] Octoparse, “Octoparse web scraper apis”, 2025. URL: <https://www.octoparse.com/>, Accessed: 2025-08-08
- [298] H. Kumar, “Classification of short text using various preprocessing techniques: An empirical evaluation”, Recent Findings in Intelligent Computing Techniques, 2018, pp. 19–30, DOI [10.1007/978-981-10-8633-5_3](https://doi.org/10.1007/978-981-10-8633-5_3)
- [299] Arpita *et al.*, “Data cleaning of raw tweets for sentiment analysis”, 2020 Indo-Taiwan 2nd International Conference on Computing, Analytics and Networks (Indo-Taiwan ICAN), 2020, pp. 273–276, DOI [10.1109/Indo-TaiwanICAN48429.2020.9181326](https://doi.org/10.1109/Indo-TaiwanICAN48429.2020.9181326)
- [300] P. Savastano and community, “Cheshire cat ai”, 2025. URL: <https://cheshirecat.ai/>
- [301] S. Ramlochan, “Complete guide to prompt engineering with temperature and top-p”, 2024. URL: <https://promptengineering.org/prompt-engineering-with-temperature-and-top-p/>, Accessed: 2025-08-06
- [302] V. Schmalbach, “Does temperature 0 guarantee deterministic llm outputs?”, 2025. URL: <https://www.vincentschmalbach.com/does-temperature-0-guarantee-deterministic-llm-outputs/>, Accessed: 2025-08-06
- [303] Vellum.ai, “What is llm streaming and how to use it?”, URL: <https://www.vellum.ai/llm-parameters/llm-streaming>, Accessed: 2025-08-06
- [304] APXML, “Llama 3 1.8b”, 2024. URL: <https://apxml.com/models/llama-3-1-8b>, Accessed: 2025-08-05
- [305] llama3 AI, “Llama 3.1 requirements”, 2024. URL: <https://www.llama3-1.com/requirements/>, Accessed: 2025-08-05
- [306] Meta, “Llama 3.1 8b instruct model”, 2024. URL: <https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct>, Accessed: 2025-08-05
- [307] G. Cloud., “Llama 3.1 8b instant (model card)”, 2024. URL: <https://console.groq.com/docs/model/llama-3.1-8b-instant>, Accessed: 2025-08-06
- [308] Llamaimodel, “Llama vs chatgpt (2025)”, 2025. URL: <https://llamaimodel.com/vs-chatgpt/>, Accessed: 2025-08-05
- [309] P. Bhavsar, “Best llms for rag: Top open and closed source models”, 2024. URL: <https://galileo.ai/blog/best-llms-for-rag>, Accessed: 2025-08-05
- [310] E. Hecks, “The benefits of open-source vs. closed-source llms”, 2025. URL: <https://opendatascience.com/the-benefits-of-open-source-vs-closed-source-llms/>, Accessed: 2025-08-05
- [311] T. L. Team, “The llama 3 herd of models”, 2024. URL: <https://arxiv.org/abs/2407.21783>
- [312] OpenAI, “Introducing gpt-4.1 in the api”, 2025. URL: <https://openai.com/index/gpt-4-1/>
- [313] OpenAI, “Gpt-4 è il sistema più avanzato di openai e produce risposte più sicure e appropriate”, 2024. URL: <https://openai.com/it-IT/index/gpt-4/>, Accessed: 2025-08-07
- [314] OpenAI, “Presentazione di chatgpt”, 2022. URL: <https://openai.com/it-IT/index/chatgpt/>, Accessed: 2025-08-07
- [315] Q. Team, “Fastembed - fast, accurate, lightweight python library to make state of the art embedding”, 2023. URL: <https://github.com/qdrant/fastembed>, Accessed:

- 2025-08-07
- [316] N. Team, “Introducing nomic embed: A truly open embedding model”, 2024. URL: <https://www.nomic.ai/blog/posts/nomic-embed-text-v1>, Accessed: 2025-08-07
 - [317] Z. Nussbaum, J. X. Morris, B. Duderstadt, and A. Mulyar, “Nomic embed: Training a reproducible long context text embedder”, 2024. URL: <https://arxiv.org/abs/2402.01613>
 - [318] J. Liang, “Evaluating open-source vs. openai embeddings for rag: A how-to guide”, 2025. URL: <https://www.tigerdata.com/blog/open-source-vs-openai-embeddings-for-rag>, Accessed: 2025-08-07
 - [319] Techtoni, “Choosing the right embedding model for your rag application”, 2025. URL: <https://techttoni.com/posts/genai/embedding-models/>, Accessed: 2025-08-07
 - [320] C. C. AI, “Llm and embedder setup video”, 2024. URL: <https://cheshire-cat-ai.github.io/docs/assets/vid/setup.mp4>, Accessed: 2025-08-28
 - [321] N. Procopio, “Decoding chunking: Notes on mastering language structure”, 2024. URL: <https://cheshirecat.ai/decoding-chunking/>, Accessed: 2025-08-28
 - [322] H. Yu, A. Gan, K. Zhang, S. Tong, Q. Liu, and Z. Liu, “Evaluation of retrieval-augmented generation: A survey”, *Big Data*, p. 102–120, Springer Nature Singapore, 2025, DOI [10.1007/978-981-96-1024-2_8](https://doi.org/10.1007/978-981-96-1024-2_8), ISBN: 9789819610242
 - [323] S. Es, J. James, L. Espinosa-Anke, and S. Schockaert, “Ragas: Automated evaluation of retrieval augmented generation”, 2025. URL: <https://arxiv.org/abs/2309.15217>