POLITECNICO DI TORINO

MASTER's Degree in COMPUTER ENGINEERING

Exploring LLM-Based Solutions for IoC Extraction and Classification in Social Media: Reddit Case Study

Supervisors

Prof. Danilo GIORDANO

Prof. Juan CABALLERO

Giordano PAOLETTI

Candidate

Giovanni GIORDANO

MARCH 2026

Exploring LLM based solutions for IoC extraction and classification in social media: Reddit case study

Giovanni Giordano

Abstract

The evolution of the internet in recent years has brought countless opportunities for global interconnection and growth, but it also provided fertile ground for malicious activities of all kinds. Every day thousands of people are victims of frauds and the identification of such activities is vital to guarantee a safer environment for the less experienced users. Social media in particular, used by users for activities prone to fraudulent exploitation, like asking for financial advice, share investment opportunities or seek technical support, have rapidly become a target for malicious actors, who quickly evolved techniques to exploit the architectures of said platforms. In cyber threat intelligence, the ability to extract indicators of compromise (IoCs) from unstructured sources, such as social media posts, forums or instant messaging chat is crucial for correctly identifying potential threats. Standard approaches rely on tools that mainly exploit regular expressions to correctly extract the relevant indicators. However, these methods often present downsides, as the lack of contextual understanding makes it impossible to differentiate between benign and malicious indicators, and threat actors have found more sophisticated ways to bypass these detection tools.

This thesis proposes a new framework based on large language models (LLM) for the extraction and filtering of indicators of compromise in social media text. Unlike traditional methodologies, this framework can use the context of the content and the knowledge inherent in the LLM to determine whether an identity should be considered a threat. This thesis studies the application of LLMs for this task, with the application of recent prompt engineering techniques in order to have a clear understanding of what works best for the task. In order to do so, we chose a popular social media platform (Reddit), from which we gathered 600 textual posts from various forums (called subreddits) with the objective to form a manually labeled balanced gold standard, serving as a benchmarking ground truth and perform analysis on the application of different approaches. This study does not stop at the application and comparison of the LLM against standard regular expression tools, but tries different methods and level of analysis, which includes the application of the LLM for both extraction and filtering of IoCs, the application for only extraction (to have a better comparison with tools using regular expressions), the application of a baseline tool (regular expression-based for only extraction and the use of a combined approach (baseline tool for extraction and LLM for filtering). These four main approaches were rigorously compared to correctly identify the strengths and weaknesses of each methodology.

Promising results have emerged from this study for the application of these new technologies in this particular domain. LLM for both extraction and filtering demonstrated a better precision and overall F1 score, good ability to reduce operational noise and superior capacity at handling complex or obfuscated entities thanks mainly to its superior context awareness and semantic abilities. Prompt studies show how important it is to carefully choose the components as they can have an impact on the performance of the model. Interesting conclusion can be drawn by the importance of some elements like few shot prompting and domain specification

This solution has proven to be ready for implementation in real world threat intelligence, optimizing human resource and reducing and reducing alert fatigue. In addition, it is a more flexible solution, which can be adapted to different scenarios by simply tuning the prompt.

ACKNOWLEDGMENTS

Grazie ai professori Juan e Danilo e al dottorando Giordano per essere stati così pazienti con me e per avermi fatto capire come dovrebbe essere condotta una ricerca scientifica. I loro consigli sono stati fondamentali per dare più rigore scientifico alla mia tesi e sono soddisfatto del risultato finale.

Ricordo bene la sensazione che provai quando, da piccolo, vidi un computer per la prima volta. Rimasi stupito, mi chiedevo come fosse possibile che da una scatola piena di fili e circuiti potessero nascere immagini, suoni e interi mondi sullo schermo. Quella sensazione di curiosità non è mai scomparsa; anzi, è cresciuta. Mi ha portato a scegliere ingegneria informatica e questo lungo percorso è finalmente giunto al termine.
Durante questi anni universitari mi è capitato spesso di pensare al giorno della laurea, a questo grande traguardo. Più ci pensavo, più mi rendevo conto che nell'immagine che avevo in mente non c'ero soltanto io. Accanto a me c'erano la mia famiglia e i miei amici, persone senza le quali tutto questo non sarebbe stato possibile, per cui mi sento di dirvi: questa laurea non è mia, questa laurea è stata possibile grazie al continuo supporto da parte vostra, nei momenti belli e specialmente in quelli difficili.

- **Mamma**, grazie per aver messo i tuoi figli al primo posto sempre; anche se spesso sembra che non lo capisca, so bene che tutto ciò che dici e fai è sempre per il mio bene, ti voglio tanto bene.

- **Lucia**, grazie di essere la migliore sorella che io possa avere. Sto seguendo la tua crescita personale da vicino e sono orgoglioso della persona che stai diventando. Per qualsiasi cosa nella tua vita potrai sempre contare su di me.

- Un ringraziamento speciale va a **Marco**. Sei in assoluto la persona che conosco di più. Ci siamo visti crescere ed evolvere, sviluppando un'affinità mentale che considero rara: spesso so esattamente a cosa stai pensando ancora prima che tu parli. Credo in te come persona e nel percorso che stai costruendo e sono convinto che raggiungerai traguardi pazzeschi, non fermarti.

- Un'altra persona che si è aggiunta alla mia vita relativamente da poco è **Uxue**. Uxu, so bene quanto odierai questo momento quando leggerò i ringraziamenti, quindi te lo dico in modo che solo tu possa capire: *Benetan pozik nago zure bezalako norbait izateaz nire bizitzan. Niretzat oso berezia zara, maite zaitut.*

- **Sam**, aver fatto questo percorso insieme ha reso l'università molto più sopportabile. Il tuo modo di fare e vedere le cose mi affascina tantissimo. Continua

così, non vedo l'ora di vedere cosa ti aspetta nel futuro, sarò in prima fila a farti il tifo.

- **Ema**, sei stato il mio primo amico alle superiori e, dieci anni dopo, siamo ancora qui. Ti considero come un fratello e la nostra amicizia è una delle cose più belle che ho nella vita.
- Ci tengo anche a ringraziare **Amedeo, Emiliano, Leonardo, Deto, Luca, Antonio, Pietro ed Enrico** per essere delle fantastiche persone con le quali ho avuto il piacere di confrontarmi e apprendere molto.
- **La mia famiglia**: nonna Adele, i miei cugini e le mie cugine, e gli zii e le zie con cui mi ritrovo ogni anno per il tradizionale pranzo di Natale.
- Un ringraziamento anche ai miei **compagni di erasmus** Viola, Eleonora, Lisa, Max, Darragh, Gabe e Joe per essere stati la mia seconda famiglia durante l'anno a Madrid; anche se le nostre vite hanno preso strade diverse, quell'esperienza me la porterò nel cuore tutta la vita.

Vi ricordate quello che ho detto all'inizio, riguardo ai computer? Bene, c'è una persona davvero speciale che ha sempre capito il mio potenziale ed ha fin da subito supportato questa mia passione: mio papà. **Papà**, grazie per aver avuto pazienza se da piccolo ti ho smontato tutti i computer cercando di capire come diamine fosse possibile che un pezzo verde potesse far vedere interi mondi su uno schermo di 15 pollici, grazie per aver avuto pazienza se ti rubavo il computer per scaricare i videogiochi probabilmente installando 30 virus. La mia passione per l'informatica è nata da un tuo input e questo traguardo è tanto mio quanto tuo.

Ce l'abbiamo fatta papà.

Questa laurea segna la conclusione di un percorso di cui sono profondamente soddisfatto e, allo stesso tempo, rappresenta l'inizio di nuove strade e nuovi orizzonti. Sono pronto e impaziente di affrontare ciò che verrà con orgoglio, determinazione e soprattutto a testa alta, come ho sempre fatto, sapendo di poter contare su persone straordinarie che mi stanno accanto e non mi lasceranno mai solo.

> *Steve Jobs diceva che bisogna restare affamati e folli. Guardando indietro a quel bambino che cercava interi mondi dentro un 'pezzo verde', capisco che quella follia era dentro di me fin dall'inizio. Spero di non perdere mai la fame di scoprire cosa si nasconde dietro ogni nuova sfida.*

Da parte mia è tutto, grazie per l'attenzione.

Giovanni

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

During recent years, the Internet has evolved from a static repository of information to a dynamic ecosystem driven by user-generated content. As of October 2025, the number of active internet users has surpassed 6 billion users [1], accounting for roughly **72%** of global population.

The nature of social media has changed drastically through the years, they were primarily intended to be a connection for people, but during their evolution they ceased to be merely tools for entertainment and have evolved into critical channels, through which terabytes of data are exchanged every day. In the context of cyber-security, this phenomenon has given rise to Social Media Intelligence (**SOCMINT** [2]), whose role is to gather, analyze and interpret social media data.

However, this accessibility and big amount of data represent a double edged sword, while social media facilitate knowledge sharing, they have become a breeding ground for malicious activities. Malicious actors are starting to increasingly exploit the anonymity and great reach of social networks to orchestrate scams, frauds, distribute malwares and launch social engineering campaigns. A prime example of these threats can be found in discussion forums, such as Reddit, where users can engage with other actors whose identity is largely anonymous. Here, reliance on manual analysis is unfeasible due to the sheer volume of data generated daily. Consequently, automated extraction techniques are required.

A crucial point to understand is that these malicious actors rely on tangible technical artifacts that can be added through social media posts.For example, a phishing campaign uses malicious *URLs* and malware distribution involves specific *file hashes*, while other scam techniques rely on different methodologies. In the domain of Cyber Threat Intelligence (CTI), thse piece of data are referred to as "**Indicators Of Compromise (IoCs)**". Detection and extraction of these IoCs is crucial for proactive defense. However, identifying valid IoCs in discussion threads is a non-trivial task. Unlike emails or other more structred forms of communication, social media data presents itself as **unstructured and noisy**. Posts may contain slang, obfuscated parts, sarcasm or ambiguous contexts. Indicators of Compromise can be differentiated in 2 main categories: **Malicious** and **Not malicious (Benign)**. distinguishing between these two categories is another challenge since it requires

semantic understanding of the text. Extraction methods based on rigid pattern matching, such as Regular Expressions, often proves inadequate, as they lack the complexity to distinguish between a malicious or not malicious IoC.

One of the latest innovation in the field of artificial intelligence may help us address this issue: **Large Language Models (LLMs)** . Thanks to their pre-trained knowledge and reasoning capabilities , LLMs are able to intepret the context of a post, potentially distinguishing between threat or non threat posts in a better way than what regular expressions can achieve. The application of LLMs does not come without a challenge. The prompt structure is essential and LLM hallucinations [3] must be minimized to ensure this is a valid option.

1.1 Goals

The object of this thesis is to research the application of LLM in the field of cyber threat intelligence, specifically in the extraction of Indicators of Compromise (IoCs) and how LLMs compare against traditional tools based on regular expressions.

While Regular Expressions may be the industry standard, this research aims at showing its limitations when dealing with noisy, unstructured social media data and how LLMs can bridge this semantic gap.

To conduct this study, we will:

- **Create a Ground Truth Dataset:** since no standardized dataset exists for IoC extraction in social media, a crucial point for the success of our study is the construction of a manually annotated dataset. We specifically targeted Reddit due to its density of technical discussions and community-driven forums (the so called "subreddits"), which distinguish it from generic social networks. By collecting and labeling raw posts from different subreddits, this work establishes a reliable baseline where we can combine heterogenerous contexts, more high-risk subreddits (often investment related), general purpose discussions and subreddits where frauds are reported.

- **Analyze the performance of the various approaches:**To establish a formal comparison between the industry standard based on **Regular Expressions (RegEx)**, and the proposed approach based on **Large Language Models (LLMs)**. We'll define different categories of architectures, which will also include mixes of the two approaches, all of this is aimed to understand what really works best for this task. Then, we will quantitatively measure the effectiveness of each approach, based on standard metrics. To really understand the difference between the approaches, a granular error analysis will be conducted by inspecting specific instances of True Positives and False Positives. This study aims to interpret the root cause of differences between the approaches, while trying to give an explanation tied to the fundamental differences of the approaches

- **Prompt engineering ablation study:** Prompt structure is fundamental when interacting with an LLM. We expect the results to vary wildly with different prompts and our goal is to isolate specific components of the prompt to determine what elements are more impactful, based on performances
- **Conduct a study on IoC presence on social networks such as Reddit:** Once the evaluation on the approaches has been conducted, we will see how the most interesting approaches perform on a bigger dataset, collected in a large time period. We will be able to see real world application of the proposed approach and give examples, other than observing how scam campaigns evolve over time.

1.2 Structure of the Thesis

The thesis is structured as follows:

Chapter 2 Provides background on the terminology and tools that will be used in the thesis

Chapter 3 introduces previous work that has already been conducted in this field using LLM, with an emphasis on the results obtained

Chapter 4 presents the data collection, with a focus on the creation of a small ground truth dataset and a general dataset that will be used to conduct more studies on the application of the approach once its performance are measured.

Chapter 5 define the approaches used to extract IoCs form the social media posts

Chapter 6 evaluates the different approaches and analyzes the LLM results, with a focus on the LLM technical aspect.

Chapter 7 draws the final conclusions from the thesis, and indicates possible future improvements.

Chapter 2

Background

This study is situated at the intersection of two main fields of computer science: **Cybersecurity,** specifically social media intelligence, and Artificial Intelligence, with particular focus on the use of LLM and prompt engineering. During the dataset creation and analysis process, different concept and tools from both fields were used, such as Python libraries

In this chapter, I will introduce the technologies and concepts needed to have a full understanding on how the different parts of the thesis were executed, firstly by describing social media structure, with a particular focus on Reddit, which is the target of our analysis, by defining the nature of cyber threats and Indicator Of Compromise (IoCs). Finally, we introduce the principle of Large Language Models (LLMs), which represents the core technology used to conduct this study.

2.1 Social media and Reddit

2.1.1 Social Media Intelligence (SOCMINT)

Social media's influence on today's society is well established, almost 5.6 billion users [1] use social networks as of October 2025. Furthermore, social media's role in today's society has evolved, they were primarily intended as a way to connect people, but rapidly evolved as a place where users get their information, **ask for financial advice** or rely on community consensus to validate their decisions.

Social Media Intelligence [4] is a subfield of Cyber Threat Intelligence (CTI), whose goal is monitoring, analysis and extracting information from social media platforms, with a particular focus in user-generated content. Unlike traditional static web sources, social media provides dynamic unstructured data shared within web communities . One of the core challenge of SOCMINT, which we will deeply address in this thesis, is the signal-to-noise ratio: with the massive amount of data passed through the various social media every day, the ability to identify and isolate threats is essential in order to conduct a good study. Analyzing social media content comes with a series of challenges: due to the large amount of teens, the use of slang, trolls, grammatical errors and abbreviations is widely spread. This introduces

another layer of difficulty when dealing with content that might present a threat. As a consequence, automatic extraction tools based on regular expressions may fail: contextual understanding is essential to isolate malicious activities

2.1.2 Reddit

Reddit is a social media platform also known as "the front page of the internet". It is structured in **subreddits**, which are independent, topic-specific communities, ranging from massive, general-interest ones to more specialized, niche, communities. Users can follow any subreddit they want and will receive the content posted in them in their home page.

Reddit is a very user-oriented platform: any content on the app is posted by users (who are often anonymous, they don't use name or surname but rather pseudonym), who can interact with it by commenting or by giving an 'upvote' (positive feedback) or 'downvote' (negative feedback). The difference of the two (upvote-downote) results in the post score and this metric directly influences the visibility of the content. Users are able to upload almost any kind of content such as images, videos and **unstructured text**.

The democratic structure of Reddit is a double-edged sword. Poorly moderated subreddits can contain any kind of textual posts inside them there may be different IoC, whichi people often exploit in order to perpetrate frauds. Communities are self-regulated and rely on manual revision of the content: in every subreddit there are special users called **moderators** are in charge of removing suspicious content that might represent a threat to others. This is not always possible due to the massive amount of posts uploaded to the various subreddits every day, so an automated solution to this problem is necessary [5].

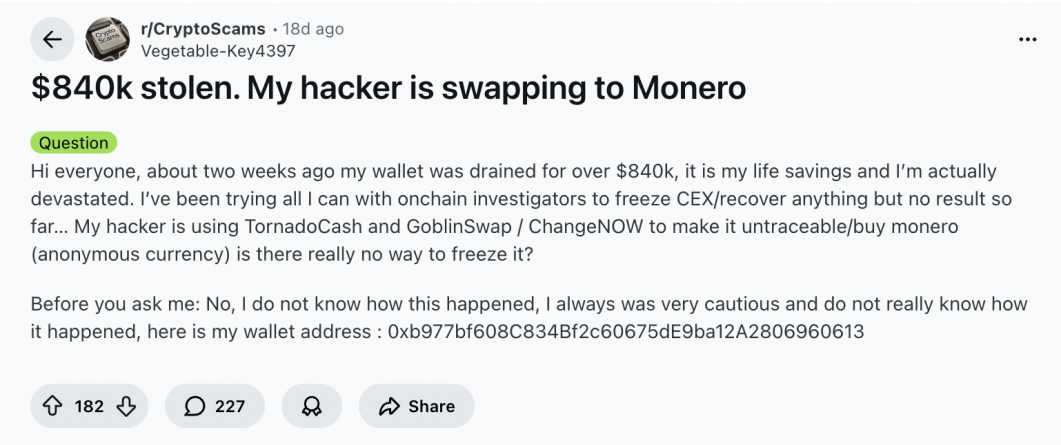

Figure 2.1: Example of a Reddit post containing unstructured text and a specific Indicator of Compromise (IoC). In this example, a user in `r/CryptoScams` shares a narrative along with an Ethereum wallet address. The proposed system aims to automatically extract such artifacts from the noisy textual data.

2.2 Cyber threats and traditional extraction mechanisms

2.2.1 Taxonomy of common social media frauds

As stated before, social media have become highly attractive for malicious actors that exploit their anonymity and poor moderation. Frauds can be divided in different types as showed in Table 2.1.

Scam Type	**Description**
Phishing	Attackers impersonate legitimate entities (e.g., banks, services) to deceive users into clicking malicious links or revealing sensitive credentials.
Romance Scam	A form of social engineering where criminals build fake romantic relationships over time to manipulate victims into sending money (often referred to as "Pig Butchering").
Lottery & Sweepstakes	Victims are falsely notified that they have won a prize or a lottery but are required to pay an "advance fee" or taxes to claim the non-existent reward.
FOMO-based Scams	Exploiting the "Fear Of Missing Out," these scams create a false sense of urgency, typical in crypto communities—to pressure users into making hasty, high risk financial transfers.
Job Scams	Fake employment offers, often for high-paying remote work, designed to harvest Personally Identifiable Information (PII) or demand payment for training/equipment.
Social Media Impersonation	Fraudsters create fake profiles posing as customer support agents, influencers, or friends to gain trust and request money or account access.
Fake Charity Scams	Exploiting real-world tragedies or natural disasters, attackers set up fraudulent donation drives to steal funds intended for humanitarian aid.
Cryptocurrency Scams	A broad category including "Rug Pulls" (developers abandoning a project after raising funds), "Pump & Dump" schemes, and fake "Giveaways" (promising to double sent funds). These exploit the irreversibility of blockchain transactions.

Table 2.1: Taxonomy of common scams found on social media platforms.

Malicious actors often use fixed schemas when conducting their frauds. Although these patterns use different psychological techniques, they all have something in common: they reach to a broader audience thanks to social media, but will often try

to redirect the victim to another less secure platform, by using **technical artifacts** whithin their message. This is crucial as these artifacts serves as gateways to the fraud: once the victim is isolated to a private, unmoderated channel, the attacker can proceed with the fraudulent activities without any form of oversight. Detecting these artifacts before the victims have access to them is crucial for mitigating financial losses and preserving the integrity of the internet.

2.2.2 Indicator of Compromise (IoC)

Indicator of Compromise (IoCs) are digital clues, like malicious IP addresses or suspicious file hashes, that indicates that a system or network has been breached by a cyberattack. **Cyber Threat Intelligence (CTI)** is the field of cybersecurity that collects and analyze data about potential cyber threats.

In the context of this thesis, we will refer to any technical artifacts extracted from the analysis of unstructured social media text as Indicator of Compromise. We divided the main IoCs in categories and they will be explained in Table 2.2

Category	Artifact Type	Description & Examples
Network	Malicious URLs	URLs, fqdn, or any other artifacts used to redirect users to external sites or fake trading platforms. Often obfuscated using URL shorteners (e.g., `bit.ly`) or "defanged" syntax (e.g., `example[.]com`).
Financial	Crypto/ibans	Alphanumeric strings used to receive stolen funds. Common formats include Ethereum addresses (starting with `0x...`), Bitcoin addresses, and Tron addresses (starting with `T...`), International Bank Account Number (iban).
Communication	Social Handles	Usernames or contact details used to migrate the victim to unmoderated private channels. Includes Telegram/instagram handles (e.g., `@username`), phone numbers, emails.
Identity	Entities	Names of legitimate **Organizations** (e.g., "Coinbase", "Tesla") or **Persons** (e.g., "Elon Musk", CEOs) referenced to establish false authority. Unlike technical artifacts, these require context awareness to be identified as part of a scam.

Table 2.2: Taxonomy of Indicators of Compromise (IoCs) used in this thesis.

In addition to what has just been said, extraction of Indicator of Compromise, especially in social media text, is a difficult task to automate. First of all, Malicious actors are aware of the ways social networks handle suspected texts by using automatic filters. IoC obfuscation (also known as defanging)is used in order not to be detected by those tools.

Another important aspect is about the nature of IoCs in social media texts. Things like URLs, social media handles or crypto wallet are shared every day and only a fraction of those may represent a threat. Distinguishing between malicious and not malicious IoCs is another critical point this thesis will try to address, employing more than traditional regexp extraction tools.

To conclude, IoCs lifecycle is very short, thanks to blacklists and automated takedown, most of the potential threats are removed pretty quickly, creating a narrow window for intelligence collection. This volatility makes manual analysis harder and underscore the necessity for automated extraction tools, capable of capturing these artifacts before the frauds can take place.

2.2.3 Regular expression based extraction techniques

Frauds have been present since the beginning of the internet, evolving in complexity alongside the technology itself. In the early stages of the internet, users were unexperienced and malleable: this changed as years went by and they started to recognize standard phising techniques. Platforms introduced **filtering** as a way to mitigate frauds on their websites.

The first form of filtering is a pretty easy concept: define a static list (**blacklists**), filled with known malicious sites, this method works well with historically identified threats and static infrastructure. However, limitations of this approach are pretty evident: especially in recent years, malicious actors can create new domains quickly, therefore launching and concluding phishing campaigns before the list can be updated. This latency potentially create a window of vulnerability where users are not protected against the newest frauds schemas.

To overcome these limitations, the industry standards has shifted to another approach: **pattern base recognition**, which is done by using Regular Expressions (Regex). While blacklists rely on external databsses, regex works on the text itself. regex is the fundamental tool for syntactic extraction, allowing systems to identify artifacts based on their rigid structure format. For instance, to identify an Ethereum wallet address the regex rely on a fixed rule: the string must start with "0x" followed by exactly 40 hexadecimal characters. The regex pattern for this looks like this:

```
^0x[a-fA-F0-9]{40}$
```

this structure makes Regular Expressions computationally efficient and have very little room of error on standardized data.

However, this rapidity and little computational costs come with some drawbacks: regex are very fragile, they rely completely on the **syntax** of the data rather than the **semantic** meaning. Various techniques to avoid detection have been created,

obfuscation/defanging being one of those. Originally created to make IoCs harmless for security researcher sharing threat intelligence, it is now weaponized by threat actors to avoid automated detection filters. If a malicious actor inserts a single space into a wallet address (e.g., `0x 123...`) or writes "dot" instead of "." in a URL, the regex fails to match that IoC. Furthermore, regexp lacks **contextual awareness**. There is no way for this tool to distinguish between benign and malicious IoCs. For example, using this approach an link to google support will be extracted just like it would with a phishing link, requiring additional layers of logic to filter the results.

To bridge the gap left by syntactic rigidity, this thesis proposes the adoption of **Large Language Models (LLM)**. Unlike previously cited methods, LLMs operate on a semantic level. Thanks to their training on vast natural language datasets, they possess the capabilities to reconstruct obfuscated IoCs and to understand the intent behind a social media post. This consent for the extraction of IoCs based on their contextual function instead of their syntactic structure. The only downside of this approach is its computational cost: LLMs require substantial processing power, unlike regex scripts that runs in less than a second. Particular attention should be given to the prompt: this is our only way of communicating with the LLMs, rigorous optimization is needed to ensure the output remains structured, consistent and free from hallucinations.

A summary of the detection methodologies can be seen below in Table 2.3

Feature	**Blacklists**	**Regex**	**AI/LLM (Proposed)**
Detection Type	Reactive (Known threats)	Syntactic (Rigid patterns)	Semantic (Contextual)
Speed to Deploy	Slow (Requires update)	Instant	Instant
Adaptability	None (New domains)	Low (Defanging)	High (Understands intent)
Computational costs	Medium (Database)	Low	High

Table 2.3: Comparison of detection methodologies.

2.2.4 iocsearcher

Iocsearcher[6] is a Python library and command-line tool to extract indicators of compromise (IOCs) from HTML, PDF, Word (.docx), and **text files**. It can identify both defanged (e.g., URL hxxp://example[DOT]com) and unmodified IOCs (e.g., URL http://example.com). This library will be our standard to compare traditional extraction techniques to the LLMs. the supported IOC are the following:

Table 2.4: List of IoC type iocsearcher is able to extract

Category	Supported IoC
Network & Infrastructure	URLs, Domain names (FQDN), IP addresses (IPv4, IPv6), IP subnets (CIDR), Tor v3 addresses (onion), Amazon Resource Names (ARN).
File & Cryptography	File Hashes (MD5, SHA1, SHA256), Universal Unique Identifiers (UUID).
Threat Intelligence	CVE vulnerability identifiers, MITRE ATT&CK Technique identifiers (TTP), Android package names.
Financial & Crypto	**Blockchain addresses:** Bitcoin, Bitcoin Cash, Cardano, Dash, Dogecoin, Ethereum, Litecoin, Monero, Ripple, Solana, Stellar, Tezos, Tron, Zcash. **Banking:** Bank account numbers (IBAN), Payment addresses (Webmoney).
Social & Identity	Email addresses, Phone numbers, TOX identifiers, Spanish NIF, Chinese ICP licenses. **Social Handles:** Facebook, GitHub, Instagram, LinkedIn, Pinterest, Telegram, Twitter, WhatsApp, YouTube.
Miscellaneous	Copyright strings, Trademarks, Advertisement/Analytics identifiers (Google AdSense, Analytics, Tag Manager).

Regarding data normalization, iocsearcher supports detection of some popular defang operations,rearming the IOCs by default so that deduplication works even if the same IOC has been defanged in different ways. However, it is not possible to support all defang operations, as every analyst can come up with their own. Furhtermore, it is important to emphasize this extraction tool does not support **filtering** of the IoCs. It does not differentiate between malicious or benign indicators.

2.3 Artificial Intelligence

2.3.1 Large Language Models (LLMs)

A **Large Language Model (LLM)** is a type of Artificial Intelligence designed to understand, generate, and manipulate human language. It represents a significant evolution in the field of Natural Language Processing (NLP), moving beyond simple rule based systems to advanced Deep Learning techniques.

Fundamentally, an LLM is a probabilistic model trained on large datasets, which include books, websites, scientific articles, and code. During the training phase, the model does not just memorize text; instead, it learns the statistical structure of language, including grammar, reasoning patterns, and facts about the world. This

semantic understanding of the context is a crucial point we'll try to demonstrate in this thesis.

The core mechanism of an LLM is the **"next-token prediction."** When the model receives an input (known as a *prompt*, we'll talk more in depth about this later), it analyzes the context and calculates the probability of which word (or "token") is most likely to come next in the sequence. For example, given the phrase "The cat is on the...", the model uses its training data to predict that "mat" or "roof" are highly probable completions, while "sky" is not.

Modern LLMs, such as GPT-5 or Llama, are built upon a specific neural network architecture called **Transformer**. Introduced by Vaswani et al. in 2017 [7], the Transformer utilizes a "Self-Attention" mechanism. This allows the model to process an entire sentence at once and understand the relationship between distant words. This capability is crucial for understanding context, sarcasm, and intent, which makes LLMs significantly more flexible and powerful than traditional extraction methods like Regular Expressions. To understand the innovation introduced by Large Language Models, it is necessary to distinguish between two fundamental approaches in machine learning: **Discriminative AI** and **Generative AI**.

Discriminative Models (Traditional NLP) Until recently, most NLP applications relied on discriminative models (such as Support Vector Machines or BERT). The primary goal of these models is **classification**. They are trained to find a decision boundary between different classes. For example, a traditional spam filter is a discriminative model: it takes an input (an email) and assigns it a specific label ("Spam" or "Not Spam"). While these models are highly efficient for specific tasks, they are rigid. A model trained to detect spam cannot detect sentiment (positive/negative) unless it is retrained from scratch on a new dataset.

Generative Models (LLMs) In contrast, Large Language Models belong to the category of Generative AI. Instead of just classifying data, these models learn the **underlying structure** and patterns of the data itself. Their objective is not to assign a label, but to generate new data that resembles the training set. In the context of language, they predict the next likely word in a sentence based on the previous context. The key advantage of this approach is **adaptability**: a single generative model can perform translation, summarization, or extraction tasks simply by changing the input instruction (**prompt**), without the need for specific retraining.

In summary, while discriminative AI asks *"Which category does this data belong to?"*, generative AI asks *"What is the most probable continuation of this data?"*. This shift is what enables the flexible, context-aware extraction proposed in this thesis.

To conclude, the advanced semantic understanding and recent architectural improvements of Large Language Models positions them as a promising alternative for IoC detection.

2.3.2 Prompt Engineering

Prompts are sequence of instruction and context that are provided as input to a Language Model, to achieve a desired task.

Prompt Engineering is the practice of developing and optimizing prompts, to efficiently use LMs for various applications.

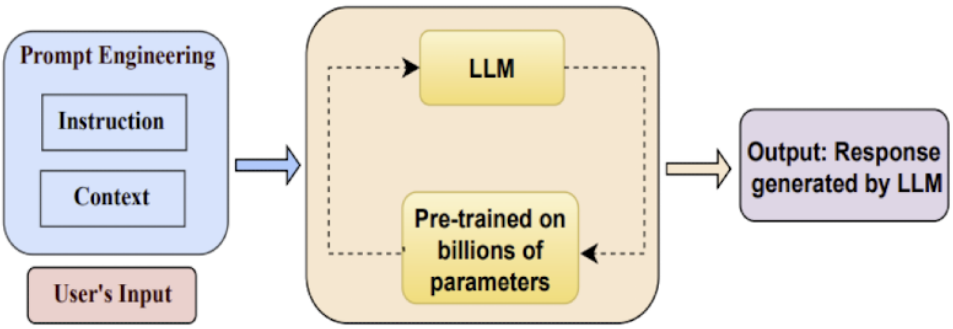

Figure 2.2: Conceptual workflow of the interaction between the user and the LLM, with the location of prompt engineering.

The main principles of prompt engineering are:

- **Specifiness:** the more the prompt is specific, the more focused output will be obtained.
- **Step-wise approach:** break large tasks into small chunks.
- **Iterate and improve:** re-work the inputs with iterative interaction with the LM.

Prompt engineering is important for research, advancements and evaluation of LLMs. Because of this, numerous prompt techniques have been developed. Table 2.5 offers a brief overview of the main techniques.

Technique	Description
Priming	Give the model an initial context or a "state" (for example by defining a role, give some rules) at the start of the input to guide the output
RGC (Role-Goal-Context)	A structured framework that specifies the *Role*, the *Task* and the *Context*.
Shot Prompting	Technique that gives the LLM some examples. It can be *Zero-shot* (no examples), *One-shot* (one example) or *Few-shot* (Some examples) to show the output format and explain the logic that has to be used while doping the requested task.
Chain of Thought (CoT)	Force the LLM to explicit the intermediate passages (often done by using the phrase "let's think step by step" before giving us the output, usually result in better performance in complex tasks.
Multi agent system	An architecture where complex tasks are decomposed into sub-tasks assigned to specialized agents. In the context of this thesis we don't use a multi agent system but more of a pipeline of LLM, where the output of one agent serves as the input for the next (chaining), creating a sequential or dynamic workflow (e.g., Extraction → Filtering → Review).

Table 2.5: Main prompt engineering techniques

Given the cybersecurity context of this thesis, other techniques can be taken into consideration.

RAG (Retrieval-Augmented Generation) consists in optimizing the output of a LLM, making it reference an authored knowledge base which is not the one on which it is trained, before generating an answer. This is because in cybersecurity threats change every day, a "frozen" LLM could not be up to date with the most recent vulnerabilities.

ToT (Tree of Thoughts) is an evolution of the Chain of Thought prompting technique. It is based on the exploration of multiple reasoning paths simultaneously, with eventual backtrack in order to find the best solution.

While Large Language Models are powerful tools, they face significant security challenges related to how they process user input. One major threat is **prompt injection**, where a malicious user inserts specific commands to manipulate the AI's behavior, effectively overriding its original programming to perform unauthorized actions. Similarly, **prompt leaking** poses a risk to privacy; this occurs when an

attacker tricks the model into revealing its hidden system instructions or confidential data that should remain private. Finally, developers must guard against **jailbreaking**, which refers to techniques used to bypass the AI's ethical safety filters. This allows the model to generate restricted, harmful, or illegal content that it is normally designed to block. Understanding these vulnerabilities is crucial for building safer and more robust AI systems.

2.4 Programming languages and libraries

The entire project was developed using the Python programming language, this choice was made because of Python's extensive ecosystem of libraries for data analysis, web scraping and natural language processing, as well as for its robust support for API integration.

2.4.1 Key libraries and tools

The development of the research relies on a specific set of software tools and libraries within the Python ecosystem.

- **Praw**[8] is a wrapper that abstracts the complexity of standard http calls. Instead of manually handling URLs, header and JSON parsing, this library allows us to work with Python objects. Praw handles rate limiting as well, ensuring the script stability and preventing the account to be banned. Authentication is controlled with **OAuth2** (secret client ID and client), making the access secure and controlled. In the context of this thesis, praw has played a crucial role for the post retrieval section. The library allowed us to easily browse through selected subreddits, posts and comments, thanks to a tree structure, and to order posts as new/hot/top. Overall this was the main tool for the data collection part of the thesis.

- **OpenAi API interaction**[9]: this project utilizes the official **OpenAI Python Library**, that behaves as a client wrapper to handle requests to the OpenAI REST API. Similarly to what praw does to for the Reddit APIs, this library facilitates the connection to the models via *http requests*. The interaction protocol requires constructing a conversation with the LLM, where specific roles are applied: **System** sets the global behaviour and constraints of the mode, while **User** contains the query to be executed.

  Different version of LLMs are available, ranging from more costly but effective models (such as the newest GPT-5o) to older versions.

- **Data Analysis and Visualization Tools:** Other useful libraries were used to help in the process:

  Pandas was used for the data manipulation phase, to organize data and make it easier to plot graphs.

Matplotlib was used in the data visualization part, this library facilitates the creation of bar charts, Venn diagrams and plots.

JSON (Built-in module) given that our datasets relied entirely on the json format, this library was essential for efficent data manipulation, allowing us to immediately store data acquired from the API interaction in a format that allowed us to conduct further analysis.

Chapter 3

Related work

The extraction of **Indicators of Compromise** is a critical topic for identifying malicious activity and is being actively studied to improve the performance. In literature, research on this domain has evolved from rigid, rule based approaches to more recent machine learning approaches, with Large language models (LLMs) starting to be taken into consideration for some tasks.

This evolution is due to already previously cited reasons. Specifically, the nature of social media unstructured text makes it hard to analyze with fixed rules, and several studies have proven how LLMs can be a better alternatives. This chapter will be structured into three main subsections that will provide a general overview of the most recent papers published:

- **Frauds on social media:** analysis of studies conducted on social media frauds.
- **LLM in cybersecurity:** focus on the most recent papers using LLMs in a cybersecurity context, with particular attention to the ones performing tasks similar to this thesis, using NLP in order to extract Indicators of Compromise.
- **prompt engineering:** Discussion of papers discussing the newest prompting techniques, crucial aspect when interacting with a LLM.

3.1 LLM in cybersecurity

The application of LLM in a cybersecurity context can give us valuable knowledge to see if they are flexible and can adapt enough to this specific domain. Some studies [10] already showed how the cybersecurity context is slowly starting to use LLMs. The main motivation that can be seen from the mentioned survey is that this specific domain has to be always up-to-date, as malicious actors can exploit newer technologies to their advantage, thus encouraging the exploration of LLM approaches in both the defensive and offensive side. Although promising results appeared, some limitations of the use of LLMs have emerged, such as *Hallucinations* and *loss of context*.

3.1.1 NLP based approaches for IoCs extraction

Although their work is based mainly on IoC extraction from threats reports, Froudakis et al. [11] perform a very similar analysis to this thesis. In this paper the authors explain how the main challenges when dealing with IoC extraction and labeling are:

- **Heterogeneous formats and information fragmentation**
- **Distinction between "artifacts" (not malicious content) and actual indicator of compromise.**

They confirm that Regular-expression-based tools lack contextual awareness and ultimately lead to an high raise of False Positives, potentially leading to **alert fatigue**. Limitations of modern approaches are also given:

Rule based and NLP approaches often struggle with complex grammatical structures and context that is far from the indicator.

Machine learning & Deep learning approaches requires Large, labeled, and high-quality datasets, which are scarce.

LLM based approaches, while promising, may be prone to hallucinations. Particular attention to the prompt must be given.

Manual labeling approaches remains the most reliable approach, but it is also very time consuming.

Particular attention is also paid to the **ground truth** creation. Various methods for the creation of a robust starting set are compared, highlighting the problems in them. Their solution to the creation of a quality ground truth is by using a mixed approach called **LANCE**, based on regular expressions which are responsible for extracting the indicators, followed by an LLM (zero-shot) used to label them, with the help of a rolling window of 8000 characters in order to provide some context, prompt structure has been shown to be a crucial point and they managed to find a prompt structure that significantly reduces error rate. It is worth noting that in this approach, an analyst must still be present, in order to give a final validation of the IoC found. By comparing a baseline annotation pass (where analysts did not have access to LLM labeling) against a guided annotation pass (where the LLM labels were given to the analysts), they demonstrated how providing the LLM labels can facilitate the ground truth creation process and give a more consistent result.

Finally, the study shows how LANCE outperforms other automated ground truth creation methods like VirusTotal and AlienVault, in both coverage and F1-score. Another interesting take from this paper is how LANCE outperforms naively prompted chatGPT, showing how a **structured pipeline** can greatly improve performance. Although in this thesis we will try to completely automate the IoC extraction and labeling process, this results give us a benchmark on the capabilities of the LLM in this specific domain.

Another interesting read is Domain and Website attribution beyond WHOIS [12]. Although the goal of this paper is different from ours (identifying the company or person that owns a specific domain), they implemented stylistic choices that provide

us Valuable knowledge, specifically in the **content attribution** part. The paper identifies 24 types of indicators that can help in identifying the owner of a specific domain and, although more expanded, the IoCs categories are similars to the ones we chose for this thesis.

In order to successfully extract the indicators, a variety of tools has been employed. Rigid structure ones such as *URLs, email and fqdns* were extracted using the regular expression tool **iocsearcher**. Acknowledging structural limitations of regular expressions like the impossibility to extract complex indicators such as identities (organization or persons names), the paper uses **NLP techniques**, specifically Named Entity Recognition (NER), showing great results for this kind of approaches in this specific domain. The Result of this studies prove how **NER** approaches works but still need some adjustment because of the amount of noise produced, thus forcing the implementation of a ranking system in order to lower it. Our thesis proposes LLMs and their native contextual awareness and superior semantic understanding as an alternative for complex entities extraction.

Muzami et al. [13] gives us valuable information about the use of LLM in the classification of crypto threats on large scale, scientifically proving how LLMs (GPT-4 and Llama three in this case) can be used in this specific domain. The paper shows great results, LLM-based approaches can reach an accuracy of **90%** when asked to distinguish between legitimate sites and scam ones. These results shows how these approaches can be considered a state of the art and we hope to find similar results for their applications in the social media domain.

TTPDrill (Husari et al.[14]) is a tool developed to automatically extract **TTP** (Tactiques,Techniques and Procedures) from unstructured intelligence report, with a rigid ontology used to define what can be considered a malicious action. Relevant for our thesis is the application of NLP techniques in order to identify verbs and nouns and to map them to the ontology, while using a regular expression tool to extract IP addresses and filenames and convert them into a more semantic rich term. This approach proves to be efficient enough in well written formal reports, although still needing a predefined ontology. What we hope to understand in this thesis is if the evolution of the LLMs can eliminate the need for the creation of such ontologies thanks to a better semantic understanding of the text. Newer models should have the ability to understand the context, even if the scammers is using metaphors or does grammatical errors. As showed by Gomez et al. [15], one of the main problem when utilizing data coming from crowdsourced data is the **pollution/noise** present in said data. Services like BitcoinAbuse reports up to 75% of their report as spam, while often wrongly classify the abuse type. It's a relevant comparison to make for our thesis: data on Reddit often presents noise, especially in scam reports subreddits and noise reduction methods are crucial, here is where the LLMs become helpful as they can not only extract the indicators, but also determine whether to consider them malicious. The results from this paper are pretty clear: LLM used for classification obtained an F1 score of **0.99** for spam identification, and a **0.95** score for abuse type classification, an unsupervised method that beats classic supervised classification

ones. Other results that should be taken into consideration are how they identified investment scams as the one with the highest economic impact, thus justifying our choice of crypto investing subreddits.

3.2 Frauds on social media

During recent years, social media have become targets of malicious actors because of the many ways it is possible to exploit such platforms. Mirtaheri et al. [16] provide us a scientific base for **cross platform manipulation**, underlining how malicious actors often use different social media in a coordinated manner. The study identifies three main components of a **Pump and Dump** scheme:

- **Malicious actors :** a group of people who plan the fraud operation.
- **Fraud platform:** private communication channels where the fraud can actually take place (Telegram/Reddit).
- **the "Hype" platform:** mainly a social media, used by malicious actors to attract victims, such as Twitter or Reddit.

This paper lends credibility to our choice of Reddit as our main social media, since it is mentioned as one of the main platform used to create hype. Telegram is also seen as more difficult to observe since the majority of channels are private.

Siddharth et al.[17] does another similar analysis that helps us have a better understanding of the problem and the urgency to act. They analyzed donation based abuse in social media and interesting results have emerged. First of all, they could rely on a large dataset with three million posts and 150000 accounts, 876 of which were identified as fraudulent, using **social engineering** tactics in order to leverage sympathy and urgency (earthquakes, war in Ukraine,cancer). The study identified social media as the prime platforms where malicious actors performs such frauds, mainly thanks to the speed it takes to create an account compared to the registration of a web domain. Similar patterns to previous studies have been found, malicious actors often engage the victims on social media, then they try to move the conversation to other platforms **email, mobile phone number, Telegram** to ask for payments using Paypal or crypto addresses. The extraction is done by using 78 keywords, while this thesis propose LLMs to not only analyze the presence of a word, but also the underlying intent.

Cryptocurrency scams aren't new, Vasek and Moore [18] already noted the dangers of using such platforms, and explored bitcoin-based scams. This study demonstrate how crypto scams are evolving and becoming a bigger threat, with malicious actors earning revenues of up to **11 million dollars**, with the bigger part of the profits deriving from a considerably small portion of users. They also extracted **Bitcoin addresses** in order to track the amount of money, indicating the importance of extracting such indicators. Although the paper is a bit dated, they also used a similar

approach to analyze criminal behavior online, using forums such as *bitcointalk.org* and *reddit.com/r/bitcoin*, proving how effective the study on these platforms can be.

3.3 Prompt engineering

Prompt structure proved to be crucial in order to obtain good performance from the LLMs. The ability of these models to generalize to new concepts thanks only to some examples has been proven with the introduction of **few-shot learning techniques** [19]. In this paper, the authors show how often "out-of-the-box" LLMs struggle with generalizing to newer or more complex tasks, thus needing fine tuning in order to have competitive performance. As an alternative they proposed a novel techniques called **few-shot learning**, in which they propose examples about the task to perform directly in the prompt text, without any fine tuning done to the model. The results showed that the ability to perform in context learning improved together with the increase of the number of parameters of the model, with performance comparable to fine tuned variations of said model. This is crucial for our thesis since these findings demonstrate the Large language models (LLMs)'s abilities to adapt to domains for which they were not trained, such as identifying indicators of compromise in social media text, without the need of fine tuning (which is hard to do because of the scarcity of a training corpus on this matter). We think that, for this thesis' specific task, it's important to provide some examples on the task that the LLM will perform (identifying IoCs in social media post), mainly because of the complexity of the task, maybe providing some edge cases such as obfuscated indicators in order to show how to handle them.

Other studies also focused on the effectiveness of role prompting [20]. In this paper the authors assessed the performance of a zero-shot model but with **strategically designed role prompting techniques**, and the results show the improvement of this approach. In the various benchmark dataset there is a rise in accuracy (from **53.5%** to **63.8%** on the AQuA dataset, from 23.8% to 84.2% on the Last Letter dataset). To conclude, authors attributed these improvement to a trigger for the **CoT** prompting. We'll use these technique to give more context to the LLM, using something like *"you are a cybersecurity expert and your task is identifying Indicators of compromise in this social media text ..."*

The previously cited **Chain of Thought prompting (CoT)** technique has been the subject of study as well. Studies [21] concluded that forcing the model to reason through a series of intermediate steps can once again improve the general performance, and it's adaptable to every specific domain. In this thesis for example, we could guide the LLM on how to reason in order to correctly execute the task of IoC extraction, including a thinking process with different steps such as *scanning of the text, check for defanged/obfuscated indicators, validation and a final output format check*

Lastly, a downside of the use of LLMs could be the **hallucinations**, as explored in the survey by Ji et al.[22]. This paper demonstrate how natural language generation

reached coherence level never seen before, but this accentuated the problems of hallucinations, where the model generate incorrect or unintentional text. Examples were provided where these hallucinations are shown, and techniques to avoid it are also shown, such as building high fidelity datasets, automatic data cleaning and information augmentation. We'll evaluate if hallucinations are a problem for our specific task and, in a positive case, try to use the above mentioned mitigation techniques

Chapter 4

Data collection

4.1 Dataset creation

Data collection is a crucial step to properly analyze the performances of different approaches in extracting various IoCs. This research focuses on unstructured social media text, as scams and frauds are often embedded in posts and can affect any user, particularly those with less experience. Various platforms were considered but we ended up selecting Reddit, a social media and online discussion platform founded in 2005.

Reddit is organized into thousands of user-created communities called subreddits, each dedicated to a specific topic such as technology, finance, memes, or science. Users can create posts, share links, and comment, with the visibility of content determined by an upvote and downvote system. Due to the self-regulated nature of its subreddits, this social network frequently serves as a host for many different types of scams, appearing in both post bodies and comments. Another relevant aspect is that Reddit combines anonymity with high user engagement. Attackers can easily create disposable accounts and spread malicious links, while victims or community members may respond in ways that provide useful contextual information.

This interaction between malicious and legitimate content creates a rich environment for the extraction of Indicators of Compromise, as IoCs may appear explicitly (e.g., IP addresses, domains, or wallet addresses) or implicitly, embedded within user discussions. Collecting and analyzing such data, therefore, provides not only the raw indicators but also valuable context that can improve detection models. Reddit was also chosen for its free and accessible APIs, which allowed us to retrieve a significant volume of posts and other relevant information for each selected subreddit. Drawing on previous research regarding Reddit dynamics and the identification of suspicious subreddits, we compiled a list of those most likely to contain scams.

Based on this analysis, the following subreddits were selected:

Subreddit	**Posts**
investing	823
trading212	817
Scams	773
coinbase	742
cryptocurrency	629
cryptomarkets	574
CryptoScams	493
CryptoMoonShots	466
NFTsMarketplace	340
cryptomoon	195
NFT	125
beermoney	69
SHIBArmy	60

Table 4.1: Selected subreddits and number of collected posts collected for August 2025

Although scams can exists in every kind of subreddits, these in particular were chosen because they focus on finance,investing and cryptocurrencies. It remains necessary to confirm that the selected subreddits contain instances of fraud, in order to do so a python script interacting with the official Reddit API is needed.

We used a simple script that uses **praw** (Python Reddit API Wrapper), a python library that provides a simple interface to the official Reddit APIs. Despite some limitations we'll talk about later, by authenticating with Reddit credentials we were able to collect a large amount of posts. The library handles pagination and rate limits automatically, facilitating the collection of large datasets for further analysis and manual labeling.

The next step is to identify the information that needs to be collected for each post, ensuring sufficient context to determine whether it constitutes an instance of a scam.

Field	Description
Subreddit	Name of the subreddit where the post was published
Post ID	Unique identifier of the post
Creation date (UTC)	Timestamp of when the post was created
Length	Number of characters in the post text
Title	Title of the post
Text	Full text content of the post
Score	Number of upvotes minus downvotes
IoCs	Extracted malicious links or data
URL	Link to the post
Author	Reddit username of the poster
Author age (days)	Age of the Reddit account in days
Number of comments	Total comments on the post
Potential scam	Label indicating if the post is suspected of being a scam
Comments	Collected data of comments associated with the post

Table 4.2: Information saved for each Reddit post

More information could be saved but we believed with this approach enough context can be obtained in order to be able to properly classify a post as potential fraud. The most relevant fields are of course the post Title and Text, that's mainly what malicious actors use to exploit victims, often trying to redirect them through other websites by using urls or other social networks handles.

4.1.1 Post collection script

As mentioned before, the use of the praw python library was essential for this project, since it made interacting with Reddit APIs easier and we could focus on other aspects. We opted for an object oriented approach, with well defined class that all served a specific purpose. The creation of a big enough dataset is essential for this kind of projects. The script also contains the possibility to include a time period for posts retrieval, by specifing the start date and the end date for the collection, this ensure that only posts published within the desired timeframe are retrieved. This feature allows for precise temporal analysis and ensures consistency when comparing data across different periods. By limiting the collection to a defined window, it is possible to monitor trends, detect sudden spikes in scam activity, and perform week-by-week or month-by-month analyses without including outdated or irrelevant posts.
The script operates as follows:

- `RedditClient:` this class is in charge of setting up the praw library, the information needed to interact with Reddit's API. It also contains the actual post retrieval methods and a call to the IoCsearcher class in order to scan for eventual IoCs. To help us have a better understanding of the various subreddits and where scams are more common, an array of `json` object is also computed every time the script is executed. The information saved for each subreddit

are the number of posts visualized for the desired time period, the number of posts that contains IoCs, the average upvote for post and the average number of comments.

- `IoCsAnalyzer`: to understand if a post should be considered as a potential scam, **IoCsearcher** and an **isscamcandidate** function are used. This class acts as a wrapper and contains the various methods used to initialize and interact with the library.

- `PostManager`: this class reads the input file `subreddits.txt` and ensures the posts identified as potential scams are correctly saved in a `.jsonl` file, with a specific format to make it easier to navigate the dataset. It also creates a `statistics.json` file where the statistics array previously computed can be saved

The list of all considered subreddits was stored in a `subreddits.txt` file, which the script would read at runtime. If a # symbol was placed before the name of a subreddit, the script interpreted it as a comment and ignored that subreddit.
Once the script started, the subreddits list was obtained and the collection could finally start.
The program pipeline is simple and can be seen illustrated in figure 4.1:

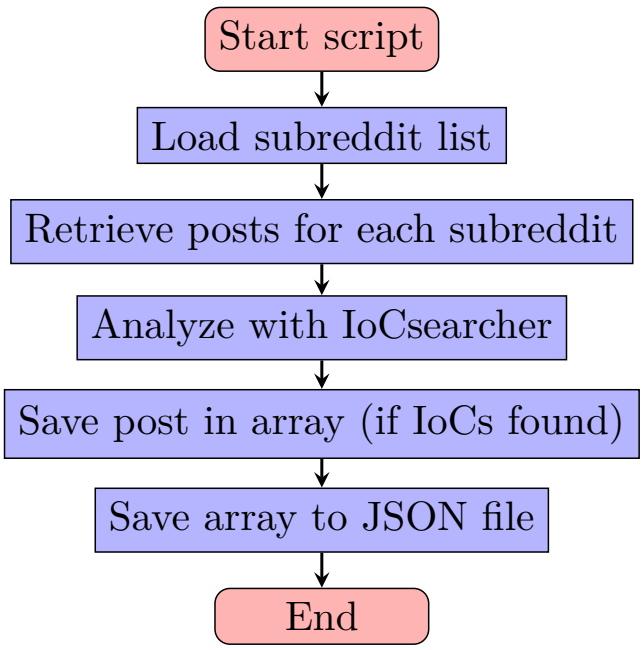

Figure 4.1: Simplified pipeline of subreddit and post analysis with IoCsearcher.

Table 4.3 presents a summary of the data obtained from the posts with IoCs collected during August 2025. It provides an overview of the number of posts analyzed, as well as the key statistics that will serve as the basis for the subsequent analysis.

The total number of posts in our dataset proved to be a solid base for our analysis. Similarly to what we did for our ground truth dataset, we chose a variety

Subreddit	Posts	Start date	End date	Posts/day
CryptoMoonShots	2587	2025-04-20	2026-02-16	8.57
Scams	1480	2025-06-30	2026-02-16	6.41
CryptoScams	1472	2025-04-20	2026-02-16	4.87
cryptomoon	980	2025-02-17	2026-02-16	2.69
cryptocurrency	993	2025-06-05	2026-02-16	3.88
investing	976	2025-05-07	2026-02-16	3.42
coinbase	815	2025-05-22	2026-02-16	3.02
NFTsMarketplace	682	2025-06-10	2026-02-16	2.72
cryptomarkets	484	2025-05-02	2026-02-16	1.67
trading212	352	2025-05-17	2026-02-16	1.28
SHIBArmy	257	2025-01-01	2026-02-16	0.63
beermoney	173	2025-01-08	2026-02-16	0.43
NFT	104	2025-06-12	2026-02-16	0.42
Total	**11355**			

Table 4.3: Number of posts with IoCs per subreddit in the dataset, with start and end dates, and posts per day density

of subreddits to differentiate between more suspicious subreddits and safer ones, including subreddits where victims do report the scams they suffered.

The average post per day column gives us insights on the most active subreddits. r/CryptoMoonShots and r/Scams are the most active ones with respectively **8.57** and **6.41** average each. Thee subreddits are usually the most prone to include malicious activity, since the amount of noise is often an occasion to perform it.

We chose this collection windows in order to also identify multiple scams campaign, where we hoped to see the evolution throughout the year of such activities. Ideally we could cross compare posts/scams in different subreddits to individuate different scam campaigns.

4.1.2 Praw library limitation

Although praw library is very useful since it takes care of interacting with Reddit APIs for us, we encountered some important limitations when we tried to push it for our post post retrieval purposes.

1. **Reddit API rate limit:** Reddit API imposes a limit on the maximum number of request possible per minute, from the official Reddit API we can observe how, for an authenticated free user, that limit is about **100 queries per minute (QPM)**. After some experimentation, We could see how exceeding that limit raised an **HTTP 429 "Too Many Requests"** exception, this is intended to ensure fair usage of the Reddit API, allowing all users to access it without any one user overloading the service. A solution to this could be adding a `sleep()` function in the code and avoid overloading the server with requests. Alternatively, we could try to catch the exception and work with it, without it crashing the script.

2. **Post retrieval limit:** Even by solving the first limitation, Reddit's API does not allow unlimited retrieval of posts from a subreddit at once, instead, it enforces pagination and time-window restrictions. This meant we had a subreddit specific limitation on how far back our post retrieval could go.

These issues prompted a revision of our post collection strategy. Instead of running a long script for a limited set of subreddits, we collected as many posts as possible for each subreddit within a shorter timeframe. A cron job was configured to run the script every Monday, retrieving posts from the previous week, ensuring consistent and automated data collection.

The methodology shifted from a vertical to a horizontal approach. Rather than focusing on large volumes of posts from a few subreddits, we expanded the range of subreddits considered. This broader coverage improved the likelihood of capturing diverse instances of potential scams and malicious content.

This approach offered multiple advantages. It mitigated the Reddit API rate limits, since the total weekly posts remained manageable. It also enabled continuous dataset growth, supporting longitudinal analysis of scam activity across communities. Additionally, spreading data collection across multiple subreddits reduced the risk of bias from overly active communities, improving the representativeness of the dataset.

The different kinds of exception the program could throw were handled in order to ensure the cronjob wouldn't be interrupted.

4.2 Indicators of Compromise in our dataset

Our search for online frauds rely mainly on the capacity of individuating IoCs (Indicator of Compromise) in the social media posts we gathered. An Indicator of compromise is any measurable artifact or piece of digital evidence that suggests a system, network, or application has been breached or is experiencing malicious activity.

4.2.1 IoCsearcher IoCs extraction

For our research we limited ourselves to a few of the most popular IoCs, they can be seen along with their explanation in Table 4.4

IoCs Type	Description	IoCsearcher support
bitcoin	Bitcoin address or related information	Yes
instagramhandle	Instagram username or handle	Yes
twitterhandle	Twitter username or handle	Yes
email	Email address	Yes
phoneNumber	Phone number	Yes
tron	Tron blockchain address	Yes
solana	Solana blockchain address	Yes
telegramHandle	Telegram username or handle	Yes
ethereum	Ethereum blockchain address	Yes
person	Person name mentioned in the post	No
organization	Organization name mentioned in the post	No
fqdn	Fully Qualified Domain Name (domain)	Yes
url	URL or link included in the post	Yes
iban	valid International Bank Account Number	Yes

Table 4.4: Types of IoCs extracted from Reddit posts, and whether they are supported by IoCsearcher

4.2.2 IoCs comparison in the dataset

Understanding the subreddits with the highest amount of IoCs is crucial for our research, this can give us indications on how to proceed with the analysis part and help us create a solid ground truth dataset.

Figure 4.2 offers a visual representation of the proportion of posts considered safe compared to those that may contain IoCs.

Subreddits such as `r/investing` and `r/trading212` contains the largest amount of posts, this is due to the fact this subreddits are finance based and are very active. The number of IoCs we found is relatively low compared to other subreddits.

`r/CryptoScams` is a subreddit dedicated to reporting scams specifically related to the cryptocurrency domain, whereas `r/Scams` addresses online fraud more broadly. Given their focus, a significant presence of IoCs can reasonably be expected in both communities.

A more interesting result can be seen in the `r/CryptoMoonShots` and `r/cryptomoon` subreddits. These communities focus on discussions related to emerging or lesser-known cryptocurrencites that are believed to have a significant growth potential. The main topics are speculative investement opportunities, new tokens and trading strategies. The presence of IoCs is not surprising, but it represents a significant portion of the content in these subreddits compared to the others in our list.

Studying these subreddits can provide valuable insights for our research because they illustrate how fraudulent actors exploit high-risk, speculative environments. By analyzing posts and identifying patterns in content containing IoCs, we can better understand the tactics and strategies used to deceive potential investors. This knowledge can improve automated detection methods, guide the creation of more accurate

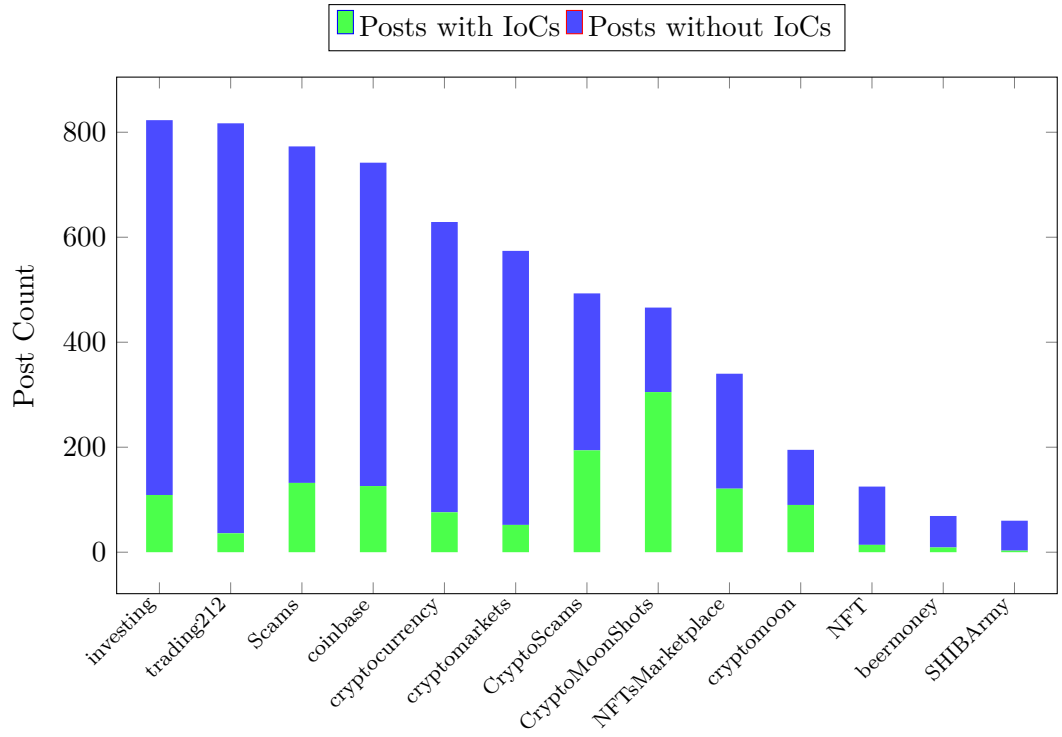

Figure 4.2: Stacked bar chart showing total posts and potential scam posts across the selected subreddits for the month of august 2025

datasets for labeling scams, and contribute to developing preventative measures for online financial fraud.

4.3 Ground truth creation

The first step in order to proceed with our analysis is the creation of a ground truth dataset. In the context of machine learning, a ground truth refers to verified, true data used for training, validating and testing artificial intelligence (AI) models. This enables us to evaluate the performances of various approaches by comparing output to the "correct answer".

The goal of our research is to see what tool is best at detecting IoCs in the text, this will help us better understand if a post can be considered fraudolent or not.

The process for our our ground truth creation is fairly simple: A `groundTruth.json` file is created, containing a sequence of `json` objects. The field we chose to include for each post in our ground truth dataset are:

- **Post ID:** a unique identifier automatically assigned by Reddit to each post, used to reference, retrieve, or distinguish individual posts within the dataset. It can also be used to access the original post on the Reddit website
- **Creation date:** the date and time (typically in UTC) when the post was published, essential for temporal analyses and reconstructing the chronological sequence of activity.
- **Title:** the post's title provided by the author, which often summarizes the content or intent of the message; it is a key field for classifying posts as potential scams or legitimate content.
- **Text:** the full textual content of the post, including descriptions, details, links, or references; this is the main field analyzed to detect IoCs (Indicators of Compromise) or signs of fraud.

Once these sequence of object was ready, our creation of the ground truth dataset could start. Each post must be carefully read and evaluated manually. The evaluator could acquire enough context from these fields to understand if the encountered IoCs should be considered malicious or not. A list of the most common IoCs is made, which facilitates our work by allowing us to quickly identify recurring patterns of malicious content and focus the analysis on the most relevant indicators.

Figure 4.4 presents an overview on the IoCs we considered during the creation of the ground truth dataset

The IoCs identified for each post are systematically recorded in a structured format, as follows: `[(type, IoCs, label), ...]` where `type` indicates the kind of IoCs, `IoCs` is the detected indicator, and `label` specifies whether the IoCs is considered malicious. We chose this format as it will help us conduct the analysis later on.

This approach was then executed for every post, manually labeling all the IoCs we could find. Reviews have been done to ensure quality and consistency of the ground truth dataset.

Another critical aspect was the subreddit choice: we used different communities for the creation of our ground truth dataset. An overview of our ground truth dataset can be seen in the table 4.5 down below:

Subreddit	**Posts**
r/CryptoScams	200
r/cryptomoon	100
r/CryptoMoonShots	100
r/Scams	50
r/investing	75
r/trading212	75
Total	**600**

Table 4.5: Number of posts collected per subreddit in our ground truth dataset.

The first subreddit we chose was `r/CryptoScams`, a community where users report and discuss scams related to the cryptocurrency world. It serves as a community-driven resource to alert others about fraudulent coins, projects, websites, or social media accounts, helping members avoid potential losses and stay informed about common crypto scams. The analysis of this subreddit helped us understandings common scam strategies, as well as facilitating the IoCs labeling. The second and third subreddits we chose are `r/cryptoMoon` and `r/cryptomoonshots`, as explained before these two subreddits are filled with many real threats, not only reports. It will be interesting to see how the different approaches will perform at finding IoCs in these posts. We chose to start in a subreddit where scams are explicitly reported because the posts already highlight malicious activity.
Although user reported content could be biased towards more obvious scam types,this approach enabled us to establish a baseline understanding on how fraudolent schemes are executed, and establish the IoCs categories that will be used to analyse other subreddits.

4.3.1 Manual labelling

As mentioned in Chapter 4, the ground truth dataset serves as the benchmark for this study. While section 4.3 details the dataset composition and its statistical properties, in this section we will focus on the techniques applied during the manual annotation phase of the study. Extracting Indicator of Compromise and manually annotating each single one into a structured threat intelligence dataset is not a simple transcription task: it requires context awareness and semantic interpretation of the post, to correctly distinguish actual threats from benign ones. Some normalization guidelines are needed to ensure a coherent annotation process. This criteria were rigorously applied to every post in the ground truth dataset:

- **Scope of the analysis:** The analysis of the post was not merely restricted to the body or title of the text: to ensure a correct understanding of the context

other factors were taken into account: this includes comments, upvote count of the post, age of the account that posted. Researches on the topic were often needed to correctly understand whether an IoCs was in fact malicious or not.

- **Normalization strategy:** although strings are already normalized before the computation of performance metrics, decisions on how to handle certain edge cases were needed. Defanged IoCs are transcripted in the "refanged" version. This ensured the analysis focuses on the semantic identity of the IoCs rather than the text transcription.

- **Human filtering:** Unlike the regular expression approaches, human annotators can evaluate the context of the IoCs. Legitimate IP addresses, benign URLs or a victim's Ethereum wallet address are all labeled as **Benign**. This distinction is crucial for our analysis of the filtering capabilities of an LLM.

- **Output format:** The final string containing the list of IoCs had to be normalized in *json* format, in order to be able to automate the analysis process. We opted to use an identical tuple format we already implemented with the LLM: `[(Type, Value, Label),...]`.

Ideally, these guidelines help reduce ambiguity in the annotation process. It provides a solid benchmark for the evaluation phase, although many iterative review of the dataset will be needed during the analysis phase to correct human errors and retroactively incorporate missed IoCs in the ground truth.

4.3.2 Limitations and bias

During this process we found various challenges we had to overcome in order to create a reliable and robust ground truth.
The following is a list with the principal challenges encountered:

- **Annotation effort:** The reading, filtering and labeling of each IoCs found inside the post is a very tedious process, since it requires the annotator to be very mindful and consistent in their judgement. Even small lapses in attention can compromise the reliability of the dataset and affect the validity of subsequent analysis

- **Consistency:** The annotator must be consistent across all the posts and try to identify common patterns that can help in the process. It is crucial to understand the context of the post and identify which IoCs are to be considered malicious. In case of multiple annotators, they need to follow clear and shared guidelines to ensure uniformity. Different interpretations of the same IoCs or post could introduce bias or reduce the overall quality of the dataset.

- **Ambiguity:** Posts often present a variety of different IoCs, it is important to evaluate the context they are written in. Some IoCs may appear suspicious but are not inherently malicious, or the contrary. It recurred many times that some

URLs, although not directly malicious, redirected the users to malicious sites. This made us ponder whether or not to consider them as potential threat.

Although these difficulties, the ground truth was created. Post insertion to expand the dataset is very easy as new entries can be appended without altering the existing structure. This ensures scalability of the dataset and allows for continuous integration of new information over time. Moreover, the use of standardized formats like `JSON` guarantees that the newly added posts remain consistent with the previously collected data, reducing the risk of errors during processing and analysis.

4.3.3 IoCs type distribution in our ground truth dataset

Once the filtering and labelling process is done, we could conduct some analysis on our ground truth dataset. An interesting metric is the frequency of our IoCs types.

Figure 4.3 illustrates the distribution of IoCs in the ground truth dataset.

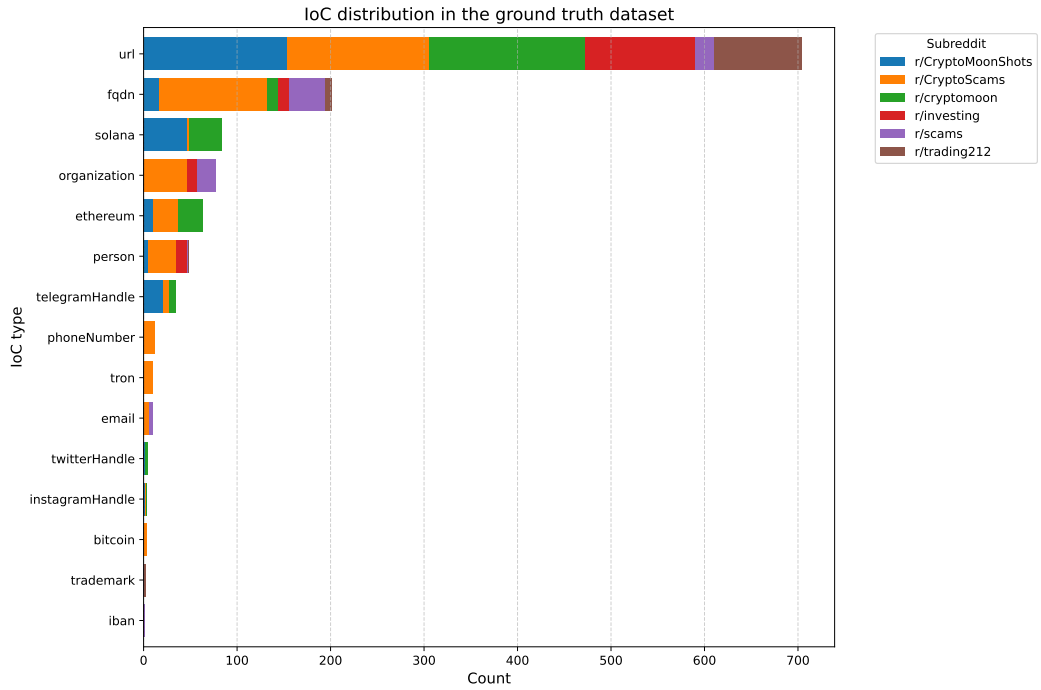

Figure 4.3: Distribution of manually identified Indicators of Compromise across the different subreddits in the ground truth dataset.

The most common IoCs type is the **url**, followed by the **fqdn**. This demonstrate malicious actors often rely on the possibility of bringing users to an external site. We noted in our investigation that many times external sites can look authentic and professional by imitating the layout, brand and design of legit sites. It's easy to see how even more experienced users can fall for this traps.

Another frequent IoCs type is **identity**, which we split in this research into **organization** and **person**. Cybercriminal often make up fake organization to increase the credibility of their scheme and appeal to more people: by presenting

themselves as part of a bigger organization, users are more incline to trust them more and share private information or transferring funds.

A relevant outcome of this research is related to cryptowallet, especially the difference in IoCs found between **Ethereum** and **Bitcoin** wallet addresses. In the crypto ecosystem Bitcoin is by far the dominant coin, followed by Ethereum. For this reason we expected scams to be conducted mainly on bitcoin wallets, but the collected data shows otherwise.
The main reasons Ethereum is more used for scam purposes can be seen in figure 4.4

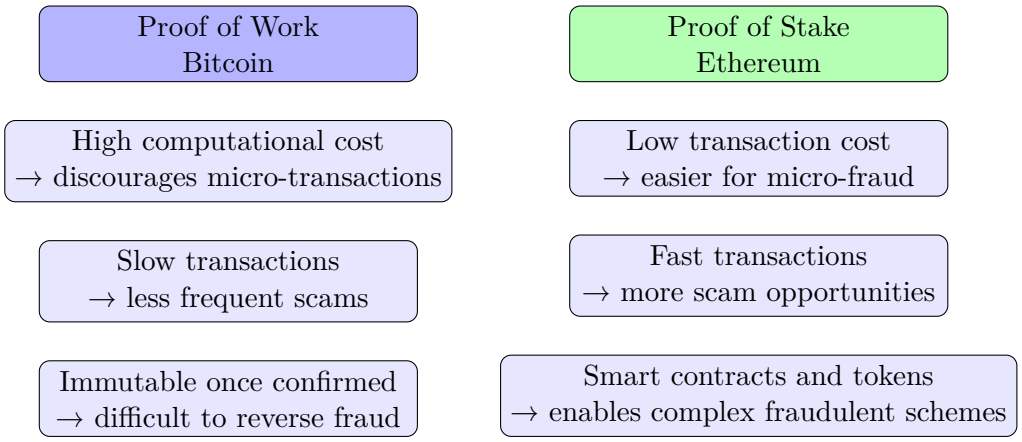

Figure 4.4: Comparison between Proof of Work (Bitcoin) and Proof of Stake (Ethereum), highlighting vulnerability to scams.

The difference between Bitcoin and Ethereum lies in their transaction validation mechanism. Bitcoin uses Proof Of Work (POW) while Ethereum uses Proof Of Stake (POS). Attackers exploits POS by taking advantages of the lower transaction cost and faster confirmation times. This makes it easier to execute micro frauds like those we expect to happens in social networks. The main goal is not to steal a large sum from a single individual, but rather to take moderate amounts from many people. Additionally, Ethereum offers smart contracts and token-based that enables more complex and automated fraudolent schemes, harder to detect compared to Bitcoin's simple transaction structure. These are the reasons we believe contribute to the higher number of scams targeting Ethereum.

Other IoCs types are present in our ground truth dataset in a smaller amount and they often correspond to social media handles or telephone number. Among these, **Telegram** handles are the most common option,which aligns with the fact that Telegram, due to its privacy and anonymity policies, is particularly prone to scams.

From this analysis we could conclude the main focus of attackers is gaining the trust of users and then redirecting them to other sites or platform where they can private interaction can take place. Another interesting point is the use of the so called "Fear Of Missing Out" (FOMO), a psychological tactic frequently employed by attackers to pressure users into making quick decisions. By creating a sense of urgency or portraying opportunities as limited, scammers can manipulate victims

into engaging with fraudulent schemes, often bypassing their usual caution and due diligence.

Chapter 5

Approach

Building upon the Ground Truth dataset established in Chapter 4, this chapter details the experimental framework used to extract Indicators of Compromise (IoCs). We describe the four distinct architectural approaches implemented, the prompt engineering strategies applied to the LLM agents, and the specific metrics selected to evaluate performance. Our goal is to determine what approach works best at extracting and/or labelling the various types of IOC's in the social media posts.
In order to do so, we defined 4 main approaches we used and compared their performances on the ground truth we created.

5.1 Architectures (the 4 approaches)

Table 5.1: Summary and comparison of the four experimental approaches for IoC detection.

Approach	Mechanism	Strengths	Limitations
1. iocsearcher (Extraction only)	Syntactic pattern matching using regular expressions.	Deterministic, efficient, defanging support.	No context awareness, high False Positive rate, no support for abstract entities (Identities).
2. LLM (Extraction + Filtering)	Single stage prompt, doing both extraction and filtering.	Semantic awareness, capable of detecting abstract entities (Identities) and non standard defanging.	Prone to hallucinations, high cognitive load, non deterministic output.
3. LLM (Extraction only)	LLM prompted solely to identify potential IoC	Semantic awareness, captures IoCs described in natural language, useful for discovery.	Noise (filtering is needed), higher token cost compared to Regex.
4. Hybrid (iocsearcher Ext. + LLM Filt.)	Two stage pipeline: Regex extracts IoC, LLM filters them.	Eliminate generative hallucinations (grounding), combines syntactic precision with semantic awareness.	Recall is capped by the Regular expression (if Regex misses it, LLM can not evaluate it).

5.1.1 Approach 1: iocsearcher - extraction

This tool consists in a python library developed by IMDEA software institute[6] and uses regular expression to extract indicator of compromise. This library is very versatile and contains a great variety of supported IOCs. It lacks of context awareness, so it will not be possible to find IOCs such as identities, this refers to the digital or real-world actors associated with a cybersecurity threat or compromise.
From the results of the analysis we will be able to understand if this will compromise too much the ability to recognize a malicious social media post. In the context of this study, this approach serves as the baseline. The objectuve is to measure the performance of syntactic extraction methods. Since this tool lacks semantic reasoning, the output will not be filtered and it will represent the capabilities of traditional extraction methodologies. By analyzing the performance of this tool, we will be able to quantify the "added value" of an LLM base approach. As mentioned before, this tool support some popular defanging operations, we will analysis how well this behaves with unstructured social media text. We expect to see an high rate of false positive due to the limitations of this tool

5.1.2 Approach 2: LLM - extraction and filtering

This second approach represent a paradigm shift, instead of relying solely on regular expression for extracting IoC, we will use the advantage of an LLM and if the semantic understanding of the context can significantly improve performances. We'll focus exactly on this aspect: semantic extraction (LLM) vs syntactic (regexp). We will be able to see how a Large language model compares in identification of IOCs. We chose GPT-5.1 as it was the latest openAI model in the moment this experiment started. This approach focus on analyzing how a large language model behaves in extraction and filtering of different kinds of IoCs. The label can either be "M" if the IoC is suspected to be malicious, or "B" otherwise. The main challenge when using a LLM inside a software project resides in the **prompt engineering**. The prompt must be carefully and meticulously crafted to ensure consistent and reliable results. We tried using different prompting techniques to maximise the metrics. For this approach, we used a single unified prompt: the LLM will need to both extract and filter the IoC, different techniques will be used based on the observed performances. The advantages we expect from this approach are multiples: as mentioned before, the LLM will be able to differentiate between malicious IoCs and not malicious ones, this will result in a significant drop in false positives. Furthermore, the LLM is better at handling defanging and abstract entities (such as organizations and threat actors).

To conclude, Particular attention must be paid to hallucination, where the LLM could find IoC that are not in the text or generating ones. Specific instruction for the output must be given to the LLM to automate the analysis part.

5.1.3 Approach 3: LLM - extraction

We chose to explore this approach to evaluate how an LLM behaves when the task is easier: IOC only extraction, without the need to filter them between "Threat" or "nonThreat". This approach has a dual objective, it tells us how the LLM performs with an only extraction purpose, as well as giving as a direct comparison with iocsearcher, since the IoCs found will be treated as all malicious in the analysis step. We expect an increase in false positives, but we'll be able to compare the defanging abilities of the LLM with the ones of iocsearcher. In this approach we try to lower the cognitive load of the model: instead of giving two different tasks (extraction and filtering), we only give the first, and see how well it compares to regular expressions.

5.1.4 Approach 4: iocsearcher - extraction + LLM - filtering

This is an hybrid approach that tries to combine the syntactic extraction of the regular expressions with the semantic understanding needed for filtering of the LLM. It behaves like a pipeline: the regular expression acts as the initial collector, it extracts everything that remotely looks like an IoC and ensure deterministic precision thanks to its pattern matching. On the other hand, the LLM is not tasked to find IoCs anymore but its sole purpose will be to filter them, this can greatly reduce the cognitive load and can focus on the reasoning part.

One of the main advantages of this approach is the elimination of generative hallucination, other than the increase in efficiency.

5.1.5 Prompt engineering techniques

Beyond the definitions of these 4 main approaches, we will try to analyze how the LLM behaves with different prompt techniques/architectures.

- **Ablation study:** we will be able to see how important different parts of the prompt are for the performances in this specific task. This process involves removing specific components of the prompt to observe the impact it has on the final metrics.
- **Chain Of Thought (CoT) prompting:** We will try this prompting technique to mitigate the risk of impulsive/lazy classification by the LLM. By forcing the LLM to actively reason step by step, it can lead to significant performance increase, other than letting us see how the model reasons in certain situations.

To conclude, we will not only stop at seeing how the LLM compares to regular expressions, but we will try to use different innovative prompting techniques and see how efficient they are on this specific task.

5.2 Prompt Engineering Strategy

5.2.1 Prompt Structure and Components

Final Optimized Prompt Structure

Prompt Content	*Component*
`You are a cybersecurity expert specialized in OSINT and Threat Hunting. Your expertise lies in identifying malicious Indicators of Compromise (IoCs) hidden within unstructured social media text. You are highly skilled at spotting obfuscated (defanged) IoCs and distinguishing between harmless references and actual threats.`	**Role Specification** *(Domain Persona & Context)*
`IoCs can be malicious or non-malicious: extract any item that fits the defined IoC categories, regardless of whether it is harmful or not. Extract IoCs from both the title and the text of the posts. Include all URLs, even if they link to images, previews, or files.`	**Task Definition** *(Scope & Constraints)*

`The IoC categories are:`
`- url: a complete web address, with path...`
`- fqdn: any complete domain name... Do NOT extract domains inside full URLs.`
`- email: a specific email address.`
`- telegramHandle: a unique Telegram username.`
`- instagramHandle: a specific Instagram username.`
`- phoneNumber: numeric sequence...`
`- ethereum: any Ethereum wallet addresses.`
`- tron: addresses starting with 'T'...`
`- solana: addresses with 32-44 base58 chars.`
`- watermark: unique identifier embedded...`
	IoC Definitions *(Semantic Disambiguation)*

`Examples (covering multiple types):`
`1. { "title": "BCH Miner", "text": "Hi..." }`
`IOC found: "[(fqdn,bchmimer.info)]"`
`2. { "title": "Treasure NFT", "text": "Check..." }`
`IOC found: "[(url,https://treasurenft.xyz/#/)]"`
`3. { "title": "Image preview", "text": "Look..." }`
`IOC found: "[(url,https://preview.redd.it/...)]"`
	Few-Shot Learning *(In-context Examples)*

`Put your IoCs in this exact format:`
`"[(type,ioc),(type,ioc),...]"`
`If no IoC is found, output "[]".`
	Output Format *(Parsing Instructions)*

`Now analyze the following posts in the same way. Output only the string object in the requested format.`	**Final Trigger** *(Execution Command)*

Figure 5.1: Breakdown of the final optimized prompt. (GPT for extraction only)

A structured prompt was utilized for the interaction with the model. As illustrated in 5.1, we can clearly see how the final optimized prompt can be divided into blocks called **components**, where each block target a specific cognitive function of the model. the structure is divided into 6 main blocks:

- **Role specification:** the prompt begins by assigning a specific persona to the LLM. This is because **Role prompting** (the name of this technique) is crucial for **Domain Adaptation**. It sets the tone and seriousness of the task the model will be doing. We prime the LLM with a cybersecurity context, thus enhancing the generation towards technical accuracy, reducing the possibility of the model to interpret ambiguous terms in a non technical way.
- **Task definition:** this section serves to set the boundaries of the analysis: it gives direction to the model on how to extract an IoC, regardless of whether it is harmful or not, giving instruction to what part of the post to analyze (the title and the body).
- **IoC definition:** given the nature of social media posts, and to uniform the analysis, we chose to include in the prompt the IoC list and the definition for each one of them. For instance, a clear distinction is given in our study between **URLs** and **fqdn**, to prevent redundancy. This is done in order to lower the hallucination rate of the LLM and give him a leeway on what IoC the model should focus on.
- **Few-Shot Learning:** a critical component of the prompt resides in few shot learning. This technique has been proven to drastically improve performances in complex task, giving the model some example where it can observe the task that it's supposed to do. This allows the model to "learn" the logic, without the need for weight updates or fine tuning.
- **Output Format:** a strict output schema is needed for our anaklysis: the model is instructed to give us a string with a proper json format, that will immediately placed in a json file, thus allowing us to conduct further analysis.
- **Final Trigger:** after the output format description, the final trigger is executed. We take advantage of the **Recency bias** effect observed in LLMs, where the instructions located at the end of the prompt text are followed more strongly than those in the middle of it.

The final prompt structure is the result of an iterative trials and error process, where small modifications where made in order to increase the model performances. This is the final version of it, although it can be modified to further improve it, or to better adapt it to the task that has to be done.

5.2.2 Few Shot Learning and Output standardization

A major challenge when using a LLM in a software enviroment, is its non deterministic behavior and output standardization. To address this issue, we implemented a strategy

combining few shot learning [23], with strict output format rules.

A special focus is needed when talking about few-shot learning (FSL) techniques , as this has been proven to greatly improve the model performances. Standard prompting techniques are called "zero shot" since no examples are provided to the model. This is not optimal since the LLM has to rely on our explanation on how to perform the task and how to format the output. During the creation of our prompt we embedded concrete examples, a set of input/output pairs. This was done in order to:

- **Give Implicit Instruction:** the examples provided show how to handle edge cases. For example, we could provide examples of defanged IoCs to instruct the model on how to retrieve it, all without giving explicit descriptions.
- **Providing Output Format:** it is crucial to conduct and automate the analysis part. By providing examples directly in the few-shot learning part, the model has a visual representation on how to output the results, this allows us to align the model the output format to our requirements. The model also learns to suppress its chatty nature and return only the raw data.

We will be able to measure the impact of the FSL techniques during the ablation study to see if it's effective in this specific task.

When using large language models in a software development environment, one of the biggest challenges to overcome is the non deterministic nature of its output. The chatty nature of LLM, combined with hallucinations can make it hard to retrieve the information needed. In order to stop this behavior, specific output instruction must be provided to the model, we obtained this by describing how the output should be provided (in combination with the examples provided in the few-shot learning component) and by instructing the model to only output the string object in the requested format.

5.3 Evaluation metrics

5.3.1 Definition of confusion matrix elements

To understand the performances of our various approaches we defined some metrics that could help us understand the impact and strengths of them. All the metrics are computed by comparing the extraction and filtering done by our architectures with the ones found by the manual labeling of out ground truth. For our evaluation, we defined a confusion matrix specific for our domain. Since the task is binary classification, we used **True Positives (TP)** and **True Negative (TN)** for correctly classified samples, while **False Positives (FP)** and **False Negatives (FN)** are the incorrectly classified samples.

Table 5.2 gives us an overview on the metrics used and their meaning.

Table 5.2: Definition of Confusion Matrix elements adapted our domain

Metric	Definition	Meaning
TP (True Positive)	A malicious IoC correctly extracted that strictly matches an entry labeled as malicious in the Ground Truth dataset.	**Successful Detection** The system correctly identified a threat.
FP (False Positive)	An entity extracted by the model that is **not** present in the Ground Truth. This includes: 1. *Filtering Errors:* Benign indicators (e.g., non threat URLs,) flagged as malicious. 2. *Hallucinations:* Entities generated by the LLM that do not exist in the posts text.	**False Alarm** Contributes to analyst alert fatigue and wastes resources.
FN (False Negative)	A malicious indicator present in the Ground Truth that the approach failed to extract or filtered out erroneously.	**Missed Threat** A security blind spot; the most critical error in CTI.

The absence of **True negatives (TN)** is due to the domain nature of this extraction task. In this context, a True Negative is every word that does not represent any threat, thus it is theoretically infinite and make it not possible to use it as a metric. Including TN would artificially inflate our metrics, rendering them meaningless.

To ensure a fair evaluation of the extraction capabilities of the various approaches, a naive exact string comparison could not be used. Some post processing of the IoC has been done in order to overcome the heterogeneous nature of social media text, since IoCs often comes in various formats.

Using exact string matching would penalize our approaches merely due to format differences. Therefore, we applied normalization to both predicted entities and our ground truth instances before computing any metrics. These normalization techniques include:

- **Case insensitivity:** All extracted indicators are converted to lowercase, this helps to reduce the inconsistencies of user generated content (Malicioussite.com vs malicioussite.com)
- **String cleaning:** Useless symbols were removed by the strings (this include "@" at the start of telegram/instagram handles, eventual spaces, or "/" at the end of URLs). Phone number were also normalized, removing eventual brackets

and spaces in between the letters.

- **Type strictness:** It is not enough For the IoC value to match with the ground truth dataset, the IoC type must match as well. An entity extracted as an *URL* containing an *IP adsress* is not considered a match.

Although this metrics can already provide valuable insights, to have a more meaningful understanding of these approaches, we adopted standard information retrieval metrics. Based on the confusion matrix elements defined above, we compute the following metrics:

$$\textbf{Precision} = \frac{TP}{TP + FP} \tag{5.1}$$

Precision is a measure of the approach's **reliability**. In our context (threat intelligence), it answers the question: *"When an indicator is flagged as malicious by the system, how often is it correct?"*. Lower precision means an higher rate of false alarm (False positives) which results in a higher extraction rate, this could be useful in a context where not loosing any potential IoC is crucial for the task.

$$\textbf{Recall} = \frac{TP}{TP + FN} \tag{5.2}$$

Recall is a measure of the system **coverage**. In our context (threat intelligence) it answers the question: *"Out of all the actual malicious indicators present in the text, how many did the model find?"*. Lower recall means the approach is missing real threats (False Negatives), potentially leaving the system vulnerable to attacks.

$$\textbf{F1} = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{5.3}$$

F1 score is the mean between Precision and Recall. There is often a trade off between the two metrics, with F1 it is possible to have a global view on the overall effectiveness of the extraction pipeline. It is especially useful in this domain, where the cost of both False Positives and False Negatives is significant.

$$\textbf{FPR} = \frac{FP}{FP + TN} \tag{5.4}$$

False positive Rate (FPR) is a measure of the system's tendency to trigger false alarms, thus potentially leading to alert fatigue. An high FPR means low contextual awareness, the system can not differentiate between benign indicators and actual threats. This metrics was used in analysis with a different granularity in order to give us another point of view on the strength of the approaches.

In conclusion, we relied on these metrics to provide a rigourous quantitative framework for the evaluation of our experiments, allowing us to conduct analysis and efficiently evaluate what approach best balances the goals of minimizing security blind spots (high recall) and reducing over extraction (high precision).

Chapter 6

Evaluation and Results

The objective of this chapter is to compare our main 4 approaches and understand what makes them different. In this first part we'll analyze our results comparing them with our ground truth, later we will simulate a real world scenario, using the approach that works best to analyze the dataset we collected.

6.1 Overall performance comparison

After defining the methodologies, the experimental evaluation was conducted on the ground truth dataset. Table 6.1 presents a comprehensive summary of the performance metrics for each approach. Each row represents the proposed methodology, while the columns are its metrics as defined in the previous chapters. It is a quick comparison of performances based on pure IoC extraction (and filtering, when possible) task.

Extraction	Filtering	TP	FP	FN	Precision	Recall	F1 Score
iocsearcher	-	645	514	184	0.56	**0.78**	0.65
LLM	LLM	504	128	331	0.80	0.60	**0.69**
LLM	-	596	495	241	0.55	0.71	0.62
iocsearcher	LLM	438	65	391	**0.87**	0.53	0.66

Table 6.1: Comparison of different approaches evaluated on 600 samples for IoC extraction (and filtering when possible) task.

This table offers interesting insights about strengths of various approaches. The second approach, LLM used for both extraction and filtering emerges as the most effective approach, with an f1-score of **0.77** and it's the most balanced approach between precision and recall. Differently from other approaches that maximize coverage, the LLM is able to efficiently filter false positives, thanks to its superior semantic understanding of the context: this enable the LLM to reduce noise, other than being able to localise obfuscated IoCs.

It is interesting to compare the results with those obtained using the LLM solely for extraction. The third approaches presents a bigger number of false positives compared to the second, this is in part due to the metrics evakuation approach we

used, in part due to the allucination of the LLM: adding the filtering instruction may have helped the LLM to be more careful and conduct a better analysis on the IoC found in the texts.

Iocsearcher used for filtering has the highest recall, **0.78**. This results is intrinsic to the nature of deterministic pattern matching: regular expressions extract every string that syntactically conforms to the definition of an ioc. Unlike LLM-based approaches which may suffer from semantic filtering, the regex approach follows strict syntatic rules and doesn't discard anything: results are optimized, hence the highest recall.It is crucial to highlight a fundamental structural difference between the evaluated approaches. iocsearcher is engineered exclusively for raw extraction: its role is to identify syntactical patterns without semantic judgment. Therefore, its lower Precision (**0.56**) should not be interpreted as a failure, but rather as a characteristic of a 'pre-filtering' stage. In contrast, the LLM-based approaches incorporate an implicit or explicit semantic filtering layer. Consequently, comparing the first and second approach's metrics is a comparison between a data collection tool and a decision-making system.

Finally, the hybrid approach (iocsearcher filtering + LLM labeling) stands out for achieving the highest Precision (**0.87**). This configuration operates as a strict sequential validation pipeline: the regex acts as a candidate generator based on syntax, while the LLM acts as a semantic verifier. This 'double-check' mechanism effectively eliminates almost all False Positives, as an entity must satisfy both rigid syntactical rules and contextual relevance to be retained. However, this high reliability comes at the cost of Recall (**0.53**), as the strict filtering pipeline inevitably discards ambiguous or obfuscated threats that do not perfectly align with both criteria. This makes the hybrid approach ideal for scenarios where False Positives are unacceptable (e.g., automated blocking), but less suitable for deep forensic analysis. Considering the **context** is very important in a cybersecurity context. If finding any potential IoC is crucial, opting for a maximum recall should be the way, whereas in a context where that is not a problem and we want a general filter against IoCs, opting for a more loose approach could be the best choice.

An advantage of the utilization of the LLMs is the fact that the prompt can be varied in order for him to act more **conservative** if the identification of IoC is crucial, or more permissive, if the system is being employed in an enviroment where alert fatigue is a problem. To

6.1.1 IoC density analysis

In this section we will analyze the IoC density distribution per post in our ground truth and for each of the four proposed approaches, distinguishing between malicious and not malicious ones. Figure 6.1 shows the average number of IoC identified per post, providing us with useful insights on the impact of semantic filtering.

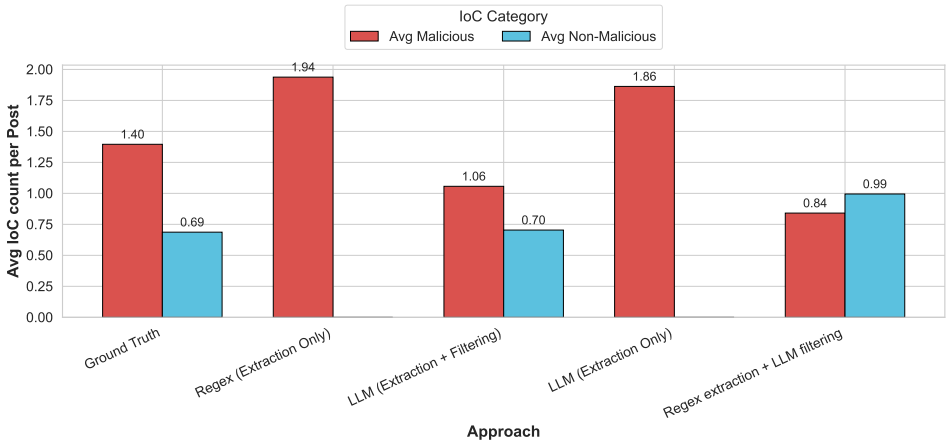

Figure 6.1: grouped bar chart showing per post IoC density.

We can clearly observe how the average number of IoCs per post in the ground truth is unbalanced, with an average of **1.40** malicious IoCs against an average **0.69** not malicious ones.

The distribution that resembles the most the ground truth is the LLM used for both extraction and filtering, with a striking **0.70** average density of non malicious IoC, very close to the ground truth one. The lower malicious IoC density does not inherently signal worse performances, but it can represent a good balance for menace visibility, without saturating the system.

The approach employing regular expression for extraction and LLM for filtering appears to be more conservative, reducing by far the malicious average. This could be explained by the nature of regular expression extraction mechanisms, which fails to extract more complex or obfuscated IoCs, thus never allowing the LLM to filter them. The LLM for both extraction and filtering does a better job since it allows, on average, to detect more menaces, while doing a good job at keeping the noise under control.

Lastly, approaches based on solely extraction of course do not present a filter, hence the absence of a dedicated bar for not malicious indicators. The graphs does a good job at showing how noise reduction is crucial: a step in recognizing which indicators are actually malicious and which are not is important to avoid **alert fatigue**. By reducing the density of non malicious IoCs, the system can help analyst to focus on real threats, optimizing the response time.

6.1.2 Post level detection

Table 6.1 focuses more on the extraction of IoC, the evaluation at post level can also provide us useful insights, since in real world applications the main concern is also related to the number of posts that are flagged with having potential malicious content in them. In this section we will discuss different analysis that can help us understand better the strenghts of each approach. Table 6.2 gives us the performances of each approach for a post level detection analysis. A True Positive **TP** is considered so if a post contains at least one malicious IoCs that is also flagged as malicious by

Extraction	Filtering	TP	FP	FN	TN	Precision	Recall	F1 score	FPR
iocsearcher	-	428	172	0	0	0.71	**1.00**	0.83	1.00
LLM	LLM	336	12	92	160	**0.97**	0.79	0.87	0.07
LLM	-	401	150	27	22	0.73	0.94	0.82	0.87
iocsearcher	LLM	300	11	128	161	0.96	0.70	0.81	**0.06**

Table 6.2: post level analysis comparison for the four approaches

the system. On the other hand, a False Positive **FP** is considered so if a benign post (which does not contain IoCs that are considered malicious in the ground truth) is flagged as malicious by the system (contains at least one malicious IoC). This different point of view is important since it provides us with a more macroscopic point of view, a real world application where the focus is mainly on the malicious post identification rather than the single indicators.

From the Table we can immediately recognize a perfect recall (**1.00**) for the first approach, this is of course due to the pattern matching nature of the iocsearcher library. This high recall is a double edged sword: while it can correctly identify every kind of artifacts that can represent a menace, it also comes with an high False Positive Ratio, giving us 172 false alarms.

Similarly to Table 6.1, the LLM for both extraction and filtering approach appears to be the most balanced, with an F1 score of **0.87**, superior to any of the other approaches. The high precision (**0.97**) means the system is capable of efficiently distinguish between actual threats and artifacts that only shares the same syntax, thus translating in a drastic reduction in alert fatigue.

Interesting results can be drawn when comparing the LLM for only extraction approach to iocsearcher. The higher precision of the third approach means the LLM still acts more conservatively, thanks to better capabilities to ignore syntactic noise, which get inevitably captured with a regular expression approach.

The mixed approach (regular expression for extraction + LLM for filtering) has the best False positive Ratio: this means the LLM filtering actually works and can help reducing **alert fatigue**. However, this approach has the lowest recall (**0.70**) which, similarly to the explanation we gave in the previous section,can be attributed to the regular expression, if it fails to extract a more complex or obfuscated indicator, the LLM can not filter it.

To conclude, the true purpose of this analysis, more focused on real world application, is centered on **alert fatigue**. In a context where it's crucial to detect any malicious indicator, a regular expression approach could work the best, while an LLM approach is better when the minimization of False Positives is essential not to saturate human resources.

6.2 Performance breakdown by IoC category

While Table 6.1 may gives ua a general overview of the performances of the various approaches, it's still interesting to notice how they perform individually

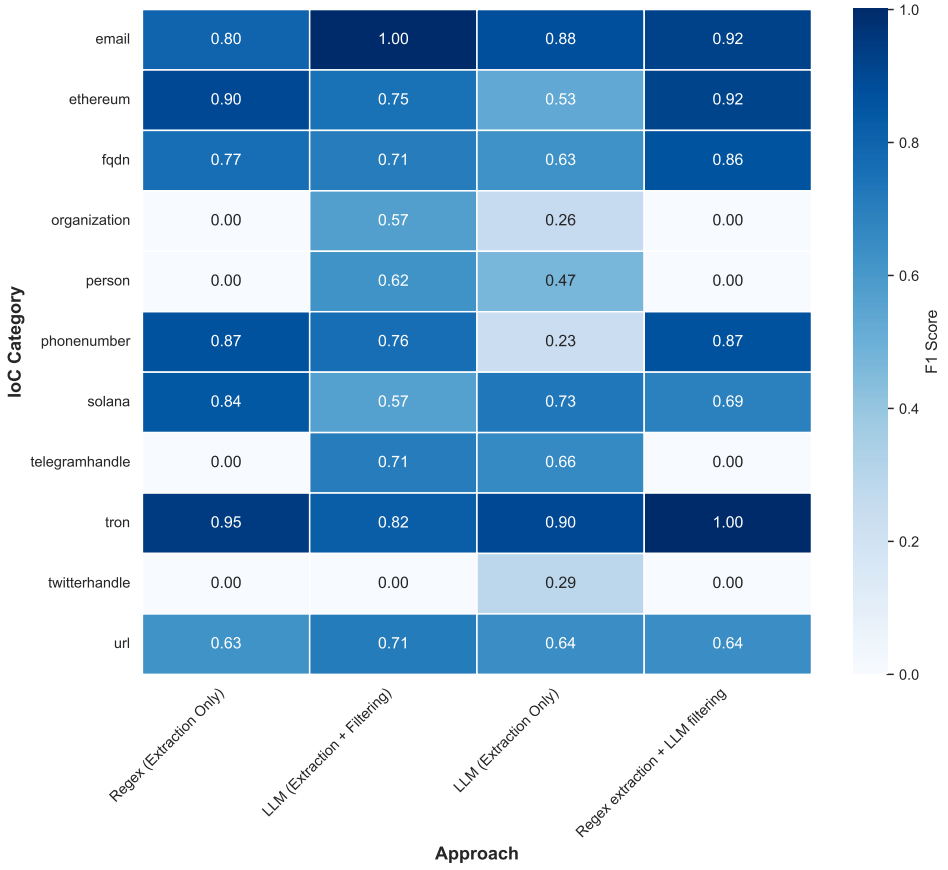

Figure 6.2: Heatmap highlighting the density of F1-Scores across different approaches. Darker colors indicate better performance, while lighter cells highlight weaknesses.

Heatmap showed in Figure 6.2 gives us great overview on the F1-score distribution for the various IoC types, providing meaningful information on what approach works best in different scenarios. The first column, **Regex (Extraction only)** shows us the limitations of simple pattern matching: this approach performs good on indicators that have a rigid syntactis structure (tron,ethereum,...), while failing completely on semantic entities. Regarding categories like organization, person, telegramHandle or twitterHandle, iocsearcher performs poorly, since often they are surrounded by a context. The second column, **LLM (extraction+filtering)** demonstrate the model's semantic capabilities. The model performs well in categories where the semantic meaning of the menace is important: performances in the recognition of organization and persons Indicators are what makes this approach overall better. Same reasoning applies to telegram/twitter handles, people often write their social username, followed or not by a @, this can be only detected by a semantic analysis of the context. The URL category seems to have a pretty stable performance across

all the proposed approaches, where the LLM for Extraction and filtering approach seems to work the best. We can confidently say this is because of a better ability in detecting obfuscated indicators. The LLM based approaches seems to struggle with cryptowallet addresses and phone numbers, probably because it gets confused when working with long hexadecimal strings or varying length phone numbers, indicating the need for specific instruction on how to handle these indicators in the prompt. The fourth approach seems to work pretty good with a good range of indicators, although suffering bottleneck from the extraction phase.

Heatmap gives us good insights on what approach works best for different indicators. Figure 6.3 presents a bar chart that provides us a similar comparison of the F1 score across all different types of indicators.

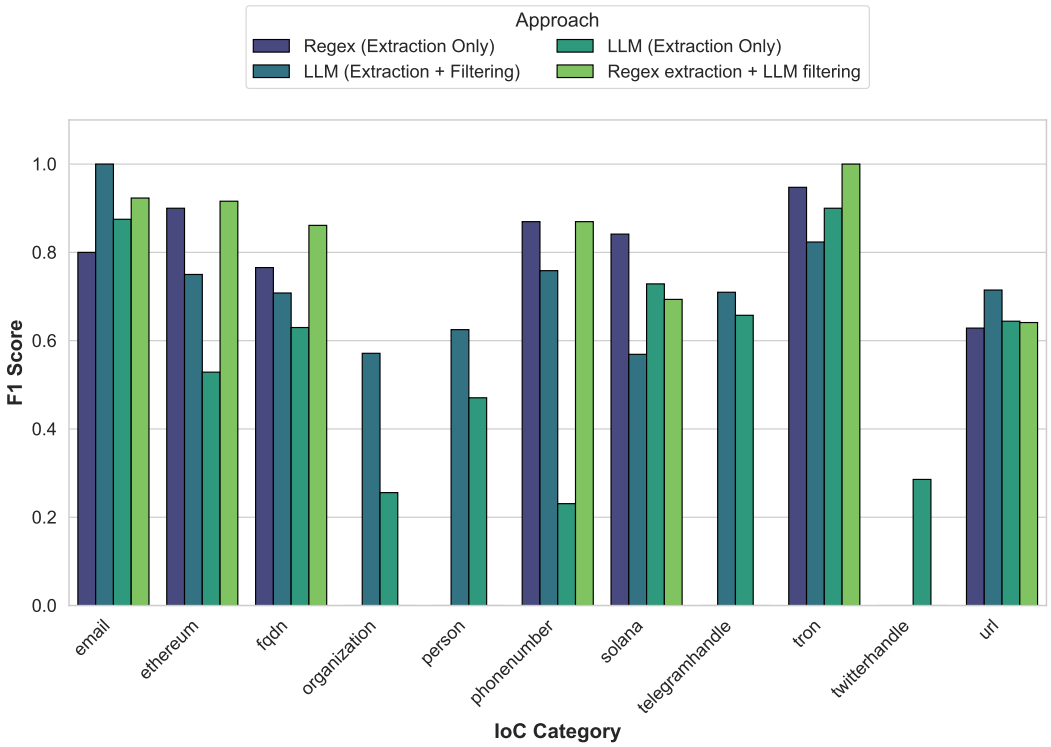

Figure 6.3: Comparative analysis of F1-Scores across different approaches breakdown by IoC category. The proposed hybrid approach and the LLM filtering approach are compared against the regex baseline.

Thanks to this different visualization, we can draw different conclusion than the ones we observe with the heatmap.

Firstly, it's interesting to visualize the **filtering gap**, when the LLM is prompted for filtering too, it mantains or improve the score of its counterpart with only extraction, mostly noticeable in the purely semantic entities, where the filtering part can potentially help the LLM to eliminate noise, thus giving us better performances.

Categories with more rigid syntax show taller and more uniform distributions, whereas categories based on semantic recognition vary wildly. This further prove the strength of the LLM approach and it's a key points of our analysis.

6.3 Comparative analysis: LLM based approach vs reg-exp based approach

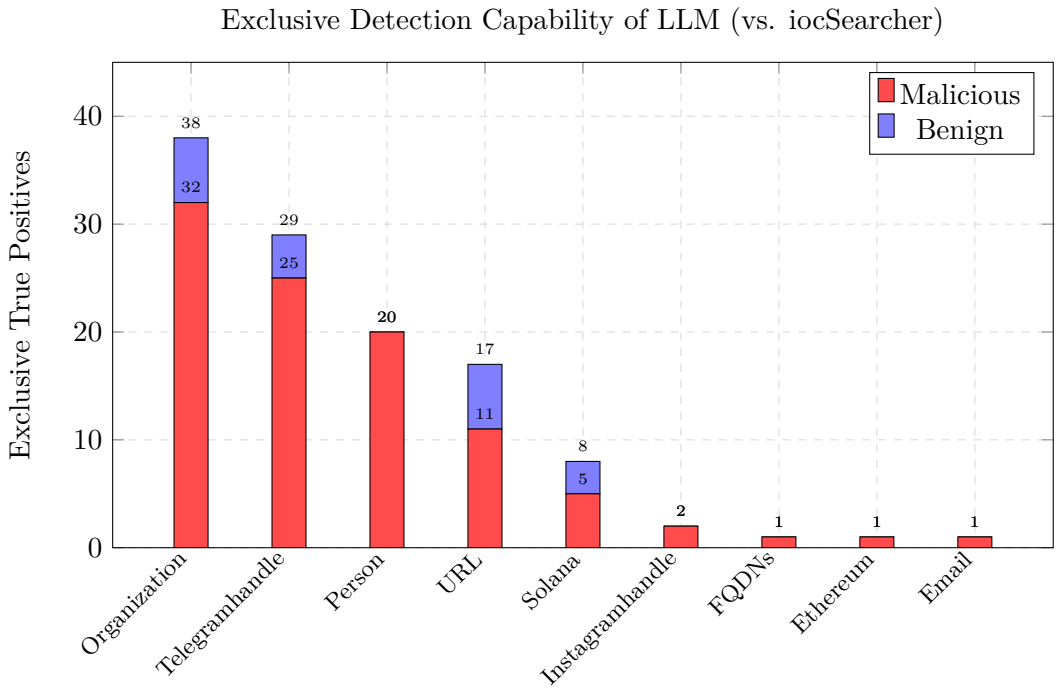

Figure 6.4: Breakdown of IoCs detected only by the LLM (extraction + filtering) compared to the iocSearcher approach, bars have been split by Malicious (red) and Benign (blue). The model's superiority in semantic categories such as Organization, Telegramhandle, and Person is clearly visible.

This graph gives us a the net value of the LLM , which is the indicators correctly identified by it and not by a regular expression tool. As expected, the LLM approach works the best for purely semantic entities: Organization and person indicators are important in a cybersecurity context since they allow us to attribute malicious activities to specific threat actors, whose name can help us recognize bigger fraud schemes. Together, *organization* and *person* represent **58** exclusive IoC detected by the LLM that the regular expression could not extract (due to limitations in the architecture making it impossible to create a general rule). We can see some examples:

- Post ID **1kbavq5** contains a user trying to expose a big fraud scheme involving two organizations called **M3 DAO** and **GC Capital** and the name of a person called **Gerald Chee**. These identities are the core of the scam operation and the LLM correctly extracts them. The regular expression tool fails on those indicators and,although the post also contains *URLs* and *FQDNss* correctly identified, this approach fails to capture the real essence of the post. Independent investigation confirmed the malicious nature of the organization cited in the post text: **M3 DAO** has been identified to be a large Ponzi scheme connected to the MarsVerse platform, counting over 40000 members in its telegram groups[24].

The article is dated at September 2025, while the reddit post was published on 30th of April, 2025. Semantic extraction from social media post could serve as early **warning signs**, helping analysts to quickly recognize fraud campaigns before they are confirmed by traditional investigation tools.

- Another example that supports the importance of extracting identities can be shown in post ID **1jmq8xq**. In this post, an user is explaining how he thought his uncle asked him money for a new project he was working on. As it turns out, the uncle fell for a crypto scam, and the user citing names like **Alpha Stock Investment Traininng Center** and **CoinBridge**. Both organization were later confirmed to be fraudolent, CoinBridge was flagged as an AI trading fraud by the Washington State Department of Financial Institutions [25], while another independent investigation exposed the connections between the two entities [26]. Once again, social media alerts went off much sooner with respect to investigation campaigns.

These examples show the importance of collecting identities such as organization and person names, because these indicators often hide bigger schemes that can not be identified by other artifacts.

The purely LLM approach works best also at identifying telegram handles, this is due to users often just writing their names, leaving no syntactic anchor point for a regular expression to catch. Here is some examples of how threat actors can use telegram handles in a scam context:

Table 6.3: Comparative analysis of Telegram handles detection: LLM (extraction + filtering) vs regular expression for extraction (iocsearcher)

ID	Text Snippet	Issue	LLM Result	iocSearcher
1	...Investigate the Telegram Scammer (**@McJohnFranklin**)...	Plaintext Label	@McJohnFranklin	*None*
2	...Telegram account with the username **"cryptozxc18"**...	**No Syntax**	cryptozxc18	*None*
3	...Telegram: **@ meme-coinjack**...	**Broken Syntax**	memecoinjack	*None*

Note: The False Negatives in iocSearcher occur because its extraction logic is strictly limited to Telegram handles embedded within full URLs (e.g., t.me/username).

The same explanation can be given for the presence of the two not identified instagram handles: without the full URLs (e.g., `instagram.com/username`), regular expression based tools have no anchor point to efficiently extract these IoCs.

Crypto wallet addresses are another interesting case study that shows differences between the two approaches.

Post ID **1ndqbf8** is a good example of *shilling*. An user was trying to promote an early stage project and, while doing so, he passed the Certificate Authority (CA), which normally is done as an act of transparency and giving a false sense of security to the people reading the post. While doing so, maybe as an error, he forgot to add the last byte. Even though it should not be considered an obfuscation attempt, the LLM still succeeded in extracting the address, while the regular expression, relying on rigid structures, fails. This shows once again limitations in approach depending too much on rigid structure extraction.

Solana addresses are another case, since they were manually reviewed and were all valid. The missing detection of them from *iocsearcher* could be hidden in its source code and maybe some cases are not detected.

The URLs and FQDNss indicators are mainly obfuscated indicators, some of which will be discussed in the context of defanging techniques (*url*: 6.4.1, *email*: 6.4.1, *FQDNs*: 6.4.1): these cases shows clear examples of the capabilities of the LLM to recover obfuscated indicators, which comes with limitations when using regular expression tool.

6.4 Prompt engineering impact analysis (ablation study)

During the analysis, we observed the importance of the prompt structure, demonstrating that specific components,such as explicit semantic definitions and few-shot examples, are not merely supplementary but essential for maintaining high precision and preventing the model from reverting to generic, error-prone extraction patterns. We identify 3 main components in our prompt and tried to run it on the whole dataset, recomputing the metrics for every combinations. This study was conducted on the third approach: LLM used for only extraction

Role Def.	IoC Spec.	Few Shot	TP	FP	FN	Precision	Recall	F1 Score
✓	✓	✓	596	495	241	0.55	0.71	**0.62**
x	✓	✓	603	564	234	0.52	0.62	0.57
✓	x	✓	662	1179	175	0.36	0.79	0.49
✓	✓	x	525	785	312	0.4	0.63	0.49
✓	x	x	611	1489	226	0.29	0.73	0.42
x	✓	x	537	716	300	0.43	0.64	0.51
x	x	✓	632	987	205	0.39	0.76	0.51
x	x	x	558	1615	279	0.26	0.67	0.37

Table 6.4: Ablation study results evaluated on 600 samples - LLM for extraction

As expected, we notice the first approach, using all components, gives us the best results: highest precision and F1-score. This demonstrate the components we chose are not redundant and have a role in helping the LLM understand the tasks that has to be done. Similar to table 6.1, the key in this metrics relies in the F1-score: the equilibrum of precision and recall is important in this tasks as it shows a good level of understanding of the task, while keeping a good disctinction between noise and actual threats

Paying close attention to the False Positives, we can notice a spike when we don't include the **IoC specification** (particularly noticeable in the third and fourth row), and the precision drop as a consequence of this. This tells us that LLM struggle to interpret what to target without an explicit description, resulting in an over extraction or hallucination.

Another interesting point to make is about the importance of few shots: even with the other two components, without some concrete example the LLM drops in performance. The example have a double role: they show how the LLM should conduct the analysis by explaining which IoC are extracted from the posts, while enforcing the output specification of the model. With few shots, the model produces less hallucination related to the output structure

Out of all the possible combinations, the one not including the role definition is the one that experiences the smallest drop in performance. Although expected, this confirms us that giving the LLM a role (Cyber threat analyst) can help set up the context but it's not enough if some more information is not given.

Finally, last row gives us an important information: when the prompt is very loose, without providing context, being specific about the IoC types it can find or without providing some examples, the results way worse than just using a regular expression tool. An "out of the box" LLM is not ready to tackle cyber security tasks without an adequate prompt engineering. The results are significant, the model triples the amount of False positives. This is a crucial result of our study: if an LLM perform worse than a regular expression, which require way less computational power, it is useless. The importance of prompt engineering should not be understimated as it can drastically improve the performance of an LLM

Table 6.4 gives us insight on the true nature of LLM, as they are generative models, not discriminative ones. If no strict information is given (removing the components), the model behaves in a greedier way: everything that vaguely resemble an IT artifact (URL,IP,FQDNs...) or other IoC is extracted. Prompt engineering components therefore acts as suppressive filters. Interesting enough, in the third row (the one without IoC specification) we notice that while the false positive increase to a large number, the false negatives actually reduce: this is coherent with what has just been said, without explicit semantic definitions the LLM cast a wider net, catching almost all real threats (low FN) but drowns them in a sea of noise (high FP), compromising the output to be basically useless without human verification.

6.4.1 Defanging and Hallucination

To qualitatively address the robustness of the approaches, it is important to dive into the ground truth and observe specific instances where an approach fails and other succeed. Here are some example of defanging:

- In post ID **1ok2lhn** of our ground truth dataset, the user suspected a potential scam and obfuscated the email, writing:

> *"...sent by OVG MEDIA LLC (email:* ***ovgbusinessmgmt //// @gmail.com****)"*

this led to the regular expression based tool to fail in recognizing the IoC, while the LLM approach, thanks to its semantic understanding, correctly classified the email as malicious.

- Another example of defanging can be seen in the entry with post ID **1ot72k3** of the ground truth:

  > *"Public dashboards and metrics are also available at the Grafana node monitor:* ***http://57 .129 .148 .132:3001.****"*

  In the instance the user (a malicious actor) promoted a fake blockchain project and redirected the users to a dashboard hosted on a raw IP address. To bypass any automatic Reddit filtering, whitespaces were employed, thus leading to a failure of the regular expression tool, while the LLM was able to reconstruct and correctly extract the original URL.

- We find another example of whitespace defanging in the title of post ID **1jslcas**:

  > *"Ever heard of* ***SnapeDex. com****?"*

  Here the user is not inherently a malicious actor, he is looking for feedback of a suspicious looking site, hence the use of whitespace defanging in order to avoid any user to accidentally click on the link.

The LLM ability to reconstruct defanged IoC is crucial in social media analysis. Regular expressions are not enough since actors know the popular filtering techniques and can always come up with innovative methods to avoid them.

The main downside when using LLM is the possibility of hallucinations. They occur when the model generates content that is semantically possible but not grounded in the sourced text. Generative AI predicts the most probable next token, which can lead to creation of false information. The following are some examples we encountered while analyzing the ground truth dataset:

- In the entry with post ID **1jv4jh4**, a user was exposing a potential scam site. While the text explicitly mentioned the fraudulent domain `http://bicoi.com`, it only referred to the source platform by its brand name, "CoinBase", without providing a URL. Despite the actual FQDNs of the CoinBase sitenot being present in the text, the LLM inferred two IoCs:

  > *"[(url,http://bicoi.com),(FQDNs,coinbase.com)]"*

  This is a clear example of the model's **Generative inference**, by mapping a recognized entity to its FQDNs. While this technically introduces a False Positive, it demonstrate a semantic understanding of the ecosystem by the LLM. This behavior happened in quite a few posts.

- Entry with post ID **1k76ykg** contains an example of **syntactic hallucination.** This happens when the model correctly identifies the semantic information (the threat indicators) but fails to adhere to the strict output formatting constraints defined in the system prompt. The post alerts other users about a potential scam campaign with clear Indicator of Compromise, the LLM output is as follow:

  ["type":"FQDNs","ioc":"simonsaffiliate.at","label":"M"]

  Although the IoC and label found were correct, this syntactic hallucination will result in a False Positive for the LLM in the analysis part.

In conclusion, with these qualitative examples we have can clearly see a fundamental tradeoff of using large language models for this task.While LLM generally outperforms regular expresison based tools in normalizing obfuscated IoCs, new challenges are introduced, related to the hallucination, (we covered examples of generative inference and syntactic hallucination). A robust post processing pipeline is needed to succesfully review the output of an LLM.

6.4.2 Chain of Thought (CoT) prompting

The baseline prompt already showed decent results and a good capability of extracting defanged IoCs.

Chain of Thought prompting is a technique that improves the performances of a model by explicitly asking the model to generate a step by step explanation or reasoning process before arriving at a final answer.By doing this, the model itself breaks down the task into smaller sections, allowing for the minimization of errors that might arise from handling too much information, thus resulting in theoretical better results.

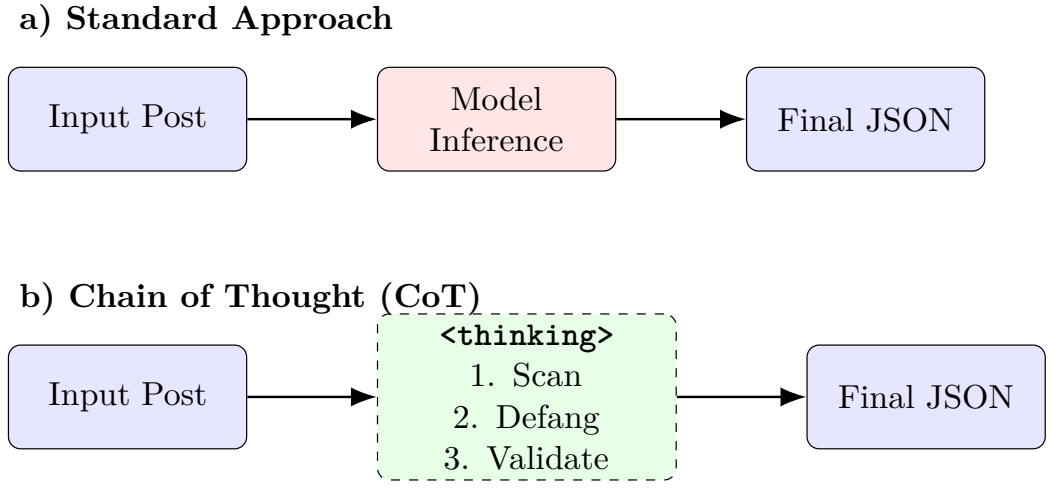

Figure 6.5: Comparison between Standard prompting (a) and Chain of Thought prompting (b). Note how the CoT block explicitly separates the reasoning phase from the output generation.

As shown in figure 6.5, we shift from a "black box" inference model to a more structured reasoning pipeline. This allows to guide the thinking phase of the mode in order to optimize the extraction task and let us understand better how the model reasons. We decided to try this approach in both the LLM for extraction approach and the LLM for extraction + filtering one. In order to implement it, a change to the prompt must be added to force the model into thinking.

We added the section shown in figure 6.6 to the prompt to enforce CoT.

CoT Instruction - LLM for extraction

Before giving the final output, a step-by-step analysis must be conducted and put inside a `<thinking>` tag:

1. **Scan & Identify**: Read the Title and Text. Identify potential entities for all categories.
2. **Defanging Check**: Look for obfuscation techniques (e.g., "example[.]com", "user at gmail dot com", spaces in IPs/URLs) and reconstruct the original IoC.
3. **Validation & Anti hallucination**:
   - Ensure FQDNss are NOT parts of extracted URLs.
   - Ensure "organization" is not just a domain name but an entity contextually identified as malicious.
   - Check if inferred entities (like inferred domains from brand names) are actually present in the text or just hallucinations (do not hallucinate IoCs).
4. **Formatting**: Prepare the final list in the requested tuple format.

Figure 6.6: Enforced reasoning done via system prompt.

The structured protocol that the model will follow has multiple purposes. It forces the LLM to read both the title and body of the post, identifying any entity that resembles IoCs (**Scan & Identify)**, the **Defanging Check** enforces an already proven good ability of the LLM to extract defanged IoCs, while the **Validation & Anti hallucination** part helps the model describing its most common mistakes. -post 426 reasoning working!

Table 6.5: Performance comparison of GPT-based approaches. **TP**: True Positives, **FP**: False Positives, **FN**: False Negatives.

Methodology	TP	FP	FN	Precision	Recall	F1-Score
Extraction Only						
GPT Extraction standard	596	495	241	0.55	0.71	0.62
GPT Extraction (CoT)	**653**	595	**182**	0.52	**0.78**	0.63
Extraction + Filtering						
GPT Extr. + Filtering standard	504	**128**	331	**0.80**	0.60	**0.69**
GPT Extr. + Filtering (CoT)	465	141	370	0.77	0.56	0.65

Although the results don't show improvement, it was still interesting to see how the reasoning process affected the LLM extraction of IoCs. In post ID **1ok2lhn** we see how the checks we force the LLM to do are working. In the post we can see an user exposing a Tiktok promotion scam saying he got called by an Italian number, thus giving us the country prefix code **+39**. The normal approach, using LLM for extraction and filtering wrongly extract it as phone number, while the Chan of Thought one gives us this explanation:

> *Phone: ∔39 ...ïs incomplete and uses ellipsis, not a specific number → cannot be taken as a valid phoneNumber IoC.*

Chain of thought reasoning allows the model to think about the actions more carefully, thus giving us a more reliable extraction. (slightly improved performances on the LLM for only extraction approach).

To conclude, although this approach does not seems to overall improve performances, further researches could be done in order to optimize it for an higher quality extraction phase.

6.4.3 Model size effects on performances

We tried to use a smaller model (GPT-5 nano) for the same extraction and filtering task, to see if it was possible to reduce the dimension and computational power while still having decent results, as shown in Table 6.6. The nano model achieves

Extraction	Filtering	TP	FP	FN	Precision	Recall	F1 Score
LLM	LLM	504	128	331	0.80	0.60	**0.69**
LLM (nano)	LLM (nano)	204	39	636	**0.84**	0.24	0.38

Table 6.6: Comparison between full-size and nano version of the model on IoC extraction and filtering.

the highest precision among all the configurations evaluated (0.84), meaning that nearly all the IoCs it extracts are correct. However, its recall drops sharply to 0.24, resulting in an F1 score of only 0.38. This pattern suggests that smaller models tend

to be overly conservative: they identify only the most obvious indicators, missing a large portion of the IoCs present in the text. Compared to the full-size LLM pipeline, the nano variant produces 305 more false negatives, meaning that the gap is not in the ability to correctly label what is found, but in the ability to recognize IoCs in the first place. This is particularly relevant in the context of social media posts, where IoCs are often embedded in informal and noisy text that requires stronger language understanding to be detected. These results suggest that model size plays a critical role in recall-oriented tasks such as IoC extraction. While a nano model may be suitable in scenarios where precision is the primary concern and computational resources are limited, a full-size model provides a significantly better trade-off between precision and recall, as reflected by the higher F1 score (0.69 vs. 0.38).

6.5 Dataset analysis

Now that we successfully compared the performances of the various approaches, we can move on to the final analysis on the dataset collected from different subreddits, as shown in 4.3. Although the main strengths of each approach have been already discussed in the ground truth analisys part of the thesis, the goal of this section is to have an overall analysis of the dataset and show real world applications, as it can still give us interesting results on the scalability ad usage of LLM base models. Picture 6.7 provides a comprehensive summary of the extraction results.

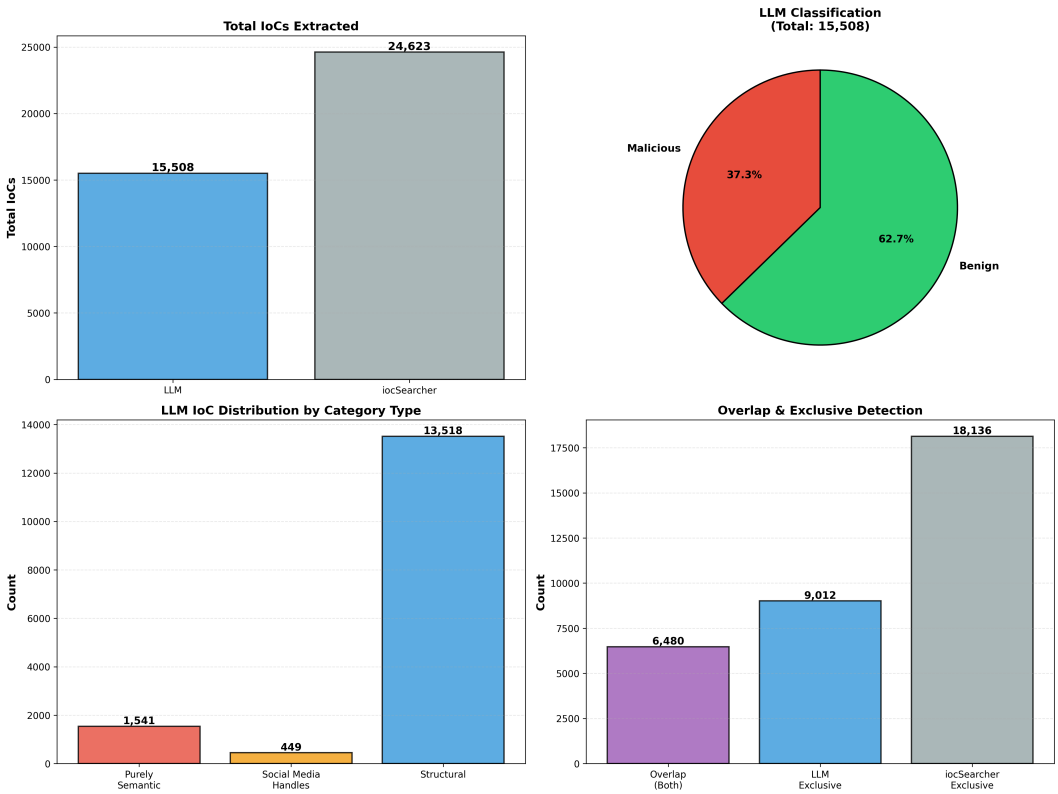

Figure 6.7: Comprehensive summary statistics: LLM vs iocSearcher on full dataset

6.5.1 IoC category distribution and analysis examples

From the result of previous analysis we concluded how the LLM based approach is often better or, in some cases, the only way to extract semantic based indicators. The goal of this section is to observe how the model performs in the extraction of all the IoCs and we divided them into 3 main categories:

- **Purely semantic indicators** such as Organization and person
- **Social media Handles** which can be also extracted from URL,so we expect some results from iocsearcher.
- **Structural IoCs** such as URLs, FQDNss, crypto wallet addresses, to do a comparison between the two methodologies with a bigger dataset.

Figure 6.8 gives us an overview of the purely semantic indicators found, making a distinction between the total indicators found and the considerd malicious ones.

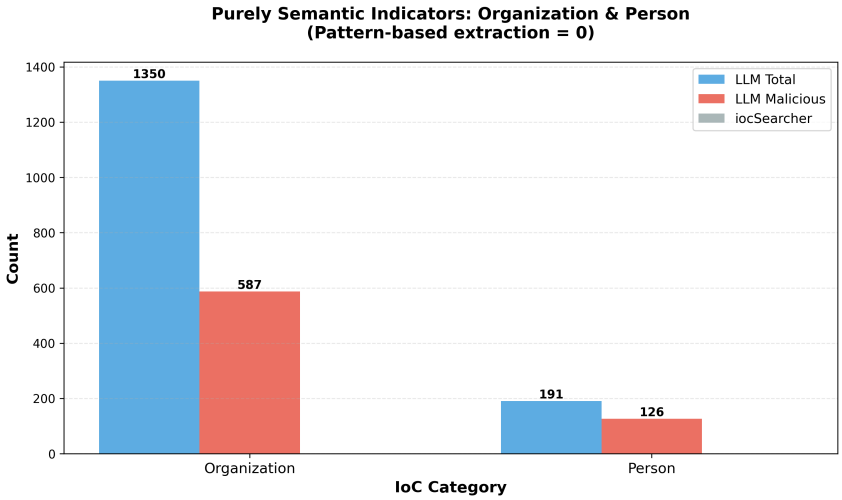

Figure 6.8: Purely semantic indicators distribution in the dataset.

As we can see, that's a total of **1541** indicators completely missed by the regular expression tools, because of limitations in its design. We already discussed why it's important to extract these type of indicators in previous sections [26],[24],[25]. Now let's see a real world application scenery and the results that can be observed, for example it would be interesting to run the study on specific subreddits considered fraudulent to try and understand patterns, such as organization names that repeat themselves many times, as showed in the Table reported below.

Table 6.7: Top 10 Entities - Organization

Organization	**M**	**B**
coinbase	133	147
kendu	45	0
powsche	37	0
binance	30	23
kraken	28	0
solfart	18	0
kendu inu	17	0
mexc	16	0
grass	15	9
trust wallet	14	0

Table 6.8: Top 10 Entities - Person

Person	**M**	**B**
mark zuckerfart	9	0
peter senius	5	0
fart mcsatoshi	5	0
elon musk	5	0
professor stephen beard	4	0
stephen beard	3	0
elizabeth holmes	2	0
billy evans	2	0
alice	2	0
claire preston	2	0

A post was considered malicious if at least one of the IoCs extracted by the LLM in said post was flagged as **"M"**. We could have used directly the label assigned by the model but our intent with this analysis is more related to understanding if the name of the organization was used in a malicious context.

Common known organizations such as **Binance** and **coinbase** appear both in benign and malicious posts (where they might be used to give a false sense of safety) and benign ones (generic crypto world discussions). Kraken, on the other hand appear exclusively in benign posts in our dataset, reflecting its lower popularity and more technical user base.

Other entities like **Kendu** and **powsche** seems to only appear in Malicious context. These organizations are a community driven meme/speculative crypto project, which is often associated with promoting fraudulent activities.

Moving on to the Person table, we can immediately notice the name of **Elon Musk**, relevant person in the crypto world and probably used by threat actors to instill **FOMO** [27]. The other names are not popular but there is an interesting trend to look into. Internet is a young community and it's common the use of memes/slang, as shown from the names picked. Mark zuckerfart is a clear parody of **Mark Zuckerberg** as a sort of identify spoofing, using such name only to attract attention. This trend highlight the **communication strategy** of malicious context online: exploiting internet slang and parody.

While frequency alone can't imply malicious intent, the asymmetric distribution shown in the tables can suggest implications of certain entities with potential harmful content rather than others.

Now let's move on to the analysis for **social media handles** extraction, since our studies showed how important it is to extract such indicators. We expect to see some results from iocsearcher, but an overall higher number of handles extracted using the LLM, because of the context semantic understanding of this approach. Extracting such handles is crucial in order to identify the contact point of malicious

actors;the most common scam techniques are conducted by moving the conversation to another platform (such as Telegram). Figure 6.9 gives us a comparison.

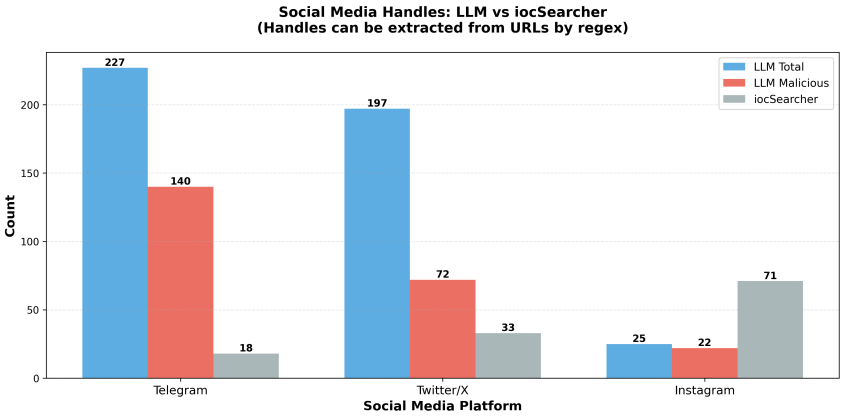

Figure 6.9: Social media handles in our dataset

As expected, Telegram is the most common social handles used in these kind of contexts, as well as having the higher amount of malicious indicators while twitter on the other hand seems to be used in normal context. Instagram seems to be better extracted by using the regular expression approach, users probably tend to give the Link directly to their account and the LLM fails to capture this more structural indicator.

Table 6.9: Mapping of malicious entities to its associated social handles

Malicious Entity (LLM)	**Associated Handles**	**Frequency**
Magnum Locked Liquidity	@sellockedliquidity, @magnumexchange	13
Sprout Capital	@catcartel_xyz, @hodicoin	6
Coinbound	@wolfannouncements, @wolf_on_sol	6
Byrrgis	@wolfannouncements, @wolf_on_sol	6
Peter Senius	@tassshub	3

In order to track scam execution flow, a table like 6.9 could come in handy in real life scenarios as it shows analysis associating the LLM's malicious extracted entities with its most frequently used social handles. Since it's a common scamming technique to try and bring the user to an external platform where the real scam can happen, this kind of study can shine a light on the correlation between organization and the cited platforms.

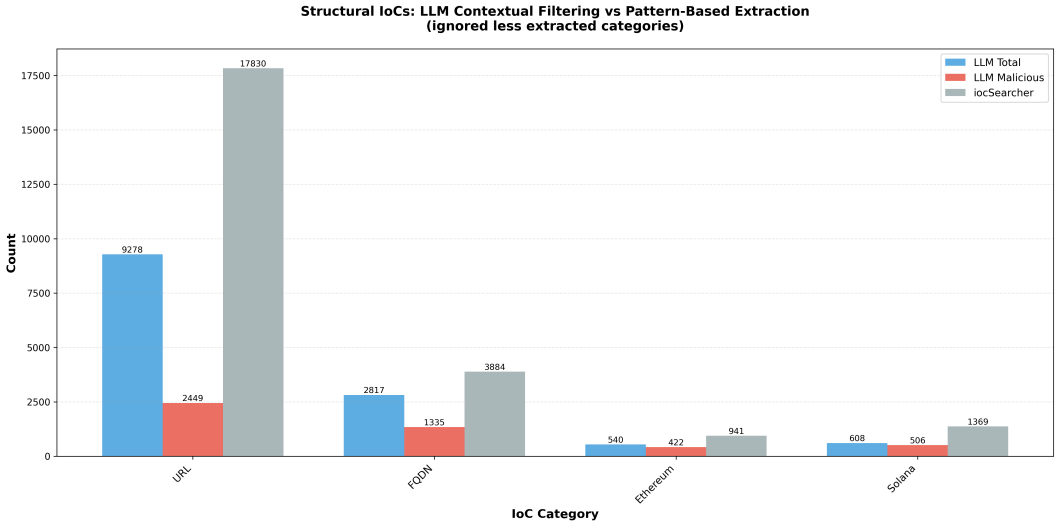

Figure 6.10: Structural IoCs extracted from our dataset

Table 6.10 contains the structural IoCs extracted from our dataset (we ignored the less extracted categories).

It's immediately noticeable how regular expression tools such as iocsearcher have better performances at extracting URLs, as studied in previous sections. The real take from this first bar chart is the **filtering** capabilities of the LLM. Ground truth analysis confirms us the performances of the two approaches are similar, so what is happening in the dataset is an extreme **noise reduction**, which would drastically lower alert fatigue in real world scenario. The LLM does not only extract strings, but it understands the role they have in the threat context. We also expect to find a great number of defanged URLs in the one rightfully identified by the LLM, and similar argument can be made for FQDNss.

Another factor to consider is that almost all crypto wallets are classified as malicious by the model, with no significant differences compared to the regular expression tool. In these contexts, the type of IoC is crucial: while noise reduction is essential for URLs to prevent data overcrowding and subsequent alert fatigue, crypto wallets do not necessarily require an LLM to be identified

Table 6.10: Examples of URL obfuscation (defanging) techniques, identified only by the LLM

Obfuscation Category	**Extracted Example**
Obfuscation	`t[.]me/BeyondOTC`
Non-standard separators	`https:\\\cryptochat.in`
Textual replacement	`fiatxgold(dot)com`
Inserted spaces	`sultanoshi.com/`
Broken protocol	`https://bazaars.app/`
Protection period	`https://.linktr.ee/catbatcoin`

Table 6.10 gives us an overview on malicious URLs that are extracted by the

LLM but not by the regular expression. Similar to the analysis we conducted on the ground truth, this proves the abilities the LLM has to extract obfuscated indicators, even if heavily modified, and shows us the model does not rely only on exact pattern matching, but understand the user **intent**.

We focused a lot on these indicators in this thesis because they usually represent a ringing bell for malicious activity, since the author of the post is actively trying to bypass social media automatic moderation systems.

6.5.2 IoC temporal distribution analysis

The last dataset analysis we wanted to include in this thesis is about the temporal distribution of the IoCs found by the regular expression and LLM approaches, to take advantage of the fact our data collection method gathered data for roughly 1 year, as shown in Table 4.3.

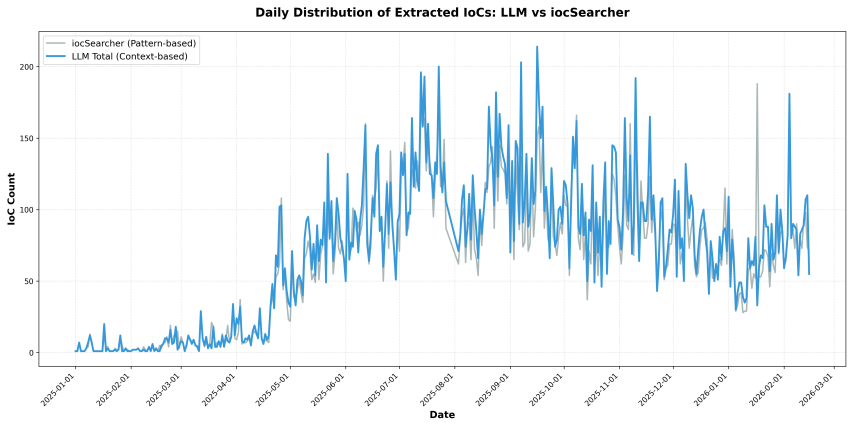

Figure 6.11: Daily distribution of extracted IoCs: LLM (context-based) vs iocSearcher (pattern-based)

Some Key observations can be extracted from Figure 6.11: Similar activity patterns can be observed by both approaches, peaks and valleys align closely suggesting a temporal distribution of the post publication rate. The LLM extracted posts are less than iocsearcher one, suggesting a contextual filtering could be done by the LLM, reinforced by the fact that the gap remains constant over time, and a peak extraction rate that reach approximately 220 IoCs/day for iocsearcher and 150 IoCs/day for the LLM, showing how higher activity periods don't affect such capabilities.

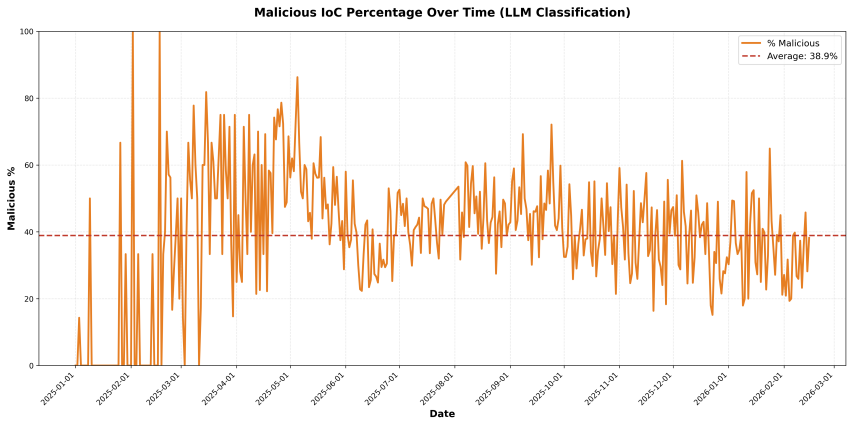

Figure 6.12: Percentage of malicious IoCs over time (LLM classification)

The malicious classification rate shown in Figure 6.12 gives us insights into LLM behavior consistency: A stable 38.9% average malicious classification rate is obtained by the LLM throughout all the analysis period, with high daily volatility (it ranges from 15% to 100%), due to the daily sample size, no significant drift can be seen in the long term trend. This results suggests that the LLM classification logic is not determined on temporal factors such as season or evolving scam tactics. One key take from this graph is about the LLM's filtering **reliability** in cyber threat intelligence applications

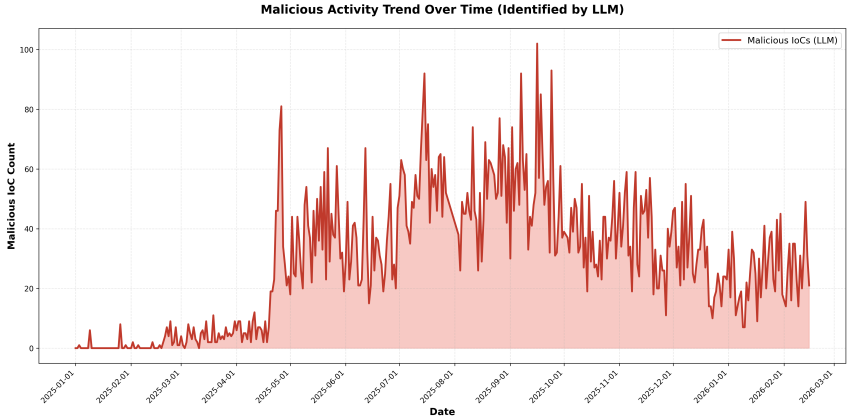

Figure 6.13: Malicious activity trend over time as identified by the LLM

Figure 6.13 reveals several temporal patterns in malicious IoC detections: The months that range from May and September have a surge in malicious activity, peaking with 100 daily malicious IoCs detected by the model. Our dataset contains a good amount of posts from crypto subreddits, thus linking the increase with market dynamics such as *bullish* market, which historically attracts coordinated scam campaigns, often successfull by taking advantage of the **Fear Of Missing Out (FOMO) [27]** (Bitcoin had one of the highest price ever in summer of 2025). A gradual decline can be noticed in October 2025, returning to a baseline level, keeping a similar behavior in early 2026. Beginning of 2025 shows a relatively low malicious activity, and this can be justified by modifications in our post gathering scripts, other

than seasonal patterns.

The temporal analysis demonstrates several key findings:

- **Contextual filtering**: compared to iocsearcher, the LLM's lower extraction rate demonstrate its contextual filtering is working and this results in a big reduction in **noise**. As already mentioned, this is crucial in avoiding analyst's **alert fatigue**.
- **Real world patterns**: By using this approach, fraudulent market campaigns can be tracked down and studies can be done to analyse them: **The LLM reflects real world trends** rather than producing random detections.
- **Stability**: The average malicious rate (39%) with no temporal drift denotes how the LLM is able to maintain consistent precision over the whole post collection time span, addressing concerns about model reliability in production environments.
- **Real world applications**: Extraction patterns align with the subreddit's activity, showing how LLM based extraction can perform in a reliable way in continuous monitoring scenarios.

Chapter 7

Conclusion and future work

The goal of this thesis was to evaluate **Large Language Models' (LLMs)** performances in a cybersecurity context, more specifically **Indicators of Compromise (IoCs)** extraction and filtering from unstructured social media text (using Reddit as a case study).

An analysis on recent "state of the art" papers has been done and we discovered how this specific application has not been studied yet, although some articles uses LLM in a correlated but different way.

This process led to the creation of two distinct datasets in order to both evaluate and assess the strengths and weaknesses of the different approaches. We quickly discovered how scarce public social media datasets are, or at least in Reddit's case, and this forced us to change our strategy, with a focus on a smaller group of different kind of subreddit that could provide a comprehensive overview on how fraudulent activities work on this platform

- **Ground truth dataset** was used to evaluate the performances of the various approaches. We collected 600 posts from a small but representative number of subreddits (Table 4.5), and manually extracted and labeled IoCs. The analyses conducted on the ground truth gave us interesting results and allowed us to directly compare the various approaches, while also providing qualitative information on the extracted IoCs.

- **General dataset:** about one year worth of 12 different subreddits was collected(Table 4.3). The analysis gave us interesting results on real-world applications and temporal analysis.

The analyses were conducted in a way to truly understand what works and what does not work. General IoC performances showed how the LLM based approach work slightly better (**0.69** F1 score against **0.65** of the iocsearcher one). Although iocsearcher capture more indicators (as showed by its recall, 0.78), the filtering abilities of the LLM approach can help to **drastically improve precision** (0.80 against iocsearcher's 0.56). These abilities are crucial: the puure extraction approach of the LLM showed to have slightly worse performances than the regular expression tools, while the hybrid one showed the overall best precision (0.87).

Furthermore, the LLM showed an IoC per post distribution (Figure 6.1) that most closely resembles the original ground truth, thus proving good noise reduction abilities without impacting the extraction performances. Post level detection analysis (Table 6.2) proves once again how the LLM model works best in real life post flagging scenarios, with a very low false positive rate: if a post contains an IoC labeled as malicious from the LLM, there is good probability the post is a threat.

To really understand what differentiates the LLM approach, we had to dig deeper to directly compare IoCs categories. Figure 6.2 showed the f1 score distribution per category and here we started to really see what makes the LLM approach better: purely semantic categories such as organization and person were exclusively extracted by the model (as expected), similar to social handles (users often pass the contact name, not the URL, this makes it impossible for a regular expression tool to extract). Surprisingly, the LLM also had better performances in structural categories such as URLs and emails, while struggling on crypto wallet addresses and phone numbers. To contextualize these numbers, we did a qualitative analysis on the IoCs extracted by the LLM approach. The importance of extracting entities is remarked by our findings: the LLM already extracted names of organizations well before they proved to be a scam campaign ([26],[24],[25]). When studying the other indicators categories, we found out one of the biggest strengths of using LLMs for this specific task: the ability to **reconstruct obfuscated indicators**, which explains the better performances in the URLs, email and social handles categories.

Since the task to perform was pretty complex, a prompt engineering study was conducted 6.4, as a way to show how we actually try to improve the LLM performance, and to understand what works and what does not. The study showed the importance of few shot learning (whose absence resulted in a drop of **0.13** in the F1 score) and the explanation of what each category of IoC is (again with a **0.13** downgrade in performances if this part was missing). Studies on other prompting techniques such as chain of thought prompting did not give meaningful improvements6.5 but were useful to understand the internal thinking of the model.

Lastly, the dataset analysis served to show the implications of this study in a real-world scenarios and to confirm some of the results we saw in the ground truth analysis. From now on, we only considered the LLM for extraction and filtering and regular expression for only extraction approaches, since it's what we wanted to focus on. We firstly conducted some analyses on the IoC distribution, divided into the 3 main categories we were interested in analyzing: purely semantic (*entitites* 6.8), social Handles (*telegram,twitter/X,instagram* 6.9), structural (*URLs,FQDNs,wallet addresses etc. etc.* 6.10). The results of this larger-scale analysis were useful in showing applications of the LLM approach (such as searching for most cited organizations , linking them to their social handles 6.9), and reinforced our previous knowledge (other examples of obfuscated URLs were presented 6.10, showing how this is a crucial difference in using this method). To take advantage of the time period our analysis was conducted on, we also conducted a temporal analysis of the IoC extraction from the LLM, with different graphs(6.12 6.13 6.11) highlighting how

the LLM extraction looks stable. This analysis also helped identify fraud campaigns, which seemed to have an increased activity in summer 2025.

Key takeaways from the use of LLMs in this domain are:

- **Semantic capabilities:** The LLM showed not only the ability to successfully execute the task that was assigned, but it also applied critical thinking in order to consider the whole post text and evaluate the indicator threat level, with a high degree of precision.
- **Organization,Person,Social handles extraction** were the categories where the LLM performed better, in part because they were exclusive to it, in part because how social media text is written. These categories proved to be crucial for deeper analysis and in order to identify scam campaigns/track their social media.
- **De-obfuscation:** The LLM proved to be able to successfully reconstruct different obfuscated indicators with good accuracy in our dataset (no defanged indicators were missed, although not many were present in it).Thanks to the dataset we were able to observe other obfuscation techniques. Hallucinations did not seem to be a big problem for this kind of task.
- **Noise reduction** is one of the main advantages of using an LLM based approach instead of a regular expression one. The filtering capabilities proved to be excellent in differentiating between benign and harmful content, thus drastically decreasing **alert fatigue**. The LLM showed a similar per post IoC distribution, which proves how an LLM can resemble human performances.
- **LLM's stability:** Temporal analysis conducted on the dataset showed how the LLM looks stable in its IoC extraction. Interesting results, such as scam campaign can be seen by the malicious indicators extraction rate.
- **Prompting techniques** had an actual impact on the performances, showing how we did not try to just use an "out of the box" LLM, but put effort in trying to see what works best for this specific application. Few shot learning and IoC explanation proved to be effective in improving the model's metrics.

A final consideration that must be made is about the *ethics* of using such an approach. While the applications of this methods looks promising, concerns may arise about the privacy of using this approach. The LLM can extract names of people, organization, and while this is crucial in order to identify scam campaigns, the risk of falsely flagging posts from innocent individuals. Other critiques could be about computational time and energy consumption. Training a model requires large amounts of time,data and power, while regular expression tools don't need all of this. Furthermore, Malicious actors will try to evolve their tactics in order to elude LLM's recognition systems and generate more evasive scam campaigns.

7.1 Future work

The results of this thesis can be expanded further. A natural next step would be integrating this analysis in a real-world scenario such as an actual deployment on Reddit, in order to see real time dynamics instead of a static repository.

Seeing performances in other social media platforms is another logical progression. From our study Telegram resulted as one of the social media with the highest concentration of malicious content, but Twitter and Instagram could as well provide interesting results.

Trying to use other fine-tuned models could be another interesting study, especially by using open source models such as Mistral or LLaMA, this would also mitigate the data privacy concern metioned previously.

Other applications of Machine Learning/ AI could be explored, like using Optical Character Recognition (OCR)[28] to extract indicators directly from screenshots that are often attached to the posts

Bibliography

[1] Ani Petrosyan. *Number of internet and social media users worldwide as of October 2025*. Accessed: 2025-12-19. Oct. 2025. URL: https://www.statista.com/statistics/617136/digital-population-worldwide/ (cit. on pp. 1, 4).

[2] OSINT Industries. *Social Media Intelligence (SOCMINT) in Modern Investigations*. OSINT Industries. 2024. URL: https://www.osint.industries/post/social-media-intelligence-socmint-in-modern-investigations (visited on 03/12/2024) (cit. on p. 1).

[3] Iguazio. *What are LLM Hallucinations?* Iguazio Ltd. 2024. URL: https://www.iguazio.com/glossary/llm-hallucination/ (visited on 03/12/2026) (cit. on p. 2).

[4] David Omand, Jamie Bartlett, and Carl Miller. "Introducing Social Media Intelligence (SOCMINT)". In: *Intelligence and National Security* 27.6 (2012), pp. 801–823. DOI: 10.1080/02684527.2012.716965 (cit. on p. 4).

[5] Vegetable-Key4397. *$840k stolen. My hacker is swapping to Monero*. Post in r/CryptoScams. Reddit. 2025. (Visited on 03/08/2026) (cit. on p. 5).

[6] MaliciaLab. *IoCSearcher: A Python library for extracting Indicators of Compromise*. https://github.com/malicialab/iocsearcher. GitHub repository, Accessed: 2025-10-20. 2024 (cit. on pp. 9, 37).

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need". In: *Advances in neural information processing systems*. Vol. 30. Curran Associates, Inc., 2017 (cit. on p. 11).

[8] Bryce Boe et al. *PRAW: The Python Reddit API Wrapper*. Version 7.7.1. Accessed: 2025-12-19. 2025. URL: https://praw.readthedocs.io/en/stable/ (cit. on p. 14).

[9] OpenAI. *OpenAI API Documentation*. Accessed: 2024-XX-XX. 2024. URL: https://platform.openai.com/docs/api-reference (cit. on p. 14).

[10] M. Siddharth, S. Dhakal, S. Chandra, M. Bhuyan, et al. "Large Language Models for Cybersecurity: A Survey". In: *arXiv preprint arXiv:2402.13520* (2024). URL: https://arxiv.org/abs/2402.16968v1 (cit. on p. 16).

[11] Evangelos Froudakis, Athanasios Avgetidis, Sean Tyler Frankum, Roberto Perdisci, Manos Antonakakis, and Angelos D. Keromytis. *Revealing the True Indicators: Understanding and Improving IoC Extraction From Threat Reports*. 2025. arXiv: 2506.11325 [cs.CR]. URL: https://arxiv.org/abs/2506.11325 (cit. on p. 17).

[12] Silvia Sebastiástian, Raluca-Georgia Diugan, Juan Caballero, Iskander Sanchez-Rola, and Leyla Bilge. "Domain and Website Attribution beyond WHOIS". In: *Proceedings of the 39th Annual Computer Security Applications Conference (ACSAC '23)*. New York, NY, USA: ACM, 2023, pp. 534–547. DOI: 10.1145/3627106.3627190. URL: https://doi.org/10.1145/3627106.3627190 (cit. on p. 17).

[13] Muhammad Muzammil, Abisheka Pitumpe, Xigao Li, Amir Rahmati, and Nick Nikiforakis. "The Poorest Man in Babylon: A Longitudinal Study of Cryptocurrency Investment Scams". In: *Proceedings of the ACM Web Conference 2025 (WWW '25)*. Oral Presentation. 2025. URL: https://openreview.net/forum?id=P0x8J5gCPP (cit. on p. 18).

[14] Ghaith Husari, Ehab Al-Shaer, Mohiuddin Ahmed, Bill Chu, and Xi Niu. "TTP-Drill: Automatic and Accurate Extraction of Threat Actions from Unstructured Text of CTI Sources". In: *Proceedings of the 33rd Annual Computer Security Applications Conference (ACSAC)*. 2017 (cit. on p. 18).

[15] Gibran Gomez, Kevin van Liebergen, Davide Sanvito, Giuseppe Siracusano, Roberto Gonzalez, and Juan Caballero. "Clean Up the Mess: Addressing Data Pollution in Cryptocurrency Abuse Reporting Services". In: *Future Generation Computer Systems* 166 (2025). arXiv preprint arXiv:2410.21041, p. 108313. DOI: 10.1016/j.future.2025.108313 (cit. on p. 18).

[16] Mehdi Mirtaheri, Sami Abu-El-Haija, et al. "Identifying and analyzing cryptocurrency manipulations in social media". In: *IEEE Transactions on Knowledge and Data Engineering* (2021) (cit. on p. 19).

[17] Alberto Poncet, Tom Vissers, and Nick Nikiforakis. "Pirates of Charity: Exploring Donation-based Abuses in Social Media Platforms". In: *2023 IEEE Symposium on Electronic Crime Research (eCrime)*. IEEE. 2023, pp. 1–14. DOI: 10.1109/eCrime60714.2023.10373204. URL: https://arxiv.org/pdf/2412.15621 (cit. on p. 19).

[18] Marie Vasek and Tyler Moore. "There's no free lunch, even using Bitcoin: Tracking the popularity and profitability of virtual currency scams". In: *Financial Cryptography and Data Security: 19th International Conference*. 2015 (cit. on p. 19).

[19] Tom B Brown, Benjamin Mann, et al. "Language models are few-shot learners". In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901 (cit. on p. 20).

[20] Aohan Kong, Kai Zhao, et al. "Better Zero-Shot Reasoning with Role-Play Prompting". In: *arXiv preprint arXiv:2308.07702* (2023) (cit. on p. 20).

[21] Jason Wei, Xuezhi Wang, Dale Schuurmans, et al. "Chain-of-thought prompting elicits reasoning in large language models". In: *Advances in Neural Information Processing Systems* 35 (2022) (cit. on p. 20).

[22] Ziwei Ji, Nayeon Lee, Rita Frieske, et al. "Survey of hallucination in natural language generation". In: *ACM Computing Surveys* 55.12 (2023), pp. 1–38 (cit. on p. 20).

[23] IBM. *Che cos'è il few-shot learning?* Consultato il: 8 Gennaio 2026. IBM. n.d. URL: https://www.ibm.com/it-it/think/topics/few-shot-learning (cit. on p. 41).

[24] Fortunato Vadala. *M3 DAO / MarsVerse: Phuket's Fake Real Estate Partnership Exposed by the Community.* https://decripto.org/en/m3-dao-marsverse-phukets-fake-real-estate-partnership-exposed-by-the-community/. Decripto.org. Sept. 2025 (cit. on pp. 50, 59, 67).

[25] Washington State Department of Financial Institutions. *Alleged AI Trading Platform CoinBridge.me Appears to be Engaged in Fraud.* https://dfi.wa.gov/consumer/alerts/alleged-ai-trading-platform-coinbridgeme-appears-be-engaged-fraud. Washington State DFI Consumer Alert (cit. on pp. 51, 59, 67).

[26] Decripto.org. *Crypto Scam Behind the Alpha Stock Investment Training Centre (ASITC): The CoinBridge Case.* https://decripto.org/en/crypto-scam-behind-the-alpha-stock-investment-training-centre-asitc-trading-course-the-coinbridge-case/. Decripto.org. 2025 (cit. on pp. 51, 59, 67).

[27] Bernama. *FOMO and social engineering techniques.* Feb. 23, 2025. URL: https://www.bernama.com/en/news.php?id=2460516 (visited on 03/03/2026) (cit. on pp. 60, 64).

[28] Amazon Web Services. *What is Optical Character Recognition (OCR)?* URL: https://aws.amazon.com/what-is/ocr/ (visited on 03/12/2026) (cit. on p. 69).

Dedications

Tutto ciò che sono lo devo a te che mi ha insegnato a vivere, amare, lottare per i miei obiettivi e a non accontentarmi mai dello status quo. Ti voglio bene papà.