POLITECNICO DI TORINO

DEPARTMENT OF MATHEMATICAL SCIENCES (DISMA)

Master Degree in Mathematical Engineering

Master Degree Thesis

Anomaly Detection at the Edge Implementing Machine Learning Techniques

Author: Ghassan El Baltaji

Advisor: Paolo Ernesto Prinetto
Co-Advisors: Vahid Eftekhari Moghadam, Nicolò Maunero

March, 2022

Abstract

The Internet of Things (IoT) refers to the process of connecting physical objects to the internet. This includes household appliances, healthcare assets like medical devices, smart industries and cities. Due to the drastic increase of data generated from the IoT devices, relying on a centralized cloud infrastructure has fundamental limitations. For example, high network latency and network bandwidth are two main constraints that should be addressed.

Edge Computing (EC) has emerged as the new computing paradigm in the IoT. Edge Computing covers the demand of the real-time response, and it moves data processing from the cloud to the Edge Nodes (ENs), hence increasing the quality of service for the IoT applications. Yet, Edge Computing has its own challenges such as cyberattacks.

IoT with edge devices is considered as an open system. Various types of cyberattacks threaten the entire system. At the terminal perception layer, such as the sensors which directly collects data from the environment, authentication attack could take place. While in the network transport layer, DoS and DDoS attacks could have severe consequences on the entire network. As for the application layer, brute force and man-in-the middle attacks are always suspected. Database attacks, malware or false data injection attacks(FDIA) and many other types of cyberattacks could be a potential threat to a specific layer,thus it could penetrate the whole network.

Artificial Intelligence (AI) advancements have opened up new possibilities for addressing security challenges. The learning capability of Machine Learning (ML) can be considered as a supportive system that identifies malicious behaviors more correctly and effectively. In this project, we apply ML techniques to create a security system for an edge device. We introduce the IoT system and its architecture with EC. Then, the convergence of Machine Learning, Cybersecurity and the IoT applications is discussed. Finally, we apply the machine learning classification model, and we embed the model in the device. Specifically, we have used the KDD Cup 1999 dataset which includes a wide variety of intrusions simulated in a military network environment. The main objective is to build an anomaly detection system. It is a predictive model that distinguishes between "good" normal connections and "bad" connections known as intrusions or attacks. Different supervised machine learning models are applied such as decision trees, K-nearest neighbors and support vector machines algorithms. The best performing classification model has been evaluated and embedded to the edge device.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The Internet of Things(IoT) refers to a distributed network that combines different sensor devices and systems, such as sensor networks, RFID devices, barcode and QR code readers, global positioning systems (GPS), etc. All these devices are connected together through wired and wireless communication technologies, enabling embedded systems to communicate and interconnect with each other. The development of IoT includes three technical routes:

- Sensing, identification and authentication technologies are the foundation of IoT. Imagined as the nerve endings of the IoT, sensors are the largest and most basic part of the chain in IoT.
- The development of transmission and communication technologies are the guarantee of IoT. The information collected by the IoT devices needs to be transmitted to the central node or the processing unit. The development of wired and wireless networks, cellular networks and other transmission and communication technologies have made it possible for large scale IoT transmission.
- Data computing and processing technologies are essential to provide services using IoT data. The development of data computing and processing technologies is the key to improve processing intelligence and effectiveness.

Having one central processing unit has aroused some limitations, among the others, high network latency and large network bandwidth. New computing technology such as edge computing have given IoT more possibilities to provide advance application services.

By definition, Edge Computing (EC) is a computing paradigm that see the shift from a centralized cloud architecture to a distributed approach, where data are processed in the same place (device) where they are collected, at the edge of the IoT network. Together, IoT and edge computing are a powerful way to rapidly analyze data in real time. Having less processing power than centralized cloud servers, edge computing has a major security challenge that should be addressed, since it is not possible to deploy complex cybersecurity solutions that are usually adopted in desktops and servers environment. While processing large amount of data, less protection leads to the inevitable temptation by bad actors to exploit edge device vulnerabilities to extrapolate sensitive information or compromise the correct functioning of the devices. The new edge security threats, which can include lateral attacks, account theft, entitlement theft, DDoS attacks, and more, can cause more than just service disruption. They pose a fundamental challenge to the way of deploying data computation at the edge to guarantee a secure and reliable flow of vital information through the whole IoT network. Machine learning can be thought as a key player in this kind of problem. With the high number of data points that the sensors collect and transfer, Machine Learning(ML) can be applied to give more insights to the system. The presence of machine learning in IoT enhances the intelligence for the whole system. The learning and predictive capabilities allows it to be one of the top use cases in edge computing.

The goal of the thesis document is to create an anomaly detection system for the edge device using machine learning techniques to improve the security level of the IoT system.

The thesis document contains 6 chapters. In the second chapter, we describe the IoT architecture network. We define the functionality for the device, network, edge and cloud layers. We also discuss about cyberattacks types that could occur at every IoT layer. In the third chapter, we discuss about how machine learning process can be adopted for enhancing IoT security. We present various machine learning models for IoT security. Moreover, state of art models for anomaly detection are represented. In the fourth chapter, we proceed to the practical work by importing the KDD 1999 Cup dataset, applying the pre-processing methods, feeding the training dataset into the model and evaluating the performance of the model. Different machine learning algorithms are performed such as Decision Trees, K-Nearest Neighbors, Logistic Regression and Random Forest. In the fifth chapter, we show the results obtained. Finally, we conclude with further work that could be extended for the whole project.

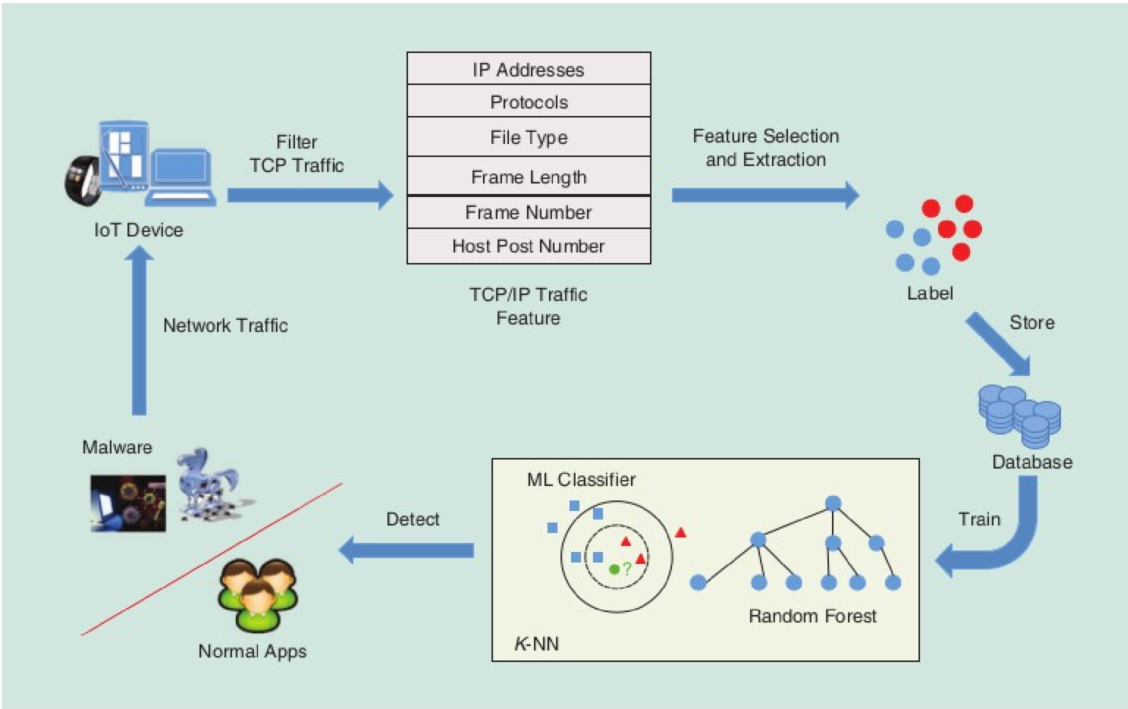

Figure 1.1. **An illustration of ML-based Malware Detection**

Chapter 2

Background

2.1 IoT Service Framework with Edge Computing

IoT service framework adopting the Edge Computing paradigm can be divided into four major layers: device layer, network layer, edge layer and cloud layer.
i) **Device layer**: Mobile phones, laptops and vehicles are all equipped with various types of sensing devices. We can feel our environment in real time using these devices, creating a vast amount of data. The majority of this information are delivered quickly, instantly and frequently.
ii) **Network layer**: This layer connects the cloud with edge devices and the end user. It is the IoT service framework's nervous system, linking sensing devices all throughout the Internet of Things and carrying out transmission tasks. The data is sent over a variety of communication technologies, including cellular networks with base stations, WIFI, ZigBee, Bluetooth and other IoT protocols or data transfer protocols, such as Hypertext Protocol and Message Queuing Telemetry Transport.
iii) **Edge layer**: This layer is the key characteristic of the IoT service framework with Edge Computing, which solves the problems of insufficient bandwidth and high delivery latency, as opposed to the standard IoT service framework. Partially transferring computational resources from the cloud to the edge, which is significantly closer to data sources. Data processing, data management and data storage are all handled by edge servers. The results are then delivered to the appropriate devices or uploaded to the cloud layer for additional analysis or storage via the network layer.
Some essential technologies, including edge operating systems, isolation strategies and data processing platforms are boosting the development of the edge layer to ensure the smooth and efficient execution of computing operations.
iv) **Cloud layer**: With Edge Computing, this layer serves as the brain of the IoT service framework. It is usually made up of huge cloud data centers with a lot of processing power. The cloud layer is typically used with EC to analyze data from the edge layer, store or update significant information and perform advanced deployment. Resource collaboration, management collaboration and safety collaboration are all examples of cloud-edge collaboration. It is a dynamic entity, it ensures that all processes are completed correctly. When a specific edge layer appears to be sending malicious traffic, the cloud layer with superior security policies can detect and stop it, preventing it from spreading further[16].

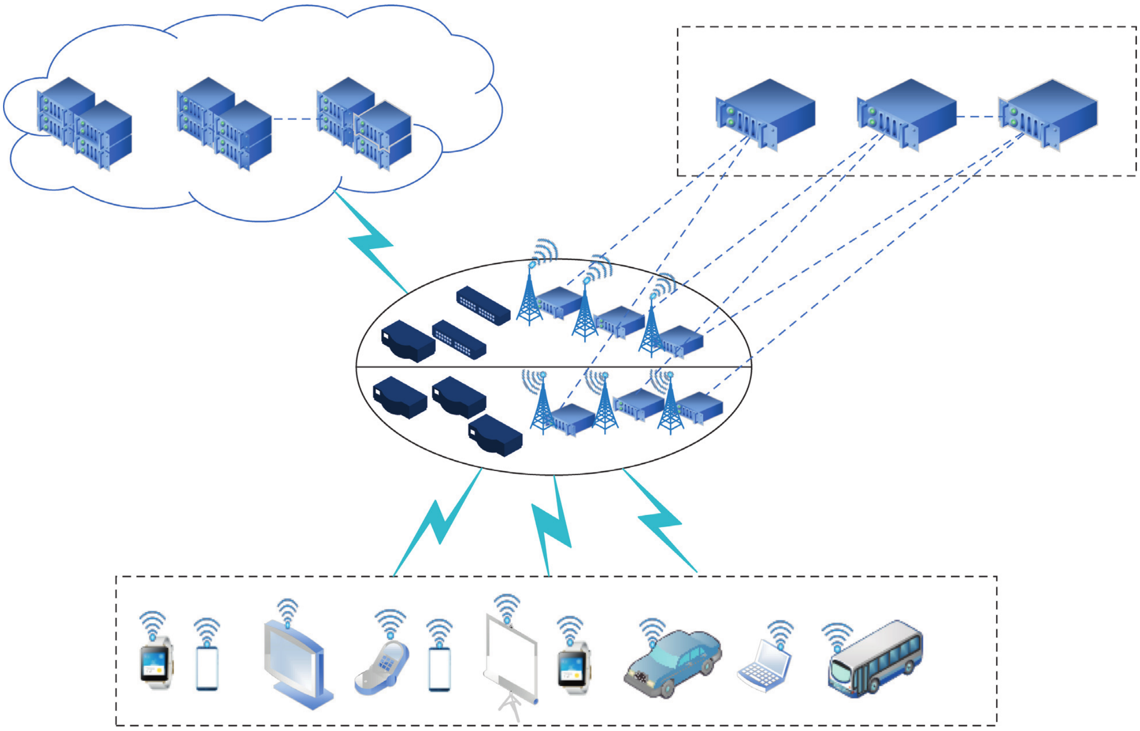

Figure 2.1. **IoT Service Framework with Edge Computing**

2.1.1 Machine Learning

Artificial intelligence (AI), often known as machine intelligence, refers to intelligence produced by machines as opposed to natural intelligence demonstrated by humans and other animals. Learning, vision, problem solving, language comprehension and logical reasoning are all areas where AI algorithms might help. The application of data-driven Artificial Intelligence techniques has been steadily increasing in engineering fields including communications. The success of such applications in recognizing patterns has given an advantage over the traditional logic-based expert systems. The success of AI systems is achieved through the unprecedented availability of data and computing resources. In scientific words, machine learning is a mathematical model that capture the physics of the system under study. An optimized algorithm is produced. It guarantees performance under the assumption that the given physics-based system is similar to the representation of reality.

Supervised learning, unsupervised learning and reinforcement learning are the main applications in machine learning.

In supervised learning, the training set consists of inputs and their outputs. The goal is to learn the mapping between input and output spaces. Example includes the application of intrusion detection classification on the basis of normal/attack connection input.

In unsupervised learning, the training set consists of inputs without labels. It aims to discover patterns within the dataset itself by clustering input points that are close to each other.

While reinforcement learning lies between supervised and unsupervised learning. Some sort of supervision exists in reinforcement learning in a way that the algorithm receives a feedback from the environment relying on an observation that results in an output. The feedback indicates the level to which the output fulfils the goals of the learner.

Advantages of data-driven machine learning tools can speed up the design cycle, reduce complexity and cost of implementation. Next, we explain more about the implementation of machine learning model, and we will present specific types of models applied in security applications.

2.2 IoT Security Threats

The flaws of IoT security are gradually uncovered as more machines and smart devices are connected to the network. Not just because of the increased use of IoT devices, but also because of their complexity, diversity and intrinsic mobility of such device application scenarios, IoT devices are more vulnerable to attack than PCs or mobile phones. Furthermore, openness is mirrored in the IoT system's numerous processes. IoT may collect data from a variety of sources, integrate different communication technologies and standards, and offer open services to consumers in a variety of industries.
Openness is beneficial to the development of the Internet of Things, but it also expands the scope of potential threats. Because of the interconnection and dependency of IoT systems, attackers can take advantage of any vulnerability to conduct large-scale, systematic attacks that will bring the entire system to a halt. For example, attackers can utilize some terminals as assault entry points and employ tools to evaluate data contained in the same type of terminal, such as source code and authentication mechanisms, in order to infiltrate

the entire system. In the terminal perception, network channels and application service levels, there exist many and different forms of security threats.

2.2.1 Security Threats in the Terminal Perception

The terminal system's key components are sensors. Their main responsibility is to collect data and keep an eye on items in real time. These tiny physical devices can be found in a wide range of technical disciplines, and they are huge in number. They become a potential attack surface for attackers because most of them are resource constrained. The three basic categories of terminal perception layer nodes are:

- Collection endpoints: primarily sensors that sense and collect data
- Information aggregation: the server that receives, processes and forwards data
- Nodes and isolated nodes: embedded equipment that performs information encryption

When information is networked among nodes, there are risks of interception, eavesdropping, counterfeiting, hijacking and node tampering due to the transmission distance. Although uniqueness and certainty of identification can significantly increase the security of IoT devices, hackers can bypass this procedure applying a variety of approaches. For example, in April 2019, the software iLnkP2P was released without any authentication or encryption protocols. Attackers can avoid the firewall and establish direct connections with IoT devices using certain serial numbers, sending malicious messages in place of any valid data supplied by the device.

2.2.2 False Data Injection Attacks (FDIA)

False data can be injected into the system through compromising physical sensors, sensor data communication lines and data processing applications. Physical sensor compromise allow physical access to the sensors, which is a time-consuming procedure. Hacking the sensor data connection channels and data processing programs, on the other hand, is a more straightforward alternative for an attacker. FDIA might cause sensors to send incorrect values to the central control, resulting in damaging consequences. They try to intercept data integrity, and they are significantly different from regular cyberattacks.[8]. The attacker can gather compromised nodes from various geographic locations of the network allowing to bypass their security regardless of having message authentication codes in the system. When an attacker reveals sensor readings, he can insert errors in the calculations of state variables and values. A countermeasure technique is to apply deep learning to learn the FDIA characteristics from the historical data. Deep learning extracts the features which helps to identify the attack.[3]

2.2.3 Security Threats in the Network Transport Layer

The Internet of Things combines sensor networks and communication networks to create large-scale network. The number of possible attacks rises in lockstep with network scale, similar to the dangers posed by the terminal perception layer. Because some network targets have inadequate security, attackers will have an easier time gaining access.

The attacker could access the platform through the software and hardware components if the system does not have security and verification procedures that risks data extraction. As a result, detecting breaches early is critical for detecting attacks and ensuring network security. Denial of Service (DoS) and Distributed Denial of Service (DDoS) are two attacking techniques that can be deployed. They work by sending traffic that exceeds the target's processing capacity, exhausting the computational and network resources, resulting in resource depletion, network blocking and a denial of service. Long-distance networks(NB-IoT) and LoRa(Long Range Radio), short-distance networks(ZigBee, Wi-Fi,etc.) and the Internet can have a security weakness which can be transferred to the Internet of Things. The Internet offers IoT users a wide range of services, but the TCP/IP-based communication infrastructure is vulnerable to security and privacy threats such as intrusion, replay attacks and identity theft. [16]

2.2.4 Security Threats in the Application Service Layer

The application service layer processes the data given by the network transport layer and then provides services for various application scenarios based on user requirements. Users can profit from the simplicity of use of the IoT system's services by using web applications or mobile apps. Application-level attackers, on the other hand, interferes with the system, the data and the software. The application service layer is made up of basic environments, components and virtual cloud platforms. Brute force and man-in-the-middle attacks will be carried out using basic environments and components such as operating systems, databases and middle-wares. The goal of these attacks is to obtaining unauthorized access, remote control and data leakage. Most IoT systems create virtual cloud platforms to decrease equipment deployment costs and increase computing performance or business throughput. By blurring the boundaries between humans and data, virtual technology offers a security risk, resulting in issues like virtual machine escape, virtual network attacks and virtual software vulnerabilities. Other types of attacks may target database attacks such as SQL injection, privilege promotion and backup theft. Data privacy protection at the application service layer is a fundamental security need. Many IoT databases, such as GPS positioning data, may contain personal information. Attackers can use this data to analyze sensitive personal information about users, such as their location, income, lifestyle, behavior or health.

Moreover, malicious programs are widely used by software attackers. The Bank of Russia, for example, identified Bespalova malware at ATMs in 2017[14], which paid automatically after entering a code. For example, attackers might use XSS(Cross-Site Scripting Attack) to inject malicious scripts into a legitimate website. Successful XSS attacks can cause IoT accounts to be hijacked and the system to become paralyzed.

Furthermore, in recent years, Android malware has become considerably frequent. Because of the openness for the Android mobile operating system, malware has proliferated across mobile devices. By exploiting security holes, malware can penetrate users' mobile phones and steal personal information.

Chapter 3

State of the Art

3.1 Machine Learning for IoT Security

Before we discuss the application of machine learning in IoT, we briefly explain the mechanisms behind useful machine learning algorithms which are performed in the implementation chapter.
a) k-Nearest Neighbor is a supervised learning technique. K-NN algorithm assumes the similarity between the new testing data point and the labeled data points. It puts the new case into the category that is mostly similar to the available categories. For the new data point that is located in the plane with respect to the training data points, we calculate the selected distance metric(Euclidean distance, Manhattan distance, Minkowski distance,..) where K number of neighbors are chosen according to the nearest distance. The new data point is assigned to the category where the number of neighbors is the highest. One disadvantage of KNN algorithm is that it has high computation cost because it calculates the distance between the data points for all the training samples.
b) Logistic regression predicts the output of a categorical dependent variable. Instead of giving the exact value as 0 or 1, it gives the probabilistic values which lie between 0 and 1. Logistic regression is similar to linear regression except that how they are used. Linear regression is used for solving regression problems whereas logistic regression is used for solving classification problems. Logistic regression is named for the function used at the core of the method, the logistic function. The logistic function, also know as the sigmoid function is a S-shaped curve (Figure 3.1) that can take any real-valued number, and it maps the number into a value between 0 and 1.

$$f(x) = \frac{1}{1 + e^{-x}}$$

Logistic regression uses an equation as the representation like linear regression. Input values are combined linearly using weights to predict an output value. A key difference from linear regression is that the output value is a binary value(0 or 1) and not a continuous numeric value. For better classification, we feed the output values from the regression line to the sigmoid function. The sigmoid function returns the probability for each output value from the regression line. Based on a predefined threshold value, we can classify the output into two classes as normal or attack. The parameters of the logistic regression algorithm are estimated from the training data using the maximum-likelihood estimation. The intuition for maximum-likelihood for logistic regression is that a search procedure seeks values for the parameters that minimize the error in the probabilities predicted by the model to those in the data. Logistic regression is easy to implement, interpret, and very efficient to train.

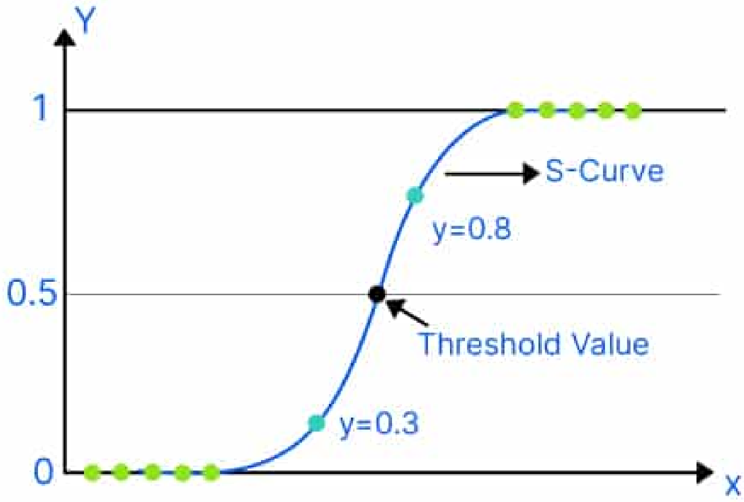

Figure 3.1. **Logistic Regression**

c) Random Forest is a supervised learning technique. It is based on the process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset, and it measures the average to improve the predictive accuracy of that dataset. Instead of relying on one decision tree, the random forest takes the prediction from each tree. Based on the majority votes of predictions, it predicts the final output. Advantages of random forest is the capability of handling large datasets with high dimensionality.

d) Support vector machine(SVM) algorithm works by constructing a hyperplane which in two dimensions it is a line that best separates the two classes form each other. This line is the decision boundary, anything that falls to one side of it is classified as normal and anything that falls to the other is classifed as intrusion. The best hyperplane would be the one that maximizes the margins from both classes. When data is non-linear, SVM classifies the distrbution of data points by mapping the space to a higher dimension.

Having explained about some of machine learning algorithms, we can proceed into the discussion of the intersection between machine learning and the Internet of Things. The focus of the Internet connectivity will shift from individuals to the Internet of Things. The new coronavirus outbreak, which began in 2020, has posed significant difficulties to the world economy and society. The Internet has been critical in the restoration of employment and production, as well as rehabilitating the economy and the maintenance of social operations. The reason of more deployment and usability of Internet-of-Things brings the importance of implementing new technologies and methods to improve the quality and the efficiency of the IoT service.

A new technology has emerged through the connection of artificial intelligence and the devices of the IoT. The two complements one another, and they make use of their respective data and technical capabilities, resulting with a new experience for the individuals. Lightweight ML algorithms can be deployed directly on the edge devices, allowing an even more powerful data analysis near the end user, without the need of sending data back to the central cloud datacenter. Moreover, artificial Intelligence now plays a significant role in both home and enterprise network security solutions. AI technologies can be used to more efficiently identify and predict hazardous behavior that could affect the functionality of the network. Artificial intelligence-based network security equipment can automatically acquire information from network resources and identify the location of network security intrusions[4].

Machine learning for IoT security diagram contains two major algorithms: decision algorithm and transaction algorithm. Data exploration and pre-processing are mostly handled by transaction algorithms. To get the general characteristics of the dataset and give the basis for decision algorithms, transaction algorithms use a small number of samples and simple models.

Decision algorithms are primarily responsible for business decisions, and they employ various decision-making processes in order to lower the ratio of misjudgement and maximize overall profit. Single decision-making, sequential decision-making and integrated decision-making are three forms of decision algorithms based on strategies and scenarios.

Machine learning can compensate for traditional security solutions' flaws in several areas, conforming to IoT characteristics, and provide new capabilities for IoT to satisfy new security requirements.

We know that the nodes of IoT devices are fixed, and their usual behaviors are predictable and structured, based on analysis for the majority of IoT security events. Machine learning methods such as supervised and unsupervised learning can help categorize normal actions and abnormal actions (cyber attacks) by identifying abnormal activity and distinguishing abnormal patterns. As a result, both supervised and unsupervised learning can be applied in IoT security. For instance, to determine whether an access device is authorized, a supervised learning method called Support Vector Machine(SVM)(figure 3.3) can be used. To identify different devices and intercept unlawful devices, Support Vector Machine can employ a hyper plane to partition points of device data into two categories: blue nodes are authorized devices and yellow nodes are unauthorized devices. [14].

Traditional security systems aren't equipped to deal with new infections or attacks, and

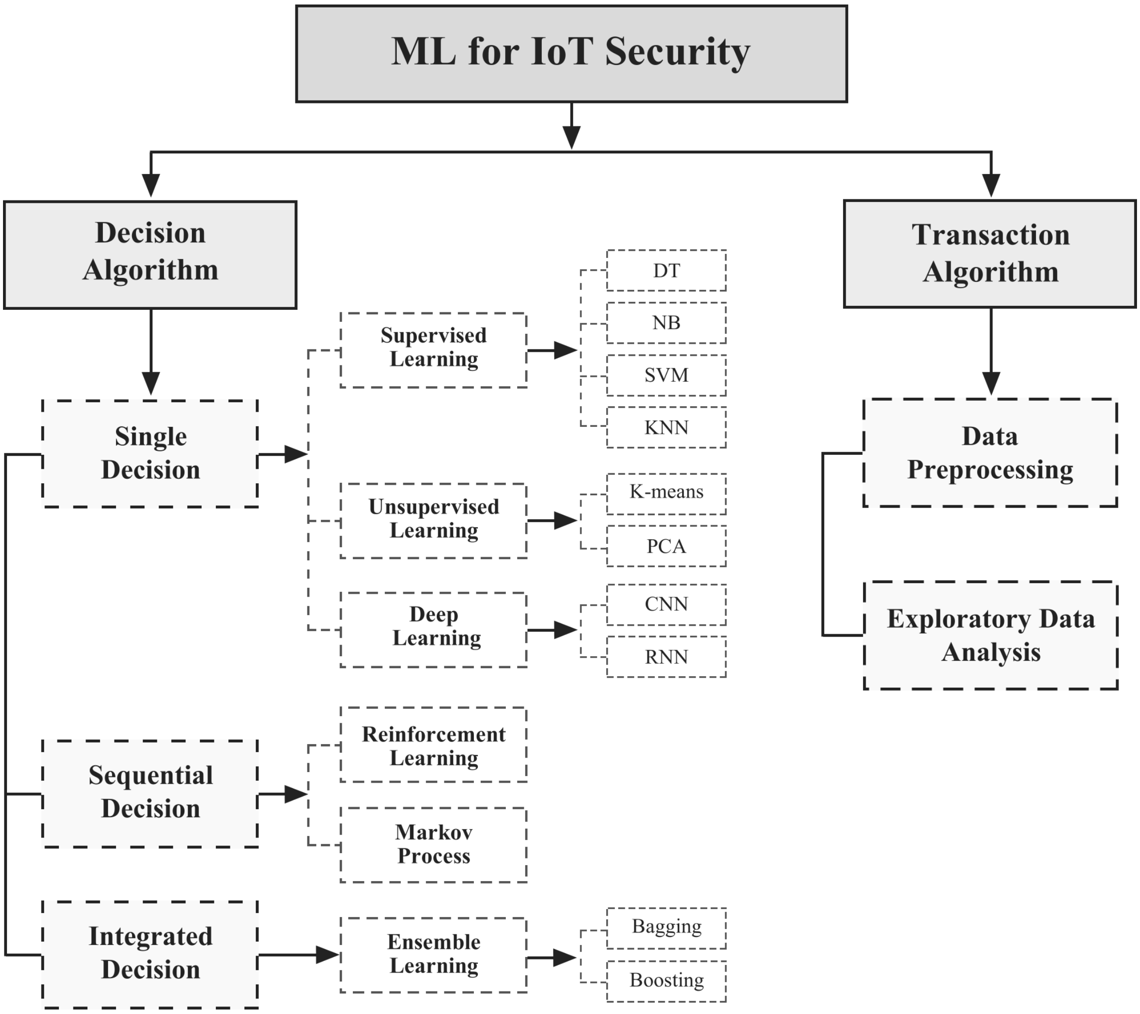

Figure 3.2. **Machine Learning for IoT Security**

they can't provide timely and effective defenses. The absence of capabilities of unknown network intrusions such as Zero Day attacks, for example, is a significant flaw in intrusion detection based on misuse detection. AI, in this sens,can provides automation and intelligence capabilities for IoT security. Unsupervised learning may automatically extract knowledge from data without the use of known tags. Ensemble learning can vote and alter the learning focus of the model based on the results of several classifiers, gradually and automatically improving the model effect and avoiding repeated human operations. While in a constantly changing environment, reinforcement learning can perform gradual model optimization through the reward/punishment mechanism and change learning procedures to optimize the benefits.

Traditional IoT security techniques are designed to function with a finite amount of data. As the amount of data generated grows, the shortcomings of these schemes, in terms of big data processing capability and computational efficiency, becomes more apparent. The major purpose of software security in the application service layer, for example, is virus detection. Traditional malware detection approaches extract malicious behavior codes from malware as signatures and use the resemblance between the software to be detected and the signature database to determine whether it is malware. When the amount and dimensions of data grow, the computational complexity grows rapidly, reducing the model's effectiveness and making it unable to detect threats in a timely and effective manner.

Traditional models lacking low accuracy is another reason why Artificial Intelligence technologies are critical in IoT security. The confidence of security models will be harmed if they are unable to detect a potential attack in time. Furthermore, certain traditional models may not function the operation mechanism of IoT in order to increase the model effect due to technical restrictions.
The robustness and generalization capabilities of Artificial Intelligence systems are highly valuable. In order for a model to be robust, it must be able to successfully reduce the impact of noise and outliers, as well as maintain its usefulness in complicated settings. The model's generalization ability shows its capacity to forecast unknown variables, ensuring that the model's efficacy is maintained when being transferred from the experimental to the application scenario. Support Vector Machine determines the classification result using a small number of support vector samples. Random forest has good anti-noise ability and is not sensitive to outliers. While linear models with L1 and L2 regularization have excellent generalization ability and can avoid over-fitting.

Device authentication, DoS/DDoS assaults and intrusions must be addressed, and with traditional solutions to these problems, there is a lack of ability to analyze large amounts of data, filled with drawbacks such as slowness and poor real-time performance. Large IoT data may be used by machine learning-based AI algorithms to infer relevant knowledge and develop predictions for unknown providing new solutions to current difficulties.

Device Authentication

For secure IoT authentication, machine learning provides a number of potential methods. These schemes use a variety of techniques to collect verified data from devices. The white and black lists are of safe and harmful devices respectively that are widely used to filter and intercept connected devices. Using random forest, Meidan et al.[7] developed an

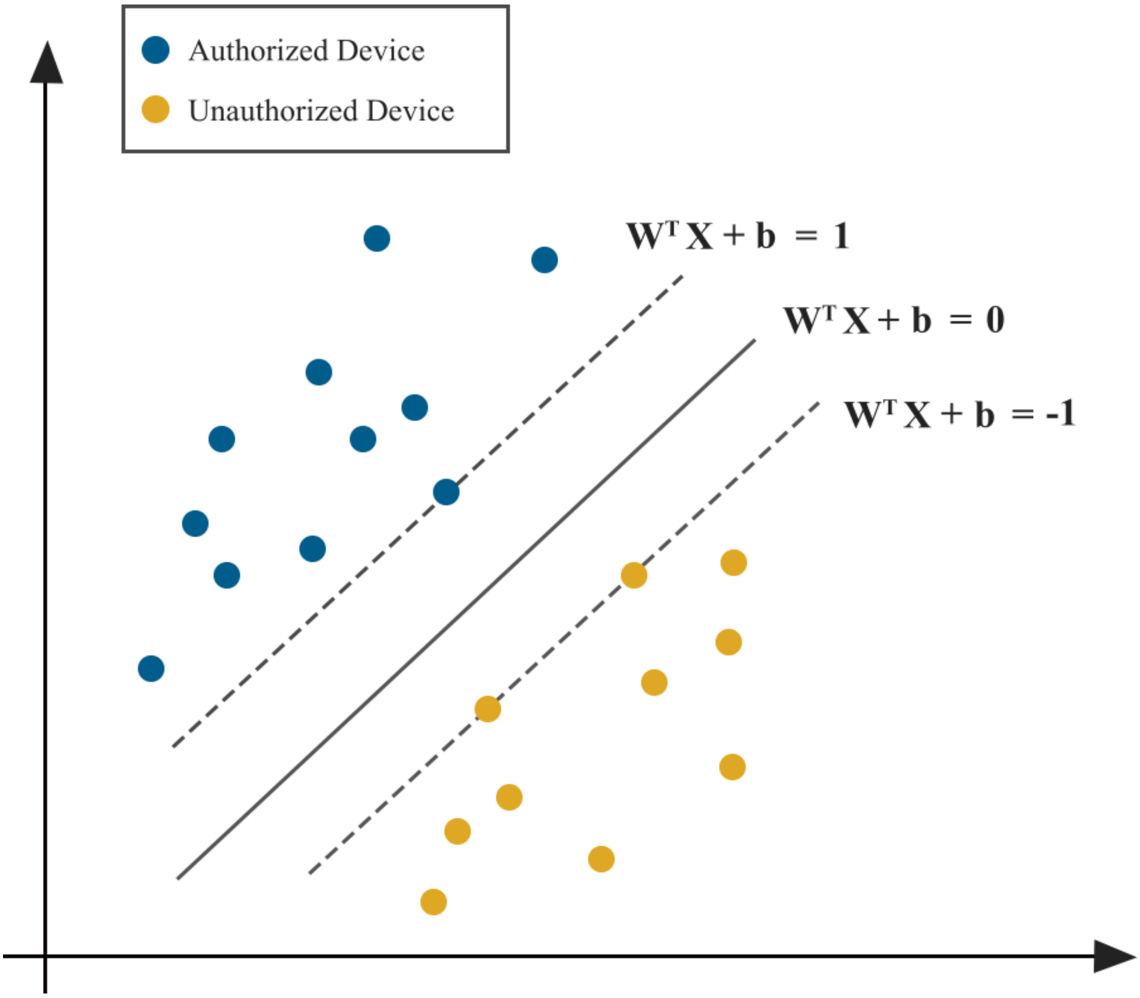

Figure 3.3. **Support Vector Machine Method**

authentication technique that combines white list and ensemble learning, as well as feature extraction and device classification of network traffic data in big enterprise IoT. This method is highly accurate where it achieved a 99% average accuracy for nine device types on testing data. DoS/DDoS
DDoS attacks harness the connectivity of many compromised devices and direct packets of data at a specific target, such as a website or internet service, with the aim of knocking it offline. On January 2022, the record-breaking 3.47 Tbps DDoS attack on Microsoft azure cloud service has originated from approximately 10,000 sources from connected devices in the United States, China, South Korea, Russia, Thailand, India, Vietnam, Iran, Indonesia, and Taiwan. "We believe this to be the largest attack ever reported in history," said Toh,a product manager on the Microsoft Azure networking team.[13] A denial-of-service attack (DoS) can be conducted in a variety of methods to prevent a victim (host, router or full network) from providing or receiving normal Internet services. When such attacks are discovered, system patches are usually released right away. Another type of DoS attack is to force a victim to perform computationally costly operations like encryption and decryption, as well as secret calculation based on Diffie-Hellman exchanges[18]. Anti-DDoS devices, firewalls and other protective settings are used in traditional DoS and DDoS defenses, which are tuned for load balancing. Through firewall, rule filtering and content filtering, these technologies determine whether external access traffic is normal. Machine learning is a strong option for detecting DoS and DDoS attacks intelligently and automatically. The software-defined network(SDN) is a network design that separates network devices' control plane and data plane. SDN can be employed as the underlying communication infrastructure for IoT because of its centralized control, flexibility and applicability. Ye et al. In [17] authors proposed a SVM-based DDoS detection technique that treats attack detection as a classification problem. The system gathers variables from the SDN switch flow table related to DDoS attacks, such as IP source, source port, flow entries speed, standard deviation of flow packets, deviation of flow bytes and pair-flow ratio to utilize as characteristic values for SVM classification.

Table 3.1 summarizes the possible machine learning techniques that can be implemented for different types of security risks[15].

Attacks	Security Techniques	ML Techniques	Performance
	Secure IoT offloading	NN	Detection accuracy
DoS	Access Control	Multi-variate correlation analysis	Root Mean Error
	Secure IoT offloading	Q-learning	Energy Consumption
Jamming		DQN	SINR
	Authentication	Q-learning	Average Error Rate
Spoofing		Dyna-Q	Detection Accuracy
		SVM	Classification Accuracy
		DNN	False Alarm Rate
		dFW	Miss-detection Rate
		Incremental Aggregated Gradient	Miss-detection Rate
	Access Control	SVM	Classification Accuracy
Intrusion		Naive Bayes	False Positive Rate
		k-NN	Detection Rate
		NN	Root Mean Error
	Malware Detection	Q/Dyna-Q/PDS	Classification Accuracy
Malware	Access Control	Random Forest	False Positive Rate
		k-NN	True Positive Rate
			Detection Accuracy
			Detection Latency
	Authentication	Q-learning	Proximity Passing Rate
Eavesdropping		Non-parametric Bayesian	Secrecy Data Rate

Table 3.1. **Machine Learning-based IoT Security Methods**

3.1.1 General Process of AI application for IoT Security

Device authentication, DoS/DDoS detection, intrusion detection and virus detection are classification types. For device authentication, AI solutions must distinguish between authorized and unauthorized devices while for intrusion detection, AI must classify between normal and abnormal behavior, etc. The following will explain the general process of Machine Learning in IoT security applications:

Data Collection

Most machine learning(ML) solutions necessitate datasets from specific environments. One must choose the suitable setting for data collection to build training and testing datasets for various challenges. To represent user differences, datasets for device authentication, for example, must include information on device configurations, user behavior and operating habits.

Data Pre-exploration and Pre-processing

The impact of solutions is directly proportional to the quality of the training data, IoT datasets are derived from a variety of sensors in a variety of industries. The original dataset, however, has a number of flaws, including irregular data distribution and incomplete data. To prepare for the next phases, it is required to mine the training data, master the data distribution and then do actions such as eliminating errors and finishing incomplete data.

Model Selection
There are numerous ML models that may be used for IoT security, but each model has its own set of scenarios that can be applied to, so we need to choose relevant models based on model attributes and task requirements. The size of the data collection and the pre-exploration outcomes will also influence the model performance. If the dataset has has fewer simple examples and the training must be done quickly, then lightweight techniques such as naive Bayes can produce predicted results.
Data Conversion
The data collected in real-world applications is frequently incompatible with the input data required by models, and it must be changed to satisfy the needs of the models we chose. The audio data acquired by voice sensors, for example, can not be directly entered into Recurrent Neural Network models, Extraction of Mel-Frequency Cepstral Coefficients(MFCC) from original audio data is one of the conversion procedures.
Training and Testing
We are required input the processed data into models for training after completing the pre-processing and choosing the model selection. We can examine the loss function value or evaluate the curve during the training process to determine the model's training trend. Possible solutions would be to modify the parameters such as the learning rate or the number of iterations to ensure the model is tuned and optimized. After training the model, test datasets are utilized to evaluate the generalization capabilities of the models. This is necessary since the trained model may be under-fitting or over-fitting, and if so, the re-adjustment of parameters can be adjusted.
Model Evaluation and Deployment
We can use many indicators to evaluate the performance of the model. To objectively evaluate the model's prediction and generalization ability, different evaluation indicators are used in classification tasks. Accuracy, precision, F1 score and AUC(Area Under the Curve) are all evaluation metrics that are calculated.

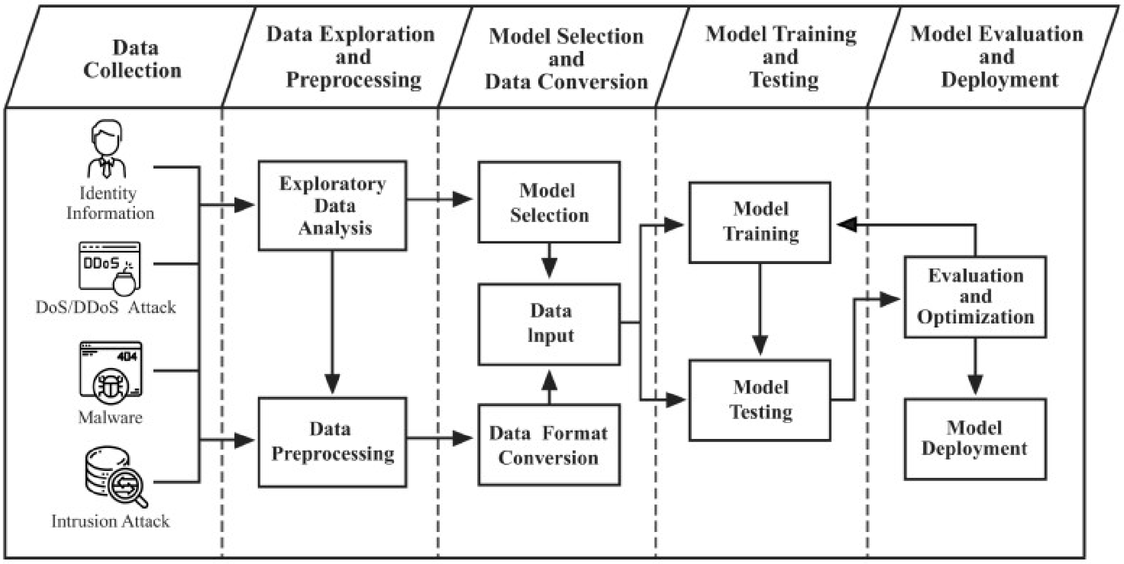

Figure 3.4. **The general process of AI solutions for IoT Security**

3.2 Anomaly Detection in IoT

In general, anomaly detection plays an instrumental role in robust distributed software systems. It can enhance communication around system behavior, improve root cause analysis and reduce threats to the software ecosystem. Traditional anomaly detection is manual. However, machine learning techniques are improving the success of anomaly detectors. Anomaly detection is any process that finds outliers of a dataset. This means it detects the items that don't belong in the reference dataset. These anomalies might point to unusual network traffic. In today's world of distributed systems, managing and monitoring the system's performance is a fundamental task to achieve. With thousands of items to monitor, anomaly detection can help point out where an error is occurring and can quickly get the technical support on the issue. Machine learning, then, suits the engineer's purpose to create an anomaly detection system that works better, handles large datasets, adapts and performs quickly. Applying machine learning to anomaly detection requires a good understanding of the problem. Anomaly detection benefits from large amounts of data because the assumption is that anomalies are rare.

In our case, we work with the supervised domain in which training data is labeled with "normal" or "anomaly". The supervised setting is the ideal setting. It is the instance when a dataset comes neatly prepared with label-annotated data points. In this case, all anomalous points are known ahead of time. Popular Machine Learning algorithms for structured data are support vector machine(SVM), k-nearest neighbor(KNN), bayesian networks, decision trees, random forest and logistic regression.
In the Internet of Thing systems, anomaly detection is known as the intrusion detection. In most cases, an intrusion results in the loss of confidentiality, integrity, resource denial or

unauthorized usage of resources. Some examples of intrusions that system administrators are concerned about are the following [9]:

- Unauthorized changes to system files that allow unauthorized access to either the system or user information.
- Unauthorized access to or modification of user data and files.
- Modifications of tables or other system information in network components that are not authorized(e.g., changes to router tables in an inter-net to prevent use of the network).
- Unauthorized use of computing resources(perhaps through the establishment of unauthorized accounts or the use of existing accounts without authorization).

In general, there are many state-of-art solutions for IoT's cyberattack detection using AI approaches. For example, in the article [1], a systematic review is applied to 80 selected studies published between 2016 and 2021. Intrusion Detection Systems (IDS) has gained a significant consideration among the best security mechanisms for safeguarding the IoT cyber infrastructures against various cyber-attacks in the last decades. That is, IDS has become one of the prominent tools to enhance security in today's IoT network-based systems. The signature or misuse based IDS, anomaly-based IDS, and hybrid approach are the well-known categories of security techniques to detect intrusions and anomalies in IoT devices.[6] The detection model is considered efficient when normal and malicious patterns can be easily identified from data using a simple mathematical model. Recently, various machine learning approaches have been already employed to develop IDS for IoT devices. The behaviour of normal traffic and abnormal traffic can be identified with relatively high detection accuracy and low false-positive rate, therefore, maximizing the security of both confidential data and IoT devices. For example, a new anomaly-based approach for IoT networks is implemented with a hybrid feature selection engine that only selects the most relevant features, and the Random Forest algorithm classifies each traffic as normal or abnormal. The performance was evaluated using IoTID20 dataset with an accuracy of 99.5% for DoS, 99.97% for MITM and 99.96% for Scanning attacks.[6] Also, Pahl[11] has introduced a system for creating site invariant IoT μS models and a highly efficient algorithm for periodicity mining. They have applied K-Means and BIRCH Clustering and achieved an accuracy around 96.3%. Moreover, using the dataset DS2OS traffic traces from Kaggle, a 99.4% accuracy is obtained by applying an Artificial Neural Network(ANN) classification algorithm[12]. Ahmad et al.[2] conducted a comprehensive analysis of different deep learning(DL) models which include convolutional neural network(CNN), recurrent neural network(RNN), long short-term memory(LSTM) using IoT-Botnet 2020 dataset to propose an effiecient anomaly detection using mutual information(MI) by considering deep neural network(DNN) for an IoT network.

In this thesis work, the goal is to apply an intrusion detection system implementing machine learning techniques. The objective is to apply different classification models that detects normal and abnormal connections. We evaluate and choose the best performing one in order to further implement it. As discussed in Section 3.1.1, we choose the dataset that is relevant to our project. Then, we apply the pre-processing needed to organize the dataset by choosing the most important features and preparing it for model building. Following these processes, we apply various training algorithms, and we evaluate the best performing machine learning classifier.

Chapter 4

Implementation

This chapter discusses the implementation of the ML model for anomaly detection in an IoT context that was the subject of the thesis work. The script has been written in Python language using Google Collaboratory notebook. The machine learning library used is scikit-learn. Scikit-learn library is very widely used across all parts of the bank for classification, predictive analytics, and many other machine learning tasks. We decided to apply binary classification model. Binary classification refers to predicting one of two classes. Class 0 indicates that the connection is normal while class 1 indicates that the connection is an intrusion.
The flow diagram of the proposed anomaly detection system is depicted in Figure 4.2.

4.1 KDD CUP 1999 Dataset

The 1998 DARPA Intrusion Detection Evaluation Program was prepared and managed by MIT Lincoln Labs.[5] The objective was to survey and evaluate research in intrusion detection. A standard set of data to be audited, which includes a wide variety of intrusions simulated in a military network environment, was provided. The 1999 KDD intrusion detection contest uses a version of this dataset. Our chosen dataset is the 10% kdd cup dataset folder. The dimension is 494021 rows × 42 columns. Each connection is labeled as either normal, or as an attack, with exactly one specific attack type. The description for some of the features is the following:

- **Duration**: length(number of seconds) of the connection
- **protocol_type**: type of the protocol, e.g., tcp, udp, etc.
- **src_bytes**: number of data bytes from source to destination
- **hot**: number of "hot" indicators
- **su_attempted**: 1 if "su root" command attempted, 0 otherwise
- **num_root**: number of "root" accesses
- **diff_srv_rate**: % of connections to different services

feature Name	**type**
duration	continuous
protocol_type	categorical
service	categorical
flag	categorical
src_bytes	continuous
dst_bytes	continuous
land	discrete
wrong_fragment	continuous
urgent	continuous
hot	continuous
num_failed_logins	continuous
logged_in	discrete
num_compromised	continuous
root_shell	discrete
su_attempted	discrete
num_root	continuous
num_file_creations	continuous
num_shells	continuous
num_access_files	continuous
num_outbound_cmds	continuous
is_host_login	discrete
is_guest_login	discrete
count	continuous
srv_count	continuous
serror_rate	continuous
rerror_rate	continuous
srv_rerror_rate	continuous
same_srv_rate	continuous
diff_srv_rate	continuous
srv_diff_host_rate	continuous
dst_host_count	continuous
dst_host_srv_count	continuous
dst_host_same_srv_rate	continuous
dst_host_diff_srv_rate	continuous
dst_host_same_src_port_rate	continuous
dst_host_srv_diff_host_rate	continuous
dst_host_serror_rate	continuous
dst_host_rerror_rate	continuous
dst_host_serv_rerror_rate	continuous
label	categorical

Table 4.1. Features Names and Types for the KDD Dataset

4.2 Data Pre-Processing

Data pre-processing is an integral step in Machine Learning as the quality of data and the useful information can be derived from it directly affects the ability of our model to learn; therefore, it is extremely important that we pre-process our data before feeding it into our model.

The labels column, in Table 4.1 has 23 types specifying the connection type, so the first change we do is to transform the connection types into '0'(good) or '1'(bad). Since our objective is to apply a binary classification, we need to assign two class labels as 0 or 1. We decide to apply a binary classification and not a multi-class classification due to simplicity, and our objective is to know if the connection is good or bad regardless of which type of connection. Having binary values in the label column, the trained model predicts 0 or 1. After inspecting the whole dataset, we analyze the categorical features in which 'protocol_type' has 3 categories(icmp, tcp, udp), 'service' has 66 categories(private, http,smtp,etc.) and 'flag' has 11 categories(SF, S0, REJ). Many machine learning algorithms expect numerical input data because they can not operate on label data directly. We need to convert our categorical data into a numerical form. One method is to use a one-hot encoding(Figure 4.1) on categorical data. One-hot encoding adds a new binary variable for each unique integer value. The binary variables are also called "dummy variables". For example, 'protocol_type' transforms into 3 binary variable columns and they are added to the main data frame.
The new dimension of the dataset becomes 494021 rows and 119 columns.

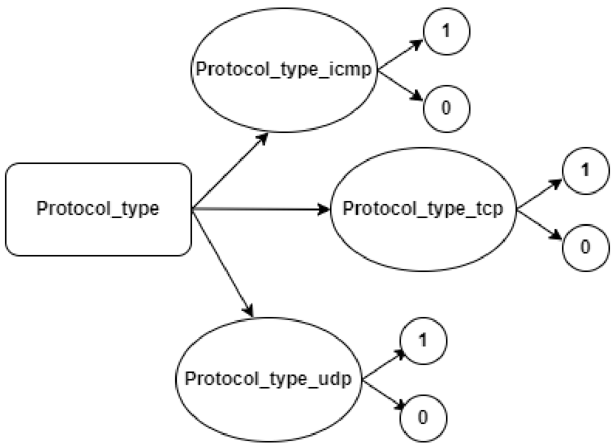

Figure 4.1. **One Hot Encoder on Protocol_type Feature**

The performance of the algorithms degrade with too many input variables. As the dimensionality increases, the number of data points required for good performance increases exponentially. The predictive power of any classifier increases as the number of dimensions increase, but after surpassing a certain number of dimensions, the performance deteriorates. This phenomenon is known as curse of dimensionality or Hughes phenomenon. We need to apply dimensionality reduction; this process refers to techniques that reduce the number of input variables in a dataset. When dealing with high dimensional data, it is often useful to reduce the dimensionality by projecting the data to a lower dimensional subspace which captures the "essence" of the data.[10] This technique is used in machine learning to simplify a classification dataset to better fit a predictive model. We select the features using uni-variate feature selection with ANOVA F-test. Uni-variate means that every feature is compared independently with the labeled data. ANOVA(Analysis of Variance) is a parametric statistical hypothesis test for determining whether the means from two or more samples come from the same distribution or not. A F-statistic, or F-test, calculates the ratio between variances values such as the variance from two different samples. The features are chosen with the highest values of separability. The selected features are reported in Table 4.2

Feature Selection
logged_in
count
srv_count
srv_diff_host_rate
dst_host_count
dst_host_same_src_port_rate
Protocol_type_icmp
Protocol_type_tcp
Protocol_type_udp
service_ecr_i
service_http
service_smtp

Table 4.2. **Features Selected**

4.3 Model Training

This is the stage where the ML algorithm is trained by feeding datasets. The training model consists of the sample output data and the corresponding sets of input data that have an influence on the output. The training model is used to run the input data through the algorithm to correlate the processed output against the sample output. Model training in machine language is the process of feeding the desired algorithm with data to help identify and learn good values for all attributes involved.
We applied 4 algorithms: Decision Trees, k-Nearest Neighbors, Logistic Regression and Random Forests. The performance metrics of the models are compared with each other, and we choose the best performing model in terms of speed and accuracy. Decision trees has achieved the best performance with its default parameters, so we will proceed by implementing it on the edge device.

4.3.1 Decision Trees

Decision trees use some cost function in order to choose the best split. We try to find the best attribute/feature that performs the best at classifying the training data. This process is repeated until a leaf node is reached and therefore, it is referred as recursive binary splitting.
Decision trees are upside down which means the root is at the top, and this root split into various several nodes. The objective of machine learning is to decrease the uncertainty in the dataset, so we introduce concepts like entropy, information gain and gini index. The main idea of a decision tree is to identify the features which contain the most information for the target feature, and then splits the dataset along the values of the features in a way that the target feature output is as pure as possible. Moving towards the downward direction leads to decreases in the level of uncertainty, and it yields in better classification split at each node.
i) Entropy defines the uncertainty in our dataset. It is a disorder measure. The equation of entropy in classification is the following:

$$Entropy = -\sum_{i=1}^{n} p_i * log_2(p_i)$$

Entropy basically measures the impurity of a node. Impurity is the degree of randomness which tells how random our data is. The entropy is calculated in a particular location without knowing the behavior of the entropy on the parent node for example. The concept of entropy plays an important role in calculating Information Gain. Information Gain is applied to quantify which feature provides maximal information about the classification based on the notion of entropy. This is done by calculating the uncertainty, disorder or impurity, with the intention of decreasing the amount of entropy starting from the top(root node) to the bottom(leaves nodes).

$$Gain(T, X) = Entropy(T) - Entropy(T, X)$$

The Gini index or Gini impurity measure is one of the methods used in decision tree algorithms to decide the optimal split from a root node and from subsequent splits. Lowering the Gini lowers the likelihood of misclassification.

$$GiniIndex = 1 - \sum_{i=1}^{n} (p_i)^2$$

Known as model tuning, there are important parameters in decision trees that impact the model in terms of over-fitting and under-fitting.
Max_depth indicates how deep the tree can be. Deep is the measuring of the number of splits. More splits capture more information about the data.
Min_samples_split represents the minimum number of samples required to split an internal node. This can vary between at least one sample at each node to considering all samples at each node.
min_samples_leaf defines the minimum number of samples required to be at a leaf node(the base of the tree).

max_features represents the number of features to consider when looking for the best split.

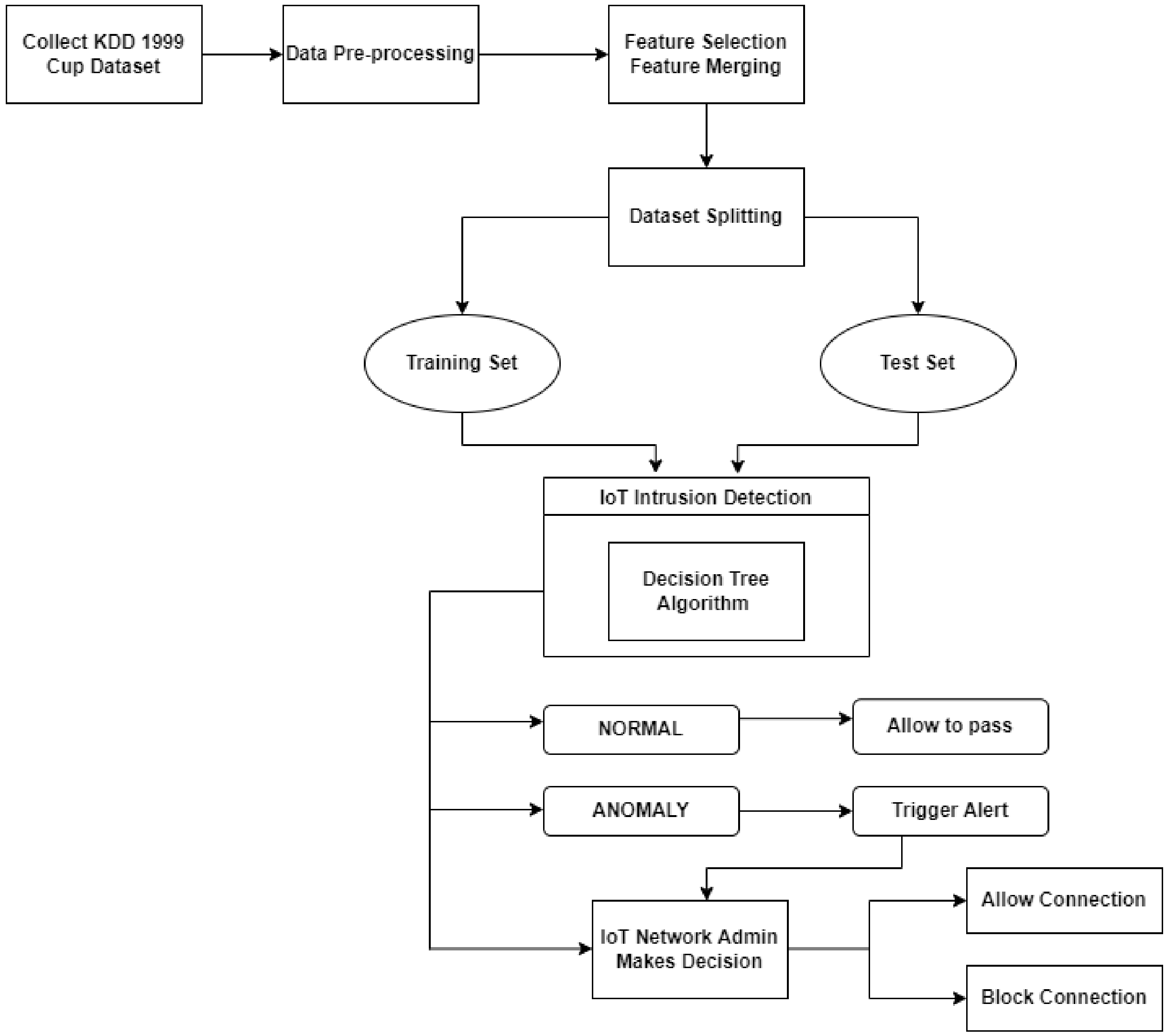

Figure 4.2. **Proposed IoT Anomaly Detection Method**

Chapter 5

Results

5.1 Model Testing & Evaluation

The final dimension of the dataset is 494021 rows and 11 columns. We split the dataset in a training set(80%) and a testing set(20%). The training set is the set that is fed to the model while the testing set is the set that allows to evaluate the model performance. To check the performance of the model, we calculate various performance metrics like accuracy, precision score, recall score and F1 score. Confusion matrix is a NxN matrix used for evaluating the performance of a classification model, where N is the number of target classes. The matrix compares the actual target values with those predicted by the model. The following metrics are derived from the confusion matrix(fig 5.1):

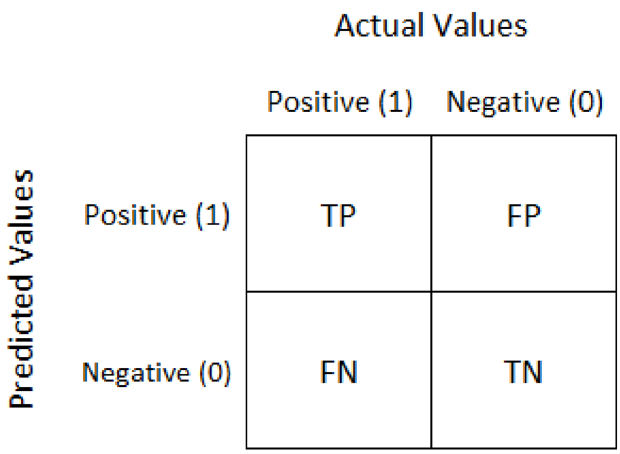

Figure 5.1. **Confusion Matrix**

a) Accuracy is the number of correctly predicted data points out of all the data points.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

b) Precision score is the ratio of correctly predicted positive data points to the total predicted positive data points.

$$Precision = \frac{TP}{TP + FP}$$

c) Recall(Sensitivity) is the ratio of correctly predicted positive data points to the all data points in actual class.

$$Recall = \frac{TP}{TP + FN}$$

d) F1 score is the weighted average of Precision and Recall.

$$Recall = \frac{2 * (Recall * Precision)}{(Recall + Precision)}$$

The Confusion matrix results are the following:
a) Decision Trees:

$$\begin{bmatrix} 19284 & 144 \\ 544 & 78833 \end{bmatrix}$$

b) K-NN:

$$\begin{bmatrix} 19206 & 222 \\ 610 & 78767 \end{bmatrix}$$

c) Logistic Regression:

$$\begin{bmatrix} 19066 & 362 \\ 1034 & 78343 \end{bmatrix}$$

d) Random Forest:

$$\begin{bmatrix} 19286 & 142 \\ 528 & 78849 \end{bmatrix}$$

ML algorithm	**Accuracy**	**Precision**	**Recall**	**F1 score**	**Test Time Complexity**	**Memory Size(MB)**
Decision Trees	99.3%	99.8%	99.31%	99.56%	O(D)	0.25
K-NN	99.13%	99.73%	99.18%	99.46%	O(K*N)	75.62
Logistic Regression	98.58%	99.54%	98.69%	99.11%	O(D)	0.0014
Random Forest	99.32%	99.57%	99.32%	99.82%	O(T*D)	19.23

Table 5.1. **Final Results**

Having achieved these results(table 5.1), we can see that Random Forest has achieved the highest accuracy of 99.31%(slightly higher than Decision Trees 99.3%). We need to choose a model in order to implement it on the edge device. The requirements for the model selection is to have a high accuracy and a fast runtime complexity. In terms of accuracy, we would proceed with Random Forest. The computational complexity at test time for a Random Forest of size T and maximum depth D is O(T * D). Prediction time complexity for k-NN is O(K * N) where N is the number of points in the dataset and K is the number of neighbors that we consider. The test time complexity of a decision tree would be O(D) where D is the depth since we have to move from root to a leaf node of the decision tree while the run time complexity of Logistic Regression is O(D) where w is the vector of weights of size D which is the fastest among the algorithms. We decide to proceed with the deployment of decision trees being the best trade-off choice.

Chapter 6

Conclusion and Future Work

In this thesis document, we apply an anomaly detection to predict cyberattacks for the Internet of Things(IoT) using machine learning(ML). Due to the rapid development of IoT systems in various domains, large amounts of data are constantly being generated, which requires an increased focus on privacy and security. Attacks in IoT include false data injection attacks(FDIA), denial-of-service(DoS) and distributed denial-of-service(DDoS), brute force and man-in-the-middle attacks. If these kinds of attacks succeed, IoT performance can be compromised in many ways such as giving false information. While in the past, traditional methods have been used for improving IoT security, the AI approach has many advantages which can be considered one of the most promising methods. We perform binary classification that refers to predicting one of two classes(normal or abnormal connection) on the KDD Cup 1999 dataset. Different machine learning algorithms are performed such as decision trees, k-nearest neighbors, logistic regression and random forest. We select decision tree to be the the best performing model according to optimal trade-off between accuracy(99.3%) and runtime complexity. Assessing the behaviour on a real case scenario, further extension would be to deploy the chosen machine learning model on the edge device. The preservation of the data context information is necessary to improve the quality of the application, so an improvement would be to create our own dataset in order to capture the new attack vectors for today's IoT.

Bibliography

[1] Alhussain Alwadain Aziz Capretz Abdullahi Baashar and Abdulkadir. «Detecting Cybersecurity Attacks in Internet of Things Using Artificial Intelligence Methods: A Systematic Literature Review». In: *Electronics* (2022).

[2] Nisar Haider Hassan Haque Tarmizi Rodrigues Ahmad Khan. «Anomaly detection using deep neural network for iot architecture». In: *Applied Sciences* 11 (2021).

[3] Rocky KC Chang. «Defending against flooding-based distributed denial-of-service attacks: A tutorial». In: *IEEE communications magazine* 40.10 (2002), pp. 42–51.

[4] C.Zhang. *Intelligent Internet of Things Service based on Artificial Intelligence Technology.* IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering.

[5] «KDD Cup 1999 Data». In: *The UCI KDD Archive* (1999).

[6] Pascal Maniriho, Ephrem Niyigaba, Zephanie Bizimana, Valens Twiringiyimana, Leki Jovial Mahoro, and Tohari Ahmad. «Anomaly-based intrusion detection approach for iot networks using machine learning». In: *2020 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM).* IEEE. 2020, pp. 303–308.

[7] Yair Meidan, Michael Bohadana, Asaf Shabtai, Martin Ochoa, Nils Ole Tippenhauer, Juan Davis Guarnizo, and Yuval Elovici. «Detection of unauthorized IoT devices using machine learning techniques». In: *arXiv preprint arXiv:1709.04647* (2017).

[8] Gautam Raj Mode, Prasad Calyam, and Khaza Anuarul Hoque. «False data injection attacks in internet of things and deep learning enabled predictive analytics». In: *arXiv preprint arXiv:1910.01716* (2019).

[9] Biswanath Mukherjee, L Todd Heberlein, and Karl N Levitt. «Network intrusion detection». In: *IEEE network* 8.3 (1994), pp. 26–41.

[10] K. Murphy. *Machine Learning: a Probabilistic Perspective.* MIT Press, 2012.

[11] Aubet Pahl. «All eyes on you: distributed multi-dimensional IoT microservice anomaly detection». In: pp. 451–466.

[12] Dr. Mukherjee Sahu. «Machine Learning based anomaly detection for IoT Network». In: IEEE. 2020.

[13] L. Tung. «Microsoft: Here's how we stopped the biggest ever DDoS attack». In: (2022).

[14] Hui Wu, Haiting Han, Xiao Wang, and Shengli Sun. «Research on artificial intelligence enhancing internet of things security: A survey». In: *Ieee Access* 8 (2020), pp. 153826–153848.

[15] Liang Xiao, Xiaoyue Wan, Xiaozhen Lu, Yanyong Zhang, and Di Wu. «IoT security techniques based on machine learning: How do IoT devices use AI to enhance security?» In: *IEEE Signal Processing Magazine* 35.5 (2018), pp. 41–49.

[16] Zhanyang Xu, Wentao Liu, Jingwang Huang, Chenyi Yang, Jiawei Lu, and Haozhe Tan. «Artificial intelligence for securing IoT services in edge computing: a survey». In: *Security and communication networks* 2020 (2020).

[17] Jin Ye, Xiangyang Cheng, Jian Zhu, Luting Feng, and Ling Song. «A DDoS attack detection method based on SVM in software defined network». In: *Security and Communication Networks* 2018 (2018).

[18] Saman Taghavi Zargar, James Joshi, and David Tipper. «A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks». In: *IEEE communications surveys & tutorials* 15.4 (2013), pp. 2046–2069.