People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

University of 8 May 1945 - Guelma -

Faculty of Mathematics, Computer Science and Science of Matter

Department of Computer Science

Master Thesis

Speciality: Computer Science

Option : Science and technology of information and communication

Theme

Analysis of the Impact of Dimensionality Reduction on the Accuracy and Performance of Intrusion Detection Systems in IoT Environments

Presented by :

ZEDOURI Amin

Supervised by :

DR. CHOHRA Chemseddine

Jury members :

DR. SOUSSI Hakim

DR. HANNOUSSE Abdelhakim

June 2023

Acknowledgment

Alhamdulillah, I am grateful for the journey that has brought me to this point, and for the knowledge and wisdom that I have gained along the way. I am particularly grateful to the good people who have supported and encouraged me throughout my journey.

Firstly, I would like to express my sincere gratitude to my supervisor, **Dr. Chemseddine CHOHRA**, for his unwavering support, guidance, and encouragement throughout the research process. His vast knowledge and expertise have been invaluable, and his unwavering dedication to my success has been a source of motivation for me, and I am grateful for the opportunity to work with you.

I would also like to extend my gratitude to the esteemed teachers who have provided me with insightful advice and guidance. Their wisdom and knowledge have helped me overcome many challenges and have shaped me into the person I am today.

I am grateful to my family for their unwavering support, encouragement, and understanding throughout my academic journey. Their love, motivation, and constant belief in me have been a great source of strength and inspiration.

Finally, I would like to dedicate my success and all my accomplishments to my parents, who have been the driving force behind everything I do, their unwavering love and support have been a constant source of inspiration for me, and I am forever grateful for everything they have done for me. I dedicate my success and all my accomplishments to them.

Thank you to everyone who has played a role in my journey, and I look forward to the opportunities that lie ahead.

ABSTRACT

The Internet of Things (IoT) has transformed technology by facilitating seamless communication and data exchange among interconnected devices. However, this increased connectivity poses security challenges, necessitating intrusion detection systems (IDS) to protect IoT environments. This study examines the influence of dimensionality reduction methods on IDS accuracy and performance in IoT. We analyze various dimensionality reduction techniques and their impact on IoT intrusion detection systems. Four machine learning models (linear regression, decision tree, SVM, MLP) are implemented with principal component analysis (PCA) as the chosen reduction method. The IoTID20 dataset is used for training and testing. Comparative evaluations with existing algorithms measure accuracy, F1-score, fit time, and score time. Results reveal that PCA significantly reduces training time without significant accuracy loss. This research offers insights into the impact of dimensionality reduction on IDS performance in IoT, highlighting PCA's advantages in optimizing training time.

Keywords: Internet of Things, security, intrusion detection systems (IDS), machine learning, dimensionality reduction methods, IoT environments, IoTID20.

RÉSUMÉ

L'internet des objets (IdO) a transformé la technologie en facilitant la communication transparente et l'échange de données entre les appareils interconnectés. Cependant, cette connectivité accrue pose des problèmes de sécurité, nécessitant des systèmes de détection d'intrusion (IDS) pour protéger les environnements de l'IdO. Cette étude examine l'influence des méthodes de réduction de la dimensionnalité sur la précision et la performance des IDS dans l'IdO. Nous analysons diverses techniques de réduction de la dimensionnalité et leur impact sur les systèmes de détection d'intrusion dans l'IdO. Quatre modèles d'apprentissage automatique (régression linéaire, arbre de décision, SVM, MLP) sont mis en œuvre avec l'analyse en composantes principales (ACP) comme méthode de réduction choisie. L'ensemble de données IoTID20 est utilisé pour la formation et les tests. Les évaluations comparatives avec les algorithmes existants mesurent la précision, le score F1, le temps d'adaptation et le temps de score. Les résultats révèlent que l'ACP réduit considérablement le temps de formation sans perte significative de précision. Cette recherche donne un aperçu de l'impact de la réduction de la dimensionnalité sur les performances des IDS dans l'IdO, en soulignant les avantages de l'ACP dans l'optimisation du temps de formation.

Mots-clés : Internet des objets, sécurité, systèmes de détection d'intrusion (IDS), apprentissage automatique, méthodes de réduction de la dimensionnalité, environnements IoT, IoTID20.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

GENERAL INTRODUCTION

The proliferation of Internet of Things (IoT) devices has witnessed a remarkable surge in recent years, revolutionizing various aspects of our daily lives. These interconnected devices offer unprecedented convenience and connectivity, enabling seamless communication and data exchange. However, the rapid expansion of IoT also brings forth significant security challenges that demand careful attention.

The security challenges associated with IoT stem from the vast network of interconnected devices, creating a fertile ground for potential security breaches and vulnerabilities. As IoT devices continue to permeate different domains, such as healthcare, transportation, and smart homes, ensuring the security and protection of these systems has become a critical concern.

To address the security challenges of IoT, machine learning techniques have emerged as powerful tools for intrusion detection. By leveraging advanced algorithms and data analysis, machine learning models can effectively detect and identify potential intrusions in real-time. These models have the ability to adapt and learn from new data, making them well-suited for identifying anomalous patterns and behaviors that may indicate unauthorized access or malicious activities within IoT environments.

However, the exponential growth of IoT also poses limitations in terms of storage and computational resources. The sheer volume of data generated by IoT devices can overwhelm traditional storage systems and hinder efficient analysis. To mitigate these challenges, dimension reduction techniques come into play. Dimension reduction aims to reduce the complexity and size of the data while preserving its essential features. By reducing the dimensionality of the data, these techniques enable more efficient storage and processing, thereby enhancing the scalability and performance of IoT systems.

We delve into the security challenges of IoT and explore the potential of machine learning and dimension reduction techniques to address these challenges. We investigate the applicability and effectiveness of various machine learning algorithms in detecting intrusions within IoT networks. Furthermore, we analyze different dimension reduction techniques to optimize the storage and processing of IoT data, considering their impact on the accuracy and performance of intrusion detection models.

By understanding the limitations of IoT in terms of storage and computational resources and exploring dimension reduction techniques, we aim to enhance the security and efficiency of IoT systems. Through our research, we seek to contribute to the development of more robust and scalable IoT environments that can withstand emerging security threats while maximizing the benefits of interconnected devices.

In our study, we aim to explore the influence of dimension reduction techniques on the performance and accuracy of machine learning algorithms for intrusion detection in Internet of Things (IoT) systems. Specifically, we focus on evaluating the effectiveness of principal component analysis (PCA) as a dimension reduction method and examine its impact on enhancing the performance of popular machine learning algorithms used for intrusion detection in IoT environments.

This dissertation is organized as follows:

— **Chapter 01:** Machine learning and Security of IoT.

— **Chapter 02:** Dimensionality reduction techniques.

— **Chapter 03:** Architecture and Implementation.

— **Chapter 04:** Results and Discussion.

CHAPTER 1 MACHINE LEARNING AND SECURITY OF IOT

1.1 Introduction

Machine learning and security of Internet of Things (IoT) have become critical research areas in the field of computer science and technology. The increasing number of connected devices, ranging from smart homes to healthcare systems, has resulted in the creation of large amounts of data that can be analyzed and used to improve decision making and the overall user experience.

However, with the increasing reliance on IoT devices, the security of these systems has become a major concern. These devices are often connected to the Internet, making them vulnerable to various cyber threats such as hacking, malware attacks, and data breaches. This is where machine learning comes in. Machine learning algorithms can be used to identify and mitigate these threats in real-time, improving the overall security of IoT systems.

The use of machine learning in IoT security can help to address challenges such as detecting and preventing unauthorized access to devices, detecting anomalies in network traffic, and classifying different types of security threats. This allows for a more proactive approach to security, as opposed to traditional security methods which are often reactive. For example, machine learning can be used to develop intrusion detection systems that monitor network traffic and identify potential security threats. Machine learning can also be used to develop authentication systems that use biometric data, such as facial recognition or fingerprint scans, to verify the identity of users and devices.

1.2 Machine Learning

According to Arthur Samuel, machine learning refers to the field of study that enables computers to learn without requiring explicit programming. Samuel gained fame for his checkers playing program.

Machine learning (ML) is employed to enhance the efficiency of data handling by machines. In some cases, we are unable to extract meaningful information from data through conventional means. This is where machine learning comes into play. Given the vast availability of datasets, the demand for machine learning is increasing. Numerous studies have been conducted on enabling machines to learn autonomously, without explicit programming [1].

1.2.1 Supervised Learning

Supervised Learning plays a crucial role in machine learning. It derives its name from the fact that the learning process relies on labeled observation variables. In this type of learning, datasets are trained using training sets to construct a machine learning model. This model is then used to label new observations from a testing set. The training set consists of input variables, known as features, which greatly influence the accuracy of the predicted variable. It encompasses both quantitative and qualitative variables. The output variable represents the label class assigned by the Supervised Learning model to new observations. Based on the nature of the output variables, Supervised Learning tasks can be categorized into two types: classification tasks and regression tasks. Classification tasks involve categorical output variables, while regression tasks involve continuous output variables. For instance, classifying images as "hot" or "not hot" represents a classification task, whereas predicting stock prices corresponds to a regression task. The procedure of Supervised Learning can be described as follows: we denote the input variables as $x(i)$ and the output variable as $y(i)$. A pair $(x(i), y(i))$ represents a training example, and the training set used for learning is denoted as $(x(i), y(i)), i = 1, 2, \ldots, m$. Here, the index i refers to an element in the training set. The space of input values is denoted as X, while Y represents the space of output values. The ultimate goal is to learn a function

$$h : X \to Y$$

so that $h(x)$ serves as an effective predictor for the corresponding value of y. In this context, h is referred to as a hypothesis [2].

1.2.2 Unsupervised Learning

Unsupervised learning is a type of machine learning where the algorithm is given a data without labels or known outputs, and the objective is to discover patterns and relationships in the data. Unlike supervised learning, where the algorithm is trained on labeled data to predict the output for new inputs, The algorithm is tasked with grouping or clustering the data points into natural categories or clusters, based on similarities and differences in the data. The goal of unsupervised learning is often to uncover hidden structures or representations in the data that can be used to gain insights or make predictions about new, unseen data.

1.2.3 Reinforcement Learning

Reinforcement learning is an online learning technology that differs from supervised learning and unsupervised learning. In reinforcement learning, the environment provides a reinforcement signal, which evaluates the quality of actions taken by an intelligent agent. The intelligent agent has the ability to sense or perceive its environment using sensors, which can be physical (e.g., cameras, microphones) or virtual (e.g., data feeds, API calls) depending on the agent and the operating environment.

Unlike other learning methods, the reinforcement signal does not explicitly instruct the intelligent agent on how to generate the correct action. Due to the limited information provided by the external environment, the intelligent agent must rely on its own experience to learn. Through this learning process, the agent acquires an appropriate appraisal value for the environment state and adjusts its action strategy to adapt to the environment.

The intelligent agent continuously interacts with the environment, perceiving the environment and selecting actions to maximize the reward value. The interactive interface between the intelligent agent and the environment consists of actions, rewards, and states. Each time the reinforcement learning system interacts with the environment, it first receives the input of the current environment state (s). Then, based on internal inference mechanisms, it outputs an action (a) that interacts with the environment. Consequently, the environment transitions to a new state (s') after accepting the action. The system then receives the input of the new state (s') and obtains the reward and punishment signal (r) from the environment [3].

1.3 Machine Learning Algorithms

Machine learning algorithms are a set of techniques that enable computer systems to learn and improve their performance on a specific task without being explicitly programmed. These algorithms use statistical models and data to recognize patterns, make predictions or decisions, and perform various tasks such as classification, regression, clustering, and anomaly detection. We list below some of the most important and widely used machine learning algorithms.

1.3.1 Decision Tree Algorithms

A decision tree is a graphical representation of choices and their outcomes, organized in the form of a tree structure. The nodes in the tree represent events or decisions, while the edges represent the decision rules or conditions. A decision tree consists of nodes and branches. Each node represents a group of attributes that are being classified, and each branch represents a possible value that the node can take [1].

1.3.2 Logistic Regression

Logistic regression is a classification technique used to predict binary or multinomial outcomes based on input variable values. It is commonly used for tasks such as predicting tumor malignancy, classifying spam emails, or determining preferred cuisine type. Unlike linear regression, which predicts continuous variables, logistic regression deals with categorical target variables. It offers advantages such as ease of implementation, computational efficiency, and regularization. Input feature scaling is not required, making it suitable for industry-scale problems. However, logistic regression has limitations, including its inability to solve non-linear problems and its susceptibility to overfitting. Additionally, all independent variables must be identified for it to work effectively. Practical applications of logistic regression include disease risk prediction, cancer diagnosis, mortality prediction, and failure probability estimation in engineering processes [4].

1.3.3 Support Vector Machine

Another extensively employed and advanced machine learning technique is Support Vector Machine (SVM). SVM is a supervised learning model with associated learning algorithms used for classification and regression analysis in machine learning [1]. While logistic regression models the probability of the output class, SVM focuses on discovering the decision boundary that maximizes the margin between the classes. SVM goes beyond modeling the probability of the output classes and aims to find the decision boundary that maximizes the separation between the classes.

1.3.4 Neural Networks

A neural network is a sequence of algorithms designed to identify underlying relationships within a dataset by emulating the functioning of the human brain. Neural networks can be comprised of either organic or artificial neurons. One key feature of neural networks is their ability to adapt to changing inputs, enabling them to generate optimal results without requiring a redesign of the output criteria. Originating from the field of artificial intelligence, the concept of neural networks is rapidly gaining traction in the development of trading systems [1].

1.3.5 K-Nearest Neighbor

The k-nearest neighbors (KNN) algorithm is a straightforward and supervised approach for machine learning that can tackle classification and regression problems. Although the method is uncomplicated and easily comprehensible, a significant drawback is that its performance slows down considerably as the data size increases [1].

1.3.6 K-Means Clustering

K-means is an unsupervised machine learning algorithm that is designed to address the clustering problem. This algorithm is relatively simple and straightforward, as it involves dividing a dataset into a specified number of clusters. The fundamental concept behind this approach is to define k centers, with each center representing a different cluster. It is crucial to strategically select the placement of these centers, as different locations can result in different outcomes. Therefore, it is best to position them as far apart from one another as possible [1].

1.3.7 Dimensionality Reduction Algorithms

Dimensionality reduction are machine learning techniques that are commonly used to reduce the size of a large dataset by identifying the most informative components and representing them with fewer features. This enables a more effective visualization of data with high dimensionality and helps to improve the efficiency of supervised classification. Examples of these techniques include Principal Component Analysis (PCA), Principal Component Regression (PCR), Partial Least Squares Regression (PLSR), Sammon Mapping, Multidimensional Scaling (MDS), Projection Pursuit, Linear Discriminant Analysis (LDA), Mixture Discriminant Analysis (MDA), Quadratic Discriminant Analysis (QDA), and Flexible Discriminant Analysis (FDA) [5]. We will get into more details about dimensionality reduction techniques in the chapter 2.

1.4 Security of IoT

1.4.1 Internet of Things

The term ”Internet of things” was coined by Kevin Ashton of Procter and Gamble, later MIT’s Auto-ID Center, in 1999. Since then, the Internet of Things (IoT) has rapidly evolved into a field involving smart objects’ interconnection and interaction [6]. IoT refers to a network of objects where everything can be identified and connected to the Internet through some kind of communication and computing device such as RFID, sensor, actuators, and mobile phone [7]. The growth of the IoT is driven by advances in connectivity, sensors, cloud computing, and analytics, as well as by the increasing affordability and accessibility of these technologies. As the IoT continues to expand, it will create new opportunities for innovation and disruption, as well as new challenges related to privacy, security, and data management.

As devices rely on different wireless technologies to communicate with each other, a multitude of security and privacy issues emerge, such as maintaining confidentiality, integrity, authenticity, and privacy. Furthermore, due to their limited energy, computation, and communication resources, IoT devices are highly susceptible to security and privacy attacks [7].

1.4.2 Usage Environments

The Internet of Things (IoT) has transformed the way we live, work, and interact with technology. IoT devices are designed to collect, exchange, and analyze data from the physical world, allowing for unprecedented levels of automation, efficiency, and convenience. The applications of IoT are vast and diverse, ranging from smart homes and cities to industrial automation and healthcare. IoT devices are used in various environments, such as homes, offices, factories, transportation systems, and public spaces. However, as IoT becomes increasingly ubiquitous, new challenges emerge, such as security and privacy concerns, interoperability issues, and the ethical implications of using and sharing personal data. Therefore, understanding the usage environments of IoT is essential for developing effective solutions that can maximize the benefits of this technology while minimizing the risks.

Healthcare

IoT technologies have significant potential to transform the healthcare industry by improving patient care, increasing efficiency, and reducing costs. The applications of IoT in healthcare can be broadly categorized into tracking of objects and people, identification and authentication of people, and automatic data collection and sensing, assist patients with chronic conditions by providing real-time feedback and reminders for medication or physical therapy. One of the primary uses of IoT technology in healthcare is to track objects and people. This includes tracking the location and movement of medical equipment, medication, and other supplies within a healthcare facility. It also involves tracking the location of patients, staff, and visitors to improve security and optimize workflows. Wearable IoT devices can also be used to monitor the health of patients, providing real-time data on vital signs, activity levels, and other metrics that can be used to monitor their health and track their recovery progress [8].

Smart Environments

A smart environment is a space that is designed to be easy and comfortable to use, thanks to the intelligence of the objects within it. This can apply to a wide range of settings, including offices, homes, industrial plants, and leisure environments. In a smart environment, objects are equipped with sensors, processing power, and connectivity, which allows them to interact with each other and with the environment in a seamless and intuitive way. For example, in a smart home, lights, thermostats, and other devices can be controlled through a single app or voice commands, making it easier and more convenient for users to manage their environment. Similarly, in a smart office, sensors can detect when rooms are occupied and adjust lighting and temperature settings accordingly, improving energy efficiency and creating a more comfortable working environment. Overall, smart environments are designed to enhance the user experience by making everyday tasks easier and more intuitive, while also improving efficiency and reducing energy consumption. In an industrial plant setting, a smart environment can include sensors that monitor the condition of machinery and alert maintenance personnel when repairs are needed. This can improve efficiency and reduce downtime.

Industrial

In the industrial sector, also known as IIoT (Industrial Internet of Things), has been rapidly growing in recent years. IIoT refers to the use of connected devices and technologies to enhance and optimize industrial processes, such as manufacturing, logistics, and supply chain management. One key application of IIoT is predictive maintenance, which involves using IoT sensors and data analytics to monitor industrial equipment and predict when maintenance or repairs are needed. Another important application is asset tracking, which involves using IoT sensors to monitor the location and condition of industrial assets, such as inventory, vehicles, and equipment. also used to optimize industrial processes and workflows, such as by automating routine tasks and providing real-time insights into performance metrics.

Transportation and Logistics

The transportation and logistics industry is rapidly adopting IoT technology to improve efficiency, safety, and sustainability. Advanced vehicles such as cars, trains, buses, and bicycles are being equipped with sensors, actuators, and processing power that enable real-time monitoring of their location, speed, fuel consumption, and other vital parameters. This information is transmitted to traffic control sites and transportation vehicles to optimize routing, reduce congestion, and improve the overall flow of traffic. Additionally, roads and rails are being instrumented with sensors that can detect traffic density, road conditions, and weather patterns, allowing for proactive maintenance and timely intervention to prevent accidents and improve safety. IoT is also being used to improve logistics management. Goods are being equipped with tags and sensors that provide real-time data on their location, temperature, humidity, and other relevant parameters. This data is transmitted to transportation vehicles and logistics management systems, allowing for real-time tracking and monitoring of the status of goods. This enables logistics providers to optimize their operations, reduce inventory costs, and improve customer satisfaction by providing timely and accurate delivery updates [8].

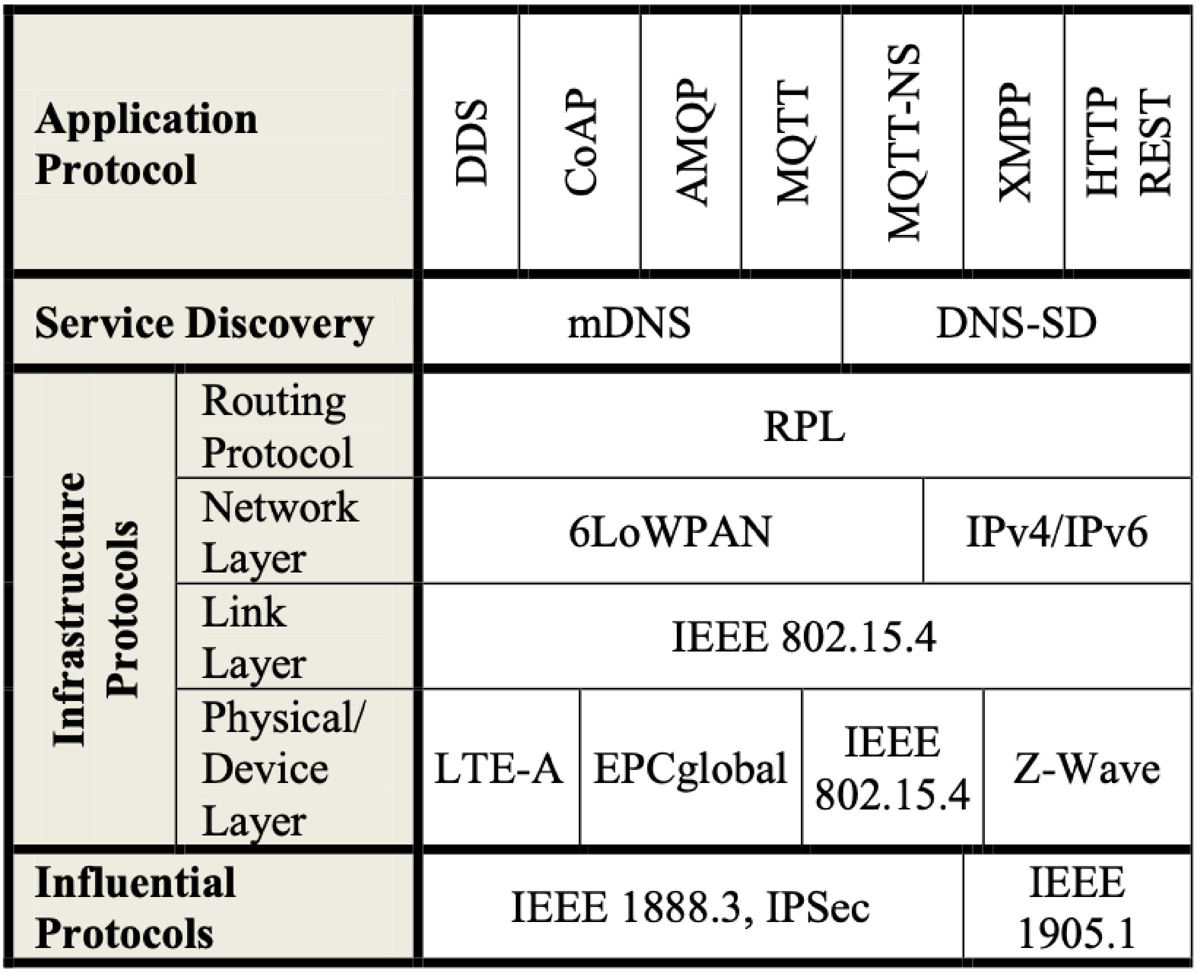
Personal and Social Domain

The personal and social domain in IoT refers to the use of connected devices and technologies to enhance and facilitate personal and social interactions. IoT devices can be used in wearable devices, such as fitness trackers and smartwatches, which can monitor and track various aspects of a person's health and fitness. They can also be used to facilitate social interactions, by enabling people to connect and communicate with each other more easily. For example, smart home devices can be used to control and automate various aspects of a person's home, such as lighting, heating, and security. In addition, IoT devices can also be used to enhance social experiences outside of the home. For example, smart city technologies can be used to facilitate public gatherings and events, such as by providing real-time information on traffic and parking, or by enabling people to connect and share information with each other through social media and other platforms [8].

1.4.3 Communication Protocols

The Internet of Things (IoT) is a rapidly growing network of interconnected devices, ranging from smartphones and wearables to industrial machinery and smart home appliances. These devices use a variety of communication protocols to exchange data and interact with each other. Communication protocols are essentially sets of rules that define how devices transmit, receive, and interpret data. In IoT, communication protocols play a crucial role in ensuring that devices can seamlessly and securely connect and communicate with each other. The choice of protocol depends on a variety of factors such as the type of device, the network infrastructure, the desired level of security, and the specific application requirements. In this context, understanding the different communication protocols used in IoT is essential for developers, engineers, and designers who are involved in building and deploying IoT systems.

The IoT protocols can be divided into four main groups: application protocols, service discovery protocols, infrastructure protocols, and other influential protocols. It's not necessary to use all of these protocols together for a specific IoT application. Additionally, depending on the type of IoT application, certain standards may not be necessary to support [9].



Application Protocol		DDS	CoAP	AMQP	MQTT	MQTT-NS	XMPP	HTTP REST
Service Discovery		mDNS	DNS-SD					
Infrastructure Protocols	Routing Protocol	RPL						
	Network Layer	6LoWPAN	IPv4/IPv6					
	Link Layer	IEEE 802.15.4						
	Physical/ Device Layer	LTE-A	EPCglobal	IEEE 802.15.4	Z-Wave			
Influential Protocols		IEEE 1888.3, IPSec	IEEE 1905.1					

FIGURE 1.1: Table provides a summary of the most prominent protocols defined by these groups

1.4.4 Security

Security Challenges

User Privacy and Data Protection Due to the omnipresence of IoT environment, user privacy is a critical concern in IoT security. With devices interconnected and data transmitted and exchanged over the internet, preserving user privacy has become a sensitive subject in numerous research studies. Despite the ample research conducted on the matter of privacy, there are still many areas that require further exploration. Topics such as privacy in data collection, sharing, and management, as well as data security, continue to be open research issues that need to be addressed [10].

Authentication and Identity Management Authentication and Identity Management (IdM) encompass a collection of processes and technologies aimed at managing and securing access to resources and information, as well as protecting object profiles. IdM plays a vital role in uniquely identifying objects, while authentication verifies the establishment of identity between two communicating parties. In the context of the Internet of Things (IoT), managing identity authentication becomes crucial due to the need for multiple devices and users to authenticate each other utilizing reliable services. This involves developing an effective identity management approach to ensure the unique identification of all objects. Furthermore, factors such as mobility, privacy, pseudonymity, and anonymity necessitate in-depth analysis and further research [10].

Trust Management and Policy Integration Trust plays a crucial role in establishing secure communication among devices in the uncertain environment of the Internet of Things (IoT). It is important to consider trust from a user's perspective to establish trust in interactions between entities and the system. In the field of IoT, the key objectives of trust research include developing new decentralized trust models, implementing trust mechanisms for cloud computing, and creating applications based on node trust. Automated and preferably autonomous trust evaluation is essential, and the reputation-based Subjective Logic (SL) approach shows promise in this regard.

Trust can be transitive between systems but should be governed by agreements. A trust device should possess the capability to prevent subversion, and a robust policy framework is necessary to incorporate the evaluated trust level and current threat level when making decisions [10].

Authorization and Access Control Authorization determines whether a person or object, once identified, is permitted to access a resource, while access control regulates resource access by granting or denying it based on various criteria. Access control is often implemented using authorization mechanisms. Both authorization and access control are crucial in ensuring a secure connection between multiple devices and services. In this scenario, the primary concern is to simplify the creation, comprehension, and manipulation of access control rules. Further details on access control are provided below [10].

End-to-End Security Endpoint security between IoT devices and Internet hosts is a critical concern. Simply employing cryptographic techniques such as encryption and authentication codes for packet transmission is inadequate for resource-constrained IoT devices. Achieving complete end-to-end security requires verifying the identity of both endpoints, using protocols to dynamically negotiate session keys (e.g., TLS and IPsec), and securely implementing algorithms such as AES and Hash. In an IoT system with end-to-end security, both endpoints can assume that their communication remains private and that data in transit cannot be tampered with by any third party. Ensuring correct and comprehensive end-to-end security is essential as many IoT applications would be impossible without it [10].

Attack Resistant Security Solution Given the diverse range of devices in the internet of things, with varying levels of memory and computation resources, it is crucial to provide lightweight and attack-resistant security solutions to protect these devices from potential attacks. In addition, mitigation measures should be implemented on the devices themselves to defend against external attacks, such as denial-of-service and flood attacks [10].

Security Attacks

1. Physical attacks on IoT systems aim at targeting the hardware components of the system, typically necessitating the attacker to be physically close or within the system's vicinity. These attacks can cause damage to the hardware's functionality or even shorten its lifespan. Various forms of physical attacks exist, including node tampering, RF interference on RFIDs, node jamming in wireless sensor networks (WSNs), malicious node injection, physical damage, social engineering, sleep deprivation attack, and malicious code injection. Node tampering involves physically replacing or modifying a sensor node, while social engineering manipulates users of the IoT system for the attacker's advantage. Sleep deprivation attacks and malicious code injection compromise both the functionality and security of the IoT system.

2. Network attacks in IoT systems are focused on exploiting vulnerabilities in the network. The attacker doesn't need to be physically close to the network to launch an attack. These attacks include traffic analysis, where confidential data is intercepted using sniffing applications like packet sniffers. Other attacks involve RFID technology, such as spoofing and cloning, which allow the attacker to gain access to the system or replicate a victim's RFID tag. Unauthorised access and sinkhole attacks breach confidentiality and deny service by dropping packets. The man-in-the-middle attack involves the attacker intercepting communication between two sensor nodes, while the Sybil attack involves a malicious node claiming the identities of multiple nodes to deceive the system. Denial of service attacks involve overwhelming the network with more traffic than it can handle. Routing information attacks can manipulate the network by spoofing or

altering routing information, causing complications such as routing loops and false error messages.

3. Software attacks are the primary cause of security vulnerabilities in computer systems, including IoT systems. These attacks involve the use of malicious software such as viruses, worms, Trojan horses, spyware, and malicious scripts to steal information, tamper with data, and deny service. Phishing attacks involve tricking users into divulging their authentication credentials through emails or fake websites. Malicious scripts can be used to shut down systems or steal data. Denial of service attacks can be executed on IoT networks, blocking legitimate users and allowing attackers access to sensitive data.

4. Encryption-based attacks on IoT systems aim to break the encryption scheme being used to secure the data transmission. These attacks include side-channel attacks, which involve retrieving the encryption key by analyzing the encryption devices through techniques like timing, power, fault, and electromagnetic analysis. Cryptanalysis attacks assume the possession of plaintext or ciphertext to find the encryption key, with examples like known-plaintext attacks and chosen-ciphertext attacks. A man-in-the-middle attack occurs when an adversary intercepts and interferes with the key exchange between two IoT system users, allowing the attacker to decrypt and encrypt any data being transmitted [11].

1.5 Conclusion

In this first chapter we have seen an overview of the Internet of Things, including its definition, its applications, and its different components. Security is a major concern in the Internet of Things, and we have seen the different threats that can be faced by these systems. We have also seen the different security mechanisms that can be used to protect these IoT devices from different attacks. We have also outlined in this chapter the different machine learning techniques that might be used to detect attacks in IoT systems. In the next chapter we focus on dimensionality reduction techniques, and we will see how these techniques can be used to improve the performance of machine learning algorithms.

CHAPTER 2 DIMENSIONALITY REDUCTION TECHNIQUES

2.1 Introduction

Dimensionality reduction is a crucial technique in the field of machine learning, which refers to the process of reducing the number of features in a dataset while retaining as much of the relevant information as possible. As datasets continue to grow in size and complexity, dimensionality reduction techniques have become increasingly important for reducing computational costs, improving model accuracy, and gaining a better understanding of the underlying structure of data. There are various approaches to dimensionality reduction, ranging from linear methods such as Principal Component Analysis (PCA) to nonlinear methods such as t-Distributed Stochastic Neighbor Embedding (t-SNE). These techniques can be applied to a wide range of applications, from image and signal processing to natural language processing and data visualization. In this dissertation, we explore the principles and applications of various dimensionality reduction techniques and their impact on machine learning performance.

2.2 Dimensionality Reduction Techniques

The increasing amount of data stored in high-dimensional data (HDD) has led to a common use of various dimensionality reduction (DR) techniques in many application areas. These techniques transform high-dimensional data into a new dataset that represents low dimensionality while preserving the original data's meaning as much as possible [12]. The two primary types of DR techniques are feature selection and feature extraction. Feature selection identifies the most relevant features to improve model performance by selecting a subset of the original features, while feature extraction transforms the original features into a lower-dimensional space by identifying new features that capture the most important information.

2.2.1 Feature Selection

Feature selection is a technique used to reduce the impact of dimensionality on a dataset by identifying a subset of features that efficiently represent the data. It involves selecting the most important and relevant features for a particular data mining task, while eliminating redundant and irrelevant features. Feature selection helps to identify an optimal subset of features that is appropriate for a given problem. The primary goal of feature selection is to create a small subset of features that represent the essential features of the input data, while minimizing the overall size of the dataset [13]. There are various ways to classify feature selection methods, with the most common being filters, wrappers, embedded, and hybrid methods. an additional type of evaluation method that has emerged recently is known as ensemble feature selection[13]. However, this classification is based on the assumption of feature independence or near-independence. For datasets with structured features that have dependencies, and for streaming features, additional methods have been developed. These methods extend the conventional classification to cover a wider range of scenarios and help to select the most appropriate subset of features for a given problem [14]. The hierarchy of feature selection techniques is illustrated in Figure 2.1.

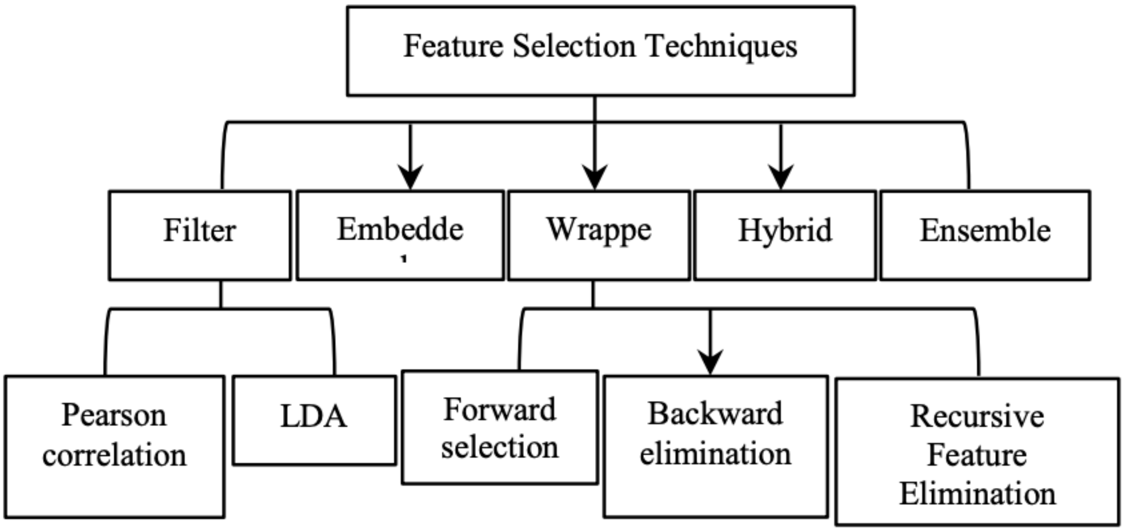


FIGURE 2.1: The hierarchy of feature selection techniques

The Filter Method

It is one of the earliest feature selection techniques and is also known as an open-loop method. It evaluates the features based on their intrinsic characteristics before the learning tasks are performed. This method mainly uses four different measurement criteria, namely information, dependency, consistency, and distance, to assess the feature characteristics. The filter method performs feature selection independently of the data mining algorithm and employs statistical standards to evaluate the ranking of the selected subset. It is highly efficient and scalable in high-dimensional datasets, and has been found to outperform the wrapper technique. However, the primary drawback of this method is that it does not consider the interaction between the selected subset and the performance of the induction algorithm [13].

The Wrapper Method

Also known as a close-loop method, integrates feature selection with the learning algorithm and uses the performance accuracy or classification error rate as a criterion for feature evaluation. It selects the most discriminative subset of features by reducing the estimation error of a specific classifier. Despite the advantages of achieving better performance and high accuracy compared to the filter method [13], wrappers are slower in finding good subsets due to their dependence on the resource demands of the modelling algorithm. Additionally, feature subsets are biased towards the modelling algorithm used for evaluation, even with cross-validation. Therefore, it is necessary to use an independent validation sample and another modelling algorithm to obtain a reliable generalization error estimate after finding the final subset. Empirical studies have shown that wrapper methods obtain subsets with better performance than filter methods because the subsets are evaluated using a real modelling algorithm [14].

The Embedded Method

Is a feature selection mechanism that integrates feature selection within the learning algorithm, using its properties to guide feature evaluation. Unlike the wrapper method, it does not require the repeated execution of the classifier or the examination of every feature subset, making it more efficient and tractable. The embedded method combines the advantages of both the filter and wrapper methods, selecting features during the implementation of the mining algorithm. As a result, it is computationally less expensive while maintaining similar performance to the wrapper method [13].

The Hybrid Method

Have been introduced to merge the advantageous features of filters and wrappers. The hybrid method usually starts with a filter approach to decrease the feature space dimension and to generate some possible candidate subsets. After that, a wrapper approach is applied to identify the optimal candidate subset. The hybrid method is renowned for providing high accuracy, which is a characteristic of the wrapper method, and high efficiency, which is a characteristic of the filter method [14]. By integrating both methods, the hybrid approach inherits the complementary strengths of filters and wrappers. The combination of filter and wrapper techniques is the most prevalent approach to building hybrid methods [13].

The Ensemble Method

Is a technique that seeks to create a collection of feature subsets and subsequently generate a consolidated outcome from the collection. This approach relies on multiple subsampling techniques, where a specific feature selection method is applied to several subsamples, and the resulting features are combined to form a more robust subset [13].
Feature selection offers several benefits [13] :

- reducing the size of the data.
- minimizing storage requirements.
- improving prediction accuracy.
- preventing overfitting.
- reducing the execution and training time by simplifying the variables.

2.2.2 Feature Extraction

Feature extraction is a type of dimensionality reduction technique that transforms high-dimensional data into a lower-dimensional space by extracting the most relevant features from the original data. In other words, it involves deriving new features from the existing set of features, which are more informative and discriminative for the machine learning algorithm. Feature extraction is particularly useful when the original data has a large number of features, some of which may be irrelevant or redundant, making it difficult to train a machine learning model. By reducing the number of features while preserving the relevant information, feature extraction can improve the performance of machine learning models, reduce computational costs, and alleviate the curse of dimensionality. Popular feature extraction techniques include Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), and Independent Component Analysis (ICA).

Principal Component Analysis

Principal Component Analysis (PCA) is widely recognized as one of the most commonly used algorithms for dimensionality reduction. It is employed to find the most optimal subspace of a given dimension, m, within a set of observations x with a dimension of M, where the objective is to minimize the least-square error. For simplicity, it is often assumed that the data is zero-mean, and the subspace to be fitted is a linear subspace passing through the origin. The essence of PCA lies in searching for orthogonal directions that can explain the maximum variance of the data. By identifying these directions, PCA effectively reduces the dimensionality of the data while preserving the most critical information. [15].

Linear Discrimant Analysis

Linear Discriminant Analysis (LDA) is a statistical technique utilized for both classification and dimensionality reduction purposes. Its primary objective is to identify a linear combination of features that maximizes the separation between multiple classes in a dataset. LDA achieves this by projecting high-dimensional data onto a lower-dimensional space while preserving the discriminative information among different classes to the greatest extent possible. Essentially, LDA aims to find a lower-dimensional representation of the data that retains the essential information necessary for distinguishing between various classes. This approach is widely employed in machine learning and pattern recognition, finding applications in diverse fields such as image recognition, bioinformatics, and natural language processing.

Autoencoder

An autoencoder is an artificial neural network architecture specifically developed to learn efficient representations of input data. It achieves this by encoding the data into a lower-dimensional latent space and subsequently decoding it back to its original form. The primary objective of an autoencoder is to minimize the reconstruction error between the input and output, which is accomplished through the use of a hidden layer stack that reduces dimensionality.

Autoencoders come in various forms, including Sparse Autoencoder, Variational Autoencoder, Denoising Autoencoder, and Relational Autoencoder. These variants highlight the versatility of autoencoders in extracting meaningful features from data. By leveraging the power of artificial neural networks, autoencoders have proven to be effective in dimensionality reduction and feature extraction tasks [16].

t-Stochastic Neighbor Embedding

t-Stochastic Neighbor Embedding (t-SNE), developed by Hinton and Roweis, is a non-linear dimensionality reduction technique (NLDRT) employed for reducing high-dimensional data to a lower-dimensional space. Unlike other methods, t-SNE operates by comparing the distances between distributions. It is particularly useful for visualizing datasets with non-linear structures and is non-parametric in nature. By preserving the local structure of the high-dimensional data, t-SNE uncovers the global structure and can be applied to manifold learning tasks. The transformation to a lower-dimensional space in t-SNE is accomplished through the utilization of conditional probability, enabling effective representation of the data in the reduced space [12].

2.3 Related Works

Vasan & Surendiran [17] investigated the effectiveness of Principal Component Analysis (PCA) in the context of network intrusion detection. The researchers aimed to evaluate the Reduction Ratio (RR), determine the optimal number of Principal Components, and analyze the impact of noisy data on the performance of PCA. To achieve their goals, the researchers conducted experiments using PCA in conjunction with different classification algorithms. They utilized two standard datasets, namely KDD CUP and UNB ISCX, which are commonly used in the field of network intrusion detection. The findings of the study revealed that utilizing the first 10 Principal Components proved to be sufficient for achieving accurate classification results. On the KDD dataset, the classification accuracy reached approximately 99.7%, while on the ISCX dataset, it achieved an accuracy of around 98.8%. These results were comparable to the accuracy obtained using the original feature sets, which consisted of 41 and 28 features for the KDD and ISCX datasets, respectively. Overall, the study demonstrated the effectiveness of PCA in reducing the dimensionality of network intrusion detection datasets while maintaining high classification accuracy.

Aksu et al. [18] presented a system aimed at detecting denial of service (DoS) attacks in intrusion detection. In their approach, they utilized the Fisher Score algorithm for feature selection and employed Support Vector Machine (SVM), K-Nearest Neighbor (KNN), and Decision Tree (DT) as the classification algorithms. The results of their system demonstrated high success rates in detecting DoS attacks. Using SVM as the classification algorithm, the intrusion detection system (IDS) achieved a success rate of approximately 99.7%. When KNN was used, the success rate was measured at 57.76%. Finally, the IDS attained a success rate of 99These findings indicate that the system showed excellent performance in detecting DoS attacks, with SVM and DT demonstrating particularly high success rates. The research highlights the importance of feature selection and the impact of different classification algorithms in building effective intrusion detection systems.

Xia et al. [19] proposed an Intrusion Detection System (IDS) that combined Principal Component Analysis (PCA) and Grey Neural Networks (GNN) to enhance network security. The IDS followed a two-step approach, where PCA was employed to reduce the dimensionality of the input data, and then GNN was utilized for classification. To evaluate the performance of the proposed IDS, it was tested on the KDDCup99 dataset, which is commonly used for intrusion detection research. The results demonstrated a detection rate of 57% for overall intrusions and a detection rate of 48% for hidden attacks. These findings indicate that the IDS was effective in detecting intrusions and outperformed other methods that achieved a maximum detection rate of 22%. The combination of PCA and GNN in the proposed IDS proved to be beneficial for network security, as it successfully reduced the dimensionality of the input data and improved the detection rates compared to other methods. These results highlight the effectiveness of the proposed approach in enhancing intrusion detection capabilities in network security applications.

Al-Qatf et al. [20], utilized Sparse Auto-Encoder (SAE) as a technique for feature learning and dimensionality reduction on the NSL-KDD dataset, which is an enhanced version of the older KDD-CUP99 synthetic netflow dataset. The objective of their approach was to enhance classification performance using Support Vector Machines (SVM). By applying SAE and SVM, the researchers achieved promising results on the NSL-KDD dataset. They reported an accuracy of 84.96% in binary classification and 99.39% accuracy in multi-class classification, specifically for five classes. These results indicate that their approach effectively learned meaningful features and reduced dimensionality, leading to improved classification accuracy. It is worth noting that this study differed from the mentioned research by utilizing the CICIDS2017 dataset. The CICIDS2017 dataset contains new attacks and intruder strategies and is based on real network traffic. This dataset offers a more realistic representation of network behavior and allows for evaluating the effectiveness of intrusion detection methods in the face of contemporary threats. By utilizing the CICIDS2017 dataset, this study aimed to assess the performance of the proposed approach in a more challenging and up-to-date context.

Vijayanand et al. [21] proposed a novel intrusion detection system specifically designed for wireless mesh networks. Their system employed multiple support vector machine (SVM) classifiers along with a feature selection technique based on genetic algorithms (GA). The authors acknowledged that the accuracy of attack detection can be hindered by the presence of redundant and irrelevant variables in the monitored data. To address this issue, they incorporated feature selection techniques, specifically using a genetic algorithm. This approach aimed to identify and select the most informative and relevant features from the dataset, improving the overall accuracy of the intrusion detection system. Experimental results were conducted on various datasets, including a 512-bit dataset, 1024-bit dataset, combined dataset, ADFA-LD dataset, and CICIDS2017 dataset. The findings demonstrated that their system achieved high accuracy in detecting attacks across these datasets. This suggests that their approach is effective and suitable for intrusion detection in wireless mesh networks. By integrating multiple SVM classifiers and employing a genetic algorithm-based feature selection technique, the proposed intrusion detection system showed promising results in terms of attack detection accuracy. This research contributes to the field of wireless mesh network security by addressing the challenges posed by redundant and irrelevant variables and providing an effective solution for intrusion detection.

Salo et al. [22] developed a novel hybrid dimensionality reduction technique that combines an ensemble classifier based on support vector machine (SVM), instance-based learning algorithms (IBK), and multilayer perceptron (MLP) with the feature selection approaches of information gain (IG) and principal component analysis (PCA). To evaluate the effectiveness of their proposed technique, they conducted experiments using three well-established datasets: ISCX 2012, NSL-KDD, and Kyoto 2006+. The goal was to assess the performance of the IG-PCA-Ensemble method in terms of accuracy. The experimental results showed that the IG-PCA-Ensemble method outperformed other approaches in terms of accuracy. Specifically, when applied to the ISCX 2012 dataset, this method achieved the highest accuracy rate of 99.01%. The combination of the ensemble classifier, instance-based learning algorithms, multilayer perceptron, and the feature selection techniques of information gain and principal component analysis proved to be effective in reducing the dimensionality of the data while maintaining high accuracy in classification. These findings highlight the potential of the IG-PCA-Ensemble method for dimensionality reduction and classification tasks in various domains, as demonstrated by the positive results obtained on the ISCX 2012 dataset.

In the same context, Jyothsna et al. [23] developed a novel approach which is a hybrid Dimensionality Reduction and Neural Network Based Classifier. they used also two feature selection techniques: IG and PCA for dimensionality reduction and a multilayer perception technique to classify the data. The proposed method was evaluated using the benchmark dataset of network Intrusion Detection System i.e., NSL-KDD.The results demonstrate that the model has improved accuracy and also offers reduced computational time and a lower false alarm rate.

Pervez & Farid. [24] proposed a novel approach that combines dimensionality reduction techniques, specifically information gain (IG) and principal component analysis (PCA), with a neural network-based classifier, specifically a multilayer perceptron (MLP). The purpose of this approach is to improve the accuracy of classification while reducing computational time and minimizing false alarms in the context of network intrusion detection. To evaluate the effectiveness of their approach, they conducted experiments using the benchmark dataset NSL-KDD, which is commonly used in the field of network intrusion detection systems. The results of the evaluation demonstrated that the proposed approach achieved improved accuracy compared to existing methods. Additionally, it showcased reduced computational time, indicating that the approach is efficient in processing and classifying the data. Furthermore, the approach showed a lower false alarm rate, suggesting that it effectively distinguishes between normal network traffic and intrusive activities. The combination of dimensionality reduction techniques (IG and PCA) and the neural network-based classifier (MLP) in this approach offers a promising solution for network intrusion detection. The improved accuracy, reduced computational time, and lower false alarm rate indicate the potential of this approach in enhancing the performance and efficiency of intrusion detection systems.

2.4 Conclusion

Dimensionality reduction techniques are important for reducing the number of features in high-dimensional data while retaining as much information as possible. There are two main categories of dimensionality reduction are feature selection and feature extraction. Feature selection methods aim to choose a subset of the original features, while feature extraction methods create new features that capture the essence of the original data. Some common techniques include PCA, LDA, ICA, t-SNE, and autoencoders. The choice of technique depends on the type of data, the goal of analysis, and the computational resources available.

CHAPTER 3 METHODOLOGY

3.1 Introduction

After establishing the foundational theoretical concepts of machine learning, IoT security, and dimensionality reduction techniques in the initial chapters, our focus shifts to the second part of our research. This phase involves an investigation into the impact of dimension reduction methods on the performance of intrusion detection systems in IoT environments. To conduct a comparative analysis, we utilize the IoTID20 database as our dataset. Multiple models are trained using various learning algorithms to assess how dimension reduction affects accuracy and processing speed. This chapter outlines the steps undertaken to obtain the study results. Firstly, we discuss the data preprocessing techniques employed to prepare it for our analysis. Subsequently, we delve into an exploration of the basic models utilized for the comparison. Furthermore, we outline the different performance metrics considered during the algorithm evaluation. Finally, we provide an explanation of the principal component analysis (PCA) method employed for data reduction.

3.2 Methodology

3.2.1 Data Pre-processing

In order to ensure accurate analysis, it is essential to perform pre-processing on the dataset and understand its characteristics. Pre-processing is a crucial step in preparing the dataset for further analysis. In our study, the initial dataset consisted of 625,783 rows and 86 columns. After removing non-numerical columns, the dataset was reduced to 70 columns. Here are the pre-processing steps that we have followed for our research on the impact of dimensionality reduction on the accuracy and performance of intrusion detection systems in IoT environments:

- First of all, We started by converting categorical labels into numerical labels using appropriate encoding techniques such as one-hot encoding or label encoding.
- Non-numerical value-containing columns were removed in the next step.
- We inspect the dataset to identify features that have a single value for all records. These features do not provide any useful information for the classification task and can be safely removed from the dataset.
- Normalization was done to equalize the weight of each feature. This step is crucial as features with a wider range of values have more impact than features with a smaller range of values.
- After normalization, the "NaN" and "Inf" values that could be present in the initial dataset or appear after normalization were removed.

3.3 Model Evaluation Measures

3.3.1 Accuracy

Accuracy is calculated by dividing the number of correctly predicted instances (true positives and true negatives) by the total number of instances in the dataset. The formula for accuracy is:

Accuracy = (Number of Correct Predictions) / (Total Number of Predictions)

It is commonly expressed as a percentage, so the accuracy value is multiplied by 100. A higher accuracy value indicates better performance of the classification model in correctly predicting the class labels of the instances in the dataset.

3.3.2 Precision

Precision is a metric used to evaluate the performance of a classification model, particularly in the context of positive predictions. It measures the proportion of true positive instances (TP) out of all instances predicted as positive, which includes both true positives and false positives (FP).

The formula for precision is:

Precision = TP / (TP + FP)

A high precision value indicates that the model has a low rate of falsely classifying negative instances as positive. It reflects the model's ability to accurately identify positive instances and is particularly useful when the cost of false positives is high, such as in medical diagnoses or fraud detection.

3.3.3 Recall

Recall, also known as sensitivity or true positive rate, is a metric used to evaluate the performance of a classification model, specifically in the context of positive instances. Recall measures the proportion of true positive instances (TP) that are correctly identified by the model out of all the actual positive instances, which includes both true positives and false negatives (FN). It quantifies the model's ability to capture and identify positive instances. The formula for recall is:

Recall = TP / (TP + FN)

A high recall value indicates that the model is effective at identifying most of the positive instances and has a low rate of false negatives. It is particularly useful in scenarios where the identification of positive instances is crucial, such as in medical diagnoses or detecting rare events.

3.3.4 F1

The F1 score is a widely used evaluation metric that provides a comprehensive measure of a classification model's overall performance by combining precision and recall into a single value. It is calculated as the harmonic mean of precision and recall, offering a balanced assessment of both metrics.

By taking the weighted average of precision and recall, the F1 score considers both the ability of the model to correctly identify positive instances (precision) and its capability to capture all positive instances (recall).

The F1 score formula is as follows:

F1 Score = 2 * (Precision * Recall) / (Precision + Recall)

The F1 score ranges between 0 and 1, with a higher value indicating better performance. It is particularly useful in situations where both precision and recall are equally important, and a balance between them is desired.

3.3.5 Fit Time

Fit time refers to the amount of time required to train a model on a given dataset. The term "fit" is used to describe the process of optimizing the parameters of a model to minimize the discrepancy between its predicted outputs and the actual outputs observed in the training dataset. Fit time is an important consideration when choosing a machine learning model for a particular task, as models that take longer to train may not be practical for real-time or time-sensitive applications.

3.3.6 Score Time

Score time is the duration it takes for a trained machine learning model to make predictions or classify new data. It represents the computational burden associated with processing and evaluating unseen instances. As a crucial performance metric, score time directly affects the model's practical usability and efficiency in real-time applications.

3.4 Cross Validation

Cross-validation is a methodology employed to assess the effectiveness of a machine learning model by systematically dividing the available data into subsets. The data is partitioned into a training set, where the model is trained, and a validation set, where the model is evaluated. This partitioning process is repeated multiple times as shown in Figure 3.1, with different subsets serving as the validation set in each iteration. The performance of the model is measured and recorded for each iteration, and the results are then averaged to obtain a more robust estimate of the model's performance.

By utilizing cross-validation, we aim to obtain a more accurate evaluation of the model's ability to generalize to unseen data. It helps in assessing the model's performance across different subsets of the data, which aids in identifying potential issues such as overfitting or underfitting. Cross-validation provides a more reliable estimate of the model's performance compared to a single train-test split, as it reduces the impact of data variability and provides a more comprehensive evaluation of the model's capabilities.

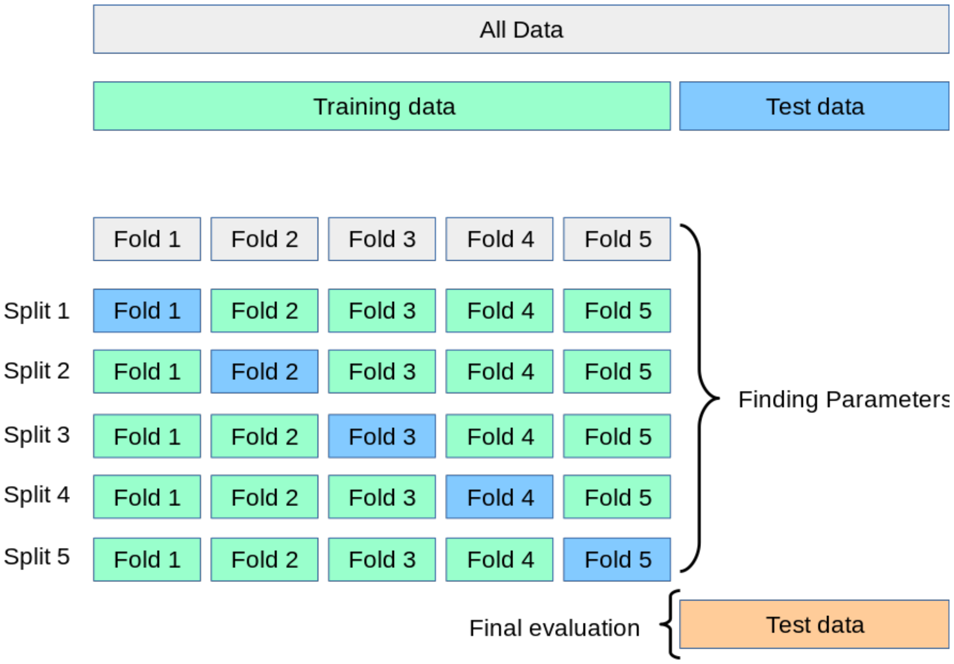

All Data
Training data
Test data
Fold 1 Fold 2 Fold 3 Fold 4 Fold 5
Split 1 Fold 1 Fold 2 Fold 3 Fold 4 Fold 5
Split 2 Fold 1 Fold 2 Fold 3 Fold 4 Fold 5
Split 3 Fold 1 Fold 2 Fold 3 Fold 4 Fold 5
Split 4 Fold 1 Fold 2 Fold 3 Fold 4 Fold 5
Split 5 Fold 1 Fold 2 Fold 3 Fold 4 Fold 5
Finding Parameters
Final evaluation
Test data

FIGURE 3.1: Cross Validation Method

One of the widely used methods of cross-validation is k-fold cross-validation. In this approach, the dataset is divided into k equal-sized folds or subsets. The model is then trained on k-1 folds and evaluated on the remaining fold. This process is repeated k times, with each fold serving as the validation set once. The performance metrics, such as accuracy or error rate, obtained from each iteration are typically averaged to provide an overall assessment of the model's performance.

3.5 The Basic Models Used for the Comparison

Choosing the appropriate learning algorithm for a case study is a crucial decision, and it is often challenging to determine the best option. In this case study, I focused on comparing and evaluating several basic models to analyze their performance and who provides the best performance and accuracy. The models I utilized for the comparison included:

— Decision Tree

— Multi-Layer Perceptron (MLP)

— Support Vector Machine (SVM)

— Linear Regression

In the upcoming chapter, we will present the evaluation measures and showcase the results obtained.

3.6 Data Compression

In our study on data compression, we employed Principal Component Analysis (PCA) as a fundamental technique. By utilizing PCA with components 3, 6, and 8, we were able to effectively reduce the dimensionality of our datasets while preserving important information. PCA is a mathematical method that transforms a set of potentially correlated variables into a smaller set of variables known as principal components. It achieves this by applying a vector space transformation to the original dataset. This transformation allows us to interpret the data using only a few principal components instead of the original numerous variables. By reducing the dimensionality of the dataset, PCA helps to simplify the analysis process. It enables us to identify significant features, trends, patterns, and outliers more easily compared to working with the original high-dimensional dataset. PCA's mathematical projection provides insights and facilitates data interpretation in a more efficient manner. By harnessing the capabilities of PCA, we were able to extract the most relevant features from our data and achieve effective data compression. This approach allowed us to reduce the complexity of the datasets while retaining critical information, thereby enhancing our understanding of the underlying patterns and structures within the data [25].

The key steps of PCA can be summarized as follows:

1. **Standardization:** The input data is standardized to ensure that each feature has a similar scale. This is achieved by subtracting the mean (μ) of each feature and dividing by its standard deviation (σ).

2. **Covariance matrix computation:** The covariance matrix (Σ) is computed to capture the relationships between different features. The element Σ ij represents the covariance between the ith and jth features, indicating how they vary together.

3. **Eigenvalue decomposition:** The covariance matrix is then decomposed into its eigenvectors (V) and eigenvalues. This can be expressed as

$$\Sigma = V \Lambda V^T$$

   where Λ is a diagonal matrix containing the eigenvalues, and V contains the corresponding eigenvectors.

4. **Eigenvector selection:** The eigenvectors represent the principal components of the data. They are ranked based on their corresponding eigenvalues, with the highest eigenvalue indicating the principal component that captures the most variance. By selecting a subset of the eigenvectors, we can reduce the dimensionality of the data.

5. **Dimensionality reduction:** The selected eigenvectors are combined to form a transformation matrix (W). By multiplying the original data matrix (X) by this transformation matrix, we obtain the lower-dimensional representation (Y = XW), where Y represents the transformed data.

3.7 Dataflow

In this project, we focused on improving the accuracy and performance of an intrusion detection system in IoT environments. To achieve this, we employed the PCA dimensionality reduction method. First, we preprocessed the IoTID20 dataset, handling missing values and outliers. Then, we applied PCA to extract the most important features and reduce the dimensionality of the dataset. Next, we trained various machine learning algorithms, such as decision trees and support vector machines on the original data and the reduced data. To evaluate the models, we split the dataset into training and testing subsets using cross-validation. we compared the accuracy and performance of the models with and without PCA to analyze the impact of dimensionality reduction. Additionally, we conducted further analysis and visualization to gain insights into the models' behavior. Overall, this architecture allowed us to investigate the effectiveness of PCA in improving the intrusion detection system's accuracy and performance in IoT environments. We summarize in Figure 3.2 the flow of our data from reading the dataset until generating our results.

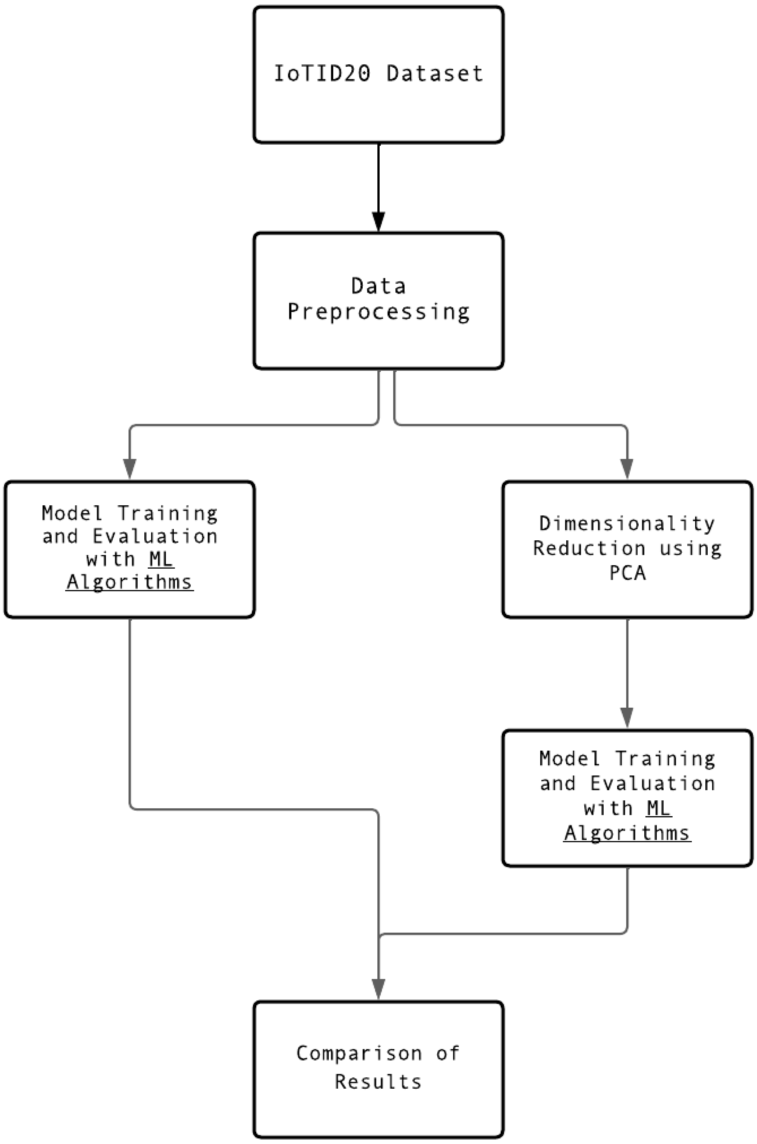

IoTID20 Dataset
Data Preprocessing
Model Training and Evaluation with ML Algorithms
Dimensionality Reduction using PCA
Model Training and Evaluation with ML Algorithms
Comparison of Results

FIGURE 3.2: Diagram illustrating the steps of applying dimensionality reduction

3.8 Conclusion

In this chapter, we provided a detailed account of our methodology for testing the impact of principal component analysis (PCA) on accuracy within the context of intrusion detection in IoT networks. To achieve this, we implement PCA in conjunction with multiple machine learning algorithms and assess its effect on accuracy. Additionally, we introduce and discuss the performance measures that we will employ to evaluate the results, including accuracy, F1 score, fit time, and score time. This approach allows us to determine the extent to which dimension reduction techniques impact the accuracy and performance of machine learning algorithms for intrusion detection in IoT networks.

CHAPTER 4 RESULTS AND DISCUSSION

4.1 Introduction

In this concluding chapter, we begin by discussing the development tools employed to obtain the results. then, we present the outcomes of the extensive tests conducted in two parts. The first part focuses on binary classification results, encompassing accuracy tests, F1 score, fit time, and score time. The second part extends the analysis to multi-class classification results, where the same tests are employed. To provide a comprehensive overview of the findings, the chapter concludes with a summarized table that consolidates the results obtained from the diverse range of tests conducted.

4.2 Development Tools

4.2.1 Python

Python is a programming language known for its simplicity and ease of learning. It operates at a high level and follows an interpreted, object-oriented, and dynamically-typed approach. Python prioritizes readability and straightforwardness in code, allowing developers to split programs into reusable modules. Additionally, it provides an extensive library of standard modules that serve as a foundation for building applications. With Python, programs can be written in a concise and readable manner, eliminating the need for explicit variable or argument declarations. [26].

Python is an extensively employed programming language that finds applications in various domains, including web development, scientific computing, data analysis, artificial intelligence, and more. The thriving developer community plays a vital role in its success, offering a wide range of libraries and frameworks. Guido van Rossum is the mastermind behind Python, releasing it for the first time in 1991. Python is freely available under the open-source Python Software Foundation License, allowing users to utilize it without any cost.

4.2.2 SKLearn

Scikit-learn, often referred to as sklearn, is a Python library designed for machine learning tasks. It presents a straightforward and efficient toolkit for data mining and analysis purposes. Built upon the solid foundations of NumPy, SciPy, and matplotlib, sklearn encompasses an extensive collection of supervised and unsupervised learning algorithms. These algorithms cover a broad range of applications, including classification, regression, clustering, and dimensionality reduction. Moreover, sklearn offers additional functionalities such as model selection, evaluation, preprocessing, and feature extraction. Its versatility and reliability have led to its widespread adoption in both industry and academia, serving as a valuable resource for the development and deployment of machine learning models. [27]. Originally known as "scikits.learn," the library now known as Scikit-learn had its beginnings in 2007 when David Cournapeau initiated its development as part of a Google Summer of Code project. Subsequently, in 2010, Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, and Vincent Michel, hailing from the French Institute for Research in Computer Science and Control (FIRCA), made significant contributions to its advancement.

4.2.3 Material Resources

The specifications of the PC used for implementing and testing are listed in the table below:

Processor	2,2 GHz Quad-Core Intel Core i7
Cache	6 MB
Ram	16 GB 1600 MHz DDR3
Operating system	macOs Big Sur

TABLE 4.1: Material Resources

4.2.4 IoTID20 Dataset

IoTID20 is a dataset that contains network traffic data captured from a real IoT network environment. It was created by researchers at the University of New Brunswick in Canada to aid in the development of Intrusion Detection Systems (IDS) for IoT devices. The dataset contains normal and malicious traffic captured from 20 IoT devices, including cameras, smart locks, smart lights, and a Google Home device.

The dataset provides network traffic at the packet level, including the packet payload, and it contains various types of attacks such as denial-of-service (DoS) attacks, command injection attacks, and reconnaissance attacks. Additionally, the researchers have provided a set of ground truth labels indicating which network traffic flows are benign and which ones are malicious.

The recently introduced dataset for IoT botnets offers a broader range of network and flow-based characteristics. The inclusion of flow-based features allows for the examination and assessment of flow-based intrusion detection systems. By serving as a benchmark, the proposed dataset enables the identification of unusual behavior within IoT networks. Ultimately, the newly introduced IoTID20 dataset establishes a solid groundwork for the advancement of intrusion detection methods in IoT networks. [28]. The dataset is publicly available and can be downloaded from the University of New Brunswick's website.

4.3 Results and Discussion

In the following section, we present and analyze the results obtained from both the original and reduced datasets using various dimensionality reduction techniques. We conducted an implementation of 4 machine learning models, namely Logistic Regression, Decision Tree, Support Vector Machines (SVM), and Multilayer Perceptron (MLP). To ensure a robust evaluation, we employed the cross-validation method to train and test these models.

Our focus lies in evaluating the accuracy and f1 score of the models employed and comparing their respective fit times and score times for learning and prediction tasks. By delving into these key metrics, we gain valuable insights into the performance and efficiency of the implemented approaches.

4.3.1 Binary Classification

We have in Figure 4.1 a showcase of accuracy metric results, where the x-axis represents the 4 machine learning models on each one we have bars representing the original and the reduced data using PCA 3, 6, and 8 dimensions, and the y-axis denotes the accuracy percentage. The legend provides color and pattern coding to indicate the number of dimensions.

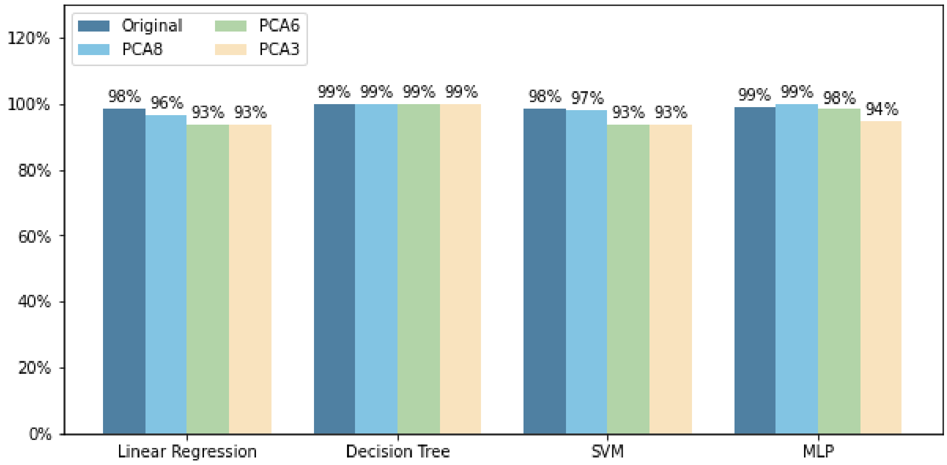


FIGURE 4.1: Accuracy metric

The Decision Tree algorithm exhibits an exceptional accuracy rate of 99% when applied to both the original and reduced data. In contrast, other algorithms such as linear regression, SVM, and MLP experience a decline in accuracy, reaching 93% when the dimensions are reduced to 3 and 6. This discrepancy highlights the superior performance of the Decision Tree algorithm in maintaining high accuracy even with reduced dimensions. While linear regression, SVM, and MLP encounter challenges in accurately capturing the underlying patterns in the data when the dimensionality is reduced, the Decision Tree algorithm proves to be more robust and effective in preserving the accuracy of classification. These findings emphasize the importance of selecting appropriate algorithms that can handle dimensionality reduction without compromising accuracy.

Despite the reduction in the number of dimensions, we observe that the accuracy remains largely consistent across all PCA components. The maximum accuracy drop observed is no more than 5%, indicating that the reduction in dimensions does not significantly impact the model's performance. This phenomenon can be attributed to the presence of a high number of correlated features in the dataset, Consequently, it becomes possible to reduce the number of dimensions without sacrificing much information.

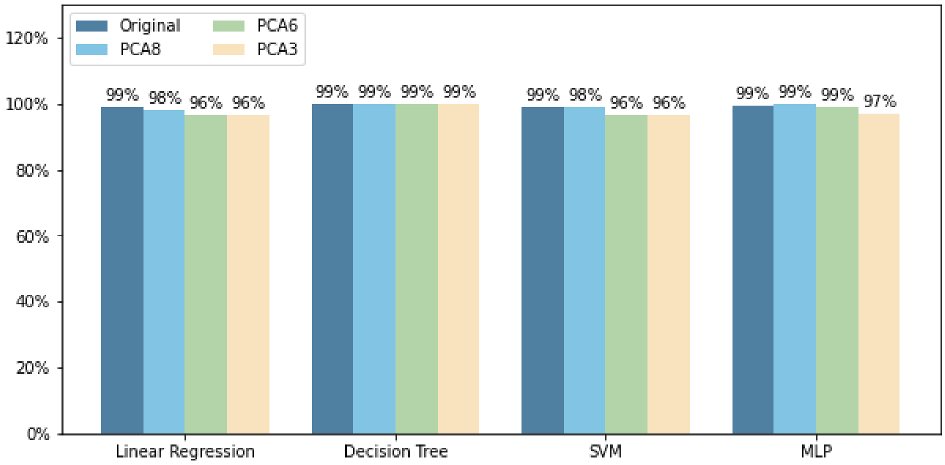


FIGURE 4.2: F1 Score metric

Figure 4.2 presents the F1 score metric for diverse scenarios, encompassing the original dataset and dimensionality reductions achieved through PCA with 8, 6, and 3 dimensions. Notably, the F1 score exhibits a consistently high performance across all original and reduced datasets, with a maximum decrease of only 3%. The F1 score holds particular significance in the context of binary classification tasks, such as the one undertaken in this study. This significance arises from the imbalanced nature of the dataset, where one class is substantially more prevalent than the other. Relying solely on accuracy as an evaluation metric can be misleading in such cases. Consequently, the F1 score offers a more dependable assessment by incorporating both precision and recall into its calculation.

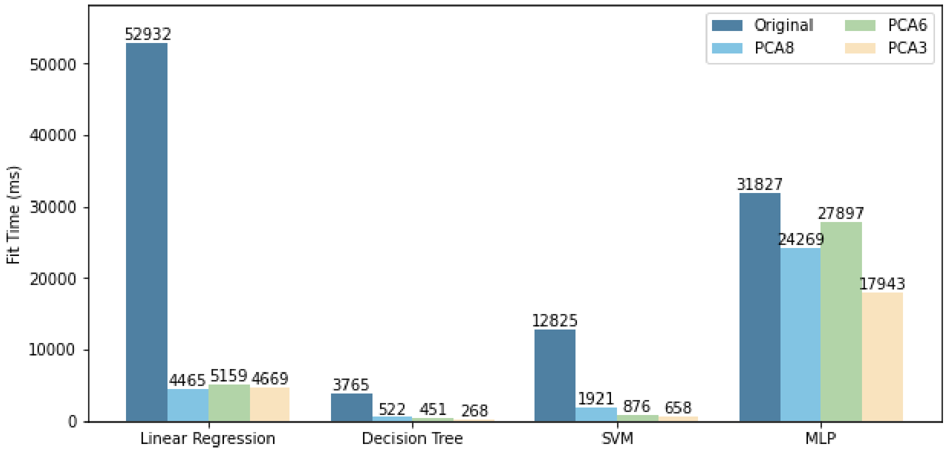


FIGURE 4.3: Fit Time metric

Figure 4.3 showcases the computational efficiency of four different algorithms, we have in the x-axis the algorithms and the y-axis represents the fit time in milliseconds. Upon analysis, we observe distinct patterns in the fit times of the algorithms. We observe that Decision Tree and SVM consistently outperform the other algorithms in terms of fit time across original and reduced data. Even with larger datasets, Decision Tree and SVM exhibit significantly faster training times compared to MLP and Linear Regression. This finding highlights the algorithm's efficiency and scalability, making it a favorable choice for training models on larger datasets.

Linear Regression shows moderately longer fit times in original data compared to SVM and Decision Tree but performs better than MLP in reduced data. Although it falls behind SVM and Decision Tree in terms of fit time, Linear Regression remains computationally feasible for datasets of the given sizes.

On the other hand, MLP exhibits the highest fit times among the algorithms analyzed. We also observe for the same model that the input dimension does not have the same impact on the training time, as the amount of operations performed in the hidden layers does not depend on the input dimension.

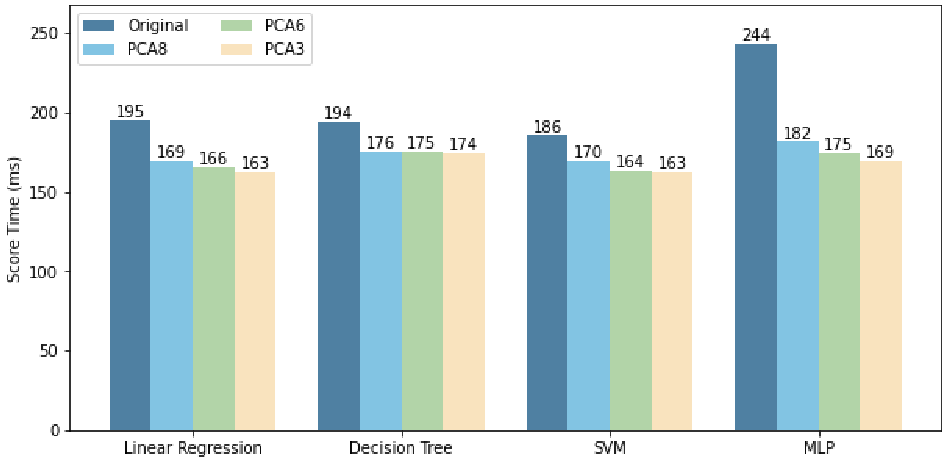
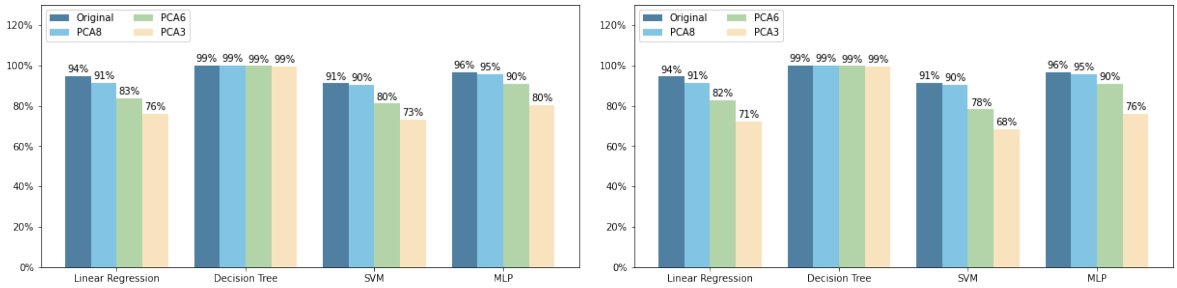


FIGURE 4.4: Score Time metric

Figure 4.4 presents the results of the score time metric, which measures the time taken by each algorithm to generate predictions for unseen data, expressed in milliseconds. Among the algorithms investigated, SVM consistently demonstrates the shortest score times across all scenarios, making it the most computationally efficient. In the original data, SVM achieves a score time of 186 milliseconds, and with PCA dimensionality reduction of 3, 6, and 8, the score times range from 163 to 170 milliseconds, indicating consistently low computational overhead. However, we note that the difference between all the tested algorithms is not notably important, either with or without dimension reduction.

4.3.2 Multi-class Classification

The multi-class classification results are presented in figures 4.5, 4.6, and 4.7 focusing on category classification.



(A) Accuracy

(B) F1 Score

FIGURE 4.5: Accuracy metrics

Figure 4.5 shows the accuracy and F1 score. Among the algorithms considered, the decision tree algorithm demonstrates the highest accuracy, achieving 99% accuracy, surpassing the performance of the other models. When comparing the results of binary classification, the decision tree consistently outperforms the other models.

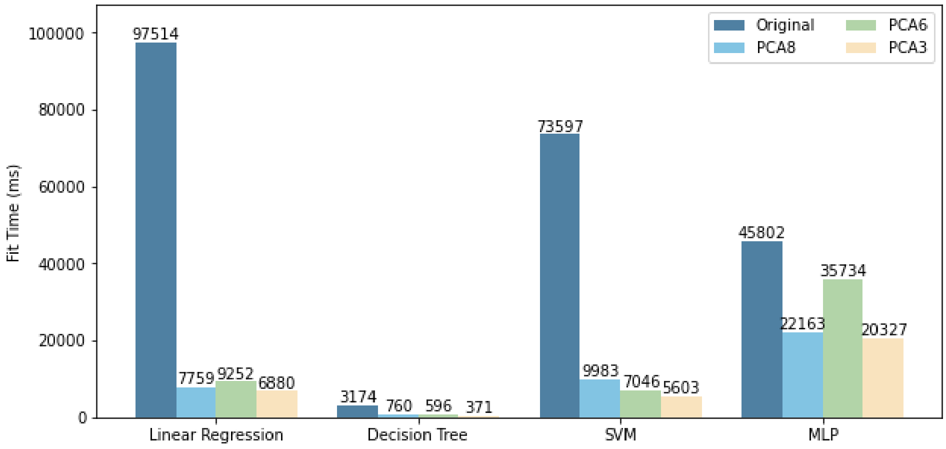


FIGURE 4.6: Fit Time metric

For linear regression, SVM, and MLP, their performance remains relatively similar across both the original and reduced datasets. However, there is a noticeable impact on accuracy when reducing the data from 8 dimensions to 6 dimensions, and further to 3 dimensions, with a drop of approximately 9% in accuracy for each reduction.

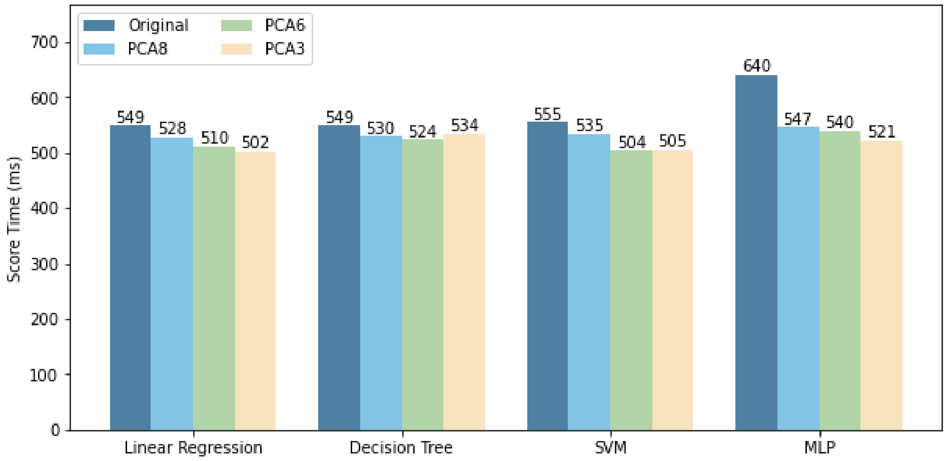


FIGURE 4.7: Score Time metric

It is worth noting that the accuracy and F1 score exhibit similar trends in this analysis. This is attributed to the dataset's balanced nature, where the classes are relatively equally represented. In such cases, the F1 score may be less informative or relevant compared to situations where the dataset is imbalanced. In imbalanced datasets, the F1 score becomes more crucial as accuracy alone can be misleading.

Figure 4.6 presents the fit time, In the original data, the fit time for multi-class classification is approximately twice as long as the fit time for binary classification. However, in the reduced data, the fit time for multi-class classification is similar to that of binary classification. As always, the decision tree algorithm exhibits the best performance.

When reducing the data dimensions to 8, 6, and 3, we observe significant improvements in fit time for linear regression, decision tree, and SVM. The fit time for MLP also improves, although not to the same extent as the other models.

In terms of score time, it is doubled compared to binary classification. However, the performance of the models remains consistent with their performance in binary classification. Specifically, linear regression, decision tree, and SVM outperform MLP in the original data.

The higher values of fit time and score time observed in category classification compared to binary classification can be attributed to the increased complexity and larger number of classes involved in category classification. In binary classification, the model only needs to distinguish between two classes, which typically requires less computational time. However, in category classification, the model must classify instances into multiple classes, which can be more computationally demanding and time-consuming, resulting in higher fit time and score time.

Algorithm	Sub-Algorithm	Accuracy%	F1%	Fit Time(ms)	Score Time(ms)
Linear Regression	Original	98	99	52932	195
	PCA8	96	98	4465	169
	PCA6	93	96	5159	166
	PCA3	93	96	4669	163
Decision Tree	Original	99	99	3765	194
	PCA8	99	99	522	176
	PCA6	99	99	451	175
	PCA3	99	99	268	174
SVM	Original	98	99	12825	186
	PCA8	97	98	1921	170
	PCA6	93	96	876	164
	PCA3	93	96	658	163
MLP	Original	99	99	31827	244
	PCA8	99	99	24269	182
	PCA6	98	99	27897	175
	PCA3	94	97	17943	169

TABLE 4.2: Table of results for Binary Classification

Algorithm	Sub-Algorithm	Accuracy%	F1%	Fit Time(ms)	Score Time(ms)
Linear Regression	Original	94	94	97514	549
	PCA8	91	91	7759	528
	PCA6	83	82	9252	510
	PCA3	76	71	6880	502
Decision Tree	Original	99	99	3174	549
	PCA8	99	99	760	530
	PCA6	99	99	596	524
	PCA3	99	99	371	534
SVM	Original	91	91	73597	555
	PCA8	90	90	9983	535
	PCA6	80	78	7046	535
	PCA3	73	68	5603	504
MLP	Original	96	95	45802	640
	PCA8	95	95	22163	547
	PCA6	90	90	35734	540
	PCA3	80	76	20327	521

TABLE 4.3: Table of results for Multi-Class Classification

4.4 Conclusion

In this chapter, our focus was on evaluating the impact of principal component analysis (PCA) on the performance of four machine learning models for intrusion detection in IoT networks using the IoTID20 dataset. We conducted a thorough comparison of these algorithms in terms of accuracy, F1 score, fit time, and score time, ensuring the reproducibility of the results. Throughout the analysis, our objective was to determine how dimension reduction affects the performance of these algorithms. Our findings indicate that PCA as a dimension reduction technique can effectively reduce training time without significantly compromising the accuracy of the models. In certain cases, it may even lead to performance improvements.

GENERAL CONCLUSION

In recent years, the rapid development of the Internet of Things (IoT) has brought about a remarkable transformation, with a multitude of interconnected devices revolutionizing various aspects of our lives. However, this unprecedented level of connectivity also brings with it substantial security challenges. The extensive network of IoT devices creates an environment ripe for potential security breaches and vulnerabilities, necessitating careful attention to ensure the security and protection of IoT systems.

In this context, machine learning techniques have emerged as powerful tools for intrusion detection in IoT networks. By leveraging advanced algorithms and data analysis, machine learning models can effectively detect and identify potential intrusions in real-time. These techniques enable the detection of anomalous patterns and behaviors that may indicate unauthorized access or malicious activities within IoT environments. With the ability to adapt and learn from new data, machine learning algorithms provide a reliable and efficient approach to enhance the security and protection of IoT networks against emerging threats.

To investigate the performance of machine learning algorithms in intrusion detection for IoT networks, our study focused on the selection of appropriate data. We chose the IoTID20 dataset, specifically designed for anomaly detection in IoT environments. In order to enhance the efficiency of our analysis, we employed principal component analysis (PCA) as a dimensionality reduction technique. This allowed us to reduce the complexity of the dataset while retaining the most informative features, thereby streamlining the learning process for our models.

Next, we evaluated the performance of several machine learning algorithms, including linear regression, decision tree, support vector machines (SVM), and multilayer perceptron (MLP), on the reduced dataset. Our objective was to identify the models that were influenced by this dimensionality reduction and analyze their performance in the context of intrusion detection. We utilized several performance metrics, with "Accuracy" and "F1" being the most commonly used. Additionally, we included "fit time" and "score time" to compare the algorithms in terms of their computational efficiency.

The results of our study demonstrated the significant impact of dimension reduction techniques on reducing training time without compromising the accuracy of our models. Interestingly, in certain cases, the accuracy of the models even improved after dimension reduction. This suggests that by reducing the dimensionality of the data, we were able to eliminate noise and irrelevant information, enabling the models to focus on the most discriminative features for effective intrusion detection.

Overall, our findings highlight the potential of dimension reduction techniques, such as PCA, in optimizing the performance of intrusion detection systems in IoT networks. By leveraging machine learning algorithms and careful data analysis, we can enhance the security and protection of IoT systems, mitigating potential threats and safeguarding sensitive data. These insights contribute to the ongoing efforts to develop more efficient and effective security solutions within the dynamic landscape of the Internet of Things.

Perspective

In future research endeavors, our intention is to explore additional dimension reduction methods, such as LDA (Linear Discriminant Analysis), t-SNE (t-Distributed Stochastic Neighbor Embedding), and UMAP (Uniform Manifold Approximation and Projection), autoencoder. The primary objective behind this undertaking is to assess the impact of these techniques on the performance of our intrusion detection system. By comparing the results obtained from these various dimension reduction methods, we hope to identify the most effective approach for enhancing the performance of our intrusion detection system. This exploration will contribute to the advancement of intrusion detection techniques and provide valuable insights for improving the detection and prevention of unauthorized activities in network security.

BIBLIOGRAPHY

[1] Batta Mahesh. “Machine learning algorithms-a review”. In: *International Journal of Science and Research (IJSR).[Internet]* 9 (2020), pp. 381–386.

[2] Lizhi Wang et al. “A Deep-forest based approach for detecting fraudulent online transaction”. In: *Advances in computers*. Vol. 120. Elsevier, 2021, pp. 1–38.

[3] Wang Qiang and Zhan Zhongli. “Reinforcement learning model, algorithms and its application”. In: *2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC)*. IEEE. 2011, pp. 1143–1146.

[4] Susmita Ray. “A quick review of machine learning algorithms”. In: *2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon)*. IEEE. 2019, pp. 35–39.

[5] Kajaree Das and Rabi Narayan Behera. “A survey on machine learning: concept, algorithms and applications”. In: *International Journal of Innovative Research in Computer and Communication Engineering* 5.2 (2017), pp. 1301–1309.

[6] Ismail Butun, Patrik Österberg, and Houbing Song. “Security of the Internet of Things: Vulnerabilities, attacks, and countermeasures”. In: *IEEE Communications Surveys & Tutorials* 22.1 (2019), pp. 616–644.

[7] Raja Benabdessalem, Mohamed Hamdi, and Tai-Hoon Kim. "A survey on security models, techniques, and tools for the internet of things". In: *2014 7th International Conference on Advanced Software Engineering and Its Applications*. IEEE. 2014, pp. 44–48.

[8] Luigi Atzori, Antonio Iera, and Giacomo Morabito. "The internet of things: A survey". In: *Computer networks* 54.15 (2010), pp. 2787–2805.

[9] Ala Al-Fuqaha et al. "Internet of things: A survey on enabling technologies, protocols, and applications". In: *IEEE communications surveys & tutorials* 17.4 (2015), pp. 2347–2376.

[10] Mohamed Abomhara and Geir M Køien. "Security and privacy in the Internet of Things: Current status and open issues". In: *2014 international conference on privacy and security in mobile systems (PRISMS)*. IEEE. 2014, pp. 1–8.

[11] Ioannis Andrea, Chrysostomos Chrysostomou, and George Hadjichristofi. "Internet of Things: Security vulnerabilities and challenges". In: *2015 IEEE symposium on computers and communication (ISCC)*. IEEE. 2015, pp. 180–187.

[12] Shaeela Ayesha, Muhammad Kashif Hanif, and Ramzan Talib. "Overview and comparative study of dimensionality reduction techniques for high dimensional data". In: *Information Fusion* 59 (2020), pp. 44–58.

[13] Rizgar Zebari et al. "A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction". In: *J. Appl. Sci. Technol. Trends* 1.2 (2020), pp. 56–70.

[14] Alan Jović, Karla Brkić, and Nikola Bogunović. "A review of feature selection methods with applications". In: *2015 38th international convention on information and communication technology, electronics and microelectronics (MIPRO)*. Ieee. 2015, pp. 1200–1205.

[15] Carlos Oscar Sánchez Sorzano, Javier Vargas, and A Pascual Montano. "A survey of dimensionality reduction techniques". In: *arXiv preprint arXiv:1403.2877* (2014).

[16] Yesi Novaria Kunang et al. “Automatic features extraction using autoencoder in intrusion detection system”. In: *2018 International Conference on Electrical Engineering and Computer Science (ICECOS)*. IEEE. 2018, pp. 219–224.

[17] K Keerthi Vasan and B Surendiran. “Dimensionality reduction using principal component analysis for network intrusion detection”. In: *Perspectives in Science* 8 (2016), pp. 510–512.

[18] Doğukan Aksu et al. “Intrusion detection with comparative analysis of supervised learning techniques and fisher score feature selection algorithm”. In: *Computer and Information Sciences: 32nd International Symposium, ISCIS 2018, Held at the 24th IFIP World Computer Congress, WCC 2018, Poznan, Poland, September 20-21, 2018, Proceedings 32*. Springer. 2018, pp. 141–149.

[19] Dong-Xue Xia, Shu-Hong Yang, and Chun-Gui Li. “Intrusion detection system based on principal component analysis and grey neural networks”. In: *2010 Second International Conference on Networks Security, Wireless Communications and Trusted Computing*. Vol. 2. IEEE. 2010, pp. 142–145.

[20] Majjed Al-Qatf et al. “Deep learning approach combining sparse autoencoder with SVM for network intrusion detection”. In: *Ieee Access* 6 (2018), pp. 52843–52856.

[21] R Vijayanand, D Devaraj, and B Kannapiran. “Intrusion detection system for wireless mesh network using multiple support vector machine classifiers with genetic-algorithm-based feature selection”. In: *Computers & Security* 77 (2018), pp. 304–314.

[22] Fadi Salo, Ali Bou Nassif, and Aleksander Essex. “Dimensionality reduction with IG-PCA and ensemble classifier for network intrusion detection”. In: *Computer Networks* 148 (2019), pp. 164–175.

[23] V Jyothsna et al. “A Network Intrusion Detection System with Hybrid Dimensionality Reduction and Neural Network Based Classifier”. In: *ICT Systems and Sustainability: Proceedings of ICT4SD 2019, Volume 1*. Springer. 2020, pp. 187–196.

[24] Muhammad Shakil Pervez and Dewan Md Farid. “Feature selection and intrusion classification in NSL-KDD cup 99 dataset employing SVMs”. In: *The 8th International Conference on Software, Knowledge, Information Management and Applications (SKIMA 2014)*. IEEE. 2014, pp. 1–6.

[25] Mark Richardson. “Principal component analysis”. In: *URL: http://people. maths. ox. ac. uk/richardsonm/SignalProcPCA. pdf (last access: 3.5. 2013). Aleš Hladnik Dr., Ass. Prof., Chair of Information and Graphic Arts Technology, Faculty of Natural Sciences and Engineering, University of Ljubljana, Slovenia ales. hladnik@ ntf. uni-lj. si* 6 (2009), p. 16.

[26] Guido Van Rossum and Fred L Drake Jr. *Python tutorial*. Vol. 620. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands, 1995.

[27] Fabian Pedregosa et al. “Scikit-learn: Machine learning in Python”. In: *the Journal of machine Learning research* 12 (2011), pp. 2825–2830.

[28] Imtiaz Ullah and Qusay H Mahmoud. “A scheme for generating a dataset for anomalous activity detection in iot networks”. In: *Advances in Artificial Intelligence: 33rd Canadian Conference on Artificial Intelligence, Canadian AI 2020, Ottawa, ON, Canada, May 13–15, 2020, Proceedings 33*. Springer. 2020, pp. 508–520.