REPUBLIC OF TÜRKİYE

ALTINBAŞ UNIVERSITY

Institute of Graduate Studies

Information Technologies

DEVELOP A ROBUST COMPUTER NETWORK ARCHITECTURE THAT IS RESISTANT TO UNAUTHORIZED ACCESS BY USING MACHINE LEARNING METHODOLOGIES

Aya Ahmed Tawfeeq TAWFEEQ

Master's Thesis

Supervisor

Asst. Prof. Dr. Abdullahi Abdu IBRAHIM

İstanbul, 2024

DEVELOP A ROBUST COMPUTER NETWORK ARCHITECTURE THAT IS RESISTANT TO UNAUTHORIZED ACCESS BY USING MACHINE LEARNING METHODOLOGIES

Aya Ahmed Tawfeeq TAWFEEQ

Information Technologies

Master's Thesis

ALTINBAŞ UNIVERSITY

2024

The thesis titled DEVELOP A ROBUST COMPUTER NETWORK ARCHITECTURE THAT IS RESISTANT TO UNAUTHORIZED ACCESS BY USING MACHINE LEARNING METHODOLOGIES prepared by AYA AHMED TAWFEEQ TAWFEEQ and submitted on 02/07/2024 has been **accepted unanimously** for the degree of Master of Science in Information Technologies.

Asst. Prof. Dr. Abdullahi Abdu IBRAHIM

Supervisor

Thesis Defense Committee Members:

Asst. Prof. Dr. Abdullahi Abdu IBRAHIM	Department of Computer Engineering, Altınbaş University	
Assoc. Prof. Dr. Sefer KURNAZ	Department of Computer Engineering, Altınbaş University	
Asst. Prof. Dr. Tareq Abed MOHAMMED	Department of Computer Science, University of Kirkuk	

I hereby declare that this thesis meets all format and submission requirements of a Master's thesis.

I hereby declare that all information/data presented in this graduation project has been obtained in full accordance with academic rules and ethical conduct. I also declare all unoriginal materials and conclusions have been cited in the text and all references mentioned in the Reference List have been cited in the text, and vice versa as required by the abovementioned rules and conduct.

Aya Ahmed Tawfeeq AWFEEQ

Signature

DEDICATION

To the spirit of my father, who instilled in me a love of learning and a passion for discovery. He always encouraged me to pursue my dreams, no matter how challenging they may seem. I know that he would be proud of what I have accomplished.

I also dedicate this work to all those who have faced obstacles and challenges in pursuing their dreams. May this work serve as a reminder that with hard work, dedication, and perseverance, anything is possible.

PREFACE

Firstly, I would want to convey my appreciation to the divine being for bestowing upon me the determination and resilience required to successfully accomplish this task.

To my whole family, with a particular mention to my mother, I express my gratitude for always motivating me to follow my aspirations and offering me the affection and assistance necessary for my accomplishments. The steadfast faith you have in me has always served as a profound wellspring of motivation.

Lastly, I would like to convey my profound appreciation to my supervisor, Asst. Prof. Dr. Abdullahi Abdu IBRAHIM, for his invaluable mentorship, unwavering support, insightful ideas, and extensive expertise during the whole of this research endeavor. His advice throughout my extensive study for my thesis has greatly enhanced my understanding of independence. I want optimal outcomes for him in all of his endeavors. Furthermore, I am grateful for the constructive criticism and support provided by the rest of my thesis committee.

ABSTRACT

DEVELOP A ROBUST COMPUTER NETWORK ARCHITECTURE THAT IS RESISTANT TO UNAUTHORIZED ACCESS BY USING MACHINE LEARNING METHODOLOGIES

TAWFEEQ, Aya Ahmed Tawfeeq

M.Sc., Information Technologies, Altınbaş University,

Supervisor: Asst. Prof. Dr. Abdullahi Abdu IBRAHIM

Date: July / 2024

Pages: 75

As the use of IT spreads rapidly into new areas, the necessity to ensure the security of these systems has grown. Cyberattacks have also become much more sophisticated as a result of the widespread availability of information technology. Consequently, traditional security measures like SIDS have failed to identify new types of assaults. Intrusion Detection Systems (IDS) make it possible to track and gather harmful data inside a network. The majority of IDSs rely on signatures to identify potential threats. They use a set of rules—either manually entered by the administrator or created automatically by the system—to identify and respond to known threats. To maintain the availability of services at all times, network security experts focus on both preventing and responding to intrusion attempts. To find and categorize suspicious actions, security professionals employ tools like IDS. Hence, to protect privacy, security, and the ongoing delivery of services, it is crucial that the IDS continually keeps up-to-date with the most recent intrusion attack signatures. Important factors to consider while evaluating IDS performance are its speed and its capacity to learn new assaults. This study demonstrates how several Machine Learning techniques may be evaluated using the Knowledge Discovery and Data Mining (KDD) dataset, which is also called Knowledge Discovery in Databases. The primary focus is on creating a comprehensive and representative dataset for experimentation, with a strong emphasis on KDD. For this analysis, we have chosen to use the K-Nearest Neighbor (KNN) and

Multilayer Perceptron (MLP) classifiers. The KNN classifier has shown the highest accuracy in recognizing and classifying all types of KDD dataset attacks (DOS, R2L, U2R, NORMAL, and PROBE), both for binary class (NORMAL vs. ABNORMAL) and multi-class scenarios. The experimental findings utilizing the proposed KNN and MLP models showed that the accuracy of binary classification using KNN and MLP was 99% and 97% respectively. Furthermore, the multi-class classification produced improved results compared to earlier work, with reported high-level accuracies of 92% and 87% respectively. This thesis investigates the effectiveness of using deep learning techniques, namely MNN and MLP designs, for network flow-based intrusion detection.

Keywords: Network Security, IDS, Classification, KNN, MLP, Feature Extraction, Data Preprocessing.

ÖZET

MAKİNE ÖĞRENME METODOLOJİLERİNİ KULLANARAK YETKİSİZ ERİŞİME DAYANIKLI, SAĞLAM BİR BİLGİSAYAR AĞ MİMARİSİ GELİŞTİRİN

TAWFEEQ, Aya Ahmed Tawfeeq

Yüksek Lisans, Bilişim Teknolojisi, Altınbaş Üniversitesi

Danışman:.Dr. Öğr. Üyesi, Abdullahi Abdu IBRAHIM

Tarih: Temmuz / 2024

Sayfa: 75

BT kullanımı hızla yeni alanlara yayıldıkça, bu sistemlerin güvenliğinin sağlanması gerekliliği de arttı. Bilgi teknolojisinin yaygınlaşmasının bir sonucu olarak siber saldırılar da çok daha karmaşık hale geldi. Sonuç olarak, SIDS gibi geleneksel güvenlik önlemleri yeni saldırı türlerini tespit etmekte başarısız oldu. İzinsiz Giriş Tespit Sistemleri (IDS), bir ağ içindeki zararlı verileri izlemeyi ve toplamayı mümkün kılar. IDS'lerin çoğunluğu potansiyel tehditleri tanımlamak için imzalara güveniyor. Bilinen tehditleri tanımlamak ve bunlara yanıt vermek için yönetici tarafından manuel olarak girilen veya sistem tarafından otomatik olarak oluşturulan bir dizi kural kullanırlar. Hizmetlerin her zaman kullanılabilirliğini korumak için ağ güvenliği uzmanları, izinsiz giriş girişimlerini hem önlemeye hem de bunlara yanıt vermeye odaklanır. Şüpheli eylemleri bulmak ve sınıflandırmak için güvenlik uzmanları IDS gibi araçlar kullanır. Bu nedenle gizliliği, güvenliği ve hizmetlerin devam eden sunumunu korumak için IDS'nin sürekli olarak en son izinsiz giriş saldırısı imzalarıyla güncel kalması çok önemlidir. IDS performansını değerlendirirken dikkate alınması gereken önemli faktörler, hızı ve yeni saldırıları öğrenme kapasitesidir. Bu çalışma, Veritabanlarında Bilgi Keşfi olarak da adlandırılan Bilgi Keşfi ve Veri Madenciliği (KDD) veri kümesi kullanılarak çeşitli Makine Öğrenimi tekniklerinin nasıl değerlendirilebileceğini göstermektedir. Öncelikli odak noktası, KDD'ye güçlü bir vurgu yaparak deneyler için kapsamlı ve temsili bir veri kümesi oluşturmaktır. Bu analiz için K-En Yakın Komşu (KNN) ve Çok Katmanlı Algılayıcı (MLP) sınıflandırıcılarını kullanmayı seçtik. KNN sınıflandırıcı, hem ikili sınıf (NORMAL vs. ABNORMAL) hem de

çok sınıflı senaryolar için tüm KDD veri kümesi saldırı türlerini (DOS, R2L, U2R, NORMAL ve PROBE) tanıma ve sınıflandırmada en yüksek doğruluğu göstermiştir. Önerilen KNN ve MLP modellerini kullanan deneysel bulgular, KNN ve MLP kullanılarak yapılan ikili sınıflandırmanın doğruluğunun sırasıyla %99 ve %97 olduğunu göstermiştir. Ayrıca, çok sınıflı sınıflandırma, sırasıyla %92 ve %87'lik yüksek düzeyde doğruluk oranlarıyla daha önceki çalışmalara kıyasla daha iyi sonuçlar üretti. Bu tez, ağ akışı tabanlı izinsiz giriş tespiti için derin öğrenme tekniklerini, yani MNN ve MLP tasarımlarını kullanmanın etkinliğini araştırmaktadır.

Anahtar Kelimeler: Ağ Güvenliği, IDS, Sınıflandırma, KNN, MLP, Özellik Çıkarma, Veri Ön İşleme.

TABLE OF CONTENTS

Pages

LIST OF TABLES

Pages

LIST OF FIGURES

Pages

ABBREVIATIONS

IDS	:	Intrusion Detection Systems
KDD	:	Knowledge Discovery and Data Mining
KNN	:	*K*-Nearest Neighbor
MLP	:	Multilayer Perceptron
FAR	:	False Alarm Rate
DT	:	Decision Trees
SVM	:	Support Vector Machines
RF	:	Random Forests
NB	:	Naive Bayes
ANN	:	Artificial Neural Networks
DDOS	:	Distributed Denial Of Service
ABID	:	Anomaly-Based Intrusion Detection
SBID	:	Signature-Based Intrusion Detection
CNN	:	Convolutional Neural Network
DNN	:	Deep Neural Networks
DFNN	:	Deep Feedforward Neural Network (
DoS	:	Denial Of Service
R2L	:	Root-To-Local
U2R	:	User To Root

MADE : Median Absolute Deviation Estimator

L-SVM : Linear Support Vector Machine

Q-SVM : Quadratic Support Vector Machine

MSE : Mean Squared Error

LSTM : Long-Short Term Memory

RNN : Recurrent Neural Networks

LDC : Linear Discriminant Classifier

AE : Autoencoder

1. INTRODUCTION

1.1 PROBLEM STATEMENT

The internet has become an integral part of contemporary society as a consequence of technological improvements and the passage of time. Forecasts for 2019 indicate that there will be more than 4.5 billion people using the internet globally. In other words, this figure [1] represents almost 60% of the global population. There is a direct correlation between the proliferation of internet users and the explosion in the volume of data sent between different types of electronic devices. On the other hand, many are worried that these innovations would compromise the privacy and security of sensitive information. These issues have arisen because information is now more accessible than ever before. This highlights the growing significance of intrusion detection systems in ensuring a truly secure setting for data transfer.

Data protection is vital for maintaining data availability, confidentiality, and integrity in light of the frequent occurrence of network security breaches. Given the prevalence of attacks that compromise network security, this is of the utmost importance. To counteract potential dangers to networks, many novel protective strategies have emerged in recent years. The implementation of firewalls is one example of this. While these systems do provide some protection, they aren't very good at analyzing network data, which means they can't stop all kinds of attacks [2]. Network security may be further enhanced with the help of an intrusion detection system. Its primary use is in the detection of previously unseen threats as well as newly exposed vulnerabilities.

As a security solution, intrusion detection systems (IDS) keep an eye on your network's resources using a variety of monitoring techniques [3]. This stands for intrusion detection systems. The orientation with respect to the sun allows us to divide these buildings into two broad classes. Hybrid systems that include elements of both host-based and network-based intrusion detection systems are the three primary categories of these systems. In contrast to network-based systems, which oversee the whole network, host-based systems just keep tabs on one host machine. In contrast to network-based systems, which are able to keep tabs on the whole network, host-based systems are only able to keep tabs on specific hosts. The second kind of intrusion detection system looks for potential dangers using signature and

anomaly detection technologies [3-5]. There can be no intrusion detection systems without it.

The primary responsibility of Intrusion Detection Systems (IDSs) in their first stages of operation is to determine if the data sent over the network is normal or associated with an attack. It is imperative that any potentially dubious data gathered in the second phase be labeled as hostile. Threats like Denial of Service, Remote-to-Local, User-to-Root, Probe, and others fall under this category. A low false alarm rate (FAR) and high accuracy in intrusion classification increase the likelihood of success and efficiency for an intrusion detection system [6]. Also, since online threats are always changing, intrusion detection systems need to be able to adapt to keep up [7, 8]. Intrusion detection systems (IDS) use a plethora of methods grounded in deep learning and machine learning to achieve these goals [9].

I find it terrible that when living conditions rise, these issues also grow, even while it is commendable that this way is effectively overcoming hurdles in less developed nations. But its outstanding contribution to alleviating problems in developing nations is ignored. Most people agree that complicated problems are hard to solve and often need cutting-edge technology. Thanks to the possibilities presented by technology, a number of issues have been resolved with the use of powerful processors. The first step on this path was to compute utilizing cutting-edge CPUs. A possible answer to the issue is machine learning. From "genetic algorithms" to "support vector machines," the phrases "machine learning," "artificial neural networks," and countless more are employed in various settings [10–12].

Machine learning is an area of study that entails a set of procedures that computers may use to learn from either pre-existing data or newly created, randomly-generated data. Following that, these machines may use the collected information to foretell and respond to future occurrences. A concept called "deep learning" [13] may be more easily implemented with the help of these methodologies. Processing quality improves with increasing data volume, but processing time also rises. With the rise of "Big Data" in the modern era, consumers now have easy and efficient ways to manage massive data sets. To capitalize on the advantages offered by contemporary civilization's ever-increasing speed, this expanding area of study was founded. However, Deep Learning has developed and expanded with the introduction of new approaches, and it is now considered its own academic field [14].

In order to correctly identify, classify, and make educated judgments, deep learning makes use of huge and varied data sets, including sounds, pictures, videos, network traffic, and human interactions. Compared to Google's projected 20,000 terabytes of data per day, Flicker's daily data production is about 3.6 terabytes [15]. There are a lot of real-time data transfers even in the most fundamental networks that never experience any downtime. When it comes to data flow management in this context, Big Data techniques are essential, and Deep Learning approaches are often used to get important results.

Machine learning refers to the process of improving performance or making predictions by analyzing large datasets using computer technology. From time to time, the process itself is referred to as "it" [15, 16]. By reviewing historical data, Intrusion Detection Systems (IDSs) are able to identify intrusion attempts. Important to the scenario at hand is classification, a cornerstone of machine learning. A variety of machine learning algorithms, including K-nearest-neighbors (KNN), decision trees (DT), support vector machines (SVM), random forests (RF), naïve bayes (NB), artificial neural networks (ANN), and others, are often used by intrusion detection systems [19, 20]. Here we see the circumstance that would be created if sufficient training data is obtained. Researchers have utilized machine learning techniques to attack detection with promising results. If the initial step in developing machine-learning techniques—feature extraction—cannot be carried out correctly, then no amount of code-based design or implementation will improve their performance [21]. The rationale for this is because feature extraction is the first step in these approaches. There is one major caveat that must be considered, even though machine learning procedures are often simple to manage and apply in code [22].

The fast expanding field of deep learning inside machine learning is being used to the particular job of detecting and averting intrusions or assaults. Machine learning is the foundation around which this application is being constructed. Using mechanical devices in a variety of ways, the process of character extraction is carried out. When applied to large datasets, however, these processes outperform machine learning approaches [23]. In particular, machine learning algorithms are designed to work well with little quantities of data. In contrast, when identifying assaults using data from small-scale events, deep learning algorithms often have a higher false acceptance rate (FAR) because to the uneven distribution of data. This is relevant even when the quantity of data being saved is little [24,

25]. We want hybrid methods that combine the best features of deep learning and machine learning to tackle asymmetrically distributed data sets, such those used in intrusion detection systems. The ability to acquire knowledge from examples (IDSs) is foundational to deep learning, but it is also essential to machine learning. Consequently, we can resolve the current problem [26].

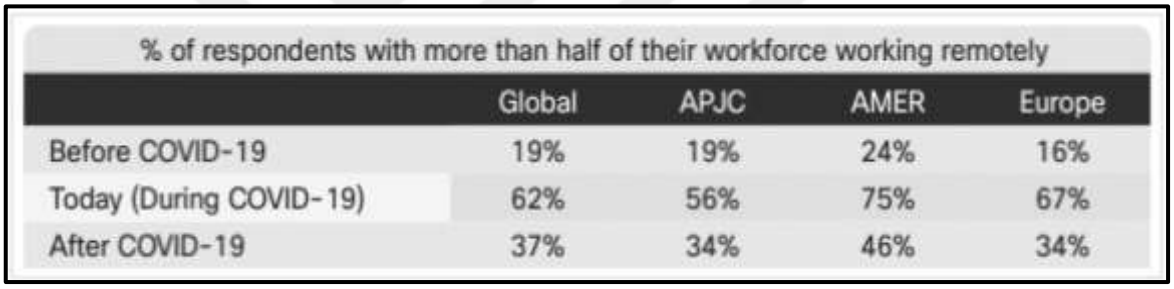
1.2 MOTIVATION

"Cybersecurity" encompasses the measures used to safeguard against malicious software and unauthorized hacking endeavors. Technology is omnipresent in the modern world. With the increasing number of individuals engaging in online shopping and conducting commercial transactions, the potential victims of cybercriminals' assaults are expanding. These systems are very vulnerable to access because they include potentially sensitive information, including personal data, government records, research and development resources, and financial papers. An such instance is the StuxNet virus's attack against the Iranian nuclear program. However, the WannaCry ransomware has significantly increased the likelihood of data loss for several companies, including those in the commercial and governmental sectors. The funds that were sent to the accounts of the hackers, who were accountable for the distribution of the dangerous software and the subsequent data breach, have been successfully retrieved and are now being returned. Cybercriminals have illicitly obtained a substantial sum of money via the use of ransomware. A different kind of cyber-attack is a Distributed Denial of Service (DDoS) attack, which is facilitated by the Mirai botnet. Reddit, Twitter, Amazon, Netflix, and the BBC were among the websites that had a breach in their access.

It is of utmost importance to assess the hourly income of a web company in proportion to the kind of service it provides and the level of confidence it has earned from its consumers, particularly in the current circumstances. This is the epitome of a disastrous collapse for any corporation. Lastly, the Carbanak breach has gained infamy for its impact on banks in over 30 nations. The assault resulted in financial losses ranging from $100 million to $1 billion, impacting more than a hundred distinct institutions. The Carbanak group executed a prolonged assault spanning from 2013 to 2018.

Due to the ongoing COVID-19 pandemic, we have had to make changes to our previous processes. Consequently, several organizations expedited their digital transformation. The growing dependence on remote employment, teleconferencing, online education and training, shopping, media, and other services, as well as comparable activities. Due to the emergence of this "new normal," hackers exhibited heightened interest, leading to a surge in the frequency of cyberattacks [27].

Figure 1.1 represents the findings of the "Future of Protected Remote Work Report," global research undertaken by Cisco. The survey included 3,000 IT decision makers ranging from small firms to huge organizations. In all, 21 markets from the AMER, APJC, and European regions were included for the research. One research found that while the pandemic was going on, the percentage of remote workers rose from 19% to 62% globally.



% of respondents with more than half of their workforce working remotely

	Global	APJC	AMER	Europe
Before COVID-19	19%	19%	24%	16%
Today (During COVID-19)	62%	56%	75%	67%
After COVID-19	37%	34%	46%	34%

Figure 1.1: Survey Results [28].

In addition, 85 percent of participants from other countries said that they believe cybersecurity to be a significant factor in their day-to-day lives. The evidence that was seen is shown in Figure 1.2.

The importance of cyber security, as well as the damage that may be caused by cyberattacks, is immediately obvious when one considers the many real attack examples that are revealed and those that are not discussed. In light of these findings, it is very necessary to make use of the appropriate technology and processes in order to improve the visibility of cyber risks and the identification of them. In the next chapters of this thesis, we will explore not only the current state of the art with regard to technology and detection methods, but also the many ways in which these approaches might be improved.

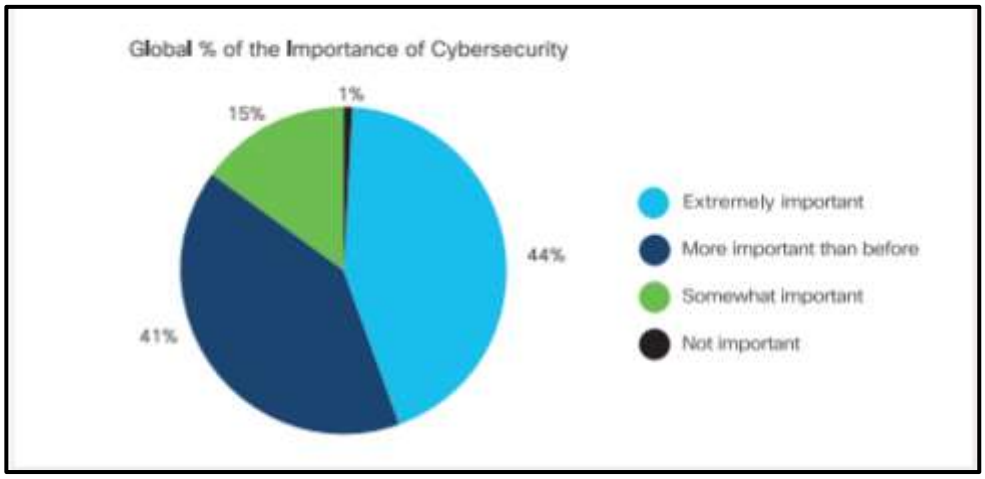


Figure 1. 2: Survey Results of Global Importance of Cyber Security [28]

1.3 RESEARCH QUESTIONS

Thesis study topics include:

i. Machine Learning-based IDS perform better than conventional signature-based IDS in detecting modern attack types?

ii. What is the most effective machine learning strategy for IDSs?

iii. Thirdly, which criteria have the most impact on the reliability with which assaults are detected?

1.4 OUTLINE OF THE THESIS

The following is the structure of this thesis:

In Chapter 2, we have a brief synopsis of relevant studies, which includes an introductory taxonomy of intrusion detection system approaches. The methods have been traditionally divided into two camps, according on whether they prioritize detection or location. There are two subcategories of location-based methods that are comparable to the others: host-based intrusion detection systems and network-based IDS. On the flip side, there are two branches within host-based intrusion detection methods: those that deal with abuse detection and those that deal with anomaly detection. In this section, we will quickly go over the following methods and how they are used in published works.

Using the Multilayer Perceptron (MLP) and K-Nearest Neighbors (KNN) algorithms, the third chapter lays out a machine learning technique for building intrusion detection systems. This chapter's fourth main idea is on. Give a brief summary of the literature outlining the various datasets available today, including their benefits and drawbacks. In the end, the thesis explains why it chose a particular dataset for its experiments.

Utilizing the KDD dataset, Chapter 5 provides a hands-on analysis of the K-nearest neighbors (KNN) and multilayer perceptron (MLP) techniques covered in this study.

We provide a brief overview of the whole thesis topic and propose future research directions in the concluding chapter.

2. LITERATURE REVIEW

2.1 DETECTING INTRUSIONS STUDY PROCEDURE CATEGORIZATION

Sorting audit reports into "normal" and "intrusion" types implies that finding intrusions may be seen as a classification job. Data mining techniques tailored to the detection of security vulnerabilities have recently attracted a lot of attention. Data mining is an essential tool for improving the efficacy of detection criteria and reducing the workload associated with reviewing massive volumes of audit data. A model for Intrusion Detection (ID) was developed by the authors of [28] via data collection and analysis. It became possible to create models for identifying abuse and outliers by studying the data's observed behavior. Data mining technologies made it easier to analyze the data, which in turn helped build these models. The fears associated with exploration provide a significant obstacle. Even if it is already the task of recognizing new threats, it is necessary to enhance classification accuracy utilizing current algorithms or approaches. An important aspect of developing a trustworthy intrusion detection system is making a sophisticated classifier.

A huge increase might be possible as a result of the widespread adoption of internet usage and the fast expansion of the number of devices linked to the internet. A conglomeration of variables, each of which contributes to the function played by this phenomenon, has substantially accelerated the proliferation of linked devices. The exponential growth in the number of people using the internet has been very beneficial to online platforms like email and social media. There has been meteoric rise in the number of people using these types of platforms over the last several years. According to reference [29], the term "big data" was coined as a result of a dramatic increase in the overall amount of data being produced. The total amount of data has grown substantially in recent years, which may be directly related to this new discovery. It is possible to determine the frightening amount of data produced every minute by taking authors' perspectives into account and examining Domo's yearly reports [30].

Internet technology enable firms to enhance customer experience and establish a prominent worldwide presence. Therefore, firms may use the many advantages offered by different internet technologies. These technologies enable companies to sell themselves globally. The extensive usage of the Internet has resulted in some positive outcomes. Nevertheless, it has

also presented a fresh array of obstacles that must be overcome to ensure the ongoing safeguarding of network security in the future. This situation has the potential to lead to both data loss and economic harm [31, 32]. This situation is very probable to occur.

The study described in [33] identifies two primary categories that may be used for intrusion detection systems. The first categorization considers the installation's placement inside the network, while the second classification considers the detection mechanism. Figure 2.1 presents a comparison that clearly demonstrates the difference between these two.

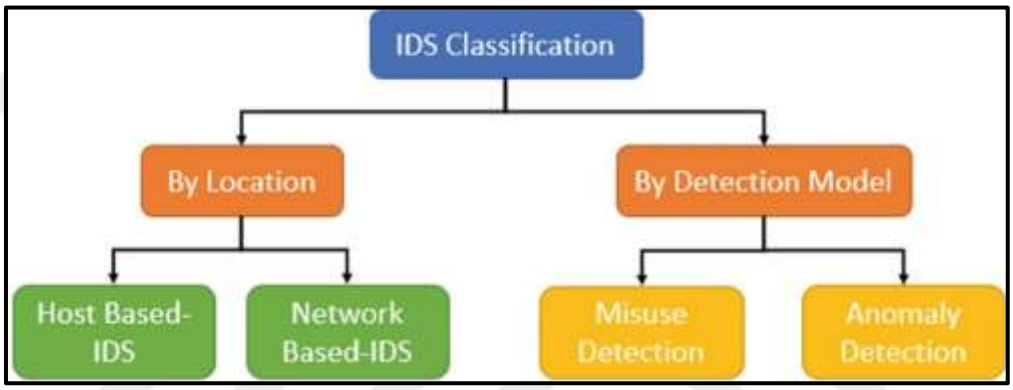


Figure 2.1: IDS System Categorization [33].

The installation of the host IDS is shown in Figure 2.2.

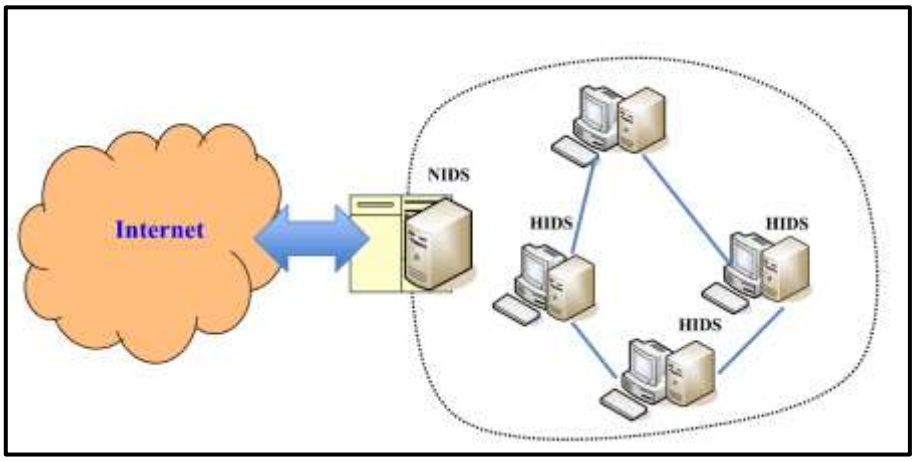


Figure 2.2: The implementation of a Host Intrusion Detection System (HIDS) with a Network NIDS is discussed in reference [33].

In the first method of categorization, there are two branches: the host branch and the network branch. In order to detect any harmful activity, the host-based intrusion detection system checks the client PC and its stored data. Log files, running details, and a list of currently active users are all possible components of the data. If the administrator makes a change to an important user file or the OS, they will be notified right away. This functionality allows the administrator to respond effectively [34]. Intrusion detection systems that operate on a network continuously scan and analyze data packets as they go over the network in order to spot and identify harmful actions, such denial-of-service attacks [35]. Just as the previous set of models, the detection model set may be broadly classified into two types: Misuse detection and anomaly detection (also called signature-based detection) take place. One approach to threat detection is signature-based detection, which involves comparing a user's actions to known hacking techniques. Behavioral matching describes this kind of identification. The goal of anomaly detection is to spot actions that don't fit within a user's or group's established pattern of behavior. This might occur in any situation. For anomaly detection to work, it is typically necessary to build knowledge bases [36, 37]. The activities that are being monitored need to have behavior profile descriptions provided by the knowledge bases. The host-based systems were the first kind of intrusion detection systems developed and deployed. Figure 2.3 is a reference.

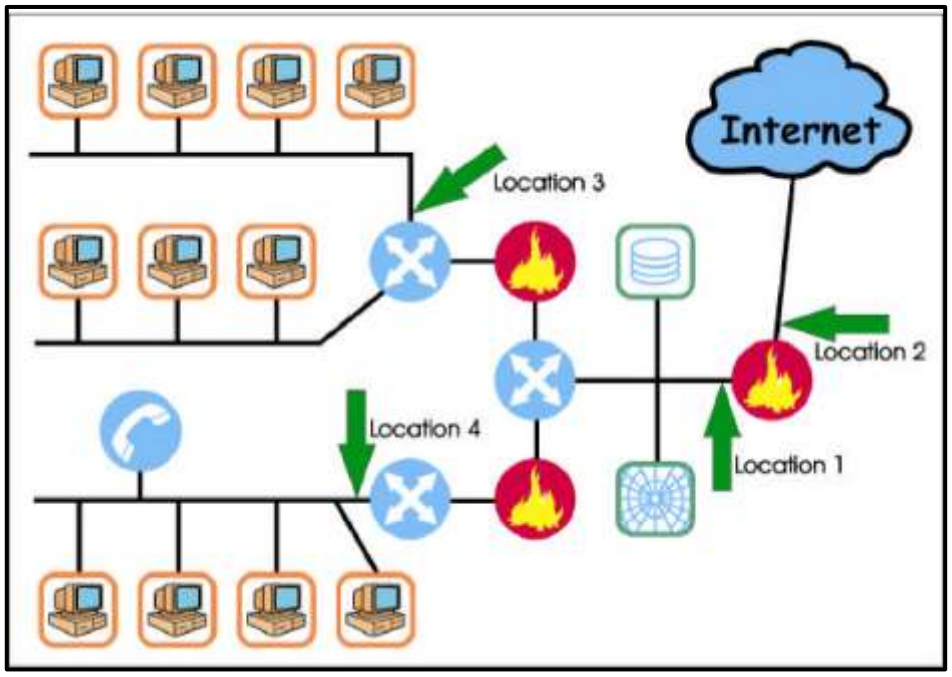


Figure 2.3: Locations of NIDS Sensors [38].

These systems oversee unusual activities that are limited to the localhost, such as unauthorized file access, local system logins, unauthorized privilege escalation, or alterations to system permissions. During the process of control, the principal system's mechanism that regulates and records activity is often used as an information source. The collected data is examined using rule-based engines. For example, "the command is the only method to obtain superuser privileges." Therefore, many login attempts as root might be seen as an attack. Host-based IDSs perform a variety of activities, including integrity checks, event correlation, log analysis, policy enforcement, rootkit detection, monitoring of CPU, RAM, HDD, and battery life, and alerting. The detection of malicious conduct is accomplished by the generation and analysis of reports on these systems. Therefore, it is essential to provide the computers with the suitable software and hardware [39].

Network-based intrusion detection systems use a distinct methodology for detection compared to host-based solutions. Instead of gathering data from individual hosts, these systems examine traffic patterns over a whole network [40]. Therefore, their operating logic is similar to that of a wiretap. Network-based systems scrutinize the contents and header information of every packets that pass over the network in order to detect attacks or abnormal activity [41]. To get precise information on the locations of NIDS, please see Figure 2.4.

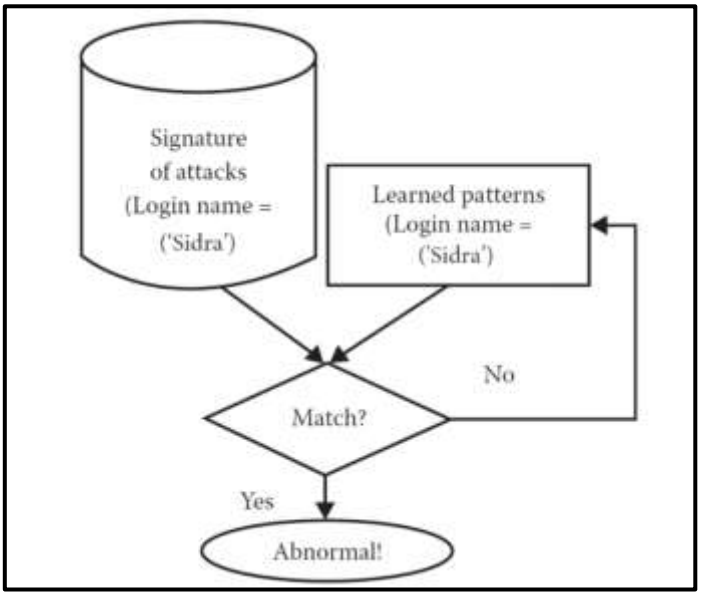


Figure 2.4: Implementation of "If-Else" Rules for Authentication or Misuse Detection [41].

Networked sensors execute the control operation. Included in network sensors are attack signatures, or criteria for determining if an attack has occurred. Skilled users can often create their own attack signatures in most network-based systems. The ability to tailor each network to specific requirements and uses is a result of this [41].

Subsequently, the sensors analyze the gathered traffic and cross-reference it with the known signatures to ascertain the presence or absence of an attack. The method of attack signature detection, also known as abuse detection, utilizes historical data to identify malicious patterns of behavior [42]. The acquired information from these patterns is then used to identify and prevent future network assaults that are similar to those that have previously taken place. Log files and data packets serve as possible repositories for the models of threats that have the ability to impact networks and hosts. A log file's signature is a distinctive arrangement of binary digits, consisting of 0s and 1s, arranged in a certain order [43]. Figure 2.4 provides a visual representation of this notion.

Anomaly detection is a supplementary technique used to identify intrusions. It involves monitoring a host or network to identify any activity that deviates from the usual patterns. These systems see assaults as atypical occurrences and react to them in a corresponding manner [44]. Thus, it is conceivable that assaults may be thwarted by using technologies that detect these disparities. Anomaly detection systems often use the approach of creating profiles. These profiles establish the typical behaviors of users, hosts, and network connections. These profiles were generated using data obtained during the system's regular operation to enable it to carry out its intended tasks. Anomaly detection detectors are tasked with collecting this information. These detectors use many criteria to ascertain the presence of aberrant activity [45]. Refer to Figure 2.5 for a detailed explanation of the conceptual functioning of anomalous intrusion detection system techniques that rely on ML, as described in reference [45].

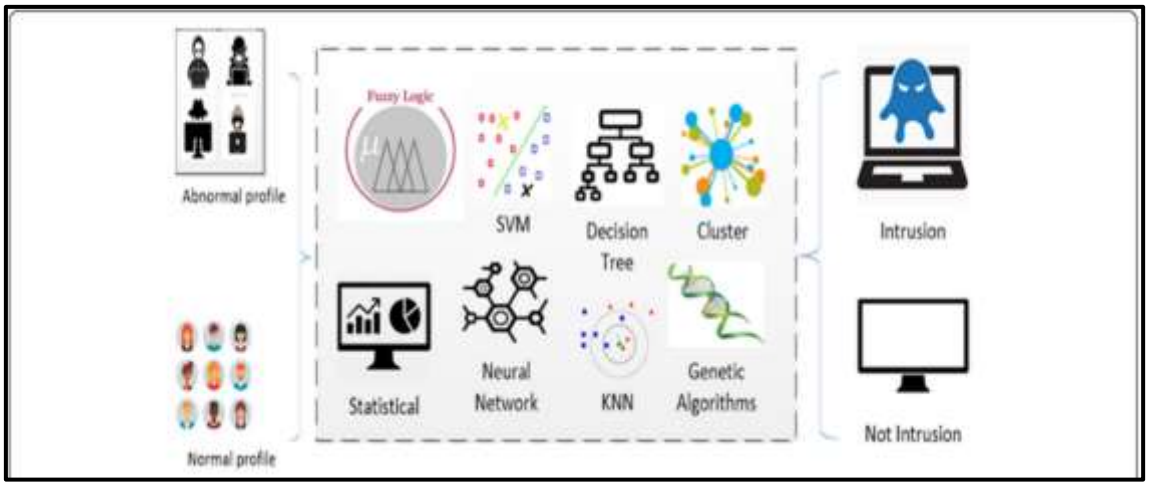


Figure 2.5: Conceptual Working of Anomaly IDS Approaches Based on ML [45].

Many academics in academia have shown interest in anomaly-based IDS because it has the potential to solve a limitation of signature-based SIDS. When developing a model to represent the behavior of a computer system in AIDS, it is customary to use techniques like as machine learning, statistical analysis, and other knowledge-based methodologies [46].

The ability of AIDS to identify zero-day attacks is the most crucial component of this security solution. This is possible because AIDS does not depend on a signature database to detect abnormal user activity [47]. AIDS will trigger an alert if it detects any abnormal activity. Furthermore, there are certain advantages to having the medical condition referred to as AIDS. One advantage is their ability to see potentially detrimental processes occurring inside their own bodies. If a thief utilizes a compromised account to carry out transactions that considerably depart from the typical pattern of activity on that account, an alert will be triggered. Furthermore, due to the presence of individual user profiles inside the system, it becomes very difficult for malicious actors to discern customary patterns of conduct [48]. The fundamental functioning of machine learning-based approaches for detecting abnormal intrusions is shown in Figure 2.5.

2.2 ML BASED IDS

In the realm of cyber security, the problem-solving process is executed through the utilization of diverse methodologies, which are chosen based on specific parameters such as the nature of the problem, the volume of data involved, the level of sensitivity associated with the problem, and the degree of tolerance for decision-making in the solution [49].

During stages of the process when the data is extensive and requires advanced methods, deep learning approaches that rely on parallel processing are used. This section analyzes and categorizes studies that have investigated the use of Deep Learning techniques in the field of Cybersecurity, based on their various subheadings. However, the bulk of research may be categorized into many groups. Figure 2.6 presents a taxonomy that represents the machine learning algorithms currently being studied in this specific topic [50].

Deep learning networks may be designed as both locally based systems and server-based systems. When working with such systems, it is crucial to implement safeguards to safeguard the integrity and security of the data, as well as ensure its reliable transmission. The study has resulted in the development of a two-phase model. The first stage of the procedure is encrypting the area inside the local environment via which the data is being transmitted to the server. The subsequent step is transferring this data to the server, followed by processing the encrypted data upon its arrival at the server [51-53].

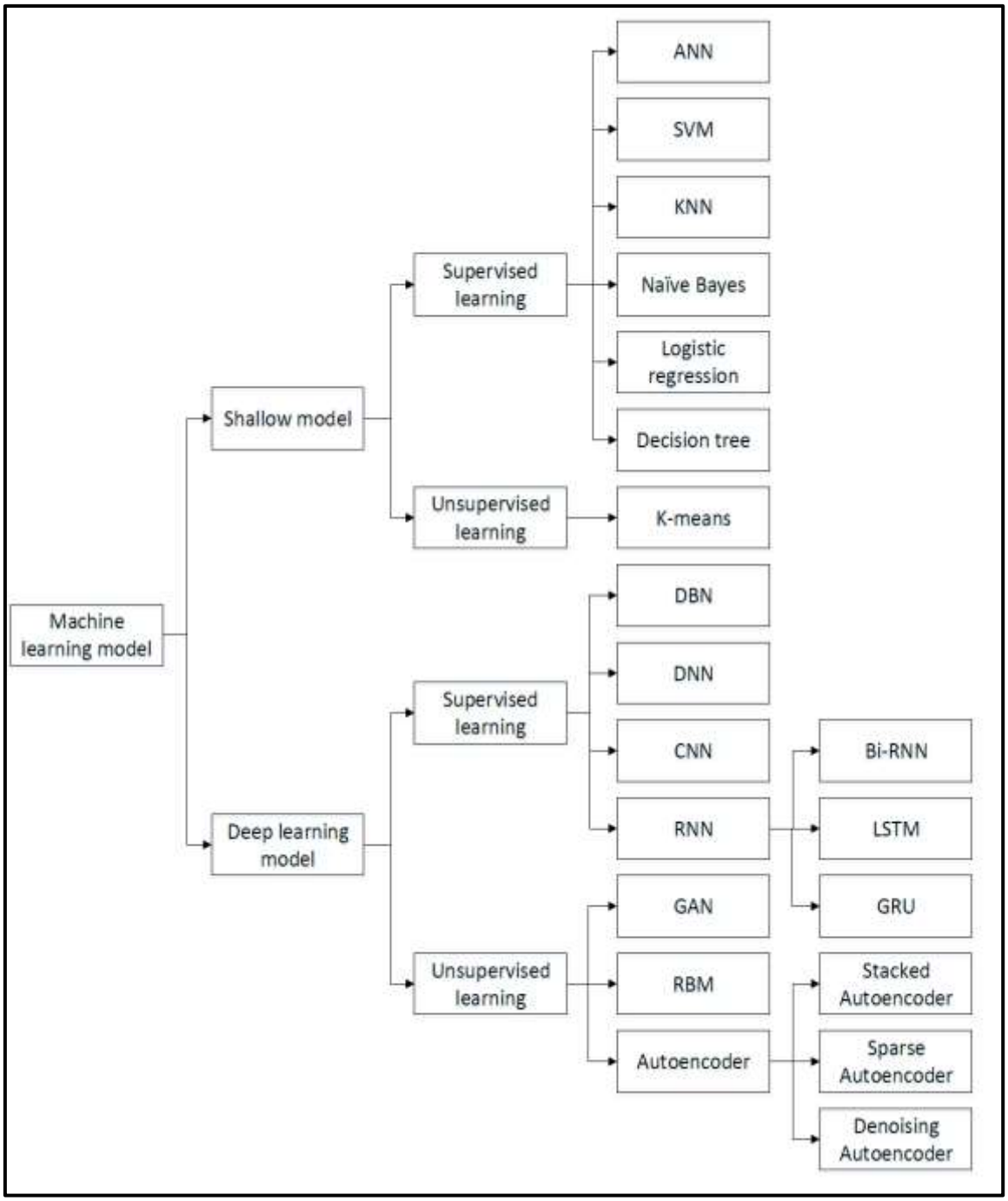


Figure 2.6: Taxonomy of Machine Learning Approaches [55].

During the two-stage character recognition procedure using photos, when the characters are encrypted and transferred, any data intercepted by a man-in-the-middle attack between the server and the local system appears as a nonsensical data stream. The reason for this is because the data is not capable of being decoded [54]. Consequently, one may achieve the secure transmission of information. All areas, with the exception of deep learning, have

tackled difficulties related to information security. However, the issue of information security in deep learning remains unresolved. The challenges that have been tackled in all other fields, save for security, are based on the fundamental principles of information security [55].

Data may be accessed and sent across several networks in the virtual realm. If the data stored and sent via networked systems is important, it is necessary to implement adequate security measures in these situations. The number of violations in a network environment fluctuates based on the degree of activity and the extent of the network. As the number of users and the size of the network increase, the amount of data being sent and requiring processing also increases. Utilizing technologies like parallel processing and deep learning [56-58] enables the efficient and accurate handling of vast quantities of data. The NSLKDD dataset, renowned for its popularity, was used to perform experiments on the efficacy of detecting network breaches using RNN in the study article. The dataset contains sub-datasets named Training, Test, and Test21. This document presents binary and categorical representations of the data for your examination [59].

In our trials, we achieved a training dataset accuracy rate of 99.81% using a learning rate of 0.1 and 80 hidden layer nodes. The learning accuracy is good, but the performance is subpar in comparison to other classification methods. Utilizing parallel operations on graphics processors enables rapid classification, resulting in increased speed [60].

Acts of physical aggression, such as targeted digital assaults in the realm of cyberspace, have the potential to undermine the integrity of the hardware and transmission systems that support cyber environments. Utilizing technical gear may also guarantee the safety of important locations via detection and evaluation approaches. In a study on video processing and evaluation, researchers used Convolutional Neural Network (CNN) technology to detect potentially suspicious activity in a susceptible geographical area. The Quadratic SVM strategy has been tested with the GoogleNet and AlexNet methods of convolutional neural networks (CNN) to classify items into four unique classes: backdrop, bag, deer, and person. Although CNN techniques have higher success probability compared to SVM, Quadratic SVM has shown subpar performance, especially in the task of human detection. When it comes to ensuring security in the cyber physical domain, the adoption of Deep Learning

algorithms may considerably improve video processing, which entails analyzing both static and dynamic pictures [61-64].

For the second time, the authors of [41, 66] recommended a hybrid strategy that makes use of deep learning methods in order to successfully escape intrusion detection systems. For the purpose of doing this, the researchers used the Adasyn methodology to ensure that the various kinds of assaults within the NSL-KDD dataset were distributed in a regular manner. The CNN model was generated after data preprocessing. The suggested model is constructed using the split convolution module, which may reduce the influence of duplicated cross-channel input during the training phase. In a recent study, a complex strategy using KNN, hyper-learning machine, and hierarchical hyper-learning machine classification methods was introduced in reference [67].

The training dataset is divided into two equal halves, with one half labeled for each kind of attack (DoS, Probe, etc.), and the other half labeled as "other," which might indicate either regular traffic or an unrecognized assault. The recommended method divides the training dataset into two equal halves at each layer. Each layer of this model utilizes just one classifier for each iteration.

Deep learning approaches include the CNN technique, first proposed by Wu et al. (Wu et al., 2018). This procedure is one of many possible ways. As part of the data preparation procedure, the normalized data was converted into a picture data format to enhance the research performance of the CNN. During the training phase, the parameters of the proposed model were identified and their optimal values were automatically computed. Figure 2.7 depicts the logical progression used by the CNN model. Jiang et al. [24, 65] advocate using a mix of CNN and BLSTM for implementing a deep hierarchical network model.

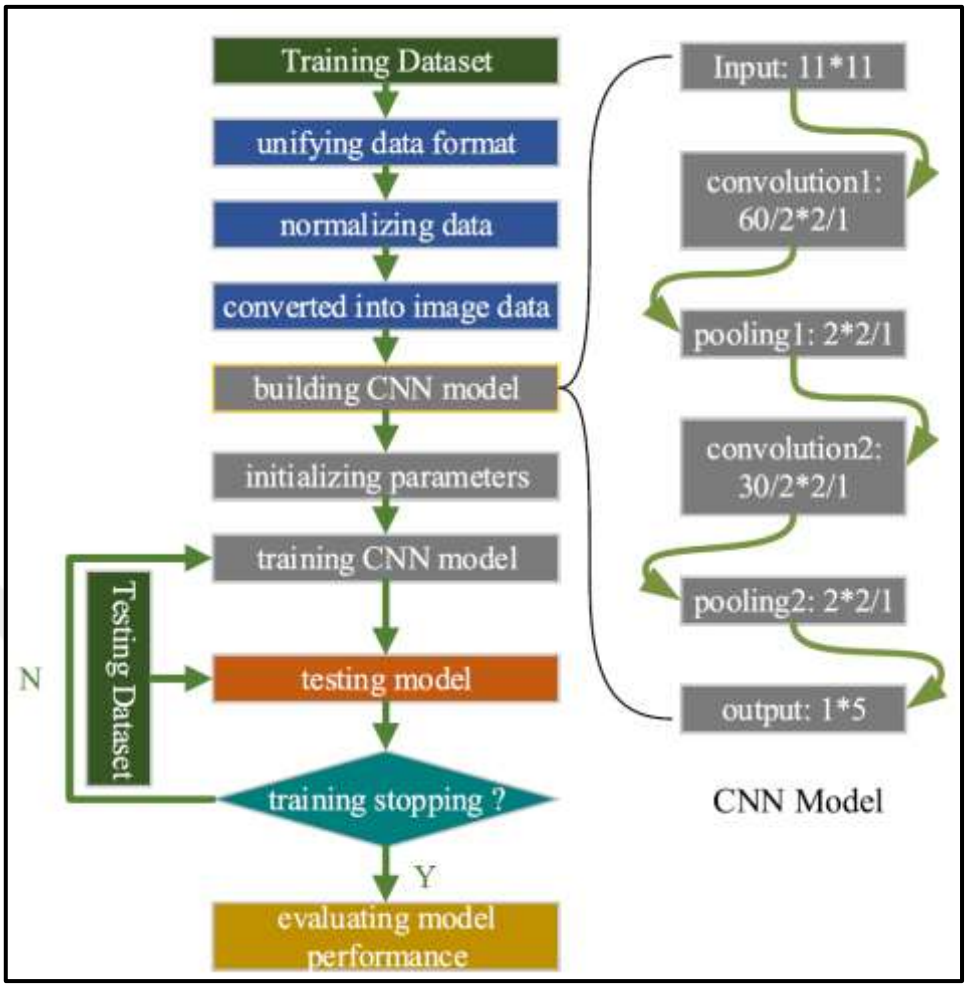


Figure 2.7: Flow Chart of the IDS Model-Based CNN [65].

The information gain strategy was used in publications [68, 69] to discover features for the NSL-KDD dataset. Next, a unique classification model, which is inspired by Support Vector Machines, is introduced. The model's effectiveness in multi-classification is shown by the meticulous selection of features and the use of an improved version of the Support Vector Field Description approach called SSPVSVDD (SVM). The NSL-KDD dataset is used as the basis for the proposed model. It had a success rate of 77.5%. The research by [70] proposed a hybrid system that combines Deep Belief Network (DBN) and weighted Support Vector Machine (SVM) approaches. The proposed methodology utilizes the Deep Belief Network (DBN) method for extracting features, together with the weighted Support Vector Machine (SVM) strategy for enhancing features and performing classification.

Using a system that is capable of detecting assaults via the use of deep learning was advised by the study that was referenced in reference [71]. In the setup that has been provided, an

entropy-based feature extraction approach is that which is used. Following the completion of the feature extraction process, we contrasted our recently developed approach, which made use of a feedforward deep neural network, to more conventional approaches to machine learning. An illustration of the structure of the suggested model may be seen in Figure 2.9. Using this strategy, which was presented in [72], classic machine learning techniques are combined with more modern deep learning approaches. Classification is accomplished by the use of the SVM approach in this strategy. Previous to this, the framework known as Self-Taught Learning (STL) was used in order to accomplish the tasks of feature learning and dimensionality reduction.

Throughout the whole of our research, we made certain that the dataset had an equal number of instances of each kind of vulnerability. In order to reduce the number of samples from the majority class, the One-Side Selection (OSS) approach was used. On the other hand, the SMOTE method was utilized in order to increase the number of samples from the minority class. It is possible to see the whole CNN model that was constructed by the researchers in Figure 2.8.

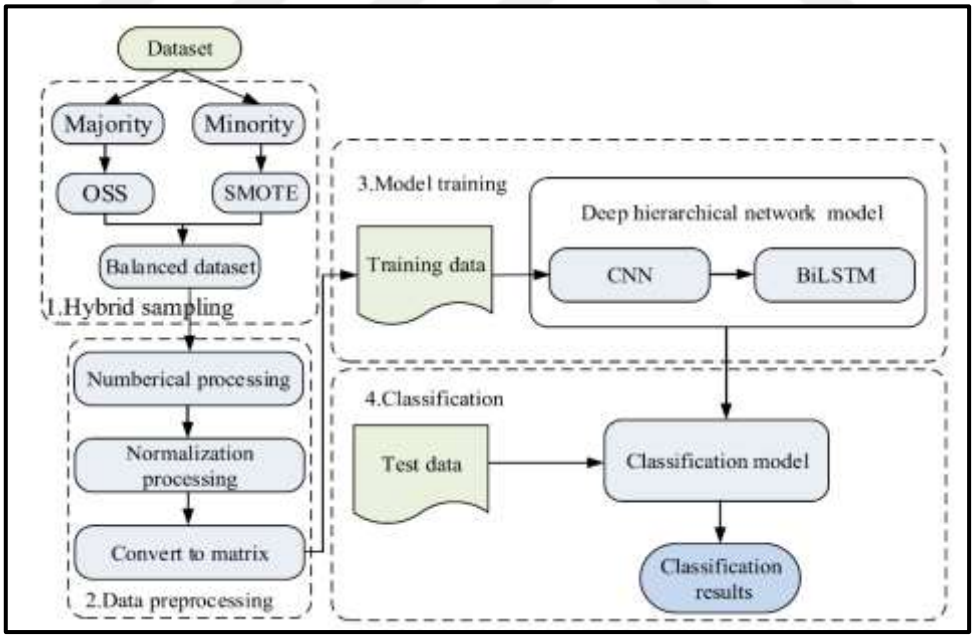


Figure 2.8: Overall Framework of NIDS [72].

The effectiveness of [69] community-based learning-based intrusion detection systems has been studied. In this approach, the NSL-KDD dataset was subjected to genetic algorithm feature selection. Then, the associated dataset was used to evaluate the efficacy of various ensemble learning techniques and Deep Neural Networks (DNNs) for binary classification, with and without feature selection [70, 72]. Machine learning approach used the FFDNN architecture for intrusion detection [73], the flowchart is presented in Figure 2.9.

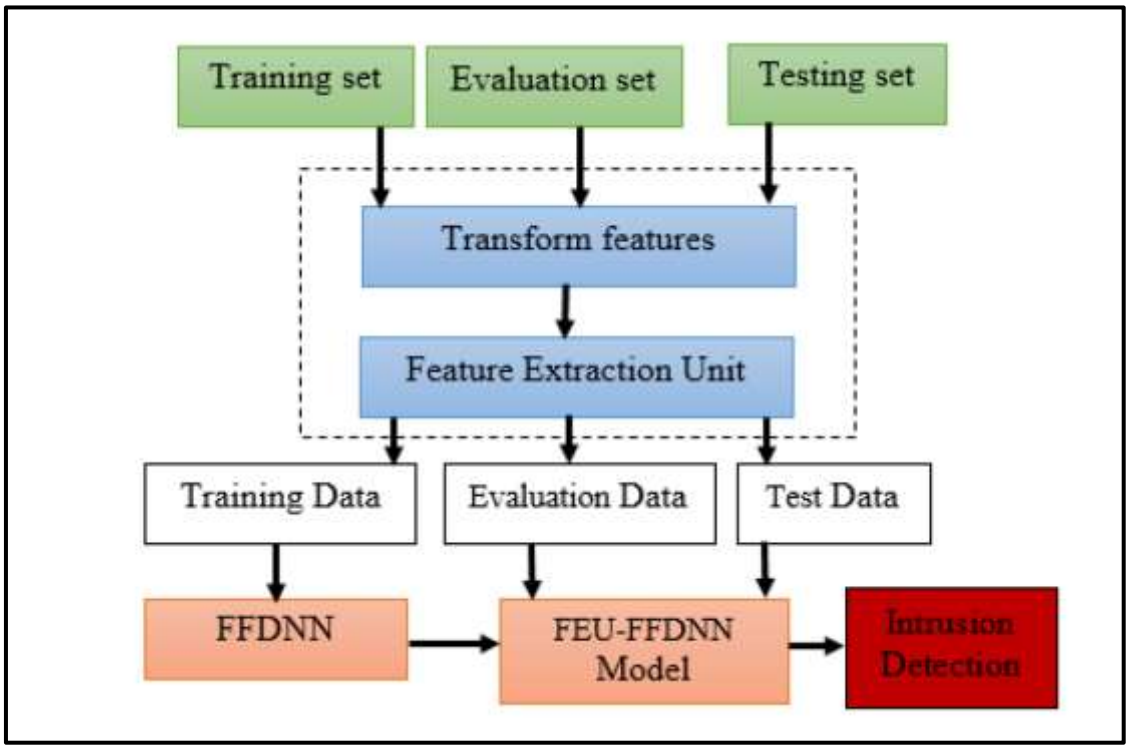


Figure 2.9: FFDNN Architecture [73].

The outcomes of several machine learning techniques [69] suggest that community-based approaches have the potential to successfully rival other methods in the field. These techniques' results support this conclusion. The unequal distribution of data flow in the network significantly impairs the performance of the models, resulting in a substantial negative effect. The research's settings were maintained and the data set remained unchanged. The model used in the study effectively minimized the impact of data imbalance on categorization. This was achieved within the limitations of the study. To accomplish this goal, the renowned KNN algorithm was used on non-uniformly distributed data. This was accomplished to fulfill the objective. This results in a significant reduction in the time needed

for the stages of model creation and testing compared to traditional models. The KNN method is specifically designed for efficient and concurrent data processing.

The successful achievement of the objectives of this thesis relies on the use of a hybrid intrusion detection system. There is a potential for achieving better results by integrating deep learning with machine learning, and our approach acknowledges this potential. Due to the implementation of the new system, the categorization process has been separated into two separate phases. To begin the process, we use the KNN and MLP classification algorithms to distinguish the attack data from the normal data.

The second step of the approach included classifying the different sorts of attacks using a Deep feedforward Neural Network (DFNN) model that was particularly created for this task. When the appropriate parameter settings are used, it has the ability to provide outcomes that are superior in terms of both performance and accuracy. These outcomes may be attained by using the essential parameters.

3. MATERIALS AND METHODOLOGY

3.1 MATERIALS

A number of IDS often utilize the NSL-KDD dataset as a benchmark. This dataset is a subset of the original KDD99 dataset. Standard operating procedure. Several of the complaints mentioned about KDD99 have been adequately addressed by NSL-KDD. Some people thought the classifiers were biased towards more common samples since there was duplicate and redundant data in both the training and testing sets.

One major issue that prevents learning algorithms from drawing broad conclusions from uncommon items is the high number of duplicate entries in the KDD dataset. Due to their susceptibility to attacks like U2R and R2L, these unusual records are often more detrimental to networks. In addition, the evaluation findings will be slanted in favor of the approaches with higher rates of detecting duplicates in the test set.

The complexity of the items in the KDD dataset was also investigated. All 21 students surprised everyone by correctly classifying over 98% of the training data and 86% of the test records.

A dataset called NSL-KDD is made available without charge by the Canadian Institute of Cybersecurity [74]. It has training and testing data sets that are here referred to as KDD_{Train+} and KDD_{Test+}, respectively, and contain 125973 and 22544 occurrences. Specifically, the occurrences belonging to such categories (3751) were excluded because the KDD_{Test+}, dataset contained seventeen more attack types that weren't present in the KDD_{Train+} for a fair classification. As a result, there were 22544 - 3751 = 18793 samples in the KDD_{Test+},. Table 3.1 provides more information on the KDD_{Train+} and KDD_{Test+}, sets. The NSL-KDD dataset includes z_f(f=1,2,..41) features, of which 38 are continuous and 3 are symbolic. In addition, the NSL-KDD dataset's attack types are grouped into four distinct attack classes:

i. DoS (Denial of Service): DoS attacks include those that flood a server with more data than it can handle, either slowing it down or causing it to shut down entirely. Disruption of service attacks disrupt valid connections to networks or services.

ii. An R2L (Root-to-Local) attack allows for unauthorized local access to a machine by sending it remotely crafted packets that fool the system into thinking they are legitimate.
iii. User to Root (U2R) vulnerabilities include those that can be used to get root access. Here, the hacker takes use of the system just like any other user after discovering its flaw.
iv. Probe (Probing): Probes are attacks that learn about a network in order to bypass security safeguards.

The NSL-KDD dataset's assault categories are detailed in Tables 3.1.

Table 3.1: NSL-KDD Dataset.

NSL-KDD	Total	Normal	DoS	Probe	R2L	U2R
KDD_{Train+}	135,973	67,373	45,927	11,656	995	52
KDD_{Test+}	19,723	9813	5841	1206	2189	47

3.2 METHODOLOGY

The area of ML is where the disciplines of statistical analysis, computer science, and artificial intelligence all come together. It is possible that you are familiar with the term statistical learning or predictive analytics; but, regardless of the name you give it, the process of gaining information from data is the same. In recent years, approaches based on machine learning have become more prevalent in a variety of fields. There are a number of cutting-edge websites and devices that make use of ML methods to do a wide range of tasks, including automatically recommending what to watch, eat, or buy, as well as identifying friends based on personal photographs. The significance of machine learning goes well beyond its commercial applications, and it plays a significant role in transforming the way data-driven research is conducted [75].

Through the process of analyzing examples, the objective of ML is to instruct computers on how to optimize a performance measure. For the goal of doing this, mathematical models that are adapted to the particulars of the circumstance are built. The process of machine learning is strongly dependent on drawing conclusions from samples; hence, statistical theory is used in the process of developing mathematical models. Because we require an

algorithm to solve the optimization problem, and because we need to store and analyze a large amount of data, both of which are made feasible by advancements in computers, computer science is vital. First, we need an algorithm to solve the optimization problem. When a model has been trained, its algorithmic solution is successful in representing and inferring it [76]. This is another advantage of the learning process.

It is possible to train the algorithms that are used in machine learning in a number of different methods. The use of each of these strategies is connected with a number of advantages as well as disadvantages. In order to choose the most effective training algorithm, the first step is to examine the data that is being entered into the machine learning system. There are two types of information that are used in the field of machine learning: labelled data and unlabeled data.

It is believed that data has been labeled when the parameters of both the input and the output are structured in a way that is totally readable by machines. In spite of the fact that computers are able to easily process data that has been labeled, the process of labeling the data itself requires a large amount of human labor. Unlabeled data is data that is either just including one parameter or none at all, and it is presented in a way that is usable by machines. However, despite the fact that this minimizes the need for human labor, it still need more intricate solutions in order to be effective.

The procedure that must be carried out in accordance with the framework of the method that is advised is shown in Figure 3.1. After removing any outliers from the NSLKDD dataset, we use the min-max normalization method to the data in order to reduce it down to a range that is between 0 and 1. This is followed by the elimination of any further outliers. The very last step involves converting symbolic (or category) qualities to their numerical equivalents. This is performed by the use of one-hot programming, which is the last stage. After that, we conduct a statistical analysis on each of the 38 numerical features in order to discover which qualities are most closely related with one another according to the results of the analysis. Shallow networks, which include MLP, L-SVM, Q-SVM, and LDA, are used in the process of determining how good the detection is in binary circumstances. When it comes to multi-classification, on the other hand, we choose to make use of deep neural networks.

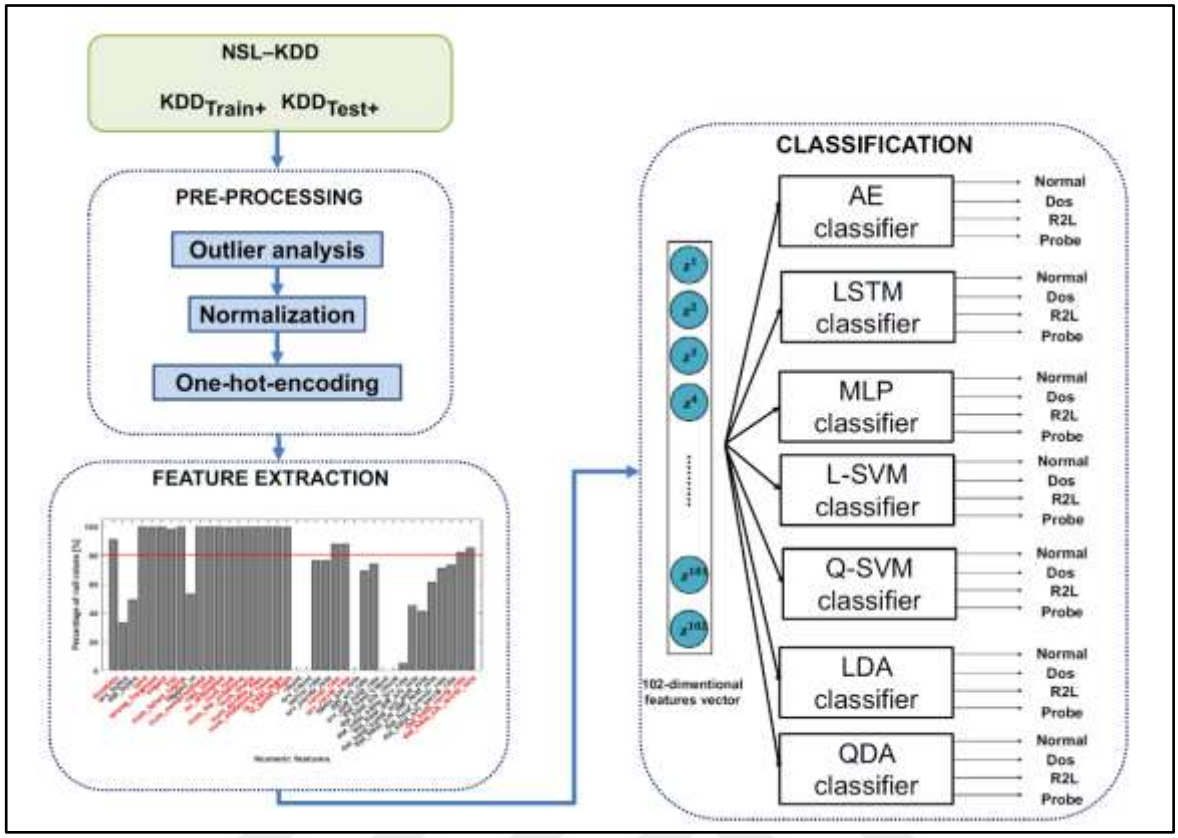


Figure 3.1: Process Flow Diagram for The Suggested Approach (Starting From Pre-Processing, Ending With Classification).

3.2.1 Data Preprocessing

The proposed pre-processing step efficiently sorts the information for use by subsequent modules. It consists of three sub-parts: an examination of outliers. It has been established that removing outliers from the NSL-KDD dataset is an essential step prior to data normalization, thus this is what is done. The proposed intrusion systems may fail to identify intrusions because outliers interfere with learning. For this purpose, we will utilize a statistic known as the Median Absolute Deviation Estimator (MADE), which has the following definition:

$$MADE = P * med(z_{fj} - |med(z_{fi})|) \quad (3.1)$$

Where med stands for the median operator, z_{fj} stands for the sample of the property z_f, and P=1.4826 is a multiplicative constant that is frequently employed when data normality is assumed. in this study, z_{fj} was an outlier if:

$$z_{fj} > p * MADE \tag{3.2}$$

with p=10. KDD_{Train+} size decreased from 125973 to 85421 after the outliers were removed, whereas KDD_{Test+} size decreased from 18793 to 11925. Table 3.2 details the new, outlier-free dataset (NSL-KDD*). The dataset has a serious sex and gender gap that has to be addressed. In reality, there are only 18 U2R assault scenarios included. Hence, the U2R group was excluded from the final data set.

For data normalization, the numeric features values z_{fj}were mapped into the range [0-1] using the min-max normalization approach, as stated in:

$$\sim z_{fj} = \frac{z_{fj} - min(z_f)}{\max(z_f) - \min(z_f)} \tag{3.3}$$

Table 3.2: The New Dataset Named NSL-KDD* After Removing Outliers.

NSL-KDD*	Total	Normal	DoS	Probe	R2L	U2R
$KDD^{*}Train+$	85,421	51,551	23,272	9683	874	12
$KDD^{*}Test+$	11,925	7341	1975	620	1971	12

The normalized feature value, z_fj, may take on values between 0 and 1, and the maximum and minimum values of the fth numerical feature, max(z_f) and min(z_f), respectively, provide the context. One hot encoding is the last stage of data preparation. The three categorical features—protocol type, service, and flag—were transformed into numerical values using the one-hot encoding approach. Specifically, every category characteristic is represented by a binary value. One example is the z_2feature (protocol type), which has three attributes: tcp, udp, and icmp. Their corresponding values are [0, 1, 0], [1, 0, 0], and [0, 0, 1] as a result of being converted into binary vectors using the one-hot encoding approach. Also converted to one-hot-encoding vectors were z_3 and z_4's service and flag characteristics. After mapping the 41-dimensional features (38 continuous and 84 with binary values associated with z_2, z_3, and z_4), a total of 122-dimensional features were generated.

3.2.2 Feature Extraction

The most correlated features are extracted by this processing module. Both the KDD^*_{Train+} and KDD^*_{Test+} sets examine the percentage of zeros for each continuous feature. The distribution of null values for each numerical variable in the KDD^*_{Train+} . This study does not go farther into feature vectors that include eighty percent or more zeros, as seen in Figure 3.2. Make a 102-dimensional features vector using only the 18 continuous features and 84 one-hot-encoding vectors instead of utilizing all 20 variables (red-highlighted).

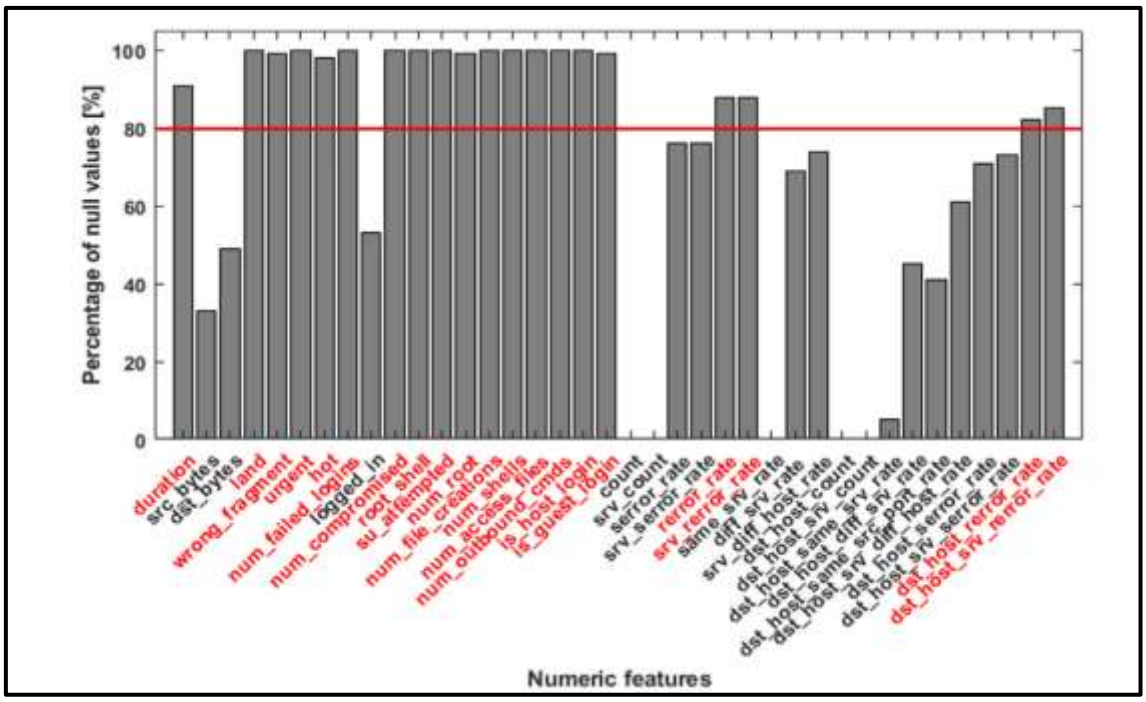


Figure 3.2: In This Histogram, We Display the Null Values With the 38 Numerical Variables That Make Up the $\boldsymbol{KDD^*_{Train+}}$ Dataset. All Red Features Have An 80% Or Above Zero That is Not Included in the Present Analysis.

These shallow classifiers (MLP, LSTM, L-SVM, Q-SVM, and LDA) and deeper classifications (MLP) use such a vector as their input (KNN).

3.2.3 Feature Classification

Within the NSL-KDD dataset, five separate designs have been created to distinguish between healthy and sick subgroups. There is a KNN-based intermediate architecture, an

MLP-based deep architecture, and an AE-based intermediate architecture with a linear and quadratic discriminant function. The set is rounded out by three shallow architectures: distributed architecture (DA), SVM (with sequential and quaternary kernel), and standard multi-layer perceptrons (MLP). All of these configurations were trained using the NSL-KDD dataset. Modes such as Normal, DoS, R2L, and Probe are included in this category. The following parts will provide a more detailed explanation of the methodology used for classification, in the order given:

3.2.3.1 Autoencoder

Autoencoders are highly valuable in the realm of unsupervised learning. You may use them to reduce the data size and lower its dimensions. Autoencoders allow us to recreate our input data from a compressed version of it, while PCA identifies paths along which you may project the data with greatest variance.

An AE-based deep classifier was constructed as part of this research project during the course of its duration. Coder and decoder operations are performed. The AE architecture is comprised of several components, which are listed below. Initial, it takes the input data vector and converts it into a representation that is often more compact, which is referred to as an encoder. The next step is for it to make an effort to decode the compressed vector in order to reconstruct the initial input. The result, which is the decoder, is finally produced by it.

Because it is trained in an unsupervised manner, the AE may get useful features out of unlabeled data [77]. A typical AE model with a single hidden layer is shown in Figure 3.3. A lower-level representation e is used to encode the input data vector z:

$$e = \varsigma(zW + b) \tag{3.4}$$

In this case, W denotes the weight matrix, b the bias vector, and & the activation function of the encoder. Afterward, decoding causes the input (z) to be reconstructed from the encoded representation (e) in the following way:

$$\tilde{z} = \varsigma(eW^T + b) \tag{3.5}$$

Where ς represents the activation function for decoder and $\tilde{z}$ is the reconstructed vector.

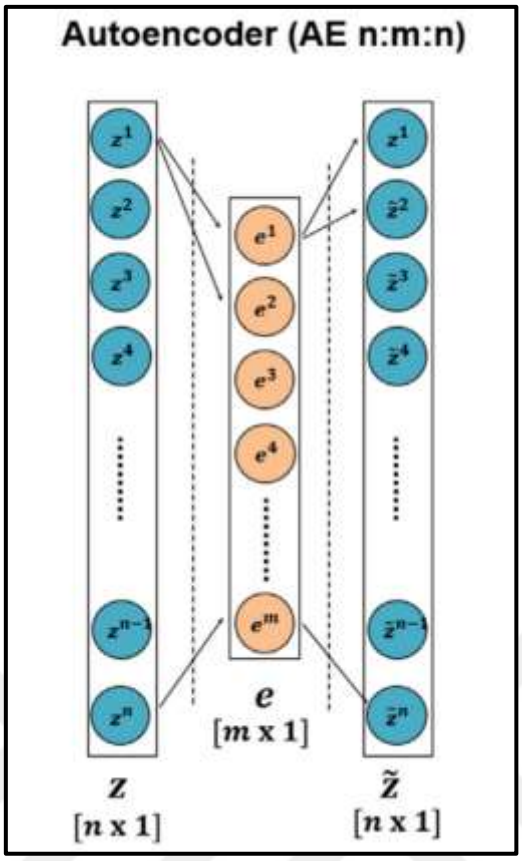


Figure 3.3: The Traditional Autoencoder Model.

3.2.3.2 Proposed autoencoder architecture

When the input and the output are identical, as is the case with autoencoders, we say that the network is feedforward. They take the input and transform it into a more compact form before reconstructing it as the final output. The code, which is also known as the latent-space representation, is a version of the input that has been compressed.

There are three basic components that make up an autoencoder. These components are the encoder, the code, and the controller. For the purpose of reconstructing the original input, the decoder makes use of the code that is generated after the input has been compressed. This occurs before the code generating process. To construct an autoencoder, three components are required: an encoding technique, a decoding technique, and a loss function to evaluate the output's similarity to the target. The next section will go into these topics. The encoder architecture presented in Figure 3.4.

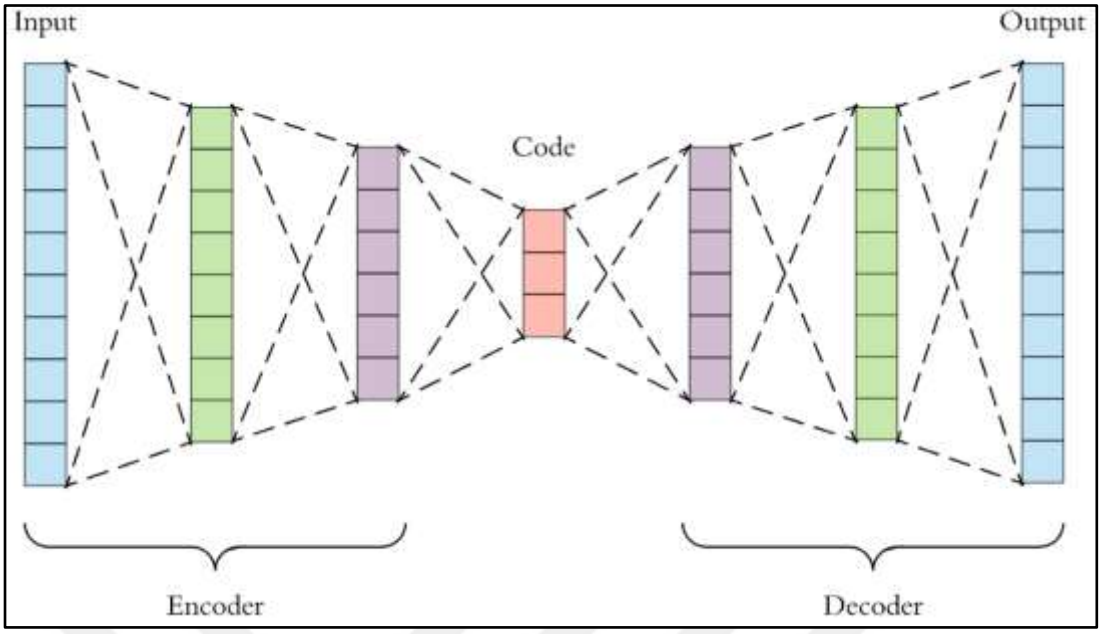


Figure 3.4: Encoder Architecture.

Autoencoders are a kind of dimensionality reduction technique, often referred to as compression, that has the following distinct features: For autoencoders to effectively compress data, the data used for training must have a high degree of similarity to the data that has to be compressed. It is crucial to acknowledge that these algorithms vary from basic data compression approaches like gzip because they acquire knowledge about features that are unique to the provided training material. Hence, it is not rational to expect that an autoencoder, which underwent training using handwritten digits, would be capable of effectively compressing landscape images. The output of a lossy autoencoder yields a representation that is similar to the input, however of worse quality. If the primary concern is achieving lossless compression, they are not a good choice.

Unsupervised learning involves training an autoencoder by directly inputting the raw data without any human supervision or guidance. Unlike supervised learning approaches, autoencoders do not need labels throughout the training process. More precisely, these models are self-supervised since they classify their own data by using knowledge from the training set.

This provides a more comprehensive visual representation of an autoencoder. The input is first sent into the encoder, which is a fully-connected artificial neural network (ANN) responsible for producing the code. The decoder utilizes the same artificial neural network

(ANN) architecture to create the output only based on the code. The objective is to establish a direct and exact relationship between the input and output. It is crucial to remember that the construction of the decoder is a complete reversal of the encoders. While it is not obligatory, it often tends to happen in that manner. The only limitation is that the size of the input and output must be the same. Anything located inside the center is considered acceptable and may be used.

3.2.3.3 Long-short term memory architecture

LSTM, an artificial recurrent neural network based deep learning architecture, is advantageous in sequence prediction problems due to its ability to learn order dependence. The outcomes derived from the Recurrent Neural Network (RNN) in the preceding step are used as input in this phase. The LSTM was developed in [79]. The RNN's deficiency in accurately predicting words from its long-term memory was rectified, but its enhanced accuracy in prediction when using more recent data persisted. As the intervals between observations increase, the effectiveness of RNN decreases. LSTM networks have the capability to retain and preserve information over extended periods. Prediction, classification, and time series processing are among the several domains in which it is used.

The feedback connections of LSTM allow it to overcome the shortcomings of classic feed-forward neural networks. In addition to processing continuous streams of data like audio or video, it can also evaluate individual data units like images. A few examples of possible LSTM-enhanced applications include unsegmented, continuous handwriting recognition and voice recognition.

The LSTM model consists of four distinct neural networks, with each network including many memory blocks, often known as cells. The fundamental components of a LSTM device are a cell, gates for inputting and outputting data, and a forget gate for deleting data. Three gates are in charge of regulating the flow of data into and out of the cell, and the information may be stored indefinitely. Figure 3.5 demonstrates that the LSTM approach performs very well in classifying, evaluating, and predicting time series of varying lengths. These are all the areas in which the algorithm is shown. Every single one of these recurrent neural networks is constructed using a series of repeated neural network modules. If a recurrent module exists in normal RNNs, it will consist of a maximum of one tanh layer. Recurrent,

as a particular kind of iteration, utilizes the outcomes of one "time step" and applies them to the subsequent step. The model takes into account both the present input and all prior inputs to determine the most effective response.

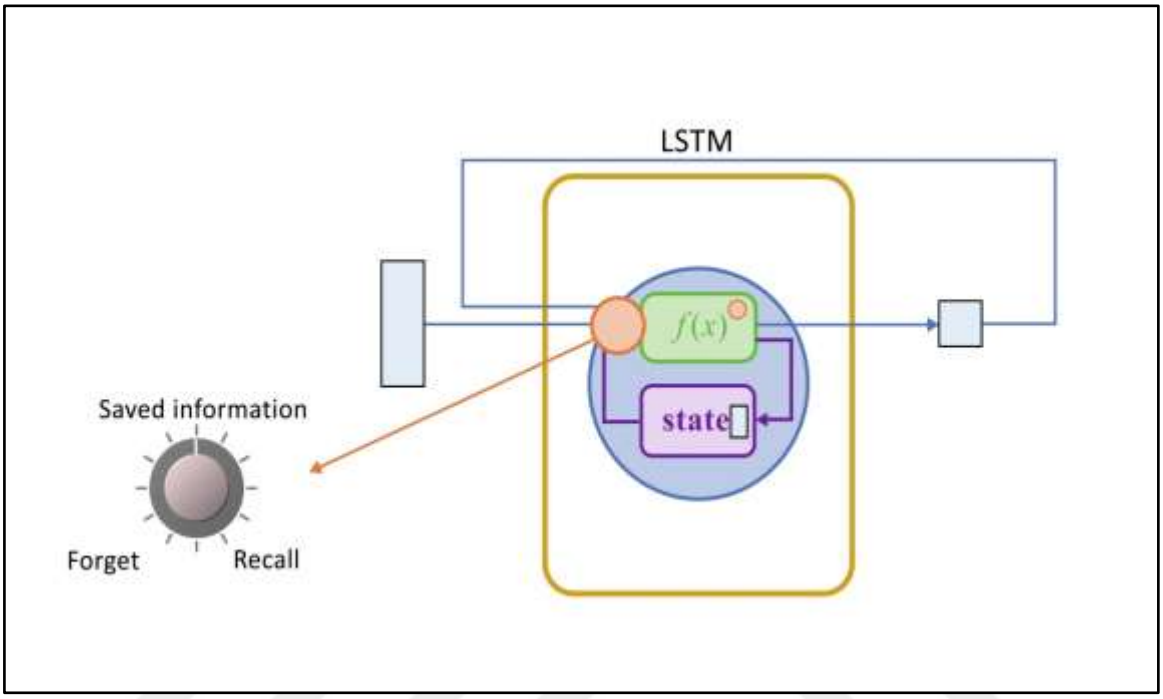


Figure 3.5: The Architecture of Basic LSTM Model**.**

The inability of traditional recurrent neural networks (RNNs) to automatically adapt to novel or unfamiliar circumstances is one of the limitations of these networks. Due to the fact that they have two hidden layers, BRNNs are able to do this duty. These hidden levels are responsible for forwarding information to the same output layer. Through the use of BRNN in conjunction with LSTM, we are able to generate a bidirectional LSTM model that is capable of successfully incorporating input from both sides, hence enabling a more extensive context window.

In order to address the problems of disappearing and exploding gradients that are often seen in conventional RNNs, LSTM and RNNs were developed. This was done with the purpose of addressing these concerns. The cyclic connections that are included in RNNs make them far more successful than Feedforward Neural Networks (FFNN) when it comes to mimicking sequences. The utility of these algorithms has been shown in a wide range of sequence labeling and prediction applications. The phonetic tagging of auditory frames, language

modeling, and handwriting recognition are some of the activities that fall under this category. In contrast to deep learning models, RNNs have focused mostly on phone recognition tasks that require a limited amount of data up to this point. This is because RNNs are more efficient than deep learning models. The objective of this study is to offer novel designs for RNN that make use of LSTM in order to optimize the use of model parameters during the training of acoustic models for speech recognition with a large vocabulary. We examine the results of LSTM, RNN, and DNN models with varying numbers of variables in order to compare and contrast the outcomes of these models with one another.

In this study, we show that LSTM models are capable of achieving state-of-the-art speech recognition performance even when the model sizes are rather modest. Figure 3.6 illustrates the multi-classification process that makes use of LSTM.

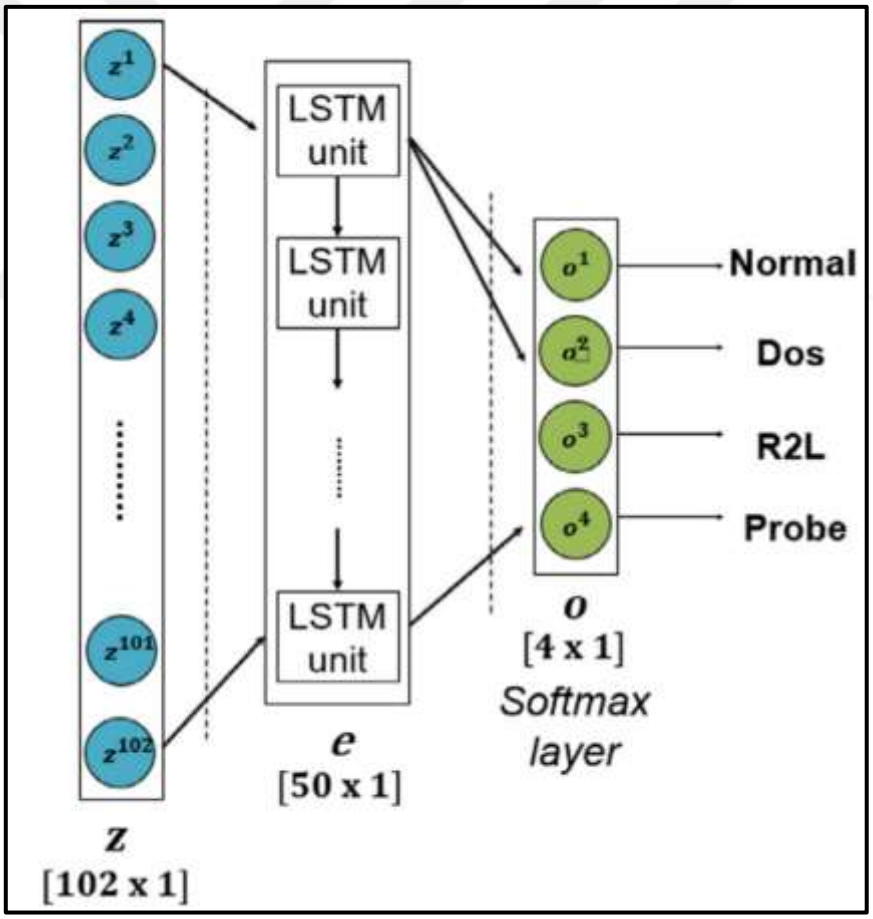


Figure 3.6: A Multi-Classification Task Aimed LSTM Architecture**.**

3.2.3.4 Traditional classifiers

In order to assess the proposed deep KNN and MLP classifier using conventional methods as well, the following shallow classifiers were developed:

i. supervised learning is used as the training approach for FFNN in MLP [47, 80]. The shallow MLP classifier that has been proposed is seen in Figure 3.7. It is important to notice that the MLP design and the AE design have the same structure so that the two may be compared fairly. In point of fact, the MLP classifier consists of a hidden layer with 50 neurons and a SoftMax output layer that is used for classification tasks.

i. ii. SVM classifier: The statistical learning theory is the basis for the SVM method [77, 81]; this theory is used by the SVM classifier. The SVM method is used to locate the best hyperplane that provides the highest level of class separation. Both a linear SVM classifier, abbreviated L-SVM, and a quadratic SVM classifier, abbreviated Q-SVM, are constructed in this study. The mathematical formulation of the SVM is discussed in full in reference [82], which you may get here.

ii. DA classifier: DA is a statistical method that is often used in the field of machine learning. The objective of DA is to reduce the dimensionality of the problem while preserving a high degree of class separability. It does this by deliberately projecting the data samples into a space with less dimensions, with the intention of increasing the class-separability and decreasing the sample dispersion within the same class as a whole. In this particular line of work, a discriminant classifier known as a QDA (linear discriminant classifier) and an LDA (quadratic discriminant classifier) are both put to use. There is presented in reference [83] a comprehensive mathematical definition of DA techniques.

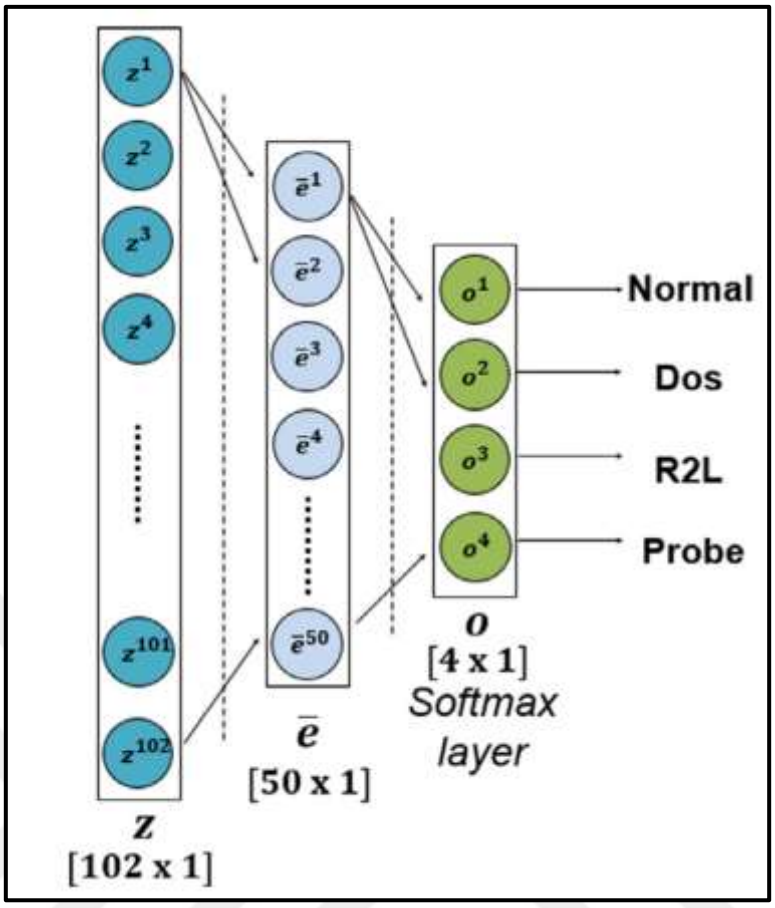


Figure 3.7: A multi-Classification Task Aimed MLP Architecture**.**

3.2.3.5 K-nearest neighbors (KNN)

In this thesis, a novel artificial intelligence framework-based method for classifying network traffic is provided. At the first step, the dataset known as the NSL-KDD Dataset was developed and given an initial processing pass. After that, the dataset is sent to the second step, at this point the dataset is automatically split into the train and test stages based on the model. The following step is the training segment instructing the KNN on various skills. After that, after the training step has been finished, the grid search technique is used to store the model with the optimal settings for its parameters. The testing part makes use of the KNN's best parameters, which have been picked from among those available. During the testing phase, the model is put to use to make inferences about the labels of the model, as is seen in Figure 3.8.

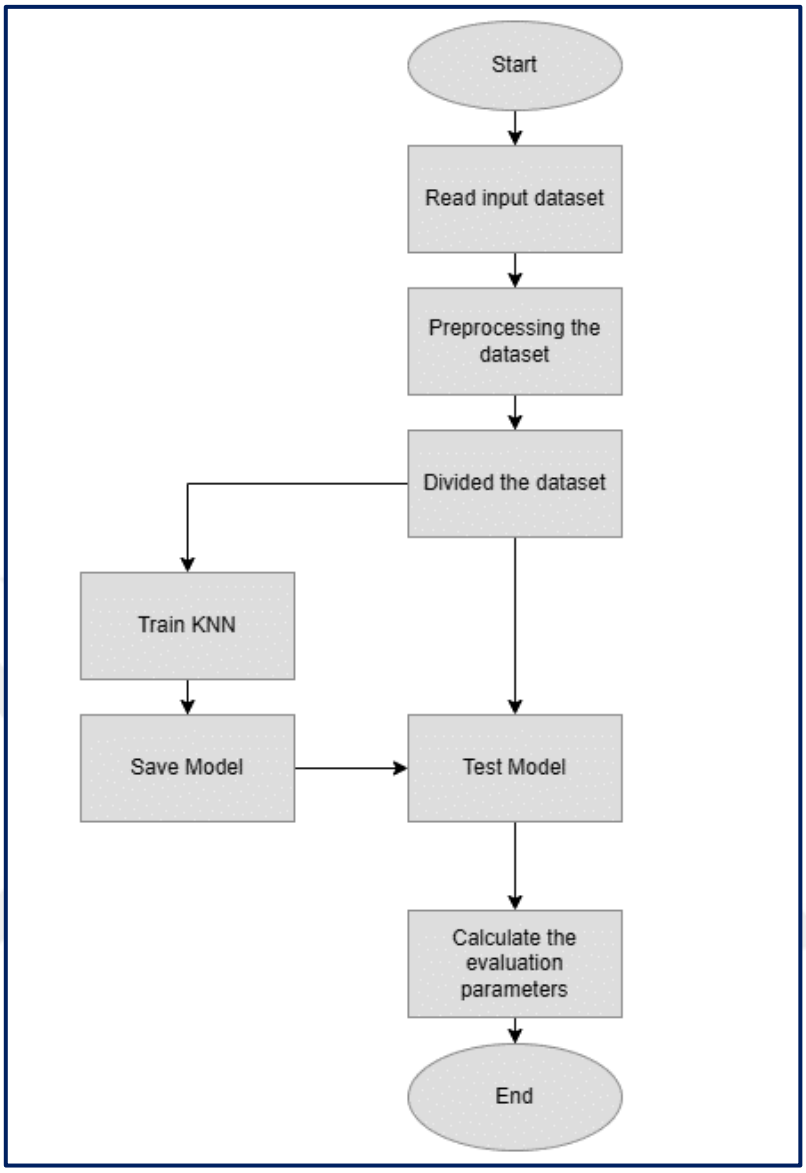


Figure 3.8: Proposed MLKNN Model.

3.3 PERFORMANCE EVALUATION METRICS

Traditional criteria such as precision, recall, F1 score (or F-measure), and accuracy are used to evaluate the performance of the suggested classifiers.

$$Precision = \frac{TP}{TP+ FP} \tag{3.6}$$

$$Precision = \frac{TP}{TP+ FN} \tag{3.7}$$

$$F1\ score = 2 * \frac{Precision*Recall}{Precision+Recall} \tag{3.8}$$

$$Accuracy = \frac{TP+TN}{TP+ FP+TN+FN} \quad (3.9)$$

The term "True Positive" (TP) refers to the number of instances that have been properly identified as abnormal, while the term "True Negative" (TN) refers to the number of cases that have been adequately classified as normal. False Positives (FP) are cases in which normal traffic patterns are wrongly detected as anomalous, while False Negatives (FN) are instances in which aberrant traffic patterns are incorrectly identified as normal. Both types of traffic patterns are referred to as "false positives." Using a binary classification system, the terms "False Positives" (TP) and "False Negatives" (FN) are used to describe findings that are inaccurate. In order to analyze the performance of the suggested architectures, namely KNN, LSTM, MLP, L-SVM, QSVM, and LDA, an investigation was carried out. The inquiry focused on evaluating the performance of these architectures in binary classification situations (Normal, Abnormal) as well as multi-classification scenarios (Normal, Dos, Probe, and R2L). To determine whether or not the classifier is capable of accurately distinguishing between ordinary and unusual attacks, this was carried out. Both accuracy and recall are taken into consideration by the F1 score metric, which is the foundation around which the subsequent arguments are built.

4. EXPERIMENTAL RESULTS

Various study issues emerge from the aforementioned knowledge, as well as from an analysis of how attributes affect the overall result of learning and the adaptive processes involved in training.

What is the impact on the final outcome of training adaptive algorithms when optimizing the number of attributes in selected data sets using different optimization approaches? How well do the proposed KNN and MLP algorithms, based on an ML system using the correspondence analysis mechanism, perform compared to other algorithms when tested on the given data sets? How can the produced MLKNN model be integrated into an existing intrusion detection system, while simultaneously considering other machine learning approaches?

A collection of models, each representing a distinct software application using machine learning technology (including the suggested machine learning system). The phase in which the algorithm is instructed in response to questions is referred to as the "training" stage. This section receives a model's description along with the training examples and applies a training technique to each model.

Following the training process, a collection of classification algorithms is created and then applied to the set of models. The input consists of a set of queries that were not included in the training data.

The correlation phase selects the most efficient adaptive model. As a result of this procedure, the model that performs the best is chosen, and examples that have been correctly classified are included in the training set.

The system proposes two modes of operation: training and categorization. In the former, several models are trained using the recommended data sets, while in the latter, the system's correctness is confirmed by evaluating user queries. The system's ultimate machine learning model is designed to excel at assessing data that was not included in the training set. This task is specifically handled by the correlation block.

4.1 NSL-KDD DATASET

The KDD99 Dataset was used before the NSL-KDD Dataset was created. KDD99 is the de facto first Dataset standard for IDSs, and it was developed for the purpose of evaluating different adaptive algorithms for use in IDSs via a series of benchmarks. KDD99 [74] was implemented, however several problems were discovered as intrusion detection systems evolved; these problems were addressed by adopting the NSL-KDD Dataset. The NSL-KDD Dataset has many benefits over KDD99, including: o the deletion of certain records to reduce the impact of frequency features (redundancy, duplication) on the adaptive mechanism; o a more deliberate approach to the construction of test and training sets; etc. Tables 4.1 and 4.2 detail the NSL-KDD Dataset's make-up and methodology, respectively.

Table 4.1: NSL-KDD Dataset.

DatSet name	Description
KDDTrain+	Training sample with attack markers and level difficulties
KDDTrain+20%	20% subset of KDD train+
KDDTest+	Test set with attack marks and level difficulties
KDDTest-21	Set from KDD test+ that does not include records the level of attacks that are higher than the value of 21

Table 4.2: Distribution of Records Across Sets in The NSL-KDD Dataset.

Dataset	Quantity					
	Entries	Entries Normal	DoS	Probe	U2R	R2L
KDDTrain+20%	25192	13449	9234	2289	11	209
		53.39%	36.65%	9.09%	0.04%	0.83 %
KDDTrain+	125973	67343	45927	11656	52	995
		53.46%	36.46%	9.25%	0.04%	0.79 %
KDDTest+	22544	9711	7458	2421	200	2754
		43.08%	33.08%	10.74%	0.89%	12.22 %

NSL-KDD Dataset objects are connections, which are defined as "a series of (TCP, UDP, ICMP) packets fixed in a specified period of time that contain the data flow from the source

IP address to the destination IP address in accordance with some certain protocol" [72]. Pie chart distribution of normal and abnormal labels (binary classes) is presented in Figure 4.1.

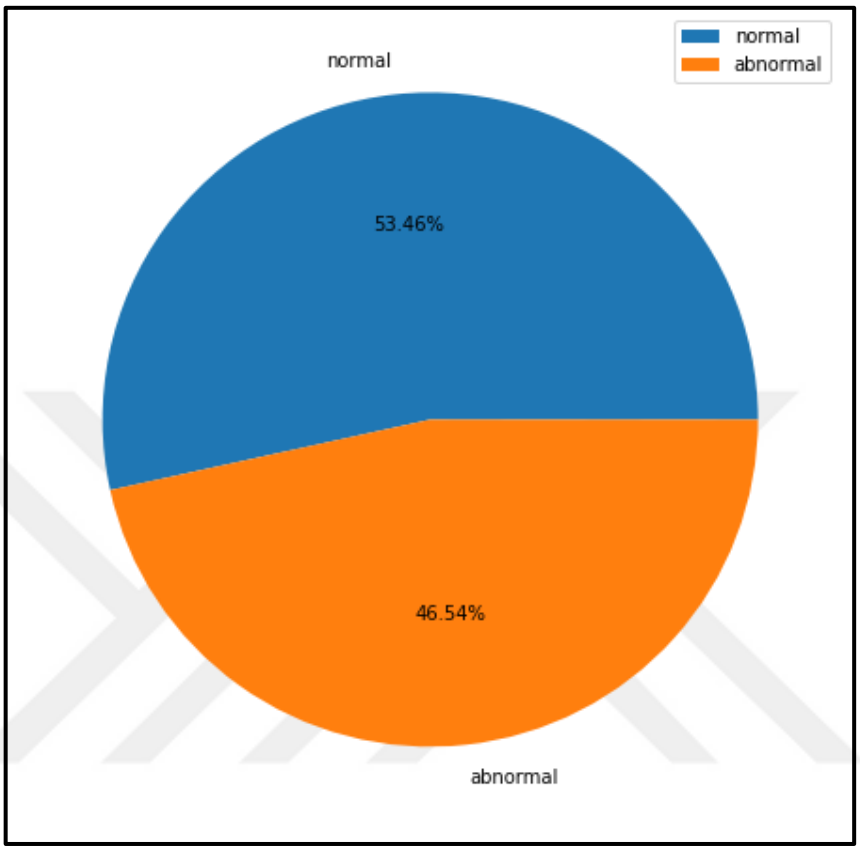


Figure 4.1: Pie Chart Distribution of Normal and Abnormal Labels.

There are four types of dangers represented in the data:

i. Lack of Service (dos). Access control attacks are a class of attacks in which an attacker uses a protocol to limit access to a service to known, authorized users (Back, Land, Neptune, Pod, Smurf, Teardrop, Apache2, Udpstorm, Processtable, Worm).

ii. Inversely, from Faraway to Nearby (r2l). A collection of attacks where the attacker attempts to access the user's local system remotely (Guess Password, Ftp write, Imap, Phf, Multihop, Warezmaster, Warezclient, Spy, Xlock, Xsnoop, Snmpguess, Snmpgetattack, Httptunnel, Sendmail, Named).

iii. Root to User (u2r). Group of exploits (Buffer overflow, Loadmodule, Rootkit, Perl, Sqlattack, Xterm, Ps) that exploits the victim's workstation to elevate their privileges.

iv. Probe. Attacks aimed at compromising a user's infrastructure in order to get insight into its inner workings (Satan, Ipsweep, Nmap, Portsweep, Mscan, Saint).

The NSL-KDD Dataset has 43 characteristics (one for each query), although only 41 are actually being utilized. Attribute 42 indicates the kind of danger, whereas attribute 43 describes the level of difficulty of the assault (from the simplest to the most complex). Thus, the NSL-KDD Dataset has a 41-dimensional vector as its formal query model. Pie chart distribution of multi-class labels is presented in Figure 4.2.

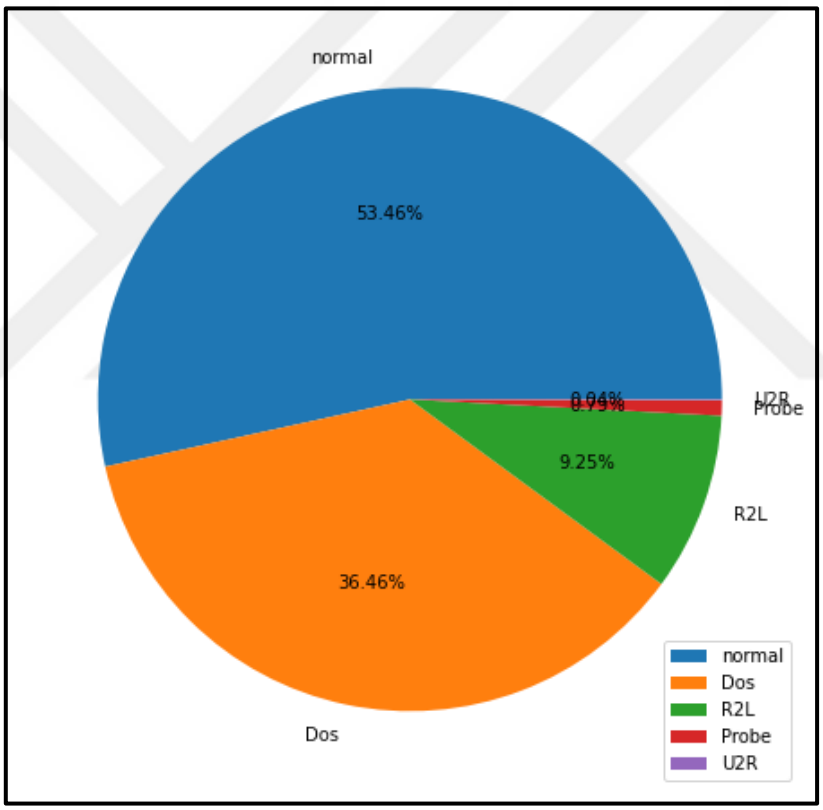


Figure 4.2: Pie Chart Distribution of Multi-Class Labels.

In correspondence analysis, a normalized (divided by the total marginal sum) tuple of data is considered the gold standard. Here and in the following, the term "object tuple" will be used to refer to a generic object tuple.

A sample of representing tuples of objects in absolute scales is shown in Table 4.3, where 3 TCP objects and 3 UDP objects of anomalous requests for 4 attributes each with

corresponding values (from the NLS-KDD test set) were taken as objects, and the nominal characteristics are indicated by numerical values.

Table 4.3: Requests in Absolute Scale.

№	Flags				M_r
	access files 1 – меньше 5 2 – от 5 до 10 3 – более 10	*root shell1 – нет 2 – да*	*srv count 1 – меньше 10 2 – от 10 до 50 3 – свыше 50*	*servic1 – ftp 2 – telnet 3 - smtp 4 - sftp*	
1	1	2	2	1	6
2	3	2	1	3	9
3	1	1	1	4	7
4	2	1	3	3	9
5	3	2	3	2	10
6	3	1	2	1	8
M_s	1 3	7	1 2	14	M = 49

For binary values, the marginal sums across all rows are the same. Applying the chosen methodology for calculating the influence of attributes through the correspondence analysis mechanism for the NSL-KDD Dataset, the following parameters were obtained (Table 4.4), as well as the distribution of combinations and presented in Figures 4.3 and 4.4.

Table 4.4: Correlation of Parameters in The Implementation of The Program and Attributes of the NSL-KDD Dataset.

Parameter	***Attribute Value***	***Example***	***Parameter***	***Attribute Value***	***Example***
p1	*duration*	124	*p2*	*protocol type*	*icmp*
p3	*service*	*ftp data*	*p4*	*flag*	*SF*
p5	*source bytes*	232	*p6*	*destination bytes*	8153
p7	*land*	0	*p8*	*wrong fragment*	1
p9	*urgent*	1	*p10*	*hot*	0
p11	*failed logins*	1	*p12*	*logged in*	1
p13	*compromised*	0	*p14*	*root shell*	0
p15	*su attempted*	0	*p16*	*root*	0
p17	*file creations*	0	*p18*	*shells*	0
p19	*access files*	1	*p20*	*outbound cmds*	0
p21	*is hot login*	1	*p22*	*is guest login*	1
p23	*count*	123	*p24*	*srv count*	32
p25	*serror rate*	0.20	*p26*	*srv serror rate*	0.11
p27	*rerror rate*	1.00	*p28*	*srv rerror rate*	0.00
p29	*same srv rate*	0.08	*p30*	*diff srv rate*	0.15
p31	*srv diff host rate*	0.43	*p32*	*dst host count*	255
p33	*dst host srv count*	26	*p34*	*dst host same srv rate*	0.17
p35	*dst host diff srv rate*	0.03	*p36*	*dst host same src port rate*	0.12
p37	*dst host srv diff host rate*	0.04	*p38*	*dst host serror rate*	0.03
p39	*dst host srvb serror rate*	1.00	*p40*	*dst host rerror rate*	0.01
p41	*dst host srv rerror rate*	0.57			

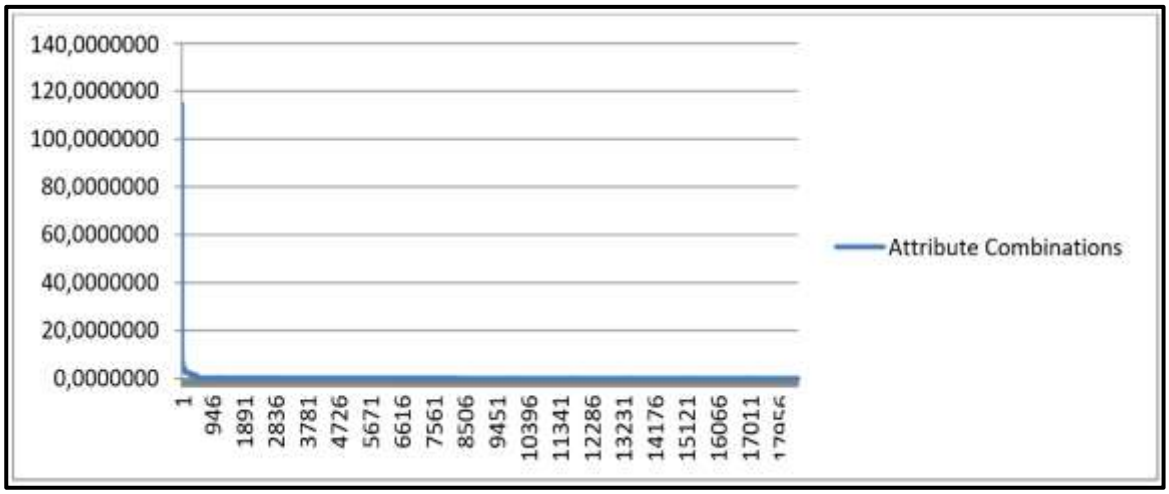


Figure 4.3: Distribution of Values by Final Attributes.

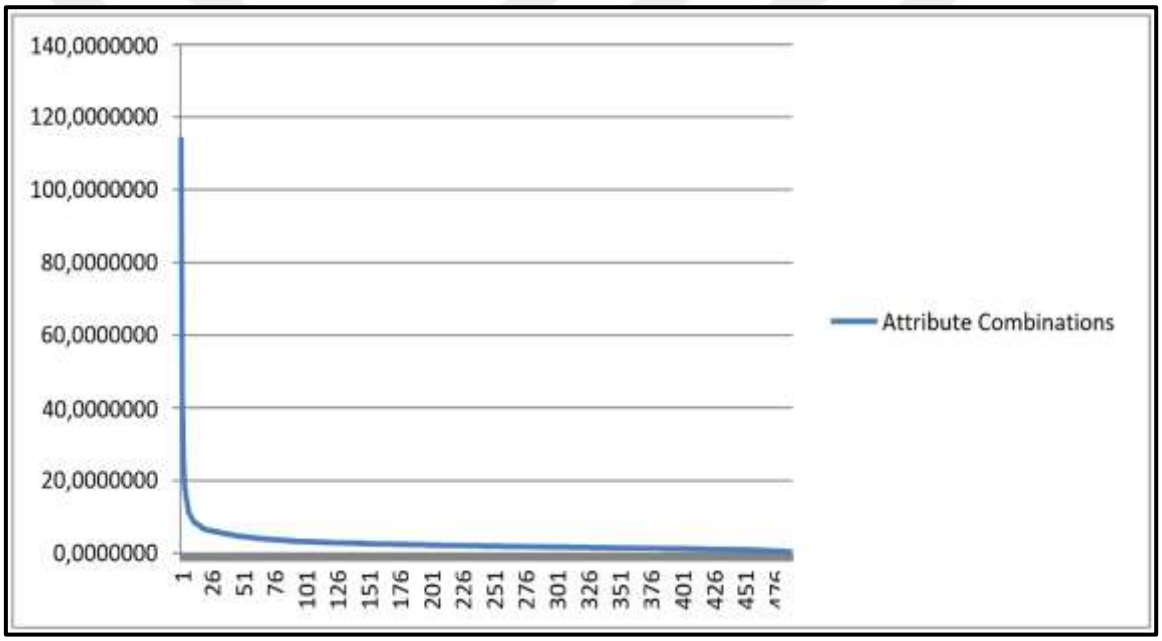


Figure 4.4: Distribution of Values by Final Attributes (Reduced To 500).

A 10-step cross-validation procedure, as described, was utilized to evaluate the performance of the built ML methods. An affinity value, calculated as the average of all element distances to cluster centres, was used to evaluate the AE and LTSM system model. Selecting the best model and evaluating its effectiveness in solving the issue are the goals of performance analysis. The following are therefore examples of metrics by which machine learning models are evaluated:

i. C0 = the proportion of non-anomalous requests that were accurately categorized;

ii. P - accuracy, which characterizes how "sure" the system is, issuing a certain signal;

iii. C1 - percentage of correctly classified anomalous requests;

iv. R_{TP} - the level of true positive signals, showing how many requests, recognized as anomalous, turned out to be really anomalous;

v. R_{FP} - the level of false positives, showing how many non-anomalous requests turned out to be anomalous.

vi. F is the F-measure, which summarizes the classifier's performance in terms of precision and recall.

The experiment's findings are shown as a model of average indicators, which includes the average values of the levels of real positive signals (the configuration with the highest TP level) and the levels of false positives (the configuration with the lowest FP level) presented in Table 4.5 and Table 4.6.

Table 4.5: Displays the Experimental Outcomes of A Logistic Regression-Based Classifier.

DataSet	**Optimization attributes**	***C0*(%)**	***C1*(%)**	***RTP*(%)**	***RFP*(%)**	***P*(%)**	***F*(%)**
NSL-KDD	-	86	78	86	15	88	84
	+	83	75	84	17	87	83

For a classifier based on AE, a two-layer perceptron model with 100 hidden layer neurons was used. This is an average version from [77].

Table 4.6: The Results of The Classifier Based on an Artificial Neural Network**.**

DataSet	**Optimization attributes**	***C0*(%)**	***C1*(%)**	***RTP*(%)**	***RFP*(%)**	***P*(%)**	***F*(%)**
NSL-KDD	-	87	89	87	13	88	90
	+	84	87	86	16	84	87

Table 4.7 shows the performance indicators of the KNN classifier with the calculated affinity threshold. Also, for each experiment, the ratio of the number of β-elements to βm-elements is shown.

On the other hand, after applying attribute reduction methods in queries, the classification quality slightly decreased (by no more than 2% for C0 and 1% for C1), with a slight increase in the level of false positives (by no more than 1%).

Table 4.7: The Results of The Classifier Based on A Developed KNN System.

DataSet	*Aff.*	β^m/β (%)	**Optimization attributes**	***C0*(%)**	***C1*(%)**	***RTP*(%)**	***RFP*(%)**	***P*(%)**	***F*(%)**
NSL-KDD DataSet	0.01	22	-	98	96	99	2	97	96
		22	+	97	94	97	3	96	94
	0.05	19	-	96	94	96	4	95	94
		18	+	94	92	95	6	93	92
	0.1	16	-	95	91	95	5	93	92
		15	+	93	90	94	7	92	91

4.2 BINARY CLASSIFICATION RESULTS

The results of binary classification studies are shown in Table 4.8, where the abnormal class consists of the DoS, Probe, and R2L categories. The KNN classifier successfully identified the normal and abnormal categories with accuracy of 99 % and 92%, respectively.

Table 4.8: Accuracy of Presented Classifiers for Binary and Multiple Classification.

Method	Binary classification	Multi classification
LSTM	86.04%	76.27%
MLP	97.32%	82.53%
L-SVM	97.32%	82.46%
Q-SVM	95.46%	84.68%
LDA	97.81%	86.17%
KNN	99.64%	91.67%

Figure 4.5 present the KNN binary classification (Normal and Abnormal), which the red line indicates the label of real values and blue line indicates the label of prediction.

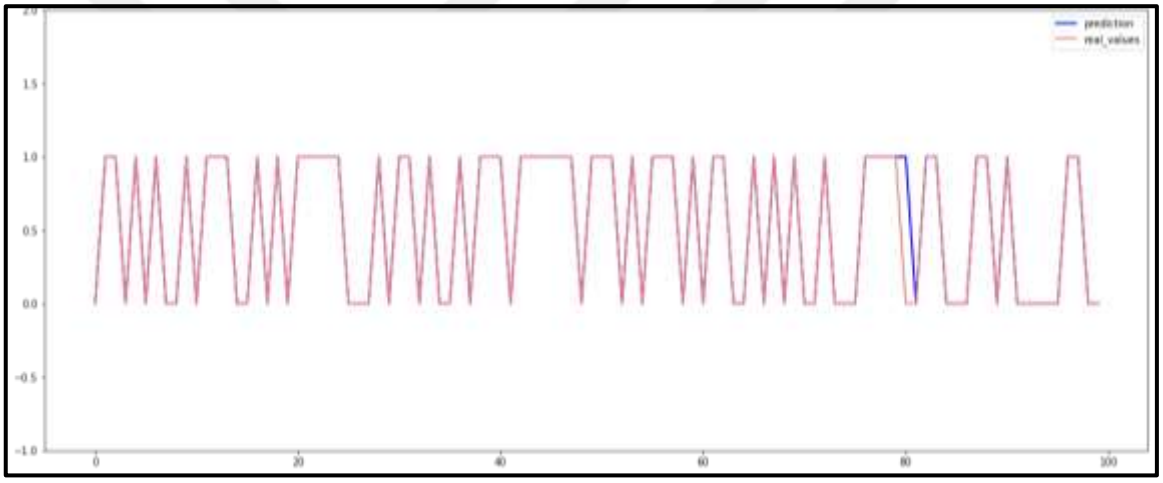


Figure 4.5: MLKNN Binary Classification.

In terms of average F1 score rate for SVM classifiers, Q-SVM surpassed LSVM (reported values of 95.46% and 84.68%, respectively). With F1 values of 97.32% and 82.46%, respectively, the Q-SVM classifier did indeed perform better in identifying both the normal and abnormal classes. In terms of the average F1 score for discriminant analysis, LDA fared better than SVM (reported values of 97.81% and 86.17%, respectively). The average F1 score for the LSTM architecture in terms of the deep classifiers was 86.04% and 76.27%, respectively. The suggested deep KNN classifier, on the other hand, had the highest average F1 score, coming in at 99.64% for binary classification and 91.67% for multiclassification. Additionally, the KNN classifier outperformed the aforementioned techniques in terms of

accuracy (Table 4.10), achieving performance rates up to 96% as opposed to the accuracy rates of the MLP, L-SVM, Q-SVM, LDA, and KNN classifiers.

Table 4.9: Performance (Precision, Recall, F1 score) of Classifiers for Binary Classification.

Attack class	Precision					
	LSVM	**QSVM**	**LDA**	**MLP**	**KNN**	**Support**
Normal	0.96	0.93	0.96	0.98	0.99	14720
Abnormal	0.97	0.99	0.97	0.96	0.99	16774
Avg.	0.97	0.96	0.97	0.97	0.99	31494
Attack class	**F1-Score**					
	LSVM	**QSVM**	**LDA**	**MLP**	**KNN**	**Support**
Normal	0.96	0.95	0.97	0.97	0.99	14720
Abnormal	0.97	0.96	0.96	0.98	0.98	16774
Avg.	0.97	0.96	0.96	0.98	0.99	31494
Attack class	**Recall**					
	LSVM	**QSVM**	**LDA**	**MLP**	**KNN**	**Support**
Normal	0.96	0.92	0.98	0.97	0.99	14720
Abnormal	0.97	0.99	0.96	0.99	0.98	16774
Avg.	0.97	0.95	0.97	0.98	0.99	31494

4.3 MULTI CLASSIFICATION RESULTS

The predicting target attribute on testing dataset of multi-classification studies are presented in Table 4.11 The shallow MLP, L-SVM, Q-SVM, LDA, and deep KNN, KNN based classifiers were compared in a manner akin to the binary classification study. According to the simulation results, the LDA classifier reported an accuracy multiclass classification is 87 %. While the MLP reported an accuracy multiclass classification is 83 %. The Q-SVM classifier performed better than the L-SVM classifier in terms of accuracy, with values of 85 % and 82 %, respectively. For the Dos, Probe, R2L, U2R, and Normal attack classes, the suggested KNN classifier returned Precision values of 0.99, 0.96, 0.89, 1.00, and 0.99, respectively, while returned the F1-Score values 0.99, 0.96, 0.89, 0.12, and 0.99, respectively. All attack types were detected by the KNN classifier with extremely strong discriminating performance, while some classes are not detected (U2R) by using another model as shown in Table 4.11.

Table 4.10: Performance Across Many Classifications (Precision, Recall, F1 Score) of The Classifiers LSVM, QSVM, LDA, MLP, And KNN.

Attack class	Precision					Support
	LSVM	QSVM	LDA	MLP	KNN	
Dos	0.95	0.96	0.94	0.95	0.99	11484
Probe	0.86	0.96	0.88	0.93	0.96	2947
R2L	0.61	0.00	0.37	0.82	0.89	274
U2R	0.00	0.00	0.03	0.96	1.00	15
Normal	0.97	0.91	0.97	0.91	0.99	16774
macro avg	0.68	0.56	0.64	0.92	0.96	Sum = 31494
weighted avg	0.95	0.92	0.94	0.95	0.98	
Attack class	**F1-Score**					**Support**
	LSVM	QSVM	LDA	MLP	KNN	
Dos	0.96	0.95	0.95	0.95	0.99	11484
Probe	0.82	0.74	0.80	0.91	0.96	2947
R2L	0.61	0.00	0.52	0.82	0.89	274
U2R	0.00	0.00	0.06	0.04	0.12	15
Normal	0.98	0.95	0.96	0.93	0.99	16774
Avg.	0.67	0.53	0.66	0.66	0.79	Sum = 31494
weighted avg	0.95	0.92	0.94	0.93	0.98	
Attack class	**Recall**					**Support**
	LSVM	QSVM	LDA	MLP	KNN	
Dos	0.96	0.94	0.96	0.95	0.99	11484
Probe	0.79	0.96	0.73	0.93	0.96	2947
R2L	0.60	0.00	0.89	0.80	0.87	274
U2R	0.00	0.00	0.47	0.02	0.00	15
Normal	0.98	1.00	0.95	0.63	0.70	16774
Avg.	0.67	0.51	0.80	0.71	0.77	Sum = 31494
weighted avg	0.95	0.93	0.93	0.94	0.98	

Indeed, the deep KNN classifier beats all other ML methods, reporting F1 score rates up to 99%, much as the binary classification studies. It is also important to note that the KNN classifier fared better in terms of accuracy than the MLP and traditional methods, obtaining

the greatest accuracy of up to 91%. Contrarily, the accuracy recorded by the LSTM, MLP, L-SVM, Q-SVM, and LDA classifiers was 81.63%, 81.43%, 82.41%, 84.66%, 86.75%, and 78.43%, respectively.

Figure 4.6 present the KNN multi classification (Dos, Probe, R2L, U2R, and Normal), which the red line indicates the label of real values and blue line indicates the label of prediction.

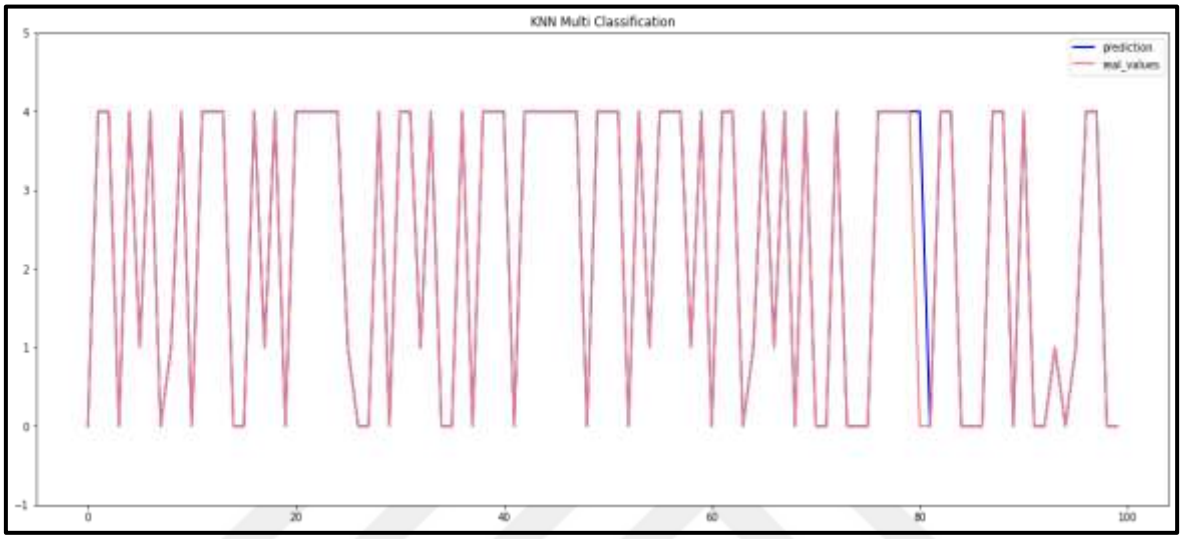


Figure 4.6: KNN Multi-Classification.

4.4 DISCUSSION

In the current work, a novel IDS is presented. This IDS is developed on top of a deep KNN, and it is statistically justified. As a standard for comparison throughout our testing and analysis, we used the NSL-KDD dataset. The proposed IDS was evaluated primarily based on its precision, recall, F1 score, and accuracy. These were some of the primary metrics that were utilized to determine the strengths and usefulness of the IDS. Likewise, the F1 score was still another significant signal to look at. Using statistical analysis, the qualities that were determined to be the most relevant were chosen out and utilized as input for both deep machine learning methods (KNN, MLP) and shallow machine learning techniques (MLP, L-SVM, Q-SVM, and LDA). In addition to this, a multi-classification analysis (Normal vs Dos versus R2L versus U2R versus Probe) as well as a binary analysis (Normal versus Abnormal) were both carried out. Experimental findings indicated that the MLP classifier had the greatest performances for binary discrimination (97% accuracy) and multi-class discrimination (82% accuracy) when compared to MLP, L-SVM, LDA, and KNN classifiers.

This was the case when discussing shallow machine learning architectures. This was the case for discriminating based on both binary and multi-class criteria. When it comes to deep machine learning architectures, the KNN classifier outperformed the MLP classifier in terms of binary discrimination (99% accuracy) and multi-class discrimination (92% accuracy). As a consequence of this, as can be shown via comparisons between the performance of shallow and deep classifiers, the deep KNN architecture performed much better than the other machine learning techniques that are provided in Table 4.12.

Table 4.11: Performance of Intrusion Detection Systems as Stated by Authors for Multi-Class**.**

Method	Accuracy
Proposed KNN	92.00%
Proposed MLP	82.67%
2018 [86]	73.23%
2017 [88]	76.04%
2016 [87]	79.10%
2022 [69]	87.29%
2018 [85]	85.42%

Parallel shallow and deep network topologies that share the same basic structure have been constructed with the intention of making accurate comparisons easier to do. Both the KNN and the MLP classifiers were made up of a total of fifty hidden units in their respective constructions. In addition, the deep KNN that was given was compared to the most current approaches that were discovered in the published study that made use of the NSL-KDD dataset. Although the majority of previous research concentrated on differentiating distinct NSL-KDD attack types, we investigated how well the KNN classifier performed when it was asked to make many classifications at once.

In contrast to these methods, we suggested an intelligent KNN classifier that was driven by statistical analysis and achieved multiclass accuracy of up to 92%. It should be noted, nonetheless, that while the suggested IDS outperformed the aforementioned efforts, an accuracy difference of roughly 5% was seen. The KNN created here, however, had a fairly straightforward design with just more than 40 hidden units on 1 hidden layer. As a result,

both the time and quantity of learning parameters for the suggested IDS were optimized. In fact, the training procedure took just 25.63s while using a high-performance GeForce RTX 2080 Ti GPU coupled with a 64 GB RAM Intel(R) Core (TM) i7-8000K CPU processor.

Designing a secure computer network to defend against intrusions using machine learning techniques involves a multi-faceted approach that combines traditional security measures with advanced machine learning algorithms. Here is a discussion on the key components and strategies for designing a secure network:

Threat Modeling:Begin by identifying potential threats and vulnerabilities in your network. Understand the types of attacks that your system may face, such as malware, phishing, and DDoS attacks.

Network Monitoring: Employ continuous monitoring tools to detect unusual patterns or behaviors on the network. This can involve analyzing network traffic, logs, and system events.

Machine Learning for Anomaly Detection: Integrate machine learning algorithms for anomaly detection. Train models on normal network behavior and use them to identify deviations indicative of a potential intrusion. Techniques such as clustering, classification, and deep learning can be employed.

Collaboration with Threat Intelligence: Stay updated with the latest threat intelligence. Collaborate with external threat intelligence services to enhance your network security measures.

Adaptive Security Measures: Implement adaptive security measures that can evolve based on the changing threat landscape. Machine learning models can be continuously trained to adapt to new attack patterns.

Several benefits may be gained by improving the system's overall security posture via the use of machine learning methods while designing a secure computer network. Some important benefits are as follows:

The use of ML algorithms allows for the early detection of abnormalities by analyzing patterns of network traffic. Security teams can react quickly to possible threats thanks to this early detection, which minimizes the impact of security events.

Machine learning models can adjust to new data and evolving attack patterns, allowing for adaptive security. They can quickly adjust to new security dangers that conventional rule-based systems miss, allowing them to effectively handle these attacks.

Machine learning algorithms may be taught to better differentiate between legitimate network activity and malicious ones, therefore reducing the number of false positives. This lessens the load of investigating false alarms, allowing security staff to concentrate on actual dangers.

Automation of Security Tasks: Machine learning enables automation of routine security tasks, such as the analysis of large datasets and identification of potential threats. This automation allows security teams to focus on more complex and strategic aspects of network security.

Behavioral Analysis: Machine learning facilitates behavioral analysis, allowing systems to understand and adapt to the typical behavior of users and devices. This helps in identifying deviations from normal behavior, which may indicate a security breach.

Immediate Security Updates: By integrating with threat intelligence streams, machine learning models can provide up-to-the-minute information on the most recent security risks. As a result, businesses can guard against emerging threats before they ever happen.

Because of their scalability, machine learning systems are well-suited for network security of varied sizes, since they can manage both simple and complicated datasets. Organizations going through modifications or expansions in their network architecture must need this scalability.

Machine learning models have the ability to learn and become more accurate with time. They get better at what they do when exposed to fresh data because they learn to distinguish between typical and unusual network activity.

Novel or zero-day attacks might be difficult for traditional security measures to detect. Since machine learning is able to spot trends and outliers that signature-based methods may miss, it can uncover risks that were previously undetected.

The amount of time that passes between the discovery of a security breach and its identification is known as "dwell time," and machine learning helps decrease this length by rapidly detecting and reacting to security problems. Intruders are less likely to inflict harm if this is done.

Improved Predictive Analysis: By using machine learning for predictive analysis, companies may better anticipate security vulnerabilities by analyzing historical data and patterns. By taking this preventative stance, safeguards may be put in place far in advance of any assault.

Customized Security Solutions: With the use of machine learning, one may create security solutions that are uniquely suited to a network's requirements and features. The flexibility of the security system improves its overall efficacy.

Although machine learning does improve network security, it still has to be integrated with more conventional security procedures and practices to be really effective. A strong protection against many cyber-attacks may be achieved by combining machine learning with existing security mechanisms.

5. CONCLUSION

Strategically, IDS provide a considerable problem in the realm of network security. A comprehensive investigation has been conducted on this subject for an extended period, and their discoveries continue to be applicable in the current era. Most of the study on this subject has primarily focused on two separate attributes. The first strategy is conventional machine learning, whereas the subsequent approach is deep learning. Furthermore, our work used a unique statistical analysis and a sophisticated intrusion detection method using K-nearest neighbors (KNN). An assessment was conducted on the efficacy of the proposed IDS utilizing data obtained from the NSL-KDD dataset. The prominent features were obtained via a deep learning model that focuses on data analysis. These features were then fed into a KNN framework, which has a single hidden layer with 40 units (KNN 40). This work introduces a novel IDS that is built utilizing a statistically guided deep K-Nearest Neighbors (KNN) method. The NSL-KDD dataset served as a baseline for our testing and analysis. The effectiveness of the suggested IDS was evaluated using metrics such as precision, recall, F1 score, accuracy, and other relevant measures. The F1 score was crucial. Through the use of statistical analysis, we have determined the most noteworthy attributes. These attributes were then utilized as input in both deep machine learning methodologies, such as KNN and Multilayer Perceptron (MLP), as well as shallow machine learning procedures, including Multilayer Perceptron (MLP), L-SVM, Q-SVM. In addition, we performed multi-classification analysis to differentiate between Normal, Dos, R2L, U2R, and Probe, as well as binary analyses to compare Normal and Abnormal. Experimental results have shown that the MLP classifier outperformed the L-SVM, LDA, and KNN classifiers in terms of accuracy for both binary and multi-class discrimination tasks in shallow machine learning architectures. This was relevant to both binary and ternary discriminating systems. The KNN classifier demonstrated superior performance compared to other deep machine learning architectures, with a 99% accuracy in binary discrimination and 92% accuracy in multi-class discrimination.

REFERENCES

[1] S. Kemp, "Digital 2019: Global digital overview," *DataReportal – Global Digital Insights*, 31-Jan-2019. [Online]. Available: https://datareportal.com/reports/digital-2019-global-digital-overview. [Accessed: 01-Apr-2023].

[2] J. L. Leevy and T. M. Khoshgoftaar, "A survey and analysis of intrusion detection models based on CSE-CIC-IDS2018 Big Data," *J. Big Data*, vol. 7, no. 1, 2020.

[3] Q. A. Al-Haijaa and A. Ishtaiwia, "Machine learning based model to identify firewall decisions to improve cyber-defense," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 11, no. 4, pp. 1688–1695, 2021.

[4] W. Meng, E. W. Tischhauser, Q. Wang, Y. Wang, and J. Han, "When intrusion detection meets blockchain technology: A review," *IEEE Access*, vol. 6, pp. 10179–10188, 2018.

[5] S. M. Kasongo and Y. Sun, "A deep learning method with filter based feature engineering for wireless intrusion detection system," *IEEE Access*, vol. 7, pp. 38597–38607, 2019.

[6] H. Liu and B. Lang, "Machine learning and deep learning methods for intrusion detection systems: A survey," *Appl. Sci. (Basel)*, vol. 9, no. 20, p. 4396, 2019.

[7] A. Zotin, Y. Hamad, K. Simonov, and M. Kurako, "Lung boundary detection for chest X-ray images classification based on GLCM and probabilistic neural networks," *Procedia Comput. Sci.*, vol. 159, pp. 1439–1448, 2019.

[8] K. Jiang, W. Wang, A. Wang, and H. Wu, "Network intrusion detection combined hybrid sampling with deep hierarchical network," *IEEE Access*, vol. 8, pp. 32464–32476, 2020.

[9] K. Wu, Z. Chen, and W. Li, "A novel intrusion detection model for a massive network using convolutional neural networks," *IEEE Access*, vol. 6, pp. 50850–50859, 2018.

[10] A. Zotin, K. Simonov, M. Kurako, Y. Hamad, and S. Kirillova, "Edge detection in MRI brain tumor images based on fuzzy C-means clustering," *Procedia Comput. Sci.*, vol. 126, pp. 1261–1270, 2018.

[11] R. A. Kemmerer and G. Vigna, "Intrusion detection: a brief history and overview," *Computer (Long Beach Calif.)*, vol. 35, no. 4, pp. supl27–supl30, 2002.

[12] Y. A. Hamad, K. Simonov, and M. B. Naeem, “Breast cancer detection and classification using artificial neural networks,” in *2018 1st Annual International Conference on Information and Sciences (AiCIS)*, 2018.

[13] “Data Never Sleeps 7.0,” *Domo.com*. [Online]. Available: https://www.domo.com/learn/data-never-sleeps-7. [Accessed: 01-Apr-2023].

[14] T. A. Alamiedy, M. Anbar, A. K. Al-Ani, B. N. Al-Tamimi, and N. Faleh, “Review on feature selection algorithms for anomaly-based intrusion detection system,” in *Advances in Intelligent Systems and Computing*, Cham: Springer International Publishing, 2019, pp. 605–619.

[15] A. Safonova *et al.*, “Individual tree crown delineation for the species classification and assessment of vital status of forest stands from UAV images,” *Drones*, vol. 5, no. 3, p. 77, 2021.

[16] J. Sen and S. Mehtab, *Machine learning applications in misuse and anomaly detection,” in Security and Privacy From a Legal, Ethical, and Technical Perspective*. IntechOpen, 2020.

[17] Y. A. Hamad, K. Simonov, and M. B. Naeem, “Brain’s tumor edge detection on low contrast medical images,” in *2018 1st Annual International Conference on Information and Sciences (AiCIS)*, 2018.

[18] A. Rai, “Optimizing a new intrusion detection system using ensemble methods and deep neural network,” in *2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184)*, 2020.

[19] M. E. Boujnouni and M. Jedra, “New intrusion detection system based on support vector domain description with information gain metric”,” *International Journal of Network Security*, vol. 20, no. 1, pp. 25–34, 2018.

[20] X. Fu, N. Zhou, L. Jiao, H. Li, and J. Zhang, “The robust deep learning–based schemes for intrusion detection in Internet of Things environments,” *Ann. Telecommun.*, 2021.

[21] C. Liang, B. Shanmugam, S. Azam, M. Jonkman, F. D. Boer, and G. Narayansamy, “Intrusion Detection System for Internet of Things based on a Machine Learning approach,” in *2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN)*, 2019.

[22] P. Devan and N. Khare, “An efficient XGBoost–DNN-based classification model for network intrusion detection system,” *Neural Comput. Appl.*, vol. 32, no. 16, pp. 12499–12514, 2020.

[23] P. Verma, S. Anwar, S. Khan, and S. B. Mane, “Network intrusion detection using clustering and gradient boosting,” in *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2018.

[24] A. Husain, A. Salem, C. Jim, and G. Dimitoglou, “Development of an efficient network intrusion detection model using extreme gradient boosting (XGBoost) on the UNSW-NB15 dataset,” in *2019 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, 2019.

[25] Y. A. Hamad, K. V. Simonov, and M. B. Naeem, “Detection of brain tumor in MRI images, using a combination of fuzzy C-means and thresholding,” *Int. J. Adv. Pervasive Ubiquitous Comput.*, vol. 11, no. 1, pp. 45–60, 2019.

[26] A. Safonova, Y. Hamad, A. Alekhina, and D. Kaplun, “Detection of Norway spruce trees (Picea Abies) infested by bark beetle in UAV images using YOLOs architectures,” *IEEE Access*, vol. 10, pp. 10384–10392, 2022.

[27] *Cisco.com*. [Online]. Available: https://www.cisco.com/c/dam/en/us/products/collateral/security/secure-remote-worker-solution/future-of-secure-remote-work-report.pdf. [Accessed: 01-Apr-2023].

[28] “Impact of COVID-19 on cybersecurity,” *Deloitte Switzerland*, 07-Dec-2020. [Online]. Available: https://www2.deloitte.com/ch/en/pages/risk/articles/impact-covid-cybersecurity.html. [Accessed: 01-Apr-2023].

[29] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, “Survey of intrusion detection systems: techniques, datasets and challenges,” *Cybersecurity*, vol. 2, no. 1, 2019.

[30] Y. A. Hamad, K. Simonov, and M. B. Naeem, “Lung boundary detection and classification in chest X-rays images based on neural network,” in *Communications in Computer and Information Science*, Cham: Springer International Publishing, 2020, pp. 3–16.

[31] H. Dong, C. Wu, Z. Wei, and Y. Guo, “Dropping activation outputs with localized first-layer deep network for enhancing user privacy and data security,” *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 3, pp. 662–670, 2018.

[32] Y. Hamad, O. K. J. Mohammed, and K. Simonov, “Evaluating of tissue germination and growth rate of ROI on implants of electron scanning microscopy images,” in *Proceedings of the 9th International Conference on Information Systems and Technologies*, 2019.

[33] C. Yin, Y. Zhu, J. Fei, and X. He, “A deep learning approach for intrusion detection using recurrent neural networks,” *IEEE Access*, vol. 5, pp. 21954–21961, 2017.

[34] A. Zotin, A. Kents, K. Simonov, and Y. Hamad, *Methods of interpretation of CT images with COVID-19 for the formation of feature atlas and assessment of pathological changes in the lungs,” in Intelligent Decision Technologies*. Singapore; Singapore: Springer, 2021.

[35] J. J. Stubbs, G. C. Birch, B. L. Woo, and C. G. Kouhestani, “Physical security assessment with convolutional neural network transfer learning,” in *2017 International Carnahan Conference on Security Technology (ICCST)*, 2017.

[36] Z. Hu, L. Wang, L. Qi, Y. Li, and W. Yang, “A novel wireless network intrusion detection method based on adaptive synthetic sampling and an improved convolutional neural network,” *IEEE Access*, vol. 8, pp. 195741–195751, 2020.

[37] M. Latah and L. Toker, “An efficient flow-based multi-level hybrid intrusion detection system for software-defined networks,” *CCF Trans. Netw.*, vol. 3, no. 3–4, pp. 261–271, 2020.

[38] M. Al-Qatf, Y. Lasheng, M. Al-Habib, and K. Al-Sabahi, “Deep learning approach combining sparse autoencoder with SVM for network intrusion detection,” *IEEE Access*, vol. 6, pp. 52843–52856, 2018.

[39] Y. Wu, W. W. Lee, Z. Xu, and M. Ni, “Large-scale and robust intrusion detection model combining improved deep belief network with feature-weighted SVM,” *IEEE Access*, vol. 8, pp. 98600–98611, 2020.

[40] A. Zotin, Y. Hamad, K. Simonov, M. Kurako, and A. Kents, “Processing of CT lung images as a part of radiomics,” in *Intelligent Decision Technologies*, Singapore: Springer Singapore, 2020, pp. 243–252.

[41] T. Saba, A. Rehman, T. Sadad, H. Kolivand, and S. A. Bahaj, “Anomaly-based intrusion detection system for IoT networks through deep learning model,” *Comput. Electr. Eng.*, vol. 99, no. 107810, p. 107810, 2022.

[42] I. F. Kilincer, F. Ertam, and A. Sengur, “Machine learning methods for cyber security intrusion detection: Datasets and comparative study,” *Comput. Netw.*, vol. 188, no. 107840, p. 107840, 2021.

[43] I. H. Sarker, Y. B. Abushark, F. Alsolami, and A. I. Khan, “IntruDTree: A machine learning based cyber security intrusion detection model,” *Symmetry (Basel)*, vol. 12, no. 5, p. 754, 2020.

[44] K. S. Kiran, R. K. Devisetty, N. P. Kalyan, K. Mukundini, and R. Karthi, “Building a intrusion detection system for IoT environment using machine learning techniques,” *Procedia Computer Science*, vol. 171, pp. 2372–2379, 2020.

[45] Y. A. Hamad, M. E. Seno, M. Al-Kubaisi, and A. N. Safonova, “Segmentation and measurement of lung pathological changes for COVID-19 diagnosis based on computed tomography,” *Period. Eng. Nat. Sci. (PEN)*, vol. 9, no. 3, p. 29, 2021.

[46] A. S. Kents, Y. A. Hamad, K. V. Simonov, and A. G. Zotin, “Methods and models for texture analysis of lung pathological changes based on computed tomography for covid-19 diagnosis,” *ISPRS - Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. XLIV-2/W1-2021, pp. 99–105, 2021.

[47] S. K. Biswas, “Intrusion detection using machine learning: A comparison study,” *International Journal of pure and applied mathematics*, vol. 118, no. 19, pp. 101–104, 2018.

[48] E. M. Kabaev, Y. A. Hamad, K. V. Simonov, and A. G. Zotin, “Methods of interpretation of data from isokinetic tests and MRI studies during rehabilitation of patients after reconstructive shoulder joint surgery,” *ISPRS - Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. XLIV-2/W1-2021, pp. 91–97, 2021.

[49] *Visualization and Analysis of the Shoulder Joint Biomechanics in Postoperative Rehabilitation*. .

[50] Y. Xin *et al.*, “Machine learning and deep learning methods for cybersecurity,” *IEEE Access*, vol. 6, pp. 35365–35381, 2018.

[51] Y. A. Hamad, J. Kadum, A. A. Rashid, A. H. Mohsen, and A. Safonova, “A deep learning model for segmentation of covid-19 infections using CT scans,” in *AIP Conference Proceedings*, vol. 2398, AIP Publishing LLC, 2022.

[52] A. Zotin, A. Kents, K. Simonov, and Y. Hamad, “Methods of interpretation of CT images with COVID-19 for the formation of feature atlas and assessment of pathological changes in the lungs,” in *Intelligent Decision Technologies*, Singapore: Springer Singapore, 2021, pp. 173–183.

[53] Y. A. Hamad, M. N. Qasim, A. A. Rashid, and M. E. Seno, “Algorithms of experimental medical data analysis,” in *2020 International Conference on Computer Science and Software Engineering (CSASE)*, 2020.

[54] M. E. Seno, A. A. Abed, Y. A. Hamad, U. M. Bhatt, B. Ravindra Babu., and S. Bansal, “Cloud based smart kitchen automation and monitoring,” in *2022 5th International Conference on Contemporary Computing and Informatics (IC3I)*, 2022.

[55] H. Alqahtani, I. H. Sarker, A. Kalim, S. M. Minhaz Hossain, S. Ikhlaq, and S. Hossain, “Cyber intrusion detection using machine learning classification techniques,” in *Communications in Computer and Information Science*, Singapore: Springer Singapore, 2020, pp. 121–131.

[56] S. Gurung, Sikkim Manipal Institute of Technology, Sikkim Manipal University, Majitar, Sikkim, India, M. Kanti Ghose, and A. Subedi, “Deep learning approach on network intrusion detection system using NSL-KDD dataset,” *Int. J. Comput. Netw. Inf. Secur.*, vol. 11, no. 3, pp. 8–14, 2019.

[57] T. Su, H. Sun, J. Zhu, S. Wang, and Y. Li, “BAT: Deep learning methods on network intrusion detection using NSL-KDD dataset,” *IEEE Access*, vol. 8, pp. 29575–29585, 2020.

[58] I. Abrar, Z. Ayub, F. Masoodi, and A. M. Bamhdi, “A machine learning approach for intrusion detection system on NSL-KDD dataset,” in *2020 International Conference on Smart Electronics and Communication (ICOSEC)*, 2020.

[59] M. E. Seno, A. A. Abed, Y. A. Hamad, S. Pundir, A. P. Srivastava, and S. Ali, “Forwarding factor routing protocol for opportunistic internet of things networks,” in *2022 5th International Conference on Contemporary Computing and Informatics (IC3I)*, 2022.

[60] J. Palimote, L. Atu, and E. Osuigbo, *A Model to Detect Network Intrusion Using Machine Learning*. 2021.

[61] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 1, 2021.

[62] S. Gamage and J. Samarabandu, "Deep learning methods in network intrusion detection: A survey and an objective comparison," *J. Netw. Comput. Appl.*, vol. 169, no. 102767, p. 102767, 2020.

[63] K. A. Taher, B. Mohammed Yasin Jisan, and M. M. Rahman, "Network intrusion detection using supervised machine learning technique with feature selection," in *2019 International Conference on Robotics,Electrical and Signal Processing Techniques (ICREST)*, 2019.

[64] M. Esmaeili, S. H. Goki, B. H. K. Masjidi, M. Sameh, H. Gharagozlou, and A. S. Mohammed, "ML-DDoSnet: IoT intrusion detection based on denial-of-service attacks using machine learning methods and NSL-KDD," *Wirel. Commun. Mob. Comput.*, vol. 2022, pp. 1–16, 2022.

[65] S. Gorgin, M. H. Gholamrezaei, D. Javaheri, and J.-A. Lee, "An efficient FPGA implementation of k-nearest neighbors via online arithmetic," in *2022 IEEE 30th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, 2022.

[66] W. Elmasry, A. Akbulut, and A. H. Zaim, "Evolving deep learning architectures for network intrusion detection using a double PSO metaheuristic," *Comput. Netw.*, vol. 168, no. 107042, p. 107042, 2020.

[67] M. Injadat, A. Moubayed, A. B. Nassif, and A. Shami, "Multi-stage optimized machine learning framework for network intrusion detection," *IEEE Trans. Netw. Serv. Manag.*, vol. 18, no. 2, pp. 1803–1816, 2021.

[68] J. Liu, B. Kantarci, and C. Adams, "Machine learning-driven intrusion detection for Contiki-NG-based IoT networks exposed to NSL-KDD dataset," in *Proceedings of the 2nd ACM Workshop on Wireless Security and Machine Learning*, 2020.

[69] A. F. Almutairi and A. Abdulghani Alshargabi, "Using deep learning technique to protect internet network from intrusion in IoT environment," in *2022 2nd*

International Conference on Emerging Smart Technologies and Applications (eSmarTA), 2022.

[70] G. Meena and R. R. Choudhary, "A review paper on IDS classification using KDD 99 and NSL KDD dataset in WEKA," in *2017 International Conference on Computer, Communications and Electronics (Comptelix)*, 2017.

[71] A. S. Ahanger, S. M. Khan, and F. Masoodi, "An effective intrusion detection system using supervised machine learning techniques," in *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, 2021.

[72] R. D. Ravipati and M. Abualkibash, "Intrusion detection system classification using different machine learning algorithms on KDD-99 and NSL-KDD datasets - A review paper," *SSRN Electron. J.*, 2019.

[73] Y. Ding and Y. Zhai, "Intrusion Detection System for NSL-KDD Dataset Using Convolutional Neural Networks," in *Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence*, 2018.

[74] "Datasets," *Unb.ca*. [Online]. Available: http://www.unb.ca/research/iscx/dataset/iscx-NSL-KDDdataset.html. [Accessed: 01-Apr-2023].

[75] R. Magán-Carrión, D. Urda, I. Díaz-Cano, and B. Dorronsoro, "Towards a reliable comparison and evaluation of Network Intrusion Detection Systems based on machine learning approaches," *Appl. Sci. (Basel)*, vol. 10, no. 5, p. 1775, 2020.

[76] E. Alpaydin, *Introduction to Machine Learning*. London, England: MIT Press, 2014.

[77] Y. Fu, Y. Du, Z. Cao, Q. Li, and W. Xiang, "A deep learning model for network intrusion detection with imbalanced data," *Electronics (Basel)*, vol. 11, no. 6, p. 898, 2022.

[78] J. Shore and R. Johnson, "Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy," *IEEE Trans. Inf. Theory*, vol. 26, no. 1, pp. 26–37, 1980.

[79] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[80] H. A. Ahmed, A. Hameed, and N. Z. Bawany, “Network intrusion detection using oversampling technique and machine learning algorithms,” *PeerJ Comput. Sci.*, vol. 8, no. e820, p. e820, 2022.

[81] G. Apruzzese, L. Pajola, and M. Conti, “The cross-evaluation of Machine Learning-based Network Intrusion Detection Systems,” *arXiv [cs.CR]*, 2022.

[82] X.-H. Nguyen, X.-D. Nguyen, H.-H. Huynh, and K.-H. Le, “Realguard: A lightweight network intrusion detection system for IoT gateways,” *Sensors (Basel)*, vol. 22, no. 2, p. 432, 2022.

[83] G. Apruzzese, M. Andreolini, L. Ferretti, M. Marchetti, and M. Colajanni, “Modeling realistic adversarial attacks against Network Intrusion Detection Systems,” *Digit. Threat.*, vol. 3, no. 3, pp. 1–19, 2022.

[84] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, “A deep learning approach to network intrusion detection,” *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 2, no. 1, pp. 41–50, 2018.

[85] Y. Imamverdiyev and F. Abdullayeva, “Deep learning method for denial of service attack detection based on restricted Boltzmann machine,” *Big Data*, vol. 6, no. 2, pp. 159–169, 2018.

[86] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, “A deep learning approach for network intrusion detection system,” in *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*, 2016.

[87] A. Basati and M. M. Faghih, “PDAE: Efficient network intrusion detection in IoT using parallel deep auto-encoders,” *Inf. Sci. (Ny)*, vol. 598, pp. 57–74, 2022.

[88] F. Khan, A. Aslam, A. Gumaei, and A. Derhab, “A novel two-stage deep learning model for efficient network intrusion detection,” IEEE Access, vol. 7, pp. 30373–30385, 2019.