REPUBLIC OF TURKEY

ALTINBAŞ UNIVERSITY

Institute of Graduate Studies

Electrical and Computer Engineering

DATA MINING AND MACHINE LEARNING FOR CYBER SECURITY INTRUSION DETECTION

Ali Mohammed Hasan AL-AMEEN

Master of Science

Supervisor

Asst. Prof. Dr. Sefer KURNAZ

Istanbul, 2022

DATA MINING AND MACHINE LEARNING FOR CYBER SECURITY INTRUSION DETECTION

Ali Mohammed Hasan AL-AMEEN

Electrical and Computer Engineering

Master of Science

ALTINBAŞ UNIVERSITY

2022

The thesis titled DATA MINING AND MACHINE LEARNING FOR CYBER SECURITY INTRUSION DETECTION prepared by ALI MOHAMMED HASAN AL-AMEEN and submitted on 2021/12/30 has been **accepted unanimously of votes** for the degree of Master of Science in Electrical and Computer Engineering

_____

Asst. Prof. Dr. Sefer KURNAZ

the Supervisor

Thesis Defense Committee Members:

Prof. Dr. A. Ahmet Mesut RAZBONYALI	School of Engineering and Natural sciences, Maltepe University	_____
Asst. Prof. Dr. Sefer KURNAZ	School of Engineering and Architecture, Altinbas University	_____
Asst. Prof. Dr. Muhammad ILYAS	School of Engineering and Architecture, Altinbas University	_____

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

Accepted by Altınbaş University Institute of Graduate Studies on the following date:

____/____/____

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.

Ali MOHAMMED HASAN AL-AMEEN

Signature

ACKNOWLEDGEMENTS

I might want to thank my administrators: Asst. Prof. Dr. Sefer KURNAZ Please let me express my profound feeling of appreciation and gratefulness to both of you for the information: direction and unrestricted help you have given me. I want you to enjoy all that life has to offer and further achievement and accomplishments throughout your life.

ABSTRACT

DATA MINING AND MACHINE LEARNING FOR CYBER SECURITY INTRUSION DETECTION

Ali Mohammed Hasan Al-Ameen

M.Sc., Electrical and Computer Engineering, Altınbaş University,

Supervisor: Dr. Sefer KURNAZ

Date: 2022

Pages: 68

In spite of the quick development in information technology, securing computer and network resources still remains as a major challenge and concern for various organizations and researchers, particularly after the growth of networks and progress of technology. Implementation and designing intrusion detection systems are become very significant in network security. Intrusion detection is the fundamental tool of network security in struggling against malicious cyber attacks and unlawful network access. Since the continuously growing of attacks, it has been a technological challenge for an intrusion detection system (IDS) to successfully recognize known attacks and unknown attacks with insufficient training data. For that reason in the present study, an innovative contributions are implemented based on data mining and machine learning techniques for accurately and professionally detecting both known attacks and unknown attacks with inaccurate or insufficient training information. Faced with the increase of more advanced attacks targeting information systems, a defense system has become vital. An intrusion detection system provides a first line of defense. A intrusion detection system monitor events within an information system or in one of the organs of the information system The objective of this research project is to design a lightweight intrusion detection system using artificial intelligence techniques, in particular deep learning techniques. The neural network will be trained and tested with the NSL KDD dataset.

Keywords: Data Mining, Machine Learning, Cyber-Attack, NSL-KDD,

TABLE OF CONTENTS

Pages

LIST OF TABLES

Pages

LIST OF FIGURES

Pages

LIST OF ABBREVIATIONS

ANN	:	Artificial Neural Networks
DARPA	:	Defence Advanced Research Projects Agency
DoS	:	Denial-of-Service attack
IDS	:	Intrusion Detection System
KDD	:	Knowledge Discovery in Databases
k-NN	:	k-Nearest Neighbour
NSL	:	Network Security Laboratory

1. INTRODUCTION

There is a colossal blast of information all over, from keeping our statement archives, dominate worksheets to the information claimed and worked by enterprises, banking/monetary areas, and numerous different spots. It is vital for secure this information from pernicious exercises. With the increasing pace of digital assault, there is a gigantic interest for effective IDS in the organization. With the invasion complicated and difficult to locate, enhanced technology is used to preserve network confidence and security.

Many networks are equipped with intrusion detection systems (IDS) to detect such attacks (e.g., in banking and educational organizations). IDS classifies these systems as host, network, and hybrid. These systems are classified as IDS. HIDS tracks the device and scans for malice, while NIDS checks the network's traffic payload for unusual incidents. Premise of strategies for identification, IDS are portrayed into two sorts, Mark-based IDS and oddity-based IDS [1] are two examples. Due to technological improvements, digital assaults have reached unprecedented heights.

On the off chance that the assault isn't recognized right off the bat, the organization and its clients are truly influenced. To defeat this calamity, shared interruption discovery frameworks (CIDS) have been planned [1]. CIDS is intended to upgrade singular IDS discovery capacities. In this case, every IDS contacts other IDS in order to work together to exchange information and manage their trust. [2].

A variety of methods are used to deter attacks and to ensure that users have a stable network. This investigation explores some of the methods used to detect intruding, such as the use of blockchains, machine learning, and deep learning technologies [3]. Blockchain technology, the secret invention behind cryptocurrencies such as bitcoin, was first introduced in 2008. In contrast to actual cash, an undisputed issue called twofold going through goes with cutting edge cash and computerized types of cash. All exchanges in a blockchain [4] are put away in blocks. This article offers extra data on blockchain innovation. Machine preparing is quite possibly the most widely recognized

interruption recognition techniques. Anomalies in the organization can be identified by running different machine preparing calculations, including K-Nearest Neighbor and SVM., [5].

One of different methodologies used to improve the abilities of AI calculations are Deep Learning procedures [6]. Deep learning methods have been proven to efficiently tackle the challenges raised by IDS [7]. Other intrusion detection techniques include statistical approaches, data mining methods [8], genetic algorithms, etc. [9].

1.1 BACKGROUND AND PROBLEM MOTIVATION

Human lives and property have for centuries been protected by the laws, manner and customs of their country with locks, walls, signatures and seals. In the present automated and electronic devices run by information system packing machines, freeze-free bank properties, remove data from the satellite and, for example, shut down militaries. Technologist's 20th century challenge was one of their most important and most difficult, is development of safety technologies to detect cyber attacks and network interference, protecting the privacy and property. [11]

The IDSs are focused on intrusion patterns that are manually generated and cyber attacks are not noted when signature-based IDS are not modified [12]. Our view of data generation is increasingly divided. The proportion of the data that people understand declines alarmingly, as the amount of information increases. [13]

A widely used intruder detection system performance testing is the 1998 Intrusion Detection Agency, (DARPA) and modified versions of this 1999 dataset (KDDCup) series and the NSL-KDD dataset. The Department for Advanced Research in Defense (KDDCup) (IDS). These datasets are made up of the simulated network traffic near to the US cyber system [14]. Many researchers have developed safety measures and explored alternate methods to detect cyber threats using those data sets; however, none of the top 10 data mining algorithms has yet investigated the performance [15] selected by data mining experts.

Cyber-attacks must be recognized to enhance network security. Additional research is required to develop new protection technologies in order to develop information about network intrusion detection in data mining.

1.2 OVERALL AIM

The ultimate objective is to increase information about intrusion detection data mining to detect cyber attacks and take sound security configuration decisions.

1.3 RESEARCH CONTRIBUTION

In this research, we have led an efficient writing study on the current interruption identification procedures. This audit is done from a period beginning from 1998 to 2018. We have given an essential prologue to the interruption identification frameworks, AI, profound learning, and blockchain innovation. Afterwards, a detailed analysis of the three tactics is presented. They're also noted for their limitations/challenges in terms of the interruption identification framework. We have investigated the current situation with the square chain innovation in digital protection in distinguishing assaults. An order of AI and profound learning innovations utilized for distinguishing pernicious client assaults in the organization is given. The methodologies utilized by different scientists to recognize any malignant exercises in the NSL KDD information utilizing instruments are featured. We have likewise ordered the distributions of the papers for as far back as 19 years by its extended time of delivery and its information base source.

Despite the fact that blockchain innovation demonstrates a promising commitment to interruption discovery, it neither gives the approaches to contrast its exhibition and AI calculations nor the dataset to assemble a powerful calculation for interruption location.

In the final round of exploration, we conduct an analysis to identify the assaults in the dataset termed NSL-KDD. Three classifiers are used to build an AI interruption location model. Exhibition of the model is estimated and assessed.

1.4 SCOPE

This thesis compares categorization algorithms among the top 10 data mining algorithms [5]. A variety of forests and plants, K-Nearest a Neighbor (K-NN), and Naive Bays are featured. There are two types of classification trees. For machine learning, no other algorithms are considered. The efficacy, error rate and average cost of the selected machine learning algorithms are assessed. In this study a review will not be included if the chosen machine learning algorithms in other environments behave differently e.g., on other military network data sets.

1.5 CONCRETE AND VERIFIABLE GOALS

Four objectives were set to examine how the use of data mining to detect network attacker and cyber-attacks:

• Assess the classification of the machine is the most accurate algorithm for intruder's detection.

• Find out if the data collection contains the number of groups.

• Examine the most important features for analyzing an anomaly.

1.6 THESIS STRUCTURE

The theory is organized as follows. The set of experiences to Chapter 2 is about Intrusion Detection Systems, various assaults and interruption identification techniques like robots, profound learning and blockchain innovation. Section 3 gives the writing overview of interruption location procedures for as long as 19 years, and Section 4 portrays the proposed procedure, including dataset depiction, calculation depiction, and succession of steps taken to lead the examination analyze. Part 6 shows the outcomes got from the investigation. At last, Chapter 5 gives the end, we finish up the ebb and flow explore and talk about future exploration headings.

2. BACKGROUND

This chapter addresses cyber attacks, data mining, and selected algorithms for the machine learning, Weka and DARPA 1998 datasets and the previous dissertation on the assessment of intrusion detection for this study.

2.1 CYBER-ATTACKS

Website vandalism and large-scale damage to military or civil infrastructure are enormous activities which can be conducted through information networks. Therefore, cyber-attack concepts differ. Efforts to modify, interrupt or destruct computer systems, networks or information or software on a cyber attack may be curbed. [6]

Intrusion detection professionals argue that most recent cyber attacks differ from existing attacks and that the recorded attacks are ideal for the detection of new attacks. [7]

Tavallaee, Bagheri, Lu and Ghorbani [8] present the following four types of cyber attacks:

- Denial-of-Service Attack (DoS) is attacks when an attacker is too busy to deal with any legitimate requests or does not allow legitimate users to access a device.
- User to Root Attack (U2R) are assaults when the intruder gets access, perhaps by social engineering, hacking or sniffing of passwords, to a user account on the device. The attacker then uses those vulnerabilities to get root system access.
- Local attack (R2L) remote attacks are attacks in which remote attackers attempt to access a local user account.
- Attacks are attacks in which an attacker seeks to obtain information on a computer network to prevent its control over security.

2.2 INTRUSION DETECTION SYSTEM

These frameworks incorporate host, network-based and half and half IDS frameworks These frameworks are (HIDS). HIDS tracks the gadget and explores vindictive exercises and NIDS investigates the payload traffic in the Internet for uncommon events. The history of IDS and CIDS and their challenges can be found in this section. Intrusion detection is a way of tracking activities in a network or device to detect an unexpected or harmful activity that violates protection or standard policies. [11]. The intrusion detectors in all networks are mounted in figure 2.1 and Figure 2.2 and provide a safety layer and track any malicious behavior.

Attackers learn how to prevent countermeasures, which is predictable. The attackers have the advantage of carefully preparing and choosing the right time for attacks and network vulnerabilities. To protect themselves from cyber-attacks, nations and organizations need information. Intrusion detection must be much improved for cyber attacks to be detected and for security configuration to be rendered best. [9]

2.2.1 HIDS and NIDS

IDS are characterized by two methods of detection: IDS-based signatures and IDS-based anomalies [1]. The IDS is primarily listed under HIDS, NIDS, and Hybrid IDS, which are HIDS and NIDS integration for further security [55].

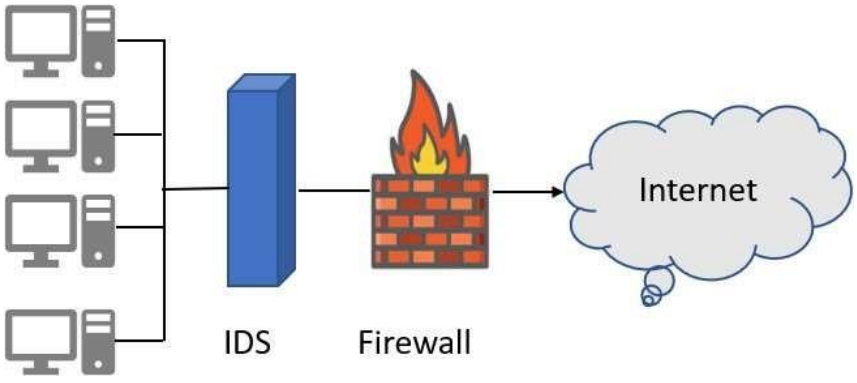

Figure 2.1: Intrusion detector in a network [55]

The HIDS and NIDS organization in an organization is displayed in Figure 2.1. An IDS is additionally sorted by location strategies as a mark, abnormality and determination [1]. A stored signature is used for the identification of the attack by the signature detection mechanism. The observed network method. Any signs of malicious activity will be detected and the anomaly-based detection system will warn these incidents.

2.2.2 Collaborative Intrusion Detection System (CIDS)

With CIDS, single IDS detection becomes more effective. This is especially true for sophisticated attacks, such as denial of service (DOS). If it isn't an assault.

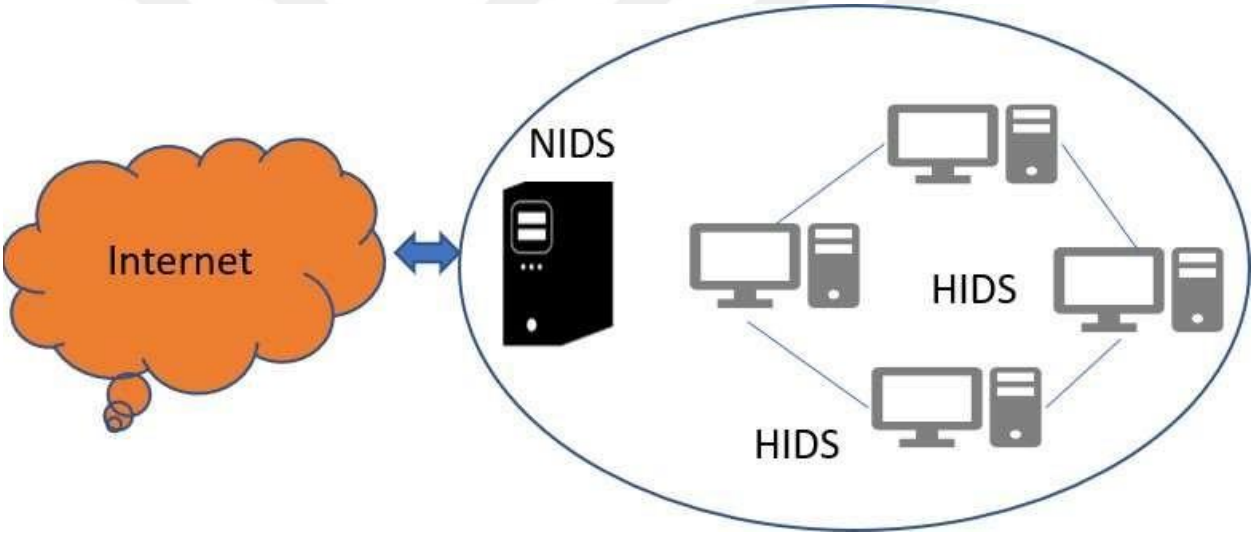

Figure 2.2: Deployment of HIDS and NIDS in a network [55]

The collaborative systems of intrusion detection (CIDS) [1] have solved this tragedy. The aim of CIDS is to improve single IDS detection capability. Every IDS interacts with other IDS in that case to cooperate on the exchange of data and to administer trust[2]. The following forms can be classified as CIDS:

• Hierarchical Frameworks of Intrusion Detection Distributed (DIDS) [12].

• Subscribe Collaboration Framework such as (Distributed Internet Overlay) [13].

• Peer-to-peer consultation device such as a question processor on Internet (PIER) [14].

2.2.3 Challenges

IDS solutions solve many issues, such as protection and confidence:

• Due to the complexity of wireless ad-hoc networks, it is impossible to perform analysis and correlation work on a centralized server [15].

• They are looking for a flexible host because there is a potential that they could be caught and then join the organization to steal the data.

2.3 DATA MINING

The society today feeds on digital and computer-generated information. In its raw form most of the information is: data. If data is defined as stored data, then the patterns behind the data are. There is a widening divide between data generation and our understanding. The proportion of data that people understand decreases as the amount of data increases. Data, potentially valuable information, is concealed in this vast volume of data, which is rarely used. [3]

The traditional approach to data transformation depends on manual analysis and interpretation in research, economics, marketing, health care, retail or any other area of expertise by one or more analysts. This manual analysis is costly, sluggish and subjective, as data volumes increase dramatically. The idea to find useful information in data is given to various names such as data mining, extraction of knowledge and the data archaeology and data pattern analysis. In 1989, a workshop developed the sentence Information Disk in Databases (KDD). If KDD is the complete search process for useful information such as data storage and access, data mining is an important step. The use of particular algorithms to collect data is data mining. Almost every statistical expert and data analyst traditionally uses the term data mining but in recent years has become more common in other fields. [10]

Data mining can be applied in many industries, such as web mining, judgment decisions, image scanning, prediction of loads, medical diagnosis, marketing, sales and development. Machine learning approaches are the scientific foundation of data mining. This is used for the finding and extraction of raw data information. The data is electronically processed in the data mining, and trends in data are automatically or semi-automatically found [3].

There are a variety of occasions when you enter a machine learning algorithm. An example is a separate, individual example of the principle we want to understand. In each case, the functionalities (also known as attributes) are predetermined, that quantifies the instance's varied features Indicators might be quantitative or nominal (also known as continuous) (also called discrete). Lines and columns are characteristic instances of a matrix. For example, "number of wheels" and "color" characteristics are characteristic of land vehicles [3].

In data mining applications, four types of machine learning are included: classification, combined learning, clustering and numerical prediction. Classification learning consists of introducing a collection of categorized instances in the learning process and thereby learning how to identify invisible instances. Association learning aims at defining some interdependent characteristics [3].

The classification of the changes is calculated by having a training set of data and an independent test data set, there is a lack of information about the true classifications How well the machine learning algorithm is able to identify unknown occurrences is measured by the "success rate [3].

2.4 MACHINE LEARNING

2.4.1 Definition

Arthur Samuel defined machine training as "a field of study that allows computers to learn without being explicitly programmed in 1959 "[21, 22], and was considered to be one of its first pioneers. He wanted to teach a computer to play machine controls.

When Samuel began, it only took a few years to create an algorithmic program before he started implementing it. In order to make choices, he introduced a system " Tree " On the basis of statistics,

the "Tree" structure calculates some progress. These figures are based again on the probability of the next move. The actual software configuration, which ranks the next phase, has been trained several times to know how to move and what the best move is next. It also takes into account what the opponent can do, so he "think" what the following moves might be. [22, 23].

In 1998 a "new" concept was created for Tom Michael Mitchells. The description of Mitchells is a more formal and mathematical clarification. Many researchers quote it because the way it is displayed is a little more scientific [23, 24].

Machine learning is an area of research often known as artificial intelligence (AI). This is because it should be possible for the computer to take decisions, taking data and established facts into account while determining. It's a way to give human potential to a computer and it's AI [25]. AI is a broad word which appreciates several different techniques. Machine learning is just one of these, there are genetic programming technology etc.

Machine learning is often known as part of data mining, since data mining is real. can be performed using tools and techniques for machine learning, but the decision-making process has not been an important part of the machine learning features [26]. Certain exceptions still occur, but mainly data mining is used to measure probabilities and statistics. That is why we are using data mining. These are examples of data mining and machine learning since it is possible to do certain calculations and, for example automated stock exchange, pumps, pacemakers, etc.

Machine learning algorithms are typically used for cyber protection in order to detect network anomalies., and high detection rates have been shown [27]. The predictive model of the training data is built through these algorithms [28]. This model forecast is utilized to follow against the given information malignant exercises [29]. This kind of learning is called administered learning [30].

The vector machine help, naive classification of Bayes, decision table, and decision tree [31] are some of these. Class marking requires supervised algorithms, whereas class marking algorithms don't need class marked data [32]. Unattended algorithms include clustering, k-means, deep neural

networks, etc. [33]. The above-mentioned two algorithms are semi-monitored. All data should not be classified [34]. Some examples of semi monitored algorithms in master learning are graphical, automated testing and generative modeling. [35]. Table 2.3 demonstrates machine learning classification [36].

2.4.2 Machine Learning Tasks

Assisting in the development of an AI calculation requires two clear steps: preparing and testing To prepare for each model, there is a specific strategy. In most cases, a structure oversees this connection and the strategy of the AI model, such as scikit-learn [14], TensorFlow [15], Pytorch [16], Matlab or Weka [17]. The system firmly affects the advancement of the calculation or the quantity of accessible boundaries.

AI models can perform many undertakings, three of which are especially fascinating for interruption discovery: arrangement, relapse, and remaking. Characterization arranges sections into a few classes, for example, "ordinary" or "assault", or even various groups of assaults. Relapse (additionally called "expectation") is utilized to decide ceaseless qualities, including a likelihood that an information is an assault. At last, recreation is explicit to a specific sort of neural organization. This assignment attempts to recreate the info information by compacting and de-pressurizing them to drive the organization to become familiar with the provisions (portrayal learning).

AI calculations are prepared in two distinctive manners: in a managed or un-regulated way. A few models can be prepared botfly, like neural organizations. Most models utilize directed preparing, where the dataset incorporates the two sources of info and the right outcomes related with them. The calculation figures out how to show the numerical capacity that connects these outcomes with the comparing inputs. Grouping and relapse are two exemplary administered preparing assignments. Then again, unaided preparation doesn't utilize any outcomes in its preparation dataset. Its motivation is to comprehend intriguing designs inside the info information. Remaking is an illustration of an unaided assignment.

When the preparation is finished, AI models should be tried to survey their exhibition. This assessment should include new information, which were not a piece of the preparation set. Something else, the assessment would be one-sided on the grounds that the model has effectively seen this information, and the right outcomes if there should arise an occurrence of regulated learning.

An approval set can likewise be utilized to analyze various upsides of a boundary (for instance, a learning rate or various neurons). When planning is complete, the value with the best results on the approval list is used. A test set will be used to test the entire organization at this time. An approval set should likewise be made out of new information, frequently a little segment of the preparation set is saved to this assignment.

2.4.3 Machine Learning techniques

The methods of machine learning are various approaches to how and how machine learning can be used. There are various ways to learn machines and the various techniques are further described below [11].

2.4.3.1 Supervised learning

Supervised learning is an automated learning process, regulated because the word means that certain helpers are used. The method of detection could be accompanied by labels that mark the data to be used. The most prevalent color can be used to identify an automobile with a traffic camera, for example. These students will learn to identify the different colors on the cars and compare them to those on camera. As you can see, this is just one example of supervised education. As long as the indications suggest that the car is a color or that traffic is a form of assault, controlled learning makes it easier to recognize. The machine learning method used is supervised learning when the data it would use is named. [14, 11].

2.4.3.2 Unsupervised learning

Instead of merely checking one of the labels which is achieved in controlled, unregulated learning is the opposite of supervised learning. It collects information and analyses the entire input such that decisions are taken on the basis of the entire information. The device does not know the correct "answers" when using unattended learning. It should take into consideration the input sent/given and provide some organized response based on patterns which it has been trained to recognize or collect knowledge classes. This is one of the main purposes of uncontrolled learning. It could be used by shops. [14, 11].

2.4.4 Datasets

Enormous measure of information is needed to prepare AI calculations. The quality and amount of information is essential in any AI issue. Undoubtedly, these issues are very information subordinate: all the greater information can regularly beat better calculations, which is the reason they are so significant. The KDD Cup 99 and NSL-KDD datasets are the two most widely used interruption recognition datasets, notwithstanding their shortcomings. Additionally, two other later datasets were employed in order to make up for some of the shortcomings of the previous ones: This year, there will be two CICIDS tests: CSE-CIC-IDS2018 and CICIDS2017.

Other datasets for interruption recognition are available [36]. We've selected to look at two well-known datasets from the past, as well as two newer datasets with a lot of sensible attacks. Other datasets, such as Kyoto2006+[37], Utwente [38], or UNSW-NB15 [39], may have been selected. A single type of attack (DoS, botnet, etc.) was not considered.

2.4.4.1 KDD Cup 99

Since its publication in 1999, KDD CUP 99 is one of the most common datasets. More specifically, training data have been processed in around five million connection records in seven weeks while testing data have been collected in two weeks, consisting of about two million connection records [40]. Preparing information should be utilized during the learning system of an AI strategy, and test information on a completely prepared answer for assess its exhibition.

A DoS attack aims at stopping a customer from having a computer or a service. A test assault is a dangerous organization action, like port checking, to find out about the engineering of the organization. A R2L assault happens when an assailant acquires neighborhood admittance to a framework by means of the organization. A U2R assault takes advantage of weaknesses in the framework to acquire super client advantages. For example, the number of DoS assaults far outweighs the number of U2R assaults in Table 2.2. A similar possibility of appropriation does not exist between test and preparation knowledge.

Table 2.1: Examples of KDD CUP 99 features [58].

Feature name	Type	Description
duration	continuous	length (number of seconds) of the connection
protocol_type	discrete	type of the protocol, e.g. tcp, udp, etc.
service	discrete	network service on the destination, e.g., http, telnet, etc.
src_bytes	continuous	number of data bytes from source to destination
dst_bytes	continuous	number of data bytes from destination to source
flag	discrete	normal or error status of the connection
land	discrete	1 if connection is from/to the same host/port; 0 otherwise
wrong_fragment	continuous	number of "wrong" fragments
urgent	continuous	number of urgent packets
hot	continuous	number of "hot" indicator
num_failed_logins	continuous	number of failed login attempts
logged_in	discrete	1 if successfully logged in; 0 otherwise

Table 2.2: Distribution of KDD Cup 99 classes [58]

	Normal	DoS	Probe	R2L	U2R
Training	97278 (19.69%)	391458 (79.24%)	4107 (0.83%)	1126 (0.23%)	52 (0.01%)
Test	60593 (19.48%)	229855 (73.90%)	4166 (1.34%)	16345 (5.26%)	70 (0.02%)

These four assault classifications can be additionally isolated into 38 test assault types, and 24training assault types. Every association additionally has 41 inferred highlights, as displayed in Table 2.1. These elements, mathematical or emblematic, are broke down by the classifier to recognize ordinary associations from assaults. KDD Cup 99, albeit mainstream, experiences a few insufficiencies as called attention to in examinations [41]. The dataset contains an immense number of excess and copied records, and different slip-ups that influence the presentation of classifiers. They become one-sided towards more successive records, whose numbers have been expanded.

2.4.4.2 NSL-KDD

The NSL-KDD dataset [42] contains answers to many of these questions. In this improved version of KDD Cup 99, there are no more repetitive and copied records (about 78 percent and 75 percent of the records). NSL-KDD also reduces the number of complete associations from 805 050 in KDD Cup 99 to 148 517. Associations are also divided into different trouble level groups. When classifiers are preparing, this sequence can assist them identify the most difficult assaults. No matter how you look at it, NSL-KDD isn't all that fantastic, and it retains some of the underlying problems of KDD Cup 1999. As an example, the assaults in this dataset are really old and do not reflect what can be found in a cutting-edge firm, according to the data. Their contrived beginning is also a problem because it can't be changed without redoing the entire dataset without any preparation.

Table 2.3: Distribution of NSL-KDD classes [58].

	Normal	DoS	Probe	R2L	U2R
Training	67343 (53.46%)	45927 (36.46%)	11656 (9.25%)	995 (0.79%)	52 (0.04%)
Test	9711 (43.08%)	7460 (33.09%)	2421 (10.74%)	2885 (12.80%)	67 (0.30%)

2.4.4.3 CICIDS2017

CICIDS2017 [44] is the Canadian Institute for Cybersecurity's dataset developed for IDSs and IPSs. This dataset includes original data that is unrelated to KDDCup 99 [56], in contrast to NSL-KDD. With the help of several existing attack scenarios, the authors propose a public IDS data collection that can be trusted by the public. Originally, it was designed to address lack of a credible intrusion detection dataset.

Five traffic days, from Monday to Friday, the 3rd July 2017, CICIDS2017 [60] offers. In the first day only the usual traffic is traffic and the next four days the traffic will include standard traffic and 14 forms, including FTP-Patator, SSH-Spatator Services, SSH-Patator Service Denial (Slowloris, Slow HTTP Test, Hulk, GoldenEye), Heartbleed, web attack (brute force, SQL injection, XSS). CICIDS2017 consists of 3 network flow labels (83 features) and 56 CICFlowMeter captured network network packets (679 networks) [43] [43]. With 83,34 percent daily flows and 0.00039 percent "heartbled" in this dataset, it is also severely unbalanced.

Information is extricated from a recreation a lot nearer to the conduct of an advanced PC organization. The assaults were made utilizing genuine apparatuses accessible on the web and solid methodologies. Its enormous measure of information additionally assists with preparing profound neural organizations, which require more data to be effective.

Table 2.4: Examples of CICIDS2017 features [60].

Feature name	Type	Description
Flow duration	continuous	duration of the flow in microsecond
total Fwd Packet	continuous	total packets in the forward direction
total Bwd packets	continuous	total packets in the backward direction
total Length of Fwd Packet	continuous	total size of packet in forward direction
total Length of Bwd Packet	continuous	total size of packet in backward direction
Fwd Packet Length Min	continuous	minimum size of packet in forward direction
Fwd Packet Length Max	continuous	maximum size of packet in forward direction
Fwd Packet Length Mean	continuous	mean size of packet in forward direction
Fwd Packet Length Std	continuous	standard deviation size of packet in forward direction
Bwd Packet Length Min	continuous	minimum size of packet in backward direction
Bwd Packet Length Max	continuous	maximum size of packet in backward direction
Bwd Packet Length Mean	continuous	mean size of packet in backward direction

2.4.4.4 CSE-CIC-IDS2018

The Communications Security Establishment and the Canadian institute for cyber security are collaborating on a collaboration for CSE-CIC-IDS2018 [59]. It offers ten traffic days, with an emphasis on Amazon Web services between Wednesday, 14 February 2018 and Freitag, 2 March 2018 (AWS). This incorporates seven diverse assault situations that are like CICIDS2017: Heartbleed, web assaults (Damn Vulnerable WebApp, XSS, savage power).

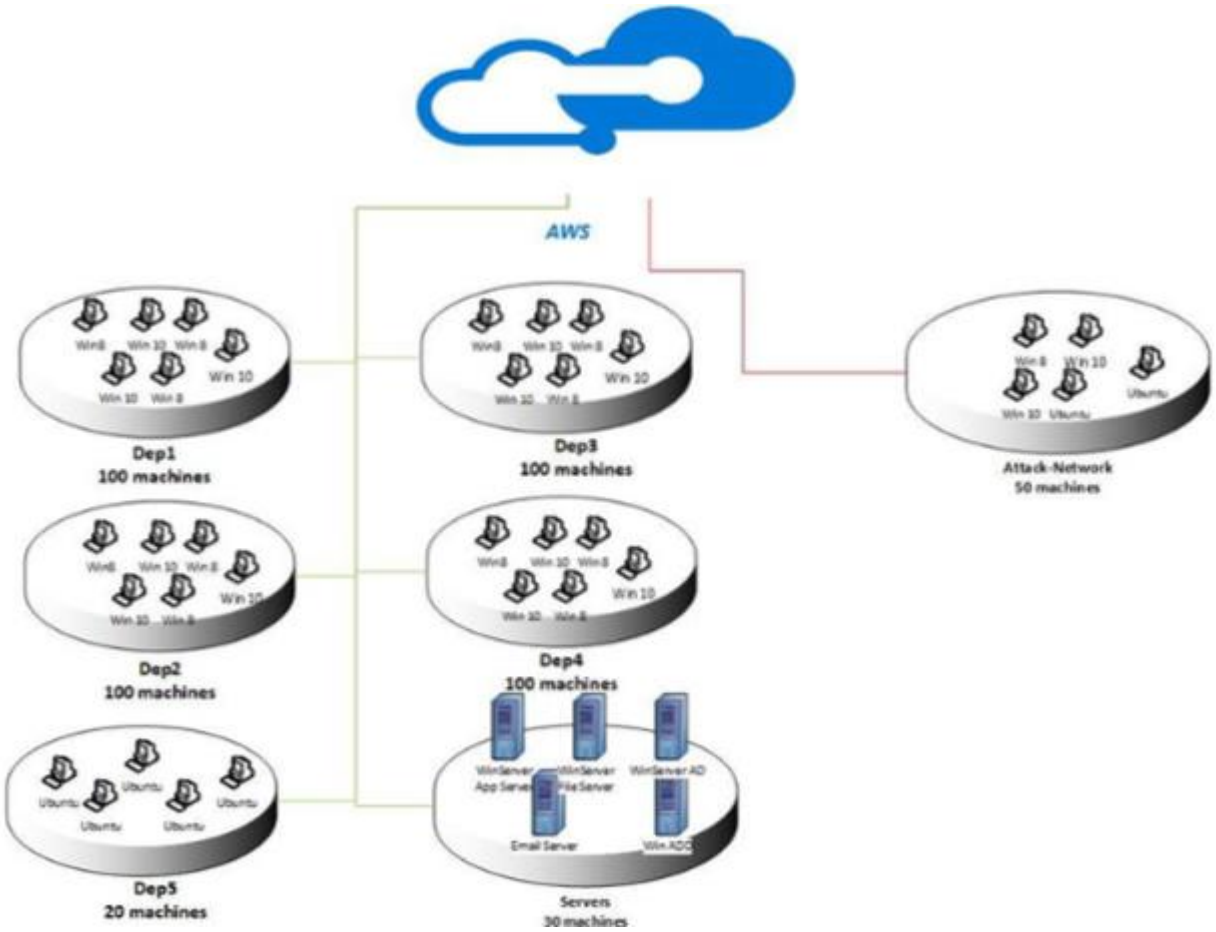

Figure 2.3: Network topology of CSE-CIC-IDS2018 [59]

Notwithstanding, the assaulting and the casualty networks have something else entirely, as displayed in Figure 2.5. The intruder uses a 50-machine infrastructure and the victims' organisation consists of 5 departments, 420 computers and 30 servers [59].

2.4.5 Deep Learning

The Neural Network algorithm was derived from Deep Learning [27]. The disadvantage of a secret layer in NN is discussed using different techniques [28]. This vast array of methods and strategies are used for deep learning. In three sub-gatherings, generative, prejudice and hybrid, the authors [29] are distorted [30] by deep learning. [27] [31].

2.4.6 Performance Metrics

A few measurements are utilized to portray the presentation of a classifier. Table 2.5 sums up the four potential results of a discovery.

Figure 2.4: Confusion matrix [9]

		Predicted	
		Normal	Attacks
Actual	Normal	True Negative	False Positive
	Attacks	False Negative	True Positive

• Detection rate (or "true positive rate", "recall", "sensitivity") is the proportion of attacks that are correctly detected.

$$Detection\ rate = \frac{TP}{TP+FN} \tag{2.1}$$

False positive rate (or "false alarm rate") is the proportion of normal traffic incorrectly flagged as attack.

$$False\ positive\ rate = \frac{FP}{TN+FP} \tag{2.2}$$

• In other words, precision is a minimal amount of clearly distinct outcomes (assault and ordinary traffic). According to the Jaccard file, in multiclass order, exactness is the convergence divided by association of mark sets.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{2.3}$$

- As a result of precision (also known as positive predictive value), a higher percentage of flagged threats are actually assaults.

$$Precision = \frac{FP}{TP+FP} \tag{2.4}$$

- F1-score is the harmonic mean of precision and recall (previously called “detectionrate”).

$$F1 - score = \frac{recall \times recall}{precision + recall} \quad (2.5)$$

- True-positive rate and false-positive rate are shown in the ROC curve (see Figure 2.5). Priority is likely to be given to regions below ROC curves over random negatives.

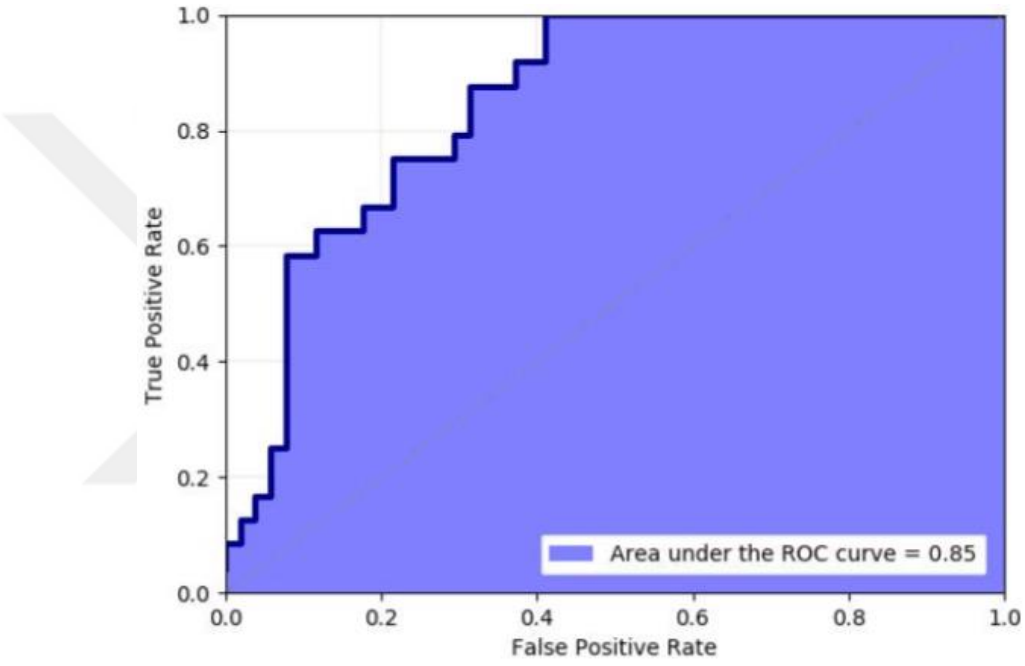

Figure 2.5: ROC curve example [10].

2.5 RELATED WORKS

A study using clustering of the KD Data Processor and Oracle 10g data miner is performed by 10 per cent of the training data set of KDD Cup'99 [56]. Its objective was to connect the various attacks to the protocol used (TCP, UDP and ICMP). It resulted in 19 out of 22 attacks affecting TCP.

In [20] with the k-means clustering algorithm at Weka 10% of the KDD Cup'99 dataset is used. In order to achieve high precision with low false positivity, a method of 7 steps is proposed. As a

result, the authors' first implementation was exactly 9,2013 percent and a false positive rate of 97 percent.

Samples taken from 10 per cent of the KDD Cup'99 datasets are used with stratified weighted sampling techniques [21]. The authors suggest a new decision tree algorithm that is compared to ID3 implemented by MATLAB. This results in a lower error rate than ID3 of their proposed algorithm.

In order to classify the most relevant characteristics for the classification of attacks, the intrusion sensing must be enhanced by [22]. In order to evaluate the 41 characteristics with the OneR machine learning algorithm, the authors used 10% of the data on the KDD Cup'99 [56] preparation. The calculated performance is true-positive and wrong. The findings were that Neptune and Smurf are very much correlated with those traits by the regular class and the attack classes. The authors comment that 98% of the training data of these tree groups are really good results, thanks to the machine-learning algorithm OneR.

In a comparison analysis of selected data mining algorithms in Weka the entire NSL-KDD dataset is analyzed [25]. Five of the famous SVM, ANN, k-NN, Naïve Bayes, and C4.5 algorithms have been compared. The accuracy, error rate and classification times are some of the performance metrics used. As a consequence, C4.5, because of the high precision and relatively rapid classification period, is ideally suited to retail classification.

[11] is performed on 10 per cent of the train data collection of the KDD Cup'99. In order to evaluate all data features in the data set, the authors quantify knowledge gain for each class (normal or attack). The author notes that the results can be challenged because of the unreal simplicity of the 10 per cent KDD Cup'99 train knowledge. 8 features therefore have very little detail (smaller than 0.001). In the opinion of the writer, the functions 9, 20 and 21 are not used for attack detection (similar to table 3). Feature numbers 15, 17, 19, 32 and 40 also play a limited role in attack detection.

While several researchers study the KDD Cup'99 dataset [58], 10 percent of train data are mostly sampled based on measurement costs. their approach. This approach provides different data sets for researchers, and thus the findings are incompatible and incomparable. Further analysis on the enhanced NSL-KDD dataset should be carried out to obtain comparable performance. There seems to be a deficient awareness of the most important features in the DARPA 1998 data set for both the entire data set and each category of attack (DoS, Probe, Un2R, R2L).

A paper by Mulay et was the first paper that can be related to this study. the.[27]. As of 2010, IDS is using SVM and decision trees. Using SVM and decision tree Algorithms, they were able to achieve excellent results using machine learning techniques on the KDD CUP99 data set. In order to see what results are best, the SVM and decision tree have also been used. An SVM-based decision-tab used the decision-tab. This is an excellent way to solve it when dealing with multi-class results. [30].

This thesis is written by X in the second document. X. Xu and.la. [46] The goal is to optimize classifier training and testing. An algorithm called PCA was used to optimize it (Principal Component Analysis). The PCA does this by bringing the unwanted data into the IDS and then running SVM on the data [46].

The third paper discussed is a paper on the use of clusters to boost the online detection anomaly network anomaly classifiers [32]. It's Ming-Yang Su &. A. [53]. A. In this paper, scientists also opted to use KDD CUP99, but this is based on the above mentioned other k-NN algorithm. It's not just the k-NN algorithm that it uses but also seeks to optimize the results across a cluster to increase the efficiency of classification. When the findings are shown, the efficiency of all the studies that have been made does not increase.

2.6 CONCLUSION

The findings are higher than without a clustering in research when two cross-validations of the data set are done [47]. During two cross-validations, the data set is shuffled approximately two times. For those who do not run the clustering, all other paper studies have the best results.

3. PROPOSED SYSTEM

Human lives have been secured by locks, clasps, signatures for centuries by national legislation, manners and customs. They were safeguarded. Bank properties, traffic light manipulation, satellite detection, and impairment are now accessible on virtually every device, including aircraft hacker IP power, and is automated and electronic. The most daunting, critical challenge of 21st century technologists was to establish security technologies defend the privacy and property by detecting cyber assaults and network intrusions. [1]

Applications are commonly used for the efficiency assessment of intrusion detection systems from Advanced Research Projects (DARPA) 1998 and revised dataset KDDCup [56] (Knowledge Discovery and the Data Mining Competition) of the year of 1999 and NSL-KDD dataset (IDS). [4]. Many researchers have developed safety technologies and have explored new methods for detecting cyber-attacks in these databases, but none of the Top 10 data mining algorithms [5] selected by experts in data mining have studied their results. You must know how to detect various cyber threats to update the network configuration. Further research is required in order to broaden the knowledge about data mining for detection of network intrusion.

3.1 SYSTEM OVERVIEW

For each part of the dataset, we utilize a similar CNN structure. Figure 3.1 shows the CNN model engineering that is carried out in the parallel grouping for interruption identification and the components of each layer for the primary piece of the dataset. The single model comprises of an information layer, two convolution and pooling layers, three totally associated layers, and a result layer. To command over-fitting, a dropout layer with a 0.5 dropout rate is utilized between the straightened model and the main completely associated layer. A Rectified Linear Unit (ReLU) actuation work is available in each covered layer. The information from the stream is changed into a 2-layered framework by the information layer. The resulting layer is the convolution layer, which concentrates input data using 32 zero-padding channels. The third layer is the most extreme pooling layer, and the data is investigated with 2.2 down inspecting with a phase size of 1. 1. The

fourth layer resembles the second, of course, really the convolution has 64 sections. As the classifier's result, a SoftMax layer is used at last, with the layer being unquestionable for equal and multiclass orders. The cross-entropy cost work is used as the incident ability to be restricted through getting ready. Adam, 128, and 0.001, respectively, are the batch size, the optimizer, and the learning rate used in training the networks.

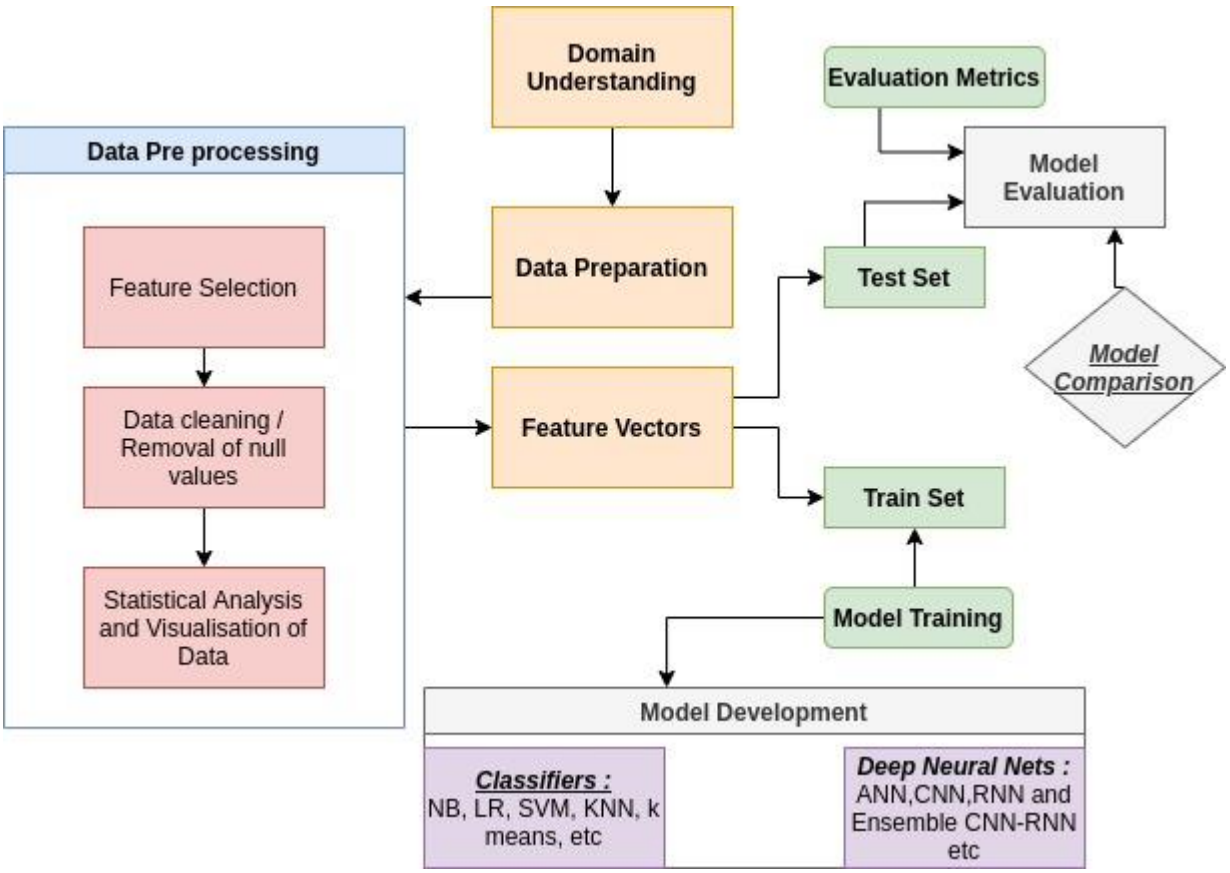

Figure 3.1: Sequence of actions for the proposed model

In this work we are using feed-forward neural networks trained on the KDDCup dataset [56] [58] to classify network connections as belonging to one of two possible categories: normal or anomalous. The goal of this work is to maximize the accuracy in recognizing new data samples, while also avoid overfitting, which happens when the algorithm is too attached to the data it learned and is not capable of correctly generalizing on previously unseen data.

It's important to understand how tasks are grouped together to achieve the desired results. This half-and-half model was constructed using a variety of methods, as shown in Figure 3.1 [48]. The following is the separated clarification of each progression in the stream chart [48] [49] [50].

- As a first step, the KDDCup 99 dataset [56] [58] [61] is stacked and pre-handled via mathematical one-Hot-encoding.
- In order to eliminate redundant and unnecessary data, the Feature determination of the information is conducted at that moment.
- For a later examination, we train the classifiers for each element as well as for the reduced components.
- The total Hybrid model is constructed utilizing these means by carrying out every one of the three classifiers, for example, Random timberland, Hybrid choice trees, SVM, and Naive Bayes classifier.

• The model is assessed by its presentation as far as Accuracy, Precision, F-measure, and Recall.

3.1.1 Data Preprocessing

3.1.1.1 Numericalization

In the KDDCup 99 dataset [56] [58] [61], there are three features which are non-numeric and 38 numeric features. Since the input values must be numeric, we convert the non-numeric features into numeric. For example, the feature 'protocol_type' can have three different types of attributes which are 'tcp', 'udp', and 'icmp'. We encode them as binary vectors (0,0,1), (0,1,0) and (1,0,0). This way, we convert the 41-dimensional feature map into a 122-dimensional feature map.

3.1.1.2 Normalization

There are several features in the dataset in which the difference between the max and min values are large. Such features are Dst_bytes [0,1.3 × 109], Src_bytes[0,1.3 × 109] and duration

[0,58329]. We apply the logarithmic scaling method to lower the differences and then use the formula below to map them to the [0,1] range:

$$xi = (xi - Min) / (Max - Min) \qquad (3.2)$$

3.1.1.3 Feature selection

The elements of a traffic record furnish data about the experience with the traffic input by the IDS and are characterized into four kinds: Intrinsic, Content-based, Host-based, and Time-based. We have considered the whole dataset for training as a four part dataset for classifying the attacks. There are many highly correlated features which hampers the model in classifying in the validation and testing time. So, we split the dataset in four part and took the first part as hybrid features.

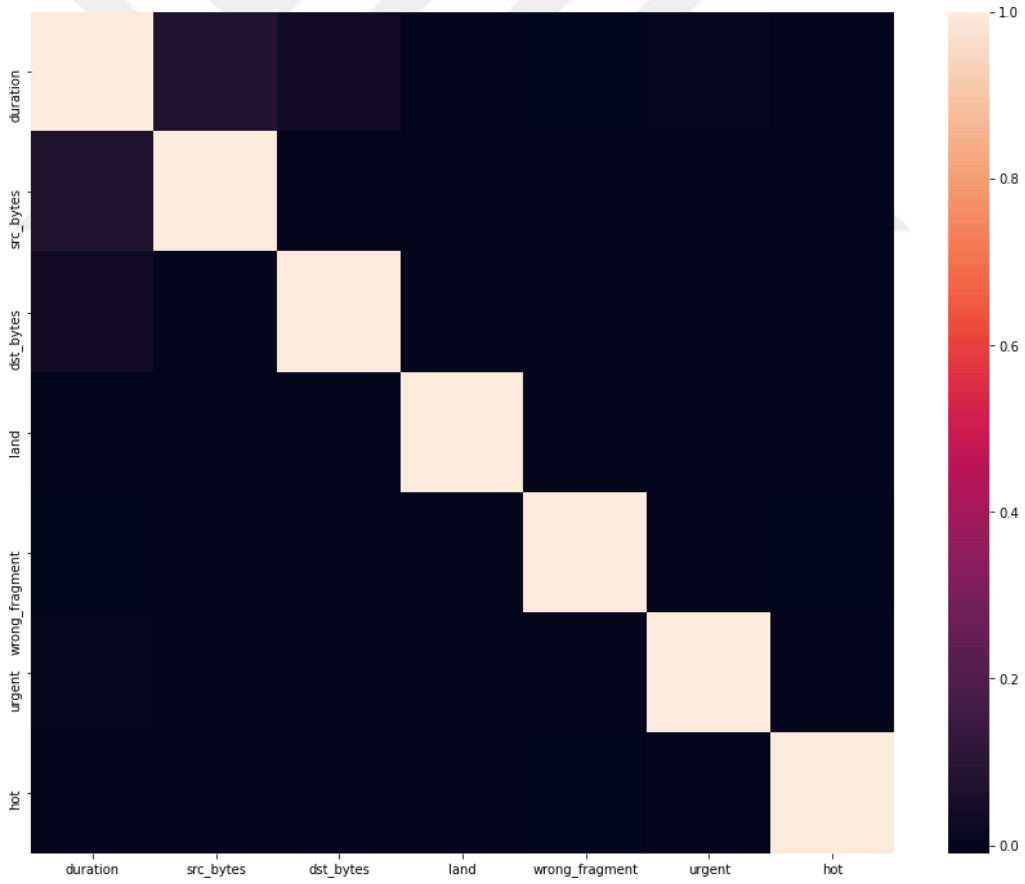

Figure 3.2: Selecting the most relevant features of NSL-KDD [56] [58] [61]

Intrinsic features can be inferred from the packet header without examining the payload itself, and they include the basic information about the packet. First 10 features of each packet give the model more generalization and inference ability which helped us to go through all this way with a better accuracy than other models.

3.1.1.4 Data cleaning

Since the NSL-KDD dataset is already an enhanced version of the older KDD ’99 CUP dataset [56], little additional cleaning had to be performed: the set had already been cleaned from redundant data and null values [5]. Also, the ratio between normal and anomalous entries is good for machine learning purposes. The only step taken at this stage, a part from loading the .arff files and decoding strings using UTF-8, was to change all the entries with http XXX as their service values into http, where XXX denotes the port number. This decision was taken since the port of an http connection, which is specified in the protocol string only for some entries, is very unlike to be correlated with an anomalous behaviour in a general case. Also, leaving this distinction could lead to new http connections on previously unseen ports to not be recognized correctly by the algorithm. This cleaning has later been proven to be an effective solution for partially reducing overfitting.

3.2 MACHINE LEARNING MODELS

3.2.1 Recurrent Neural Network

Recurrent Neural Networks (RNNs) are a type of neural network that is used for sequential prediction. RNNs can use previous outputs as inputs and can process inputs of arbitrary length by not expanding the model size where weights are shared across time [60].

3.2.2 Bidirectional Recurrent Neural Network

A form of RNN is Bidirectional Recurrent Neural Networks. Which offers all of the benefits of classic RNN design plus increased speed and the ability to capture long-term dependencies.

Bidirectional RNN is unique in Natural Language Analyzing because of its capacity to incorporate future and previous output when processing present output.

3.2.3 Long Short Term Memory

Long Short Term Memory, otherwise called LSTMs is an adjusted variant of RNN having the capacity of learning long haul conditions. LSTM manages the evaporating inclination issue experienced by conventional RNN. Recollecting the data for significant stretches of time is the default conduct of LSTM. It enjoys the benefit of failing to remember something from the past data when it needs to, and refreshes the fundamental data for the future time steps [57].

3.2.4 Convolutional Neural Network

Artificial Neural Networks are supervised machine learning algorithms inspired by the human brain. The main idea is to have many simple units, called neurons, organized in layers. In particular, in a feed-forward artificial neural network all neurons of a layer are connected to all the neurons of the following layer, and so on until the last layer, which contains the output of the neural network. This kind of networks are a popular choice among Data Mining techniques in now days, and have already been proven to be a valuable choice for Intrusion Detection [3, 4].

3.3 CNN MODEL BULDING

CNN has two operations which are convolution and pooling. Con- volution changes input data to output using a set of kernels or filters. The produced output showcases the features of the input data. That is why the output is known as the feature map. An activation function processes the convolution output further and down-sampling trims off irrelevant data using pooling. Pooling removes glitches in the data. That is how the learning improves for the following layers.[8][9] CNN adjusts the kernels/filters using rounds after rounds of learning so that the feature map can functionally represent the input data. We use ID convolution since the network packet in the dataset is represented in an ID format.

$$(f * g)(i) = \sum_{m}^{j=1} g(j).f(i - j + m/2) \qquad (3.1)$$

Here, i denotes the position of the values in sequence data. The activation function is ReLU. Finally, we use max pooling.

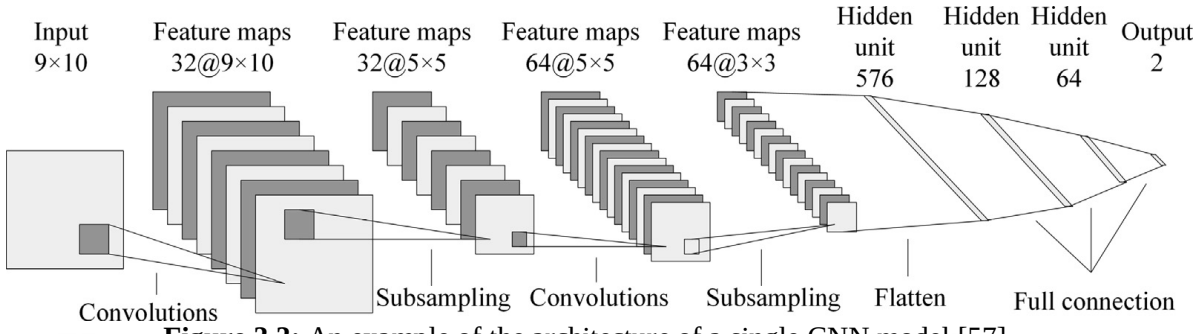

Figure 3.3: An example of the architecture of a single CNN model [57]

In a conventional neural organization, the information initially goes to the info layer. Then, at that point, it goes to the secret layer and from that point to the result layer. The layers are completely associated, and in a similar layer, there is no association between hubs. Customary neural organizations, there-front, have many issues that can not be tackled. The engineering of convolutional neural organizations [57] is an improvement over the architecture of standard neural networks. CNN has achieved extra- ordinary results in fields such as image classification and speech analysis because of it [13].

In CNN, there are:

(1) One or more convolutional layers

(2) Pooling layers at the top

(3) Fully connected layers

(4) Dropout layers which serve as regularization layers [57].

In view of this design, CNN can exploit the 2D construction of the information. As such an organization can accept a picture as an info. Thusly, we keep away from confounded component extraction and un-vital information remaking of conventional acknowledgment calculations. The demonstrating productivity can be expanded, and the trouble of handling information physically can be diminished through pooling, shared loads and inadequate availability. CNN can learn from various levels of features from a vast amount of data that is unlabeled. Therefore, the ways CNN

can be used in a field such as a network intrusion detection are comprehensive. The architecture of a method similar to the proposed method used in this paper is shown in Figure 3.3 [18].

3.4 DATASET

The NSL-KDD dataset isn't the first of its sort. The KDD Cup 99 [58] [56] [61] was a worldwide rivalry for Knowledge Discovery and Data Mining Tools. This opposition was held in 1999 to gather traffic records. The opposition point was to make an organization interruption finder, which is a forecast model fit for recognizing "terrible" associations, known as interruptions or assaults, and "great" typical associations. Because of this opposition, an immense number of web traffic records were gathered and ordered into an information assortment known as the KDD'99. The NSL-KDD informational index was created as a refined and tidied up form of the University of New Brunswick's KDD'99 information assortment.

These informational collections involve web traffic records saw by a simple interruption location organization, and they are phantoms of the traffic experienced by a genuine IDS, with proof of its quality. The dataset has 43 qualities for every record. The initial 41 highlights are traffic inputs, while the last two are marks. The primary name indicates whether the information is an ordinary or an assault bundle. The score is addressed by the last name (the seriousness of the traffic input itself). The informational collection contains four separate assault classes: User to Root (U2R), Denial of Service (DoS), Probe, and Remote to Local (R2L). The following is a speedy portrayal of each assault

A forswearing of-administration (DoS) assault expects to end the progression of traffic from and to the objective framework. The IDS gets a strangely huge amount of traffic, which the framework can't deal with and subsequently closes down to ensure itself. This disallows normal traffic from getting to an organization. For instance, on a day with an enormous deal, an internet based store might be overpowered with online orders, and in light of the fact that the organization can't deal with the solicitations as a whole, it will close down to keep paying customers from buying anything. This is the most well-known assault in the informational index.

A probe is an endeavor to acquire information from an organization. The objective is to carry on like a criminal and take fundamental information, like individual client data or banking data. U2R is an assault that starts with a standard client record and afterward endeavors to gain admittance to the framework or organization as a super-client (root). The assailant endeavors to take advantage of framework weaknesses to get root honors/access. R2L is a sort of assault that looks to obtain nearby admittance to a far-off machine. An aggressor doesn't have nearby admittance to the framework/organization and tries to "hack" their direction into it. We can see that DoS is different from the other three attacks. DoS tries to shut down a system to stop traffic flow completely. In contrast, the other three attempts to infiltrate the system undetected. There are [56] :

- 11 types of DoS attack
- 6 types of Probe attacks
- 7 types of U2R attacks
- 15 types of R2L attacks

A breakdown of the various subclasses of each attack existing in the dataset is shown in table number 4.

Table 3.1: Multiclass attacks in NSL-KDD [58][61]

Classes	DoS	Probe	U2R	R2L
Sub-Classes	apache2	ipsweep	buffer_overflow	ftp_write
	back	mscan	loadmodule	guess_passwd
	land	nmap	perl	httptunnel
	neptune	portsweep	ps	imap
	mailbomb	saint	rootkit	multihop
	pod	satan	sqlattack	named
	processtable		xterm	phf
	smurf			sendmail
	teardrop			snmpgetattack
	udpstorm			spy
	worm			snmpguess
				warezclient
				warezmaster
				xlock
				xsnoop
Total	11	6	7	15

Notwithstanding the way that these assaults exist in the informational collection, the dissemination is considerably slanted. A breakdown of the record dispersion is displayed in the table beneath. The greater part of all informational collection's records are conventional traffic, and the U2R and R2L disseminations are shallow. While this is a low number, it precisely addresses the dissemination of advanced web traffic assaults, where DoS and U2R and R2L are seldom considered to be the most widely recognized. The traffic record characteristics provide information to the IDS about the experience with the traffic input. They are classified into four types: intrinsic, content-based, host-based, and time-based. Below is a description of the various feature categories:

- Without inspecting the payload, intrinsic characteristics can be extracted from the packet's header and contain the relevant packet information. This category includes features 1–9..
- Because the content characteristics are sent in numerous chunks rather than a single piece, they contain information about the original packets. With this information, the system can gain access to the payload. This category includes features 10–22.
- Time-based features hold a two-second window to analyze the traffic input and contain information such as how many connections it was trying to make to the same host. Instead of data about the content of the traffic input, these features are mostly counts and rates. Features 23–31 is included in this category.

It is possible to break down the feature types in this data set into four types.:

- Features: 2, 3, 4, 42 -> 4 Categorical
- Features: 7, 12, 14, 20, 21, 22 -> 6 Binary
- Features: 8, 9, 15, 23–41, 43 -> 23 Discrete
- Features: 1, 5, 6, 10, 11, 13, 16, 17, 18, 19 -> 10 Continuous

All the features are shown in the table below:

Table 3.2: Features of NSL-KDD [61][56]

No.	Features	No.	Features
1	duration	22	is_guest_login
2	protocol_type	23	count
3	service	24	srv_count
4	flag	25	serror_rate
5	src_bytes	26	srv_serror_rate
6	dst_bytes	27	rerror_rate
7	land	28	srv_rerror_rate
8	wrong_fragment	29	same_srv_rate
9	urgent	30	diff_srv_rate
10	hot	31	srv_diff_host_rate
11	num_failed_logins	32	dst_host_count
12	logged_in	33	dst_host_srv_count
13	num_compromised	34	dst_host_same_srv_rate
14	root_shell	35	dst_host_diff_srv_rate
15	su_attempted	36	dst_host_same_src_port_rate
16	num_root	37	dst_host_srv_diff_host_rate
17	num_file_creations	38	dst_host_serror_rate
18	num_shells	39	dst_host_srv_serror_rate
19	num_access_files	40	dst_host_rerror_rate
20	num_outbound_cmds	41	dst_host_srv_rerror_rate
21	is_host_login		

3.5 PERFORMANCE METRICS

The mixture AI model is tried like other AI models by estimating its arrangement exactness, accuracy, review, f-measure and by considering, it's blunder rate, genuine positive, bogus positive . We have utilized the accompanying boundaries and measurements , and the disarray lattice is used for showing the characterization.

Using a disarray lattice, you can see how a calculation is performing. As soon as the classifier has been created, it is used to test the information that was collected. A lattice of disorder is used to visualize the odds of a given event. With the conditions below, the exactness, precision, recall, and f-measure are broken down. Is also shown in relation to each assault class.

(a) The truth is out there. The classifier successfully predicts positive instances.

(b) Ignorance is bliss However, the classifier incorrectly classifies them as positive.

(c) Negative in the true sense Instances analyzed are negative and are categorized accurately as negative instances.

(d) Positively False Classifier misclassified the instances as negative while being positive.

(e) Accuracy Equation 2.3 calculates the average accuracy rate.

3.6 BINARY CLASSIFICATION

We converted 41-dimensional characteristics into 83-dimensional ones. In the binary classification experiment, the CNN-IDS model includes 122 input nodes and 2 output nodes. The number of epochs is set to 100, and the learning rate is set to 0.01. To determine the best model, set the number of hidden nodes to 60, 80, or 120. The no. of hidden layers is 2, and the batch size is 64. From the table below, we have determined that a hidden node value of 64 achieves the best result.

Table 3.3: Confusion matrix of 2-category classification on KDDTest+ [58][56]

	Anomaly	Normal
Anomaly	8721	990
Normal	2576	10238

The studies reveal that when given 100 epochs for the KDDTrain dataset, the CNN-IDS model has a good detection rate (84.16 percent). The outcomes got by J48, Naive Bayesian, Random Forest, Multi-layer Perceptron, Support Vector Machine, and other order calculations, just as the fake neural organization calculation, give 81.2 percent, which is steady with late writing on ANN calculations utilized in interruption location. Luckily, these results depend on a similar benchmark - the NSL-KDD dataset. The maximum accuracy achieved by the authors using traditional approaches is 99.40 percent in [3]. In binary classification, the CNN-IDS model outperforms conventional classification algorithms.

3.7 MULTICLASS CLASSIFICATION

Inspired by [4], We have used both Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) for multiclass classification. In CNN, the number of hidden layers that we have taken is 2. The number of epochs is 15 and 150 respectively. In RNN, almost all the values are similar to those used in the binary classification experiment. The difference is that the number of hidden nodes is taken as 80 and 120 respectively. It is observable from the table below that CNN with the number of epochs being 150 gains the best results.

Table 3.4: Confusion matrix for multiclass classification [56]

	Normal	DoS	R2L	U2R	Probe
Normal	9017	633	53	2	6
DoS	126	980	0	0	0
R2L	93	101	5528	0	0
U2R	33	0	0	2	2
Probe	1823	0	0	0	376

In this experiment, the CNN-IDS model achieves a higher accuracy testing dataset detection rate, not only than the NSL-KDD dataset detection rate, but also than other neural network models. The experimental results reveal that the fully connected model outperforms the reduced-size CNN model in terms of modeling ability and detection rate.

Table 3.5: CNN model accuracy for multiclass classification

Our model		
Train	Validation	Test
99.11	84.70	65.45

The accuracy of 99.11 that this model has achieved is better than both [4]’s 93.8%.

4. EXPRIMENT RESULTS AND DISCUSSION

4.1 EXPERIMENT DETAILS

During the trial, 30 documents are thoroughly examined on a Windows HP double center PC with 8 GB RAM. An Anaconda stage provides a Jupyter workbench on which the application is built. Python is the programming language that is used to implement this interruption model. An 80 percent train and a 20 percent test set of 125,972 and 22,543 standards, respectively, comprise the NSL-KDD dataset [61] for this exploration. This interference recognition model was made by consolidating various shrewd choice computations, like the Random Forest classifier, the Naive Bayes classifier, and Decision trees. Table 4.1 shows the aftereffects of utilizing these classifiers on 13 things from the NSL-KDD dataset for attack types DOS, Probe, U2R, and R2L.

4.2 DATASET ANALYSIS

This section provides a complete analysis of the NSL-KDD dataset [61] [56].

4.2.1 Dataset Summary

The NSL-KDD dataset [58] [56] [61] is provided in two forms: .arff files, with binary labels, and .csv files, with categorical labels for each instance. Since the object of this work is to build a binary classifier, we will focus only on the .arff files. The provided .arff files are:

- KDDTrain+.ARFF: The full NSL-KDD train set with binary labels in ARFF format
- KDDTrain+ 20Percent.ARFF: A 20% subset of the KDDTrain+.arff file
- KDDTest+.ARFF: The full NSL-KDD test set with bi- nary labels in ARFF format
- KDDTest-21. ARFF: A subset of the KDDTest+.arff file which does not include records with difficulty level of 21 out of 21.

The NSL KDD dataset's classes or marks are isolated into four classifications, three of which demonstrate the assault class and one as would be expected traffic [56]:

1) Denial of Service (DoS): This kind of assault tries to forestall or confine admittance to a PC framework, network assets, or administrations.

2) Probe: For this situation, the gatecrasher expects to filter an organization or PC framework for data or shortcomings that will later be used to lead attacks.

3) Remote to Local (R2L): For this situation, the assailant acquires far off unapproved admittance to a PC framework through an organization by conveying information bundles to that framework.

4) User to Root (U2R): For this situation, the gatecrasher accesses a client with ordinary honor and afterward endeavors to get sufficiently close to a client with head or root honor. Tables 2 and 3 uncover the investigation of the assault classes and sorts in the NSL KDD dataset [56] [61], just as the quantity of individual cases and records in the preparation and testing sets.

Table 4.1: Number of instances in the training set [58][61]

Attack Classes or Labels	Attack types (number of instances)	Total of instances
DoS	back (956), land(18), neptune(41,214), pod(201), smurf(2,646), teardrop(892)	45,927
Probe	satan(3,633), ipsweep(3,599), nmap(1,493), portsweep (2,931)	11,656
R2L	guess_passwd(53), ftp_write(8), imap(658), phf(4), multihop(7), warezmaster(20), warezclient(890), spy(2)	1,642
U2R	buffer_overflow(30), loadmodule(9), rootkit(10), perl(3)	52
Grand Total	59,277	

The normal traffic contains 67,343 instances which brings a total of 126,620 instances in the training set.

Table 4.2: Number of instances in the test set [58][61]

Attack class or label	Attack types (number of instances)	Total of instances
DoS	back(359), land(7), neptune(4,657), apache2(737), pod(41), smurf(665), teardrop(12), udpstorm(2), processtable(685), worm(2), mailbomb(293)	7,460
Probe	Satan(735), ipsweep(141), nmap(73), portsweep(157), mscan(996), saint(319)	2,421
R2L	guess_passwd(1,231), ftp_write(3), imap(307), xsnoop(4), phf(2), multihop(18), warezmaster(944), xlock(9), snmpguess(331), snmpgetattack(178), httptunnel(133), sendmail(14), named(17)	3,191
U2R	Buffer_overflow(20), loadmodule(2), xterm(13), rootkit(13), perl(2), sqlattack(2), ps(15)	67
Grand Total	13,139	

The normal traffic contains 9,711 instances which brings a total of 22,850 instances in the test set. More details on the features names and descriptions can be found at [58]. To avoid redundancy, we use only the KDDTrain+.ARFF and KDDTest+.ARFF files, which contain a total of about 148,500 entries. Table I contains a summary of the most important attributes of the dataset.

4.2.2 Data Cleaning and Visualization

For the purpose of our study, we are only interested in detecting the intrusion. That is, we want to find out if any intrusion has occurred in the network system or not. So, we approach this problem as a Binary Classification problem. The network system characteristics can denote either Normal (No Intrusion detected) or Intrusion (Intrusion detected). For this, we encode all normal values from attack column as 0 which denotes no intrusion and all other attack values as 1 which denotes that there has been an intrusion. We add a new column named is intrusion that will contain the aforementioned flags for each record. The column is intrusion is now set as our target column.

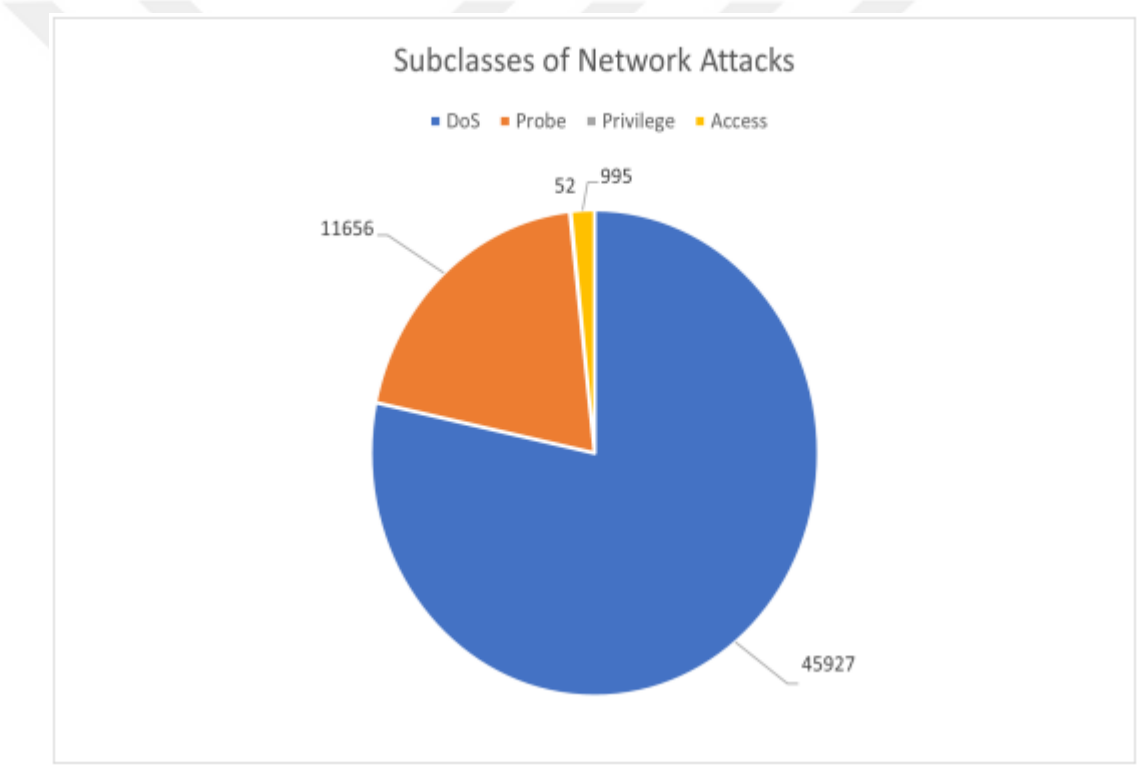

Figure 4.1: Distribution of various types of attacks

After specifying the target column, we perform feature selection to find the most relevant fields from the input feature set. As discussed before, we have a total of 42 input features. The feature set contains of both Numeric and Categorical fields. We use a Statistics-based approach to study the relations among the input features and their relationship with the target column. We compute

the correlations of the input features with the target column. For this, we encode the categorical variables using one-hot encoding technique. We remove any columns that have a very high (> 0.9) or a very low (~ 0) correlation with the target column. Then, we select the top 15 features that are most correlated with the target column ordered by their absolute value (to account for both positive and negative correlations). We also perform a co-linearity analysis as shown in Figure 4.2 on the remaining features in order to check if there are any linearly dependent pairs of features.

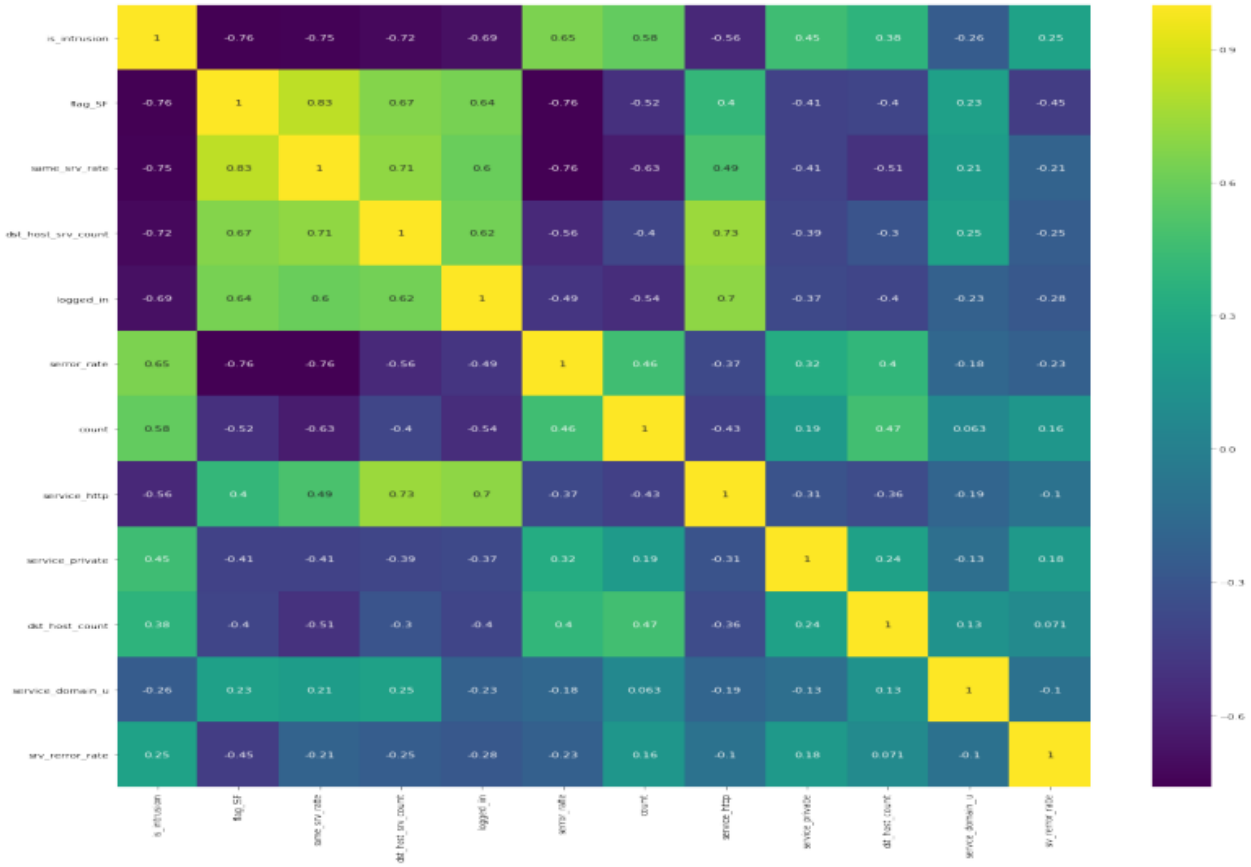

Figure 4.2: Most correlated features

4.2.3 Numerical Data Distribution

Numerical features have very different meanings in this dataset, and consequently different ranges. Continuous features are used for rates (e.g., error rates) and discrete features give information about the number of bytes in the packet, the connection duration, the number of reconnections and

many other variables. Figure 4.3 represents the normalized dataset: each column's value has been normalized between 0 and 1 in order to visualize how the different values of each feature are distributed. Note that this normalization takes into account both the train and the test set, hence it can be used only for data analysis.

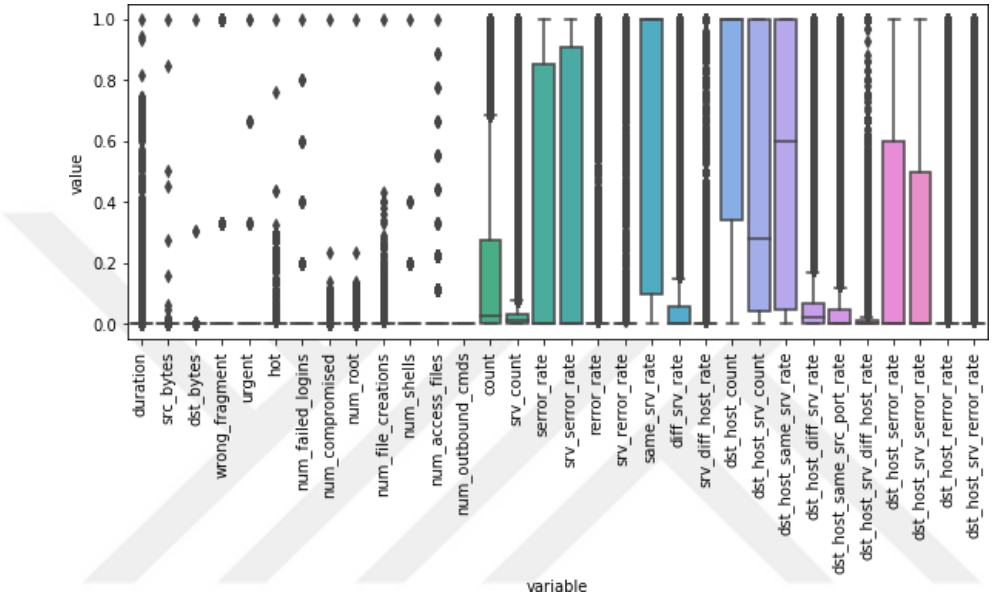

Figure 4.3: Distribution of discrete and continuous values in the normalized dataset.

4.2.4 Correlation Matrix

As a final step of the data analysis, we ran a correlation analysis for all the features of the dataset. Figure 4.4 illustrates the correlation matrix. Focusing on the class column, we can see how each feature is correlated with the target variable, either directly or inversely. This analysis can be extremely useful when performing feature selection to estimate how many important features there are in the dataset.

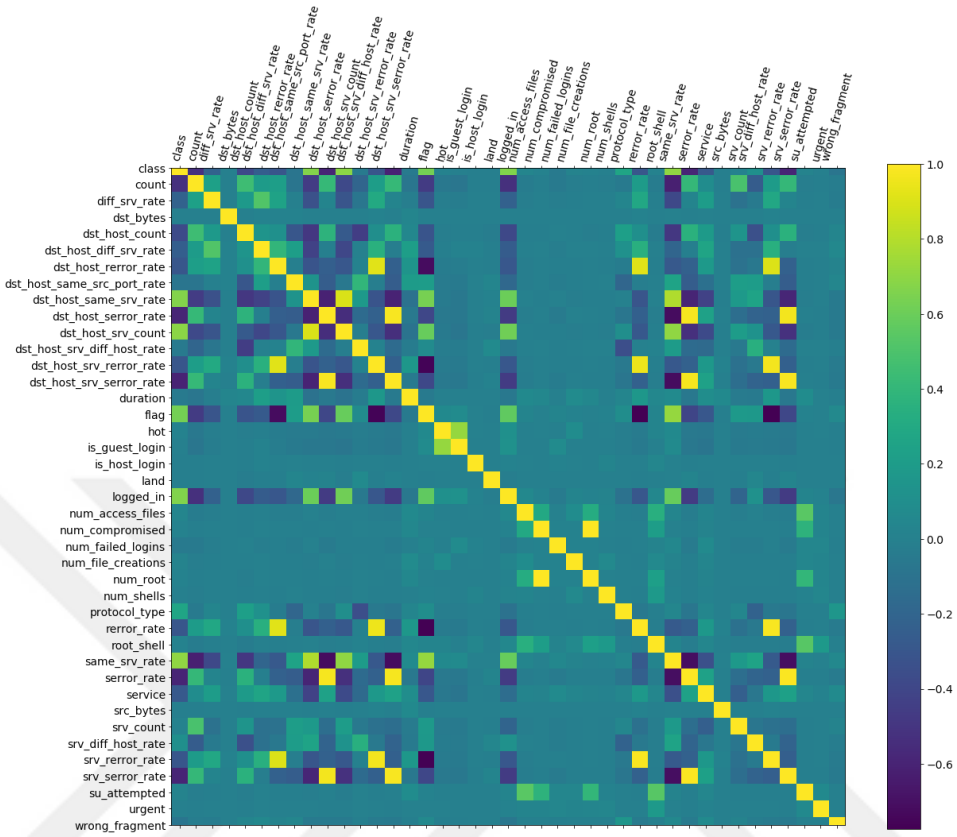

Figure 4.4: Correlation matrix between each feature of the dataset. A brighter color means higher correlation.

4.3 EXPERIMENT RESULTS

We have used Jupyter Notebook plan to conduct all of our experiments. We use two experiments to study the performance of the IDS model. First, binary classification. And secondly, five-category classification such as Normal, DoS, R2L, U2R and Probe.

Table 4.3: Performance of the CNN model and other traditional machine learning models in binary classification.

	KDDTest+	KDDTest-21
Naive Bayes	76.56	55.77
J48	81.05	63.97
NB Tree	82.02	66.16
Random Forest	80.67	63.26
SVM	69.52	42.29

RNN	83.28	68.55
Our Model CNN	84.16	70.26

J48, Naive Bayesian, Random Forest, Multi-layer Perceptron, Support Vector Machine, and other AI calculations are utilized to prepare models through the preparation set (utilizing 10-layer cross-approval) to look at the exhibition of various grouping calculations on the benchmark dataset for multiclass characterization as the twofold arrangement tests.

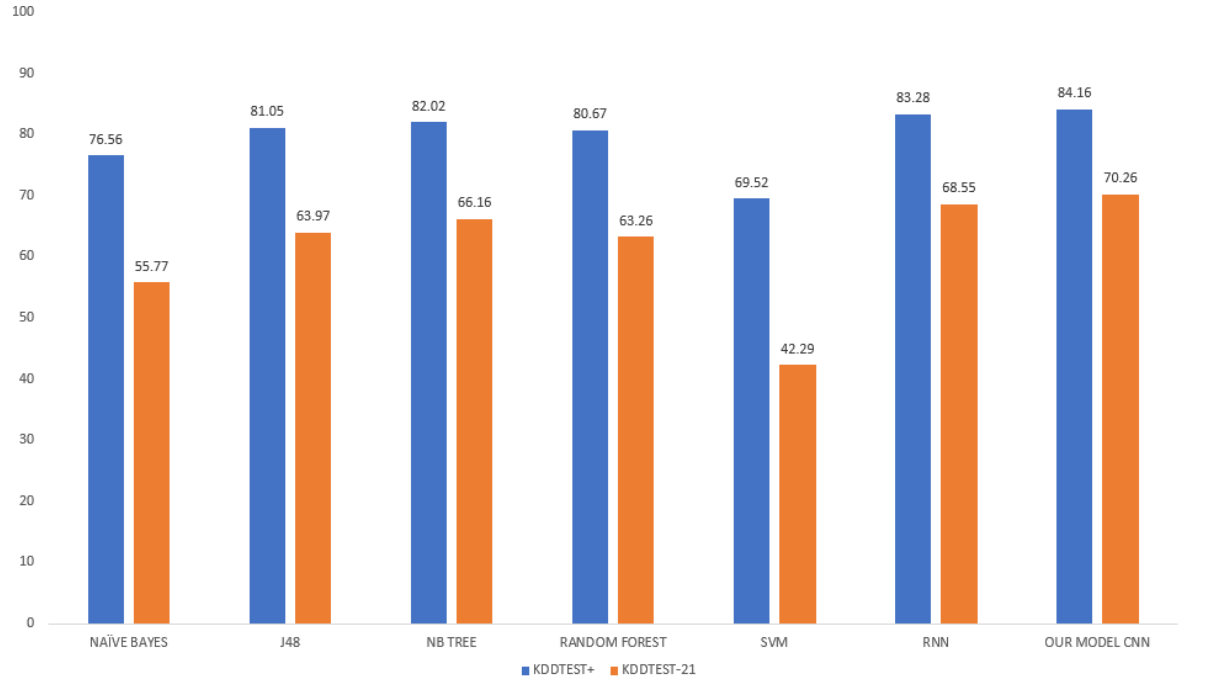

Table 4.4: Binary classification

The models are then applied to the testing set. The outcomes are depicted in Figure 4.5. The accuracy of classification algorithms in the five-category classification is lower than in the binary classification. The CNN-IDS confusion matrix on the test set KDDTest+ in the five-category classification trials is shown in Table 4.6. The experiment reveals that the model's accuracy is 84.70 percent for the test set KDDTest+ and 65.45 percent for KDDTest-21, which is higher than the accuracy of J48, Naive Bayes, Random Forest, Multi-layer Perceptron, and other classification

methods. We compared the performance of Naive Bayes, Support Vector Machine, Random Forest, and K-Nearest Neighbors models in binary classification studies.

Table 4.5: Performance of the CNN model and other traditional machine learning models in multiclass classification.

	KDDTest+	KDDTest-21
Naive Bayes	74.40%	55.77%
J48	74.60%	51.90%
NB Tree	75.40%	55.77%
Random Forest	74.00%	50.80%
SVM	74.00%	50.70%
RNN	81.29%	64.67%
Our Model CNN	84.70%	65.45%

We have evaluated the performance of different classification and neural network models to compare with our proposed model and we have found that we have achieved better result then other model by using Adam optimizer and epoch number 500. We have compared our model with other existing models and found that our model achieves better than any other neural network model in intrusion detection system but classification algorithms perform better for this specific problem because of labelled data and for the structure of the dataset.

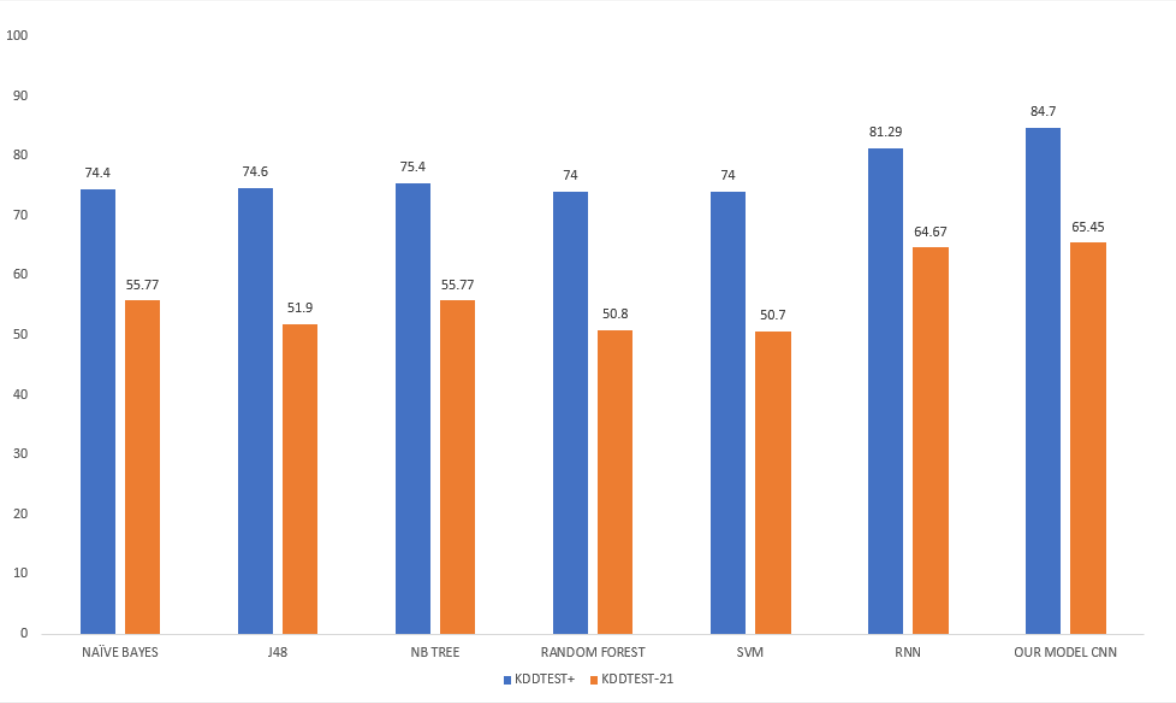

Figure 4.5: Multiclass classification

Similarly, we examine the presentation of multi-order of the CNN-IDS model against Naive Bayes, Support vector mama chine, Random Forest and K-Nearest Neighbors and RNN. In view of a similar benchmark, and with KDDTrain+ as the preparation set and KDDTest+ and KDDTest-21 as the testing sets, the exploratory outcomes show that the CNN-IDS interruption recognition model beats the other AI models as far as experimental outcomes and precision, even in the multiclass characterization issue. Obviously, the model we gave will take more time to prepare, but utilizing GPU speed increase can abbreviate the preparation time.

4.4 DISCUSSION

Our experiment results say that in both binary classification and multiclass classification, the intrusion detection model using neural networks achieve higher accuracy then tradition machine learning models using the same dataset. Though our models require more computation time, additional hardware can reduce that to a great extent. These theories are important to enhance information on data mining to detect network intrusion to detect and security configuration of

cyber-attacks. This study shows that k-Nearest Neighbor's Top 10 algorithms for the detection of cyber attacks are the most effective classification machine learning, as discussed in [5] and [3].

The results show how important a number of classifications are to be chosen. Although the particular cyber attacks that you are exposed to are fascinating to know, The wide range of classification groups reduces the accuracy of all machine learning algorithms. In order to increase the chances of making a sound security configuration option, the best categories are cyber attack categories such as DoS, testing,R2L and U2R attacks. This categorization of videos helps you to spot new threats and improves the information about cyber attacks.

Basic functions such as packet size information, the service used, and a flag indicating the link status are the most important characteristics for detecting cyber attacks. In addition, it is important to examine time-based traffic characteristics for cyber attack detectors, such as details about the percentage of connections with services other than current connections within the last 2 seconds. It is important to study content features to detect R2L and U2R attacks.

5. CONCLUSIONS AND FUTURE WORKS

5.1 CONCLUSION

Because of the colossal development in digital assaults, there is a necessity of a powerful interruption recognition framework to ensure the information and the organization. This proposition is a work towards the advancement of a critical interruption model. First and foremost, we have explored more than 30 papers on interruption location methods from 2009 to 2019. We tracked down that a portion of the discovery strategies, for example, AI, profound learning, and blockchain innovation, assume an imperative part in building these life-saving frameworks. A writing audit furnishes foundation on these strategies with their applications and impediments in the space of interruption discovery. This was followed by a review of over 30 studies and the development of three classifiers for an interruption location model using the NSL-KDD dataset.

The results indicate the significance of selecting a number of appropriate groups. Although the particular cyber attacks that you are exposed to are fascinating to know, The wide range of classification groups reduces the accuracy of all machine learning algorithms. Cyber attack types, including DoS, Poking, R2L and U2R, are the best categories for classification. to improve the likelihood of making sound choices about security configuration. This categorization of videos helps you to spot new threats and improves the information about cyber attacks.

Basic functions such as packet size information, the service used, and a flag indicating the link status are the most important characteristics for detecting cyber attacks. In addition, it is important to examine time-based traffic characteristics for cyber attack detectors, such as details about the percentage of connections with services other than current connections within the last 2 seconds. It is important to study content features to detect R2L and U2R attacks.

5.2 FUTURE WORKS

A recommendation for future work is to compare real network traffic machine learning algorithms in various settings, to investigate whether they do better or worse with changing circumstances.

As part of our research, we looked at three types of classifiers: Random Forest, Naive Bayes, KNN, SVM, and decision trees. Next, we propose using computations such as K-Nearest Neighbor and other deep learning calculations for a larger number of components in the dataset to achieve high arrangement precision. Moreover, well known information mining methods like profound neural organizations and dbscan can be utilized to work on the outcomes later on.

REFERENCES

[1] Weizhi Meng, Elmar Wolfgang Tischhauser, Qingju Wang, Yu Wang, and Jin- guang Han. When intrusion detection meets blockchain technology: a review. Ieee Access, 6:10179–10188, 2018.

[2] Nikolaos Alexopoulos, Emmanouil Vasilomanolakis, Natalia Reka Ivanko, and Max Muhlhauser. Towards blockchain-based collaborative intrusion detection systems. In Gregorio D'Agostino and Antonio Scala, editors, Critical Informa- tion Infrastructures Security, pages 107–118, Cham, 2018. Springer International Publishing.

[3] Yang Xin, Lingshuang Kong, Zhi Liu, Yuling Chen, Yanmiao Li, Hongliang Zhu, Mingcheng Gao, Haixia Hou, and Chunhua Wang. Machine learning and deep learning methods for cybersecurity. IEEE Access, 6:35365–35381, 2018.

[4] Zibin Zheng, Shaoan Xie, Hongning Dai, Xiangping Chen, and Huaimin Wang. An overview of blockchain technology: Architecture, consensus, and future trends. In Big Data (BigData Congress), 2017 IEEE International Congress on, pages 557–564. IEEE, 2017.

[5] Chih-Fong Tsai, Yu-Feng Hsu, Chia-Ying Lin, and Wei-Yang Lin. Intrusion detection by machine learning: A review. Expert Systems with Applications, 36(10):11994–12000, 2009.

[6] Bo Dong and Xue Wang. Comparison deep learning method to traditional methods using for network intrusion detection. In 2016 8th IEEE International Conference on Communication Software and Networks (ICCSN), pages 581–585. IEEE, 2016.

[7] Ahmad Javaid, Quamar Niyaz, Weiqing Sun, and Mansoor Alam. A deep learning approach for network intrusion detection system. In Proceedings of the 9th EAI International Conference on Bio-inspired Information and Commu- nications Technologies (Formerly BIONETICS), BICT'15, pages 21–26, ICST, Brussels, Belgium,

Belgium, 2016. ICST (Institute for Computer Sciences, Social- Informatics and Telecommunications Engineering).

[8] Zheng Wang. Deep learning-based intrusion detection with adversaries.IEEE Access, 6:38367–38384, 2018.

[9] Uzair Bashir and Manzoor Chachoo. Intrusion detection and prevention system: Challenges & opportunities. In Computing for Sustainable Global Development (INDIACom), 2014 International Conference on, pages 806–809. IEEE, 2014.

[10] D.H. Lakshminarayana J. P. Philips, N. Tabrizi. A survey of intrusion detection techniques. IEEE International Conference on Machine Learning Application, 175(542):7–9, 2019.

[11] Andersson, Ross. 2008. Security engineering: A guide to Building De- pendable Distributed Systems. 2nd edition. U.S: John Wiley & Sons.

[12] De Ocampo, Frances Bemadette C.; Del Castillo, Trisha Mari L.; Gomez, Miguel Alberto N. 2013. Automated signature creator for a signature based intrusion detection system with network attack detec- tion capabilities (pancakes). International Journal of Cyber-Security and Digital Forensics, Jan, 2013, Vol.2(1).

[13] Witten, Ian H.; Frank, Eibe; Hall, Mark A. 2011. Data mining – Practical Machine Learning Tools and Techniques. Third Edition. USA: Elsevier Inc.

[14] Lincoln Laboratory, Massachusetts Institute of technology. 2014. Cyber Systems and Technology – DARPA Intrusion Detection Data Sets – 1998 dataset. URL http://www.ll.mit.edu/ideval/data/. Retrieved 2015-06-25.

[15] Wu, Xindong; Kumar, Vipin; Ross Quinlan, J.; Ghosh, Joydeep; Yang, Qiang; Motoda, Hiroshi; McLachlan, Geoffrey; Ng, Angus; Liu, Bing; Yu, Philip; Zhou, Zhi-Hua; Steinbach, Michael; Hand, David; Steinberg, Dan. 2008. Top 10 algorithms in data mining. Knowledge and Information Systems, 2008, Vol.14.

[16] Karen Scarfone and Peter Mell. Guide to intrusion detection and prevention systems (idps). NIST special publication, 800(2007):94, 2007.

[17] Steven R Snapp, James Brentano, Gihan V Dias, Terrance L Goan, L Todd Heberlein, Che-Lin Ho, Karl N Levitt, Biswanath Mukherjee, Stephen E Smaha, Tim Grance, et al. Dids (distributed intrusion detection system)-motivation, ar- chitecture, and an early prototype. In Proceedings of the 14th national computer security conference, volume 1, pages 167–176. Washington, DC, 1991.

[18] Vinod Yegneswaran, Paul Barford, and Somesh Jha. Global intrusion detection in the domino overlay system. In NDSS, 2004.

[19] Ryan Huebsch, Brent Chun, Joseph M Hellerstein, Boon Thau Loo, Petros Ma- niatis, Timothy Roscoe, Scott Shenker, Ion Stoica, and Aydan R Yumerefendi. The architecture of pier: an internet-scale query processor 2005.

[20] Paul Brutch and Calvin Ko. Challenges in intrusion detection for wireless ad-hoc networks. In null, page 368. IEEE, 2003.

[21] Arthur L Samuel. “Some studies in machine learning using the gameof checkers.” In:IBM Journal of research and development3.3 (1959),pp. 210–229.

[22] Andres Munoz. Machine Learning and Optimization 2014.

[23] T.M. Mitchell. Machine Learning. McGraw-Hill international editions- computer science series. McGraw-Hill Education, 1997.ISBN:9780070428072.URL:https : / / books . google . no / books ? id =xOGAngEACAAJ.

[24] AndrewNg.MachineLearning.URL: http://openclassroom.stanford.edu /main folder/ course page. php? course=machine learning.

[25] Michael Negnevitsky. Artificial intelligence: a guide to intelligent systems. Pearson Education, 2005.

[26] Ian H Witten and Eibe Frank.Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, 2005.

[27] Marti A. Hearst et al. “Support vector machines.” In: Intelligent Systems and their Applications, IEEE13.4 (1998), 18bibrangedash 28.

[28] Leif E Peterson. “K-nearest neighbor.” In:Scholarpedia4.2 (2009),p. 1883.

[29] Padraig Cunningham and Sarah Jane Delany. “k-Nearest neighbour classifiers.” In:Multiple Classifier Systems(2007), pp. 1–17.

[30] Yihua Liao and V Rao Vemuri. “Use of k-nearest neighbor classifierfor intrusion detection.” In: Computers & Security21.5 (2002),pp. 439–448.[22]KDD Cup 1999 Data.URL:http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[31] Maoguo Gong et al. “An efficient negative selection algorithm withfurther training for anomaly detection.” In: Knowledge - Based Systems30 (2012), pp. 185–191.

[32] Terry Brugge UC Davis.KDD Cup ’99 dataset considered harmful.URL:http://www.bruggerink.com/~zow/gradschool/kddcup99harmful.html.

[33] Matthew V Mahoney and Philip K Chan. “An analysis of the 1999DARPA/Lincoln Laboratory evaluation data for network anomaly detection.” In: Recent Advances in Intrusion Detection. Springer. 2003,220bibrangedash 237.

[34] Mahbod Tavallaee et al. “A detailed analysis of the KDD CUP 99 dataset.” In: Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications 2009. 2009.

[35] Samir Kant Sahu, Saumendra Sarangi, and Sanjaya Kumar Jena."A detail analysis on intrusion detection datasets." In: Advance Computing Conference (IACC), 2014 IEEE International. IEEE. 2014,1348bibrangedash 1353.

[36] Hadi Sarvari and Mohammad Mehdi Keikha. Improving the accuracy of intrusion detection systems by using the combination of machine learning approaches. In 2010 international conference of soft computing and pattern recognition, pages 334–337. IEEE, 2010.

[37] Phurivit Sangkatsanee, Naruemon Wattanapongsakorn, and Chalermpol Charn-sripinyo.Practical real-time intrusion detection using machine learning approaches. Computer Communications, 34(18):2227–2235, 2011.

[38] T Poongothai and K Duraiswamy. Intrusion detection in mobile adhoc networks using machine learning approach. In International Conference on Information Communication and Embedded Systems (ICICES2014), pages 1–5. IEEE, 2014.

[39] David Endler. Intrusion detection. applying machine learning to solaris audit data. In Proceedings 14th Annual Computer Security Applications Conference (Cat. No. 98EX217), pages 268–279. IEEE, 1998.

[40] Arkadiusz Warzyn´ski and Grzegorz Ko~laczek. Intrusion detection systems vul-nerability on adversarial examples. In 2018 Innovations in Intelligent Systems and Applications (INISTA), pages 1–4. IEEE, 2018.

[41] Deyban Perez, Miguel A Astor, David Perez Abreu, and Eugenio Scalise. Intru- sion detection in computer networks using hybrid machine learning techniques. In 2017 XLIII Latin American Computer Conference (CLEI), pages 1–10. IEEE, 2017.

[42] Bisyron Wahyudi, Kalamullah Ramli, and Hendri Murfi. Implementation and analysis of combined machine learning method for intrusion detection system. International Journal of Communication Networks and Information Security, 10(2):295–304, 2018.

[43] MA Jabbar, Rajanikanth Aluvalu, and S Sai Satyanarayana Reddy. Intrusion detection system using bayesian network and feature subset selection. In 2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), pages 1–5. IEEE, 2017.

[44] Tahir Mehmood and Helmi B Md Rais. Machine learning algorithms in context of intrusion detection. In 2016 3rd International Conference on Computer and Information Sciences (ICCOINS), pages 369–373. IEEE, 2016.

[45] Kwangjo Kim and Muhamad Erza Aminanto. Deep learning in intrusion de- tection perspective: Overview and further challenges. In 2017 International Workshop on Big Data and Information Security (IWBIS), pages 5–10. IEEE, 2017.

[46] Snehal A Mulay, PR Devale, and GV Garje. "Intrusion detection system using support vector machine and decision tree." In: International Journal of Computer Applications3.3 (2010), 40bibrangedash 43

[47] Xin Xu and Xuening Wang. "An adaptive network intrusion detection method based on PCA and support vector machines." In: Advanced Data Mining and Applications. Springer, 2005, 696bibrangedash 703.

[48] Nabila Farnaaz and MA Jabbar. Random forest modeling for network intrusion detection system. Procedia Computer Science, 89:213–217, 2016

[49] Mrutyunjaya Panda, Ajith Abraham, and Manas Ranjan Patra. A hybrid intelli-gent approach for network intrusion detection. Procedia Engineering, 30:1–9, 2012.

[50] Preeti Aggarwal and Sudhir Kumar Sharma. Analysis of kdd dataset attributes- class wise for intrusion detection. Procedia Computer Science, 57:842–851, 2015.

[51] Himadri Chauhan, Vipin Kumar, Sumit Pundir, and Emmanuel S Pilli. A com- parative study of classification techniques for intrusion detection. In 2013 Inter- national Symposium on Computational and Business Intelligence, pages 40–43. IEEE, 2013.

[52] S Revathi and A Malathi. A detailed analysis on nsl-kdd dataset using various machine learning techniques for intrusion detection. International Journal of Engineering Research & Technology (IJERT), 2(12):1848–1853, 2013.

[53] Muhammad Shakil Pervez and Dewan Md Farid. Feature selection and intrusion classification in nsl-kdd cup 99 dataset employing svms. In The 8th International Conference on Software, Knowledge, Information Management and Applications (SKIMA 2014), pages 1–6. IEEE, 2014.

[54] S Revathi and A Malathi. A detailed analysis on nsl-kdd dataset using various machine learning techniques for intrusion detection. International Journal of Engineering Research & Technology (IJERT), 2(12):1848–1853, 2013.

[55] Alkadi, O., Moustafa, N., & Turnbull, B. (2020). A review of intrusion detection and blockchain applications in the cloud: Approaches, challenges and solutions. *IEEE Access*, *8*, 104893-104917.

[56] Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009, July). A detailed analysis of the KDD CUP 99 data set. In 2009 IEEE symposium on computational intelligence for security and defense applications (pp. 1-6). IEEE.

[57] Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., ... & Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. Journal of big Data, 8(1), 1-74.

[58] KDD 99 dataset, Accessed December 2015, http://kdd.ics.uci.edu/databases/kddcup99

[59] Kanimozhi, V., & Jacob, T. P. (2019, April). Artificial intelligence based network intrusion detection with hyper-parameter optimization tuning on the realistic cyber dataset CSE-CIC-IDS2018 using cloud computing. In 2019 international conference on communication and signal processing (ICCSP) (pp. 0033-0036)

[60] Panigrahi, R., & Borah, S. (2018). A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems. International Journal of Engineering & Technology, 7(3.24), 479-482.

[61] Revathi, S., & Malathi, A. (2013). A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection. International Journal of Engineering Research & Technology (IJERT), 2(12), 1848-1853.