

**IMPROVING THE ACCURACY OF
CONVOLUTIONAL NEURAL NETWORK BY
APPLYING RANDOM SAMPLING METHODS ON
NSL-KDD DATASET**

AYE THAWTA SANN

M. C. Sc.

SEPTEMBER 2022

**IMPROVING THE ACCURACY OF
CONVOLUTIONAL NEURAL NETWORK BY
APPLYING RANDOM SAMPLING METHODS ON
NSL-KDD DATASET**

By

AYE THAWTA SANN

B. C. Sc.

**A Dissertation Submitted in Partial Fulfillment of the
Requirements for the Degree of**

**Master of Computer Science
(M. C. Sc.)**

**University of Computer Studies, Yangon
September 2022**

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my sincere thanks to all my teachers who gave me many valuable advice and information. I am also grateful to all respectable people who directly or indirectly contributed towards the success of this thesis.

I would like to express my respectful thanks to **Dr. Mie Mie Khin**, Rector of the University of Computer Studies, Yangon for her kind permission to conduct this thesis.

I would like to express my gratitude and appreciation to Course Coordinators, **Dr. Si Si Mar Win** and **Dr. Tinzar Thaw**, Professors of Faculty of Computer Science, University of Computer Studies, who offer me an unrestricted support and valuable and timely advice and suggestions for the completion of this work.

I would also like to offer my deep and sincere gratitude to my supervisor, **Dr. Zin Thu Thu Myint**, Associate Professor, Faculty of Information Science, the University of Computer Studies, Yangon, her effort, time in reading and patience to help me in accomplishing this paper. It was a great privilege and honor to work and study under her guidance.

I also thank **Daw Aye Aye Khine**, Associate Professor & Head, Department of English University of Computer Studies, Yangon, for her kind suggestion in writing my thesis documentation.

I would like to thank all my teachers for their motivated, encouragement and recommending the thesis.

Furthermore, I am extremely grateful to my companions who have given me their precious ideas and invaluable knowledge throughout this thesis. Finally, I am extending my heartfelt thanks to my family for their encouragement and support to accomplish this work.

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

.....

Date

.....

Aye Thawta Sann

ABSTRACT

An Intrusion Detection System (IDS) acts as a cyber security system which monitors and detects any security threats for software and hardware running on the network. Although there have many existing IDS but still face challenges in improving accuracy in detecting security vulnerabilities, not enough methods to reduce the level of alertness and detecting intrusion attacks. Machine learning methods can detect data from past experience and differentiate normal and abnormal data. In this system, the Convolutional Neural Network (CNN) in deep learning method is used for solving the problem of identifying intrusion in a network. NSL – KDD dataset is used to train the data with the CNN algorithm. The system implementation is performed for balanced and unbalanced nature of NSL – KDD dataset. The analysis of evaluation results describes the achievement of the proposed system with the accuracy of 83% in balanced dataset and 80% in unbalanced dataset. The proposed system is implemented by Python programming language on Tensorflow platform.

TABLE OF CONTENTS

	Page
ACKNOWLEDGEMENTS	i
STATEMENT OF ORIGINALITY	ii
ABSTRACT	iii
TABLE OF CONTENTS	iv
LIST OF FIGURES	vi
LIST OF TABLES	vii
LIST OF EQUATIONS	viii
CHAPTER 1	INTRODUCTION
1.1	Related Works 2
1.2	Objectives of the Thesis 4
1.3	Organization of the Thesis 5
CHAPTER 2	THEORETICAL BACKGROUND
2.1	Intrusion Detection System 6
2.1.1	Signature-based Intrusion Detection System 7
2.1.2	Network-Based Intrusion Detection System 8
2.1.3	Anomaly-Based Intrusion Detection System 9
2.1.4	Host-Based Intrusion Detection System 9
2.2	Cyber Attacks 10
2.2.1	Forms of Cyber Threats 10
2.3	Data Mining for Intrusion Detection and Prevention 12
2.4	Machine Learning 13
2.4.1	Fundamental Aspects of Machine Learning 14
2.5	Types of Machine Learning Algorithms 16
2.5.1	Supervised Learning 16

	2.5.2	Unsupervised Learning	18
	2.5.3	Semi-supervised Learning	18
	2.5.4	Reinforcement Learning	19
	2.6	Deep Learning	19
	2.6.1	Recurrent Neural Networks	20
	2.6.2	Long Short-Term Memory	21
	2.6.3	Convolutional Neural Networks	23
CHAPTER 3	DESIGN OF THE SYSTEM		
	3.1	Overview of The Proposed System	26
	3.1.1	Random Sampling	28
	3.1.2	Undersampling	28
	3.1.3	Oversampling	29
	3.1.4	Convolutional Neural Networks	29
	3.2	Evaluation of the Performance of Methods	31
	3.2.1	Confusion Matrix Performance	31
CHAPTER 4	IMPLEMENTATION OF THE SYSTEM		
	4.1	Experimental Setup	35
	4.2	Implementation of the System	35
	4.3	Evaluation of Experimental Results	43
CHAPTER 5	CONCLUSION		
	5.1	Advantages	46
	5.2	Limitations and Further Extensions	47
		AUTHOR'S PUBLICATIONS	48
		REFERENCES	49

LIST OF FIGURES

	Page
Figure 2.1 Taxonomy of Intrusion Detection System	7
Figure 2.2 Taxonomy of Supervised Learning	17
Figure 3.1 System Architecture	25
Figure 4.1 NSL-KDD Training Dataset	36
Figure 4.2 Anaconda Activating	37
Figure 4.3 Changing Directory	37
Figure 4.4 Running Main Program	38
Figure 4.5 Main Page	38
Figure 4.6 Next Page	39
Figure 4.7 Loading Test File	39
Figure 4.8 Loading Test File and Input Unbalanced Model	40
Figure 4.9 Results of Unbalanced Model	40
Figure 4.10 Loading Test File and Input Balanced Undersampling Model	41
Figure 4.11 Results of Balanced Undersampling Model	41
Figure 4.12 Loading Test File and Input Balanced Oversampling Model	42
Figure 4.13 Results of Balanced Oversampling Model	42
Figure 4.14 Performance Results for Abnormal	43
Figure 4.15 Performance Results of Normal	43
Figure 4.16 Performance Comparison of Unbalanced and Balanced on Abnormal	44
Figure 4.17 Performance Comparison of Unbalanced and Balanced on Normal	44
Figure 4.18 Accuracy Results of Unbalanced and Balanced	45
Figure 4.19 Accuracy Comparisons of Unbalanced and Balanced	45

LIST OF TABLES

		Page
Table 3.1	Different Features	26
Table 3.2	Features Under Various Categories	27
Table 3.3	Confusion Matrix	32

LIST OF EQUATIONS

	Pages
Equation 2.1 Equation for Recurrence Relation	20
Equation 2.2 Equation for Updated Hidden State	21
Equation 2.3 Equation for Output Vector	21
Equation 2.4 Equation for Forget Gate	22
Equation 2.5 Equation for Sigmoid Layer	22
Equation 2.6 Equation for New Candidate Vector	22
Equation 2.7 Equation for Store Gate	22
Equation 2.8 Equation for Sigmoid Layer	23
Equation 2.9 Equation for Hidden State	23
Equation 3.1 Equation for Rectified Linear Unit	30
Equation 3.2 Equation for SoftMax	31
Equation 3.3 Equation for Accuracy	33
Equation 3.4 Equation for Precision	33
Equation 3.5 Equation for Recall	33
Equation 3.6 Equation for F-measure	34

CHAPTER 1

INTRODUCTION

The worldwide business and economic development are directly relating with the internet and enterprise networks. Moreover, cyber-attacks are popular that is a potential security issue. Due to this issue, network security specialists and technicians are giving attention for the specification of network attacks. The private and government organizations need the answers providing the stability in performance for the prevention of information assets holding from various authorized attempts in the detection and prevention of intrusions. The intrusion detection system is an application for monitoring and classification of network flows for the decision in which they are the abnormal activities that threaten the security of information systems or normal activity that could frequently occurs in a network.

Network Security can be monitored by administrators in network and security officers to provide a protected environment for user accounts, their online resources, personal details and passwords. IDS can be divided into two types by their approach:

- 1) Misuse Detection: It always uses signatures from previous data to detect intrusion; it may not be effective for new types of attacks.
- 2) Anomaly Detection: It uses an unusual pattern to detect attacks.

After increasing the cyber-crime types, the anomaly detection system becomes better than the misuse detection system for building a network intrusion detection system. An anomaly detection system is more suitable for detected unknown attacks.

Deep learning is a kind of artificial intelligence has occurred applied in the fields of pattern recognition and classification. Many information-operating layers in the hierarchical design is utilized by deep learning. Moreover, deep learning is a popular research referring that various deep neural networks like recurrent neural networks, autoencoders, convolutional neural networks, and deep belief nets containing denoising, variational autoencoders, sparse, and contractive have been observed and produced. Convolutional neural networks are efficient for complex jobs like identification of objects and faces and computing self-controlling cars in various deep learning methods. So, various researchers have tried for utilizing convolutional neural networks for handling the puzzle in intrusion detection system. DNNs contain an attractive essential intrusion detection function called learning by training, to

deduce new information to provide a decision, this makes DNNs distinguishes from all the traditional programming techniques and make it an expert system.

Convolution neural network is a kind of deep learning and is popular for the identification in complex assets with unusual patterns. The automated methods are provided by deep learning in order to extract deep features. The accurate data representation is supported permitting for the better model generation. The application of deep learning is provided for the intrusion detection systems. As the rise of the CUP 99 dataset in 1999, various researchers have attempted in deep learning intrusion detection system. The KDD dataset is most applied for the implementation of intrusion detection system. The four groups of attacks are probe, remote to local, denial of service, and user to root in KDD. Deep learning methods are depending upon the function, structure, and the operation of biological neurons in the nervous system. In the comparison with machine learning methods, deep learning relates with big data by various attributes. Moreover, huge amount of data with feature vector in NSL-KDD dataset, a military network environment generated with a group of TCP/IP data and attack for the extension of air force local area network. This dataset contains 41 network characteristics, and each characteristic divide into one of four types of assets. The CSE-CIC-IDS-2018 and CIC-IDS-2017 datasets were developed with the canadian institute of cyber security applying intrusion traffic classification methods. They are classified to 7 attack kinds that represents the current asset event. The NSL-KDD dataset contains 41 groups that is more than the CIC-IDS-2018 dataset. The NSL-KDD dataset is a network dataset which is the latest option of previous dataset, KDD CUP 99. This NSL-KDD dataset is generated for solving the issues of KDD CUP 99. Convolutional neural network is applied for the data classification. An intrusion detection system is developed with convolutional neural networks for unbalanced dataset. Then, an intrusion detection system is developed with convolutional neural networks for balanced dataset by applying random sampling techniques: undersampling and oversampling.

1.1 Related Works

Various researchers are trying in the introduction of intrusion detection system by deep learning. In this paper [1], an improved intrusion detection system according to hybrid feature selection and the ensemble of two-step classification was proposed.

This feature selection approach contains: ant colony, genetic, and particle swarm optimization methods. These methods are applied for the reducing of feature in training events. In this system, UNSW-NB15, and NSL-KDD are used as input data sources. The selection of features is done according to the efficiency of reduced error pruning tree. After, the ensemble of two-step classification: bagging and rotation forest is performed. This system achieved the sensitivity of 86.8%, the detection rate of 88%, and the accuracy of 85.8% in the NSL-KDD dataset, and the state of the art of accuracy, detection rate, and sensitivity in the UNSW-NB15 dataset. However, this paper did not consider effective classification for balanced dataset in intrusion detection system.

The authors introduced a comparative analysis of the intrusion detection system efficiency with random forest in terms of false alarm rate and accuracy [10]. In this system, UNSW-NB15, GPRS, and NSL-KDD are used as input data sources for the implementation. The consideration in the ensemble of various tree types are done whereas another most optimum parameters are provided by applying grid. The system implementation evaluated that random forest is the best outperforming for intrusion detection system as the significant performance of this classifier in terms of k-cross validation with the other ensembles of naïve bayes and neural network, and naïve bayes and random forest. However, this paper did not consider unbalanced data nature for achieving desired accuracy in intrusion detection system.

The authors proposed the deep learning approach, nonsymmetric deep autoencoder for unsupervised feature learning in intrusion detection system [11]. This system is built based on random forest method and stacked nonsymmetric deep autoencoders. This system utilized NSL-KDD and KDD Cup '99 as input data sources for the performance evaluation. The implementation of this system is done with TensorFlow on graphics processing unit (GPU). This system obtained the promising accuracy for intrusion detection system. The system implementation evaluated that this approach achieved the higher precision, recall, and accuracy and reduced the time of training. The comparison of mainstream DBN method and stacked nonsymmetric deep autoencoder was performed. The comparison results showed that this system improved the accuracy by 5% and reduced the training time by 98.81%.

The authors proposed the combination of convolutional neural network with the TensorFlow like cognitive computing method in intrusion detection system [7].

NSL_KDD dataset is used as input data source for this system. The presentation of network traffic according to the connections of TCP/IP is performed and the training of this technique is done with the signatures of known attack. The performance evaluation showed that this system achieved the promising precision by 99.82%, the F1-score by 96.34%, the accuracy by 98.92%, and the recall by 92.34%. This approach performed the integration of existing system as big data and TensorFlow by providing scalability for huge amount of data.

This paper presented an approach for intrusion detection system according to temporal convolutional neural network establishing the best detection that has the ability for solving huge amount and high dimension data [8]. Moreover, this system can be applied for not only host-based intrusion detection system but also network-based intrusion detection system. The experimental results proved that the proposed approach performed better than other detection techniques by lower false positive rate and higher accuracy. This proposed approach promised the desired accuracy by 90.5% for NSL-KDD dataset with the requirement of only 543KB storage. Moreover, the proposed system provided the decisions to data preprocessing for the systems that require complexity reduction and the efficiency improvement.

According to the knowledge that gained from the previous related works, it is found that machine learning model is widely used in intrusion detection system. And it is also found that intrusion detection system with deep learning provides accurate results than using machine learning model. Therefore, intrusion detection system is done with convolutional neural network deep learning model for unbalanced dataset and balanced dataset with undersampling and oversampling.

1.2 Objectives of the Thesis

The main objectives of the thesis are:

- To develop the intrusion detection system using convolutional neural network model
- To detect malicious patterns over the network
- To improve the accuracy of the intrusion detection system
- To prevent different types of attacks by using the intrusion detection system
- To protect the network from unauthorized users

1.3 Organization of the Thesis

This thesis contains five chapters.

Chapter 1 presents the introduction in intrusion detection system, the related works, the objectives and the organization of the thesis.

Chapter 2 describes the theoretical background concerning with intrusion detection system, machine learning, and deep learning classification algorithms in detail.

Chapter 3 presents the design of the proposed system describing system flow, the overall design of system, convolutional neural network-based intrusion detection system, and performance evaluation used.

Chapter 4 presents the implementation of the proposed system including the experimental setup, the implementation of the system by convolutional neural network with unbalanced dataset and balanced dataset and the experimental result.

Finally, Chapter 5 concludes this study with further extensions of the proposed system.

CHAPTER 2

THEORETICAL BACKGROUND

As the development in information technology at various fields, supporting for achieving the reliability to these systems has become the more crucial. Therefore, the attacks' complexity has the increment due to this growth. So, the inefficiency of traditional security services as signature-based intrusion detection systems has degraded in the unknown attack detection. In order to perform the capturing and monitoring of dangerous traffic, intrusion detection systems (IDS) are applied. Conventional signature-based intrusion detection systems have the ability on taking actions against known threats utilizing only predefined or custom rule sets. Conventional intrusion detection systems cannot detect unspecified attacks with no correspondence of static signature. The incoming behavior-based unknown actions detection methods like deep learning support with signature-based intrusion detection systems for performance increment in the unspecified attacks detection and false negative and false positive rates elimination. As the learning pattern by deep learning methods, the possible increment in specification rate of unknown actions with the estimation of normal or attack. They have the ability for automatic specification procedure with no manual specification for false alarms elimination.

2.1 Intrusion Detection System

The intrusion detection system is a software application for analysis and monitoring computer system for intrusion detection before the serious corruption to the network system. The efficient security system possesses an intrusion detection system like core thing as the detection and recognition of attacks before the operation will save the system against substantial service loss and downtime. Various development in intrusion detection started from 1980 by Anderson's paper that initiated the primary concepts of computer threats surveillance and monitoring. Dorothy E. Denning introduced the real-time intrusion detection system in 1986. This system specified various security violations from outside the system breaking-in attempts and inside the system hateful cases and data misuse occurrence. The rule-based pattern matching was applied in where the maintenance of no abnormal pattern records by safe library and the comparison of usage pattern audition is performed for

alerting any dangerous patterns. The execution commands, logins and device and file attempts are the standard functions monitored on the targeted system. Various functions like trojan horses, leakage, abuse of unauthorized users, virus and masquerading attempts can be detected by the intrusion detection system.

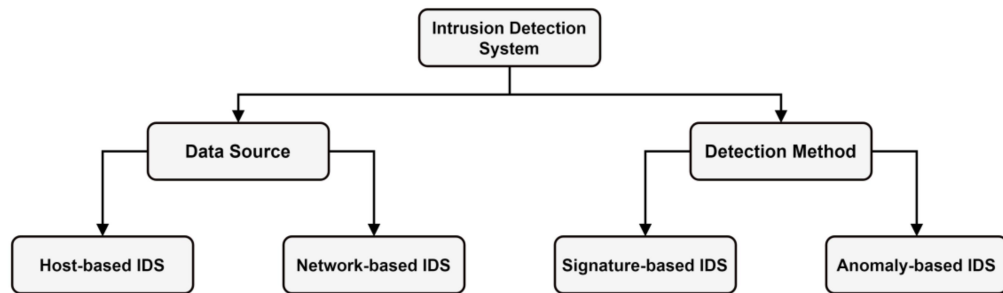


Figure 2.1 Taxonomy of Intrusion Detection System

The evaluation of the intrusion detection system is divided into various classes depending upon its ability and nature. The intrusion detection system is generally divided into network-based and host-based based on the data resources. The classification of anomaly-based and signature-based intrusion detection system is done according to the approach of intrusion detection system as shown in Figure 2.1.

2.1.1 Signature-based Intrusion Detection System

For searching a known attack, the conventional signature-based intrusion detection systems perform mostly based on pattern matching approaches. This method can be called as abuse detection or knowledge-based detection. The learning for misuse patterns or the attack is performed based on last actions in this method. After that, the detection of same patterns is taken by the gained knowledge from the learned patterns. The signatures are the misuse patterns and learned attack. For specifying the patterns of attacks, user-defined and pre-defined signatures are utilized as the matching of unknown network traffic with the corresponding signatures, the notification is produced by the signature-based intrusion detection system. The known signatures of known attack types are contained in the signature-based intrusion detection system for the comparison of current actions and it notifies as the occurrence of matching. In addition, the updating of signature database should be performed to most recent attacks for the preservation of detection effectiveness. This

approach is very suitable and it provides higher detection accuracy as the occurrence of last known attacks.

However, this approach fails in the defining popular attacks which spanning many packets. The suitable last packets are maintained by the intrusion detection system and make the communication between ordered packets is provided for the intrusion specification [3]. Various methods for handling the issues of the signature-based intrusion detection system presented which the creation of signatures like state machines formal language semantic events or string patterns. In addition, the signature-based intrusion detection method suffers the detection inefficiency as the occurrence of recent sophisticated cyber-attacks or zero- day attacks applying abnormal patterns in the bypass security systems. Furthermore, the anomaly-based intrusion detection systems have the ability for normal pattern modelling, therefore for the specification of deviations in normal traffic. The comparison of the signature-based intrusion detection with the anomaly-based intrusion detection systems is performed to prove the effectiveness of the anomaly-based intrusion detection systems in unknown attacks detection. In order to specify known attacks, the signature-based intrusion detection is principally applied. The computing is performed with a pre-programmed list of known attacks and its indicators of compromise (IOCs). The indicators of compromise may be an exact pattern which normally precedes the unauthorized network attack, malicious domains, known byte sequences, the content of email subject headings or file hashes. The comparison of those packets to the database of known indicators of compromise or attack signatures is performed for alerting many unknown patterns when the network traversal of the packets is monitored by the signature-based intrusion detection system.

2.1.2 Network-Based Intrusion Detection System

The monitoring of network actions and the analysis of unknown actions in the data traffic are done by the network-based intrusion detection systems. The header information of new network packets and the content are the principal resource of the examination for the network-based intrusion detection systems. The network-based intrusion detection systems exist at the principal points in a network infrastructure which are obtaining huge amount of external traffic. The network-based intrusion detection system is efficient for the monitoring of large-sized network as the

worldwide network protocols: UDP/IP and TCP/IP standardization, they own higher portability. Their development is done no constraints in various network device and manufacturer. As the last deliberation with the huge arising of the internet, every device is at one type or another is connected to an external network for the service delivery. Various software at one host performs the self-sharing of data by various external APIs to process data. As the development of serverless and cloud computing architectures, conventional hardware-based computing is becoming obsolete. The external hardware provisioners performs the connection with the edge devices for enabling access to computing services.

2.1.3 Anomaly-Based Intrusion Detection System

For anomaly detection system, the behavior of network is the one kind of important parameter. If this behavior is fall in some of the pre-collected behavior, then the input transaction is determined as accepted or not by triggering the alert coming from the detection system [4]. Anomaly-based Intrusion detection system can be established by analyzing on normal usage activities or patterns which are deviated from original usage pattern and define these deviated patterns as a possible intrusion pattern. So, the methods for anomaly detection can analyze the transaction patterns which are not accessible to ordinary users. In this case, it needs to consider the performance of computer and make report if abnormal trend or behavior occurs in the network. The advantage of anomaly based detection is that it is strength to identify novel attacks. This technique also has some disadvantage that the training data of this system may contain noise. It may increases the rate of false alarms while an intrusive process may take the missed detections process. And also, anomaly-based systems may face with difficulty in order to classify or name each kind of attack. It does not work properly when the incoming attack is completely new. When trying to overcome this problem, it needs to update itself and consume a lot of time.

2.1.4 Host-Based Intrusion Detection System

Host-based intrusion detection systems are applied for the detection on abuse and anomalies in the internals of a specific host installed. Host-based intrusion detection systems was the primary intrusion detection system developed for the

execution on the mainframe computers in where the rare and occasional external communication happens. The used input data source is for the derivation of the deviation patterns collected with the audit trails: the operating system mechanism. Other resources like filesystem data, other processing data, and log files created with a host are utilized by host-based intrusion detection systems. The host-based intrusion detection system is needed for the design of operating system and the machine that restricts the effectiveness due to the lack in cross-platform as various kinds of operating system and vendors. The development cost in security infrastructure is increased by each iteration in manufacturer design. Host-based intrusion detection systems need to be upgraded providing it economically unfeasible. They are not developed for working with network traffic. They are limited in the scope of protecting the system that is connected to an external network interface.

2.2 Cyber Attacks

The cyber-attack tries for gaining unknown attempts to an information asset with the intent for disruption, steal or modifying the data asset in computer security terminology. Such unknown actions may be wide-ranging forms of cyberterrorism or cyberwarfare applying computer networks. Based on the 2017 word threat assessment report by US DNI, many countries view cyber abilities as a path for projecting their global influence and are continuing for funding and developing their cyber arsenal.

2.2.1 Forms of Cyber Threats

- DoS: A denial-of-service (DoS) is a common cyber threat type which refers to the condition at where the attackers propose for the traffic overflow on a network or host infrastructure for providing services and sources inaccessible to authorized users. The theft in data assets is not led by the attack however costs the target victim association money and time resources. The physical harmfulness in systems can be occurred by the continuous crashing and debilitating of services when they are solving other critical infrastructures and control networks [9]. Distributed-denial-of-service (DDoS) attack is a variety of DoS attack that applies a distributed system known a botnet to orchestrate the cyber-attack, increasing its overall potential and severity.

- R2U: A remote-to-user is a cyber-attack in where the access is gained by the attackers like a local user for infiltrating the organization by a remote machine. To search various weak points, unknown packets to the local user's target host are sent by the attackers which can perform the attacker for exploiting the local user's existing privileges. This weak point is a overdue to more corruptive User-to-Root (U2R) attacks.
- U2R: The footing at hosting machine is first gained by the attacker like an owner by restricted rights in a User-to-Root attack and it proceeds for the rights escalating utilizing many approaches for becoming the root user. The attacker has the ability for providing many user accounts and creating backdoors for easily re-entering the association's network and detection. The attacker access to every command lists in the system and the management of data assets in the file system are provided by the root rights based on their directives.
- Port Scanning: This cyber-attack is an investigation type approach applied by attackers for scanning all ports of host. The sending and receiving all transmission information is applying many ports to defined services. All the information on analysis and redirection is gained by the attackers for obtaining further entrap the targeted user with cyber-attacks forms applying port-scanning. Other weak points can be detected the attackers with the ports' mapping in order to investigate and obtain remote access.
- Backdoor Attacks: This is malware attacks purposed for providing attackers unlimited access to the database and server of the organization. Dislike with another types of access, discreet are remained by backdoors, and this are applied by the attackers for stealing huge amount of financial and competitive data when remaining unspecified. Depending upon on the state of malware report 2019, the backdoors performs the continuous process for potential attack matrix in cybercrime on all the business entities and government, by staggering 173% increment in their detection rate in business organizations.
- Fuzzers: This kind of threat focus on error or fuzz outside the general operating host server with delivering it many faculty commands by brute force mode, that will occur in the systems for discharging many codes of error. The objective is not for falling in the system however creating the error logs which can be analyzed by the attackers for finding the locations and resources can be

utilized for continuing unknown actions for searching weak points. Conventional fuzzer methods are re-developed applying machine learning techniques for various test events and seed files creation and covering a huge place of code for finding additional risks efficiently.

- Computer Worm: This is an unauthorized software for self-replicating themselves in order to distribute to another systems and networks in their vacuum. The operation of computer depends upon the existing risks and backdoor investigates for remaining hidden as continuing on their onslaught of the entire network. The main direction is for the gradual draining the resources in the system and congesting the network infrastructure. Various worms' types have payloads directed at stealing precious data. Known worms are applied for achieving access to the system and escalating the rights for continuing with another cyber-attacks.

2.3 Data Mining for Intrusion Detection and Prevention

The information security in the computer systems is still face with continual risk. The daily internet usage over the world is extensively growth and so, the availability of tricks and tools for attacking or intruding networks are increasing. This case promotes the process of intrusion detection and prevention to play in important role in networked systems. An intrusion means any actions that threaten the integrity, confidentiality, or availability of a network resource. Both intrusion detection systems (IDS) and intrusion prevention systems (IPS) monitor the traffic in a network and, detect and remove the malicious activities. Data mining techniques can help an IDS and IPS for enhancing the performance of these systems. There are various ways for providing intrusion/detection system such as

- Signature-based and anomaly-based detection can be raised as new data mining algorithms for intrusion detection: Data. For signature-based detection system, training data are pre-collected and labeled each records as either “normal” or “abnormal (intrusion).” A classifier that can be derived from such models can detect known attack. The application area includes the system which applies association rule mining, cost-sensitive modeling and classification algorithms. For Anomaly-based detection system, It builds the

models by analyzing on normal behavior and detects the significant deviations from which behavior. The application area includes the system which apply classification algorithms, statistical approaches, clustering and outlier analysis. The techniques are scalable and efficiently handle the high volume of network data and high dimension data.

- Association, correlation, and discriminative pattern analysis can help to select and build the model of classifiers which can find relationships between the data on network data.
- Stream data analysis can detect the dynamic and transient nature of intrusions and malicious patterns. In some case, an event may be considered as malicious by analyzing the part of a sequence of this event whereas this event may be normal in original. For this reason, it is necessary to observe the frequently occurred sequences of event and it identifies outliers.
- Distributed data mining techniques can detect intrusions or attacks by launching from different locations and targeted to different destinations. These methods can analyze the data from several networks and detect the attacks come from distributed network.
- Visualization and querying tool can analyze the data by confirming with view and any anomalous patterns can be detected. These tools have the feature for viewing clusters, outliers, associations and discriminative patterns, These tools should also have a graphical user interface(GUI) that can make the security analysts

Computer systems are at continual risk of breaks in security. Data mining technology can be used to develop strong intrusion detection and prevention systems, which may employ signature-based or anomaly-based detection.

2.4 Machine Learning

Machine learning is a kind of artificial intelligence directing for enabling machines in order to execute their tasks with skillfulness with the programs building that learning from last experience. The principal factors of artificial intelligence are the capabilities for pursuing purposes and planning to next activities. The division of data to various groups and giving prediction on next information events according to last data are included in the possible actions. It is the expected appealing fact for the

manual building and the application of machine learning has increased among computer science in the most recent decade. Data is the most principal in machine learning and the application of the learning algorithm is done for achieving and studying properties or knowledge by the data. The amount of dataset has the actions on the forecasting and learning performance. In machine learning, the learning methods can be divided into taxonomy according to the expected result of the method.

2.4.1 Fundamental Aspects of Machine Learning

Machine learning is a kind of artificial intelligence in which the adaptable models are constructed by a given dataset with minimal human intervention. Machine learning is a group of approaches utilized for automatic detection of patterns in data and then applying the extracted patterns in order to forecast the next data or performing other types of decision-making tasks depending upon Murphy. A machine learning model may be descriptive when the aim is for gaining the knowledge from the data given or predictive when it is providing predictions to next situations or be both descriptive and predictive. The principle job of machine learning method is for investigating inference from the providing sample applying the statistical concepts on the construction of the mathematical models. Various facts are need to be decided in the design of machine learning method pointing out for gaining the good performance. Designing the machine learning contains the followings:

(i) Pre-processing

The primary stage of machine learning is preprocessing at where the raw data are required to preprocess and prepare by utilizing preprocessing methods according to some earlier system than feeding into the system. For instance, the conversion of document images into binary format is performed before the putting with various kinds of optical character recognition or layout analysis. If the machine learning platform does not solve incomplete datasets, the input raw data may be incomplete as missing values in other events. For this condition, the suitable developed preprocessing stage can solve by filling this gap in the dataset with applying other optimal statistical methods or averaging. Many steps of preprocessing contain for removing outliers or noise in the dataset. So, the selection and providing usage of

suitable preprocessing approach may possess the potential effects on other steps done by machine learning.

(ii) Feature Selection

The processing of patterns is performed and the patterns divided are described by many measurement metrics known as features. The suitable features set must be selected for the pattern recognition. The selected features must provide the satisfaction specific facts such that they may be most efficient. With the irrelevant features of the input data removing, they reduce resource and memory consumption and computation time. The feature selection may need the prior knowledge on the domain of the problem however selecting the suitable features is risky job and it includes many implementations.

(iii) Model Selection

The accuracy of the classification depends upon the choice of model as many kinds of models may provide many estimations on various problem domains. The most accurate estimations suffer in forecasting improvement and higher classification rates. Not only selecting a model takes place in the performance evaluation but also the quality and amount of instance data are important for the deciding the accuracy of the classifier.

(iv) Training, Testing and Optimization

They are implemented with data sample after choosing the classification approach. Data training is the model training taken on a data subset. The generation of the model from the training step is done for the further decision and testing in a continuous testing stage by utilizing testing instance. The problem is that the operation time for each step may occur excessive according to the methods applied. Generally, many iterations happen during training and testing stages within the parameter optimization may need the significant operation needs. The answer to this problem is the efficient application of machine learning methods can be provided with using the parallelization in the system. As a consequence, the system is authorized for

applying various processors and large amount of processing machines within the same time. This is at which the aspects of distributed operating occur in the picture.

2.5 Types of Machine Learning Algorithms

Machine learning techniques are grouped into three main types: unsupervised learning, reinforcement learning, and supervised learning. In this section, the overview of supervised learning techniques is described and further various original techniques which will constructing for understanding more complex techniques in the field of deep learning.

2.5.1 Supervised Learning

Supervised learning is a model for the creation of the known outputs by a training instance of the original dataset (e.g. classification of automobile types on photos). The system receives both input data and output data in the first stage. The job of this procedure is to provide rules which provides the outcome of the provided input. Till the performance of the model is higher effective, the training phase must obey for continuing. This system must be ready for the associate classification to outcome objects hidden along the training step after the complement of training procedure. Generally, the classification procedure is basically accurate and quick.

In supervised learning approaches, the training of the model is done by a dataset including labelled instances. Therefore, this learning approaches point out the forecasting of the output values for the new data with dependencies and connection modelling among input features and labels in the dataset that may be forecasting after the training stage. Popular supervised learning approaches are bayesian statistics, lazy learning, support vector machines, bayesian networks, artificial neural network, nearest neighbor algorithm, gaussian process regression, decision trees, hidden markov model, boosting, ensembles classifiers, linear classifiers etc. The classification of supervised learning approaches into regression and classification [2].

- **Regression:** The aim of regression method is for the forecasting of the outcome of associated input. For example, in order to forecast the value of some product, as the price of a stock or the price of a house in a specified town. There have various elements that can be provided the forecasting by applying regression.

- **Classification:** The aim of classification method is for providing class assignments. This has the ability to forecast the outcome value and the data is classified into “classes”. For instance, the recognition of an automobile type in a photo, of today’s weather, and of the mail spam.

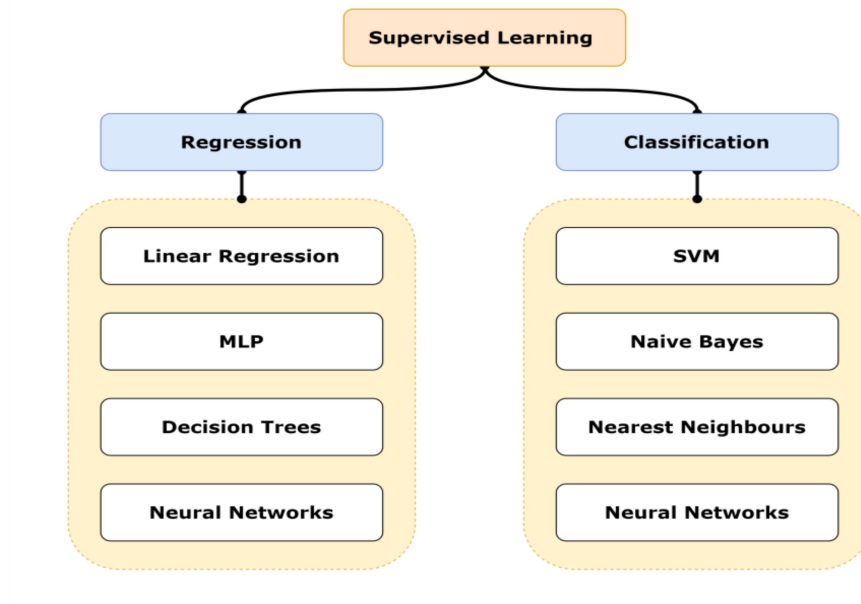


Figure 2.2: Taxonomy of Supervised Learning

The aim of the method is for learning the mappings from input x to outputs y , with a labeled set by the pair of input-output $\mathcal{D} = \{(x^i, y^i)\}_{i=1}^N$ and produce a forecasting function in the supervised learning method. \mathcal{D} is called the training set, and N is the number of training instances. The nature of training input relies on the type of problem the method is handling. The x_i is the \mathcal{D} -dimensional matrix describing the simple attributes or features. But, x_i is complex structured objects like a time-series, image, e-mail, graphs, etc,. The output y_i is in various forms according to the problem.

If the value of y_i is a categorical variable from a finite group, $y_i \in \{1, \dots, C\}$, like normal or abnormal, then the issue is called pattern recognition or classification. Also, as y_i is a real value, the issue is decided as a regression. By simple terms, regression contains forecasting a real value, happening to a label estimation whereas, classification contains specifying class membership of a given instance. The function learned in the training stage is called as a classification model. In Figure 2.2, the

taxonomy of supervised learning methods is described according to the aspects of classification and regression.

2.5.2 Unsupervised Learning

Unsupervised learning is a learning where the forecasted outcome labels are unknown. The training of the model is done applying this unlabeled data. These approaches direct for searching the hidden layers as realizing sets of photos with similar cars but there is a risk for the implementation and cannot be utilized like supervised learning method. This learning can be applied like a preceding stage before using supervised learning. For producing accurate results, the internal data structure may provide the information. The training of model is done by unlabeled dataset in this learning. The most commonly applied approach is the clustering in this unsupervised learning. Clustering is commonly applied for detection of pattern and modeling of description. As there has no labels for learning, the approaches applied in this learning searches the unlabeled input data with grouping and summarization of associated data points, with patterns detection for achieving specific information and producing forecasting. Mostly applied unsupervised learning methods are hierarchical clustering, apriori algorithm, outlier detection, clustering, self-organization map, and eclat method. Clustering is one kind of unsupervised learning methods.

- **Clustering:** This can be applied for discovering variations and similarities. It assembled same things together. However, it doesn't require for understanding any class labels, but the system can know data itself and cluster it well. In comparison with classification, the outcome labels are not pre-known. This learning type algorithm can handle various issues, as provide clusters of similar tweets according to their content, recognize sets of photos with approximate cars, or decide various news types.

2.5.3 Semi-supervised Learning

Something is required among these two types of machine learning methods, unsupervised and supervised learning for all the investigations. This can be applied semi-supervised learning in such conditions, that refers to a learning process where many outcome values are missing. This requires for using each unsupervised and supervised path as for producing useful outcomes. It is usually the event in medical

fields, where medical doctors do not own for performing the manual classification all ways of health issue based on the overwhelming huge volume of data.

2.5.4 Reinforcement Learning

The expected outcome value is explicitly unknown but the system can provide feedback to the expected outcome. Reinforcement learning is learning provided such feedback. It is applied in order to train artificial intelligence of gaming on the nero game and might be searched in schools. The specific title is studied by the students after they providing sitting an exam and the students are provided by the teacher with grades with no defining what answers were right or not. Reinforcement learning is also a kind of artificial intelligence. In this learning, the continuous learning method is done from the environment it is operating depending upon the reward thing. The main objective is for maximizing the cumulative reward until the full range in the possible events achievement. Mostly applied reinforcement learning methods are; temporal difference, deep adversarial network, and q-learning.

2.6 Deep Learning

Deep learning is a kind of artificial intelligence and machine learning for imitation the path humans achieving specific kinds of knowledge. This learning is a potential thing in data science that contains modelling of predictive and statistics. This is more efficient for data scientists who are tasked by analysis, translation, and collection huge volume of data; this learning provides this procedure easier and faster. Operational models are permitted by comprising many operation layers for learning descriptions in the data by many abstraction levels. This learning model contains many fully connected hidden layers so such models are called being deep learning models in the comparison with models by the pair of hidden layers called shallow learning models. The classification of deep neural networks is done according to the flow of information. When the flow of information from the input layer to the output layer is performed with no feedback replies, this network is known as feedforward deep learning neural network. Conversely, when the integration of neural network to operation is done with many feedback loops, this network is called recurrent neural network. One main activities of deep learning neural networks are for learning

descriptions by a raw dataset. The neural network model has the ability for automatic discovering the descriptions in data needed to detection of feature and classification is called a feature learning or description. Deep learning neural network may be specified with a class of machine learning methods which applies many layers of operational elements for continuous learning and extracting features by a raw dataset, at which the movement from the lower end to the higher end of the layers is done, the extracted features start resulting more and more pronounced in the learning model for inferring desired results for the given classification or forecasting job. The division of deep learning into three classes;

- Deep neural networks for generative or unsupervised learning: deep boltzmann machine, autoencoders, restricted boltzmann machine, deep autoencoders, deep belief network
- Deep networks for supervised learning: recurrent neural networks, convolutional neural networks
- Hybrid deep networks: Comprised by two or more deep learning methods

2.6.1 Recurrent Neural Networks

Recurrent neural networks are a type of supervised deep learning method. It applies time-series or sequential data for the model construction. They are commonly utilized in various fields like natural language processing, image captioning, language translation, and speech recognition [12]. Many applications like voice search, google translate, and Siri applies recurrent neural networks. Recurrent neural networks make input x_t at a time interval t for producing \hat{y}_t that is the result of this network. Moreover, this network is also calculating an internal event at time interval t expressed by h_t , that it moves from one-time interval to other intervals internally in the network where,

$$h_t = f_w(h_{t-1}, x_t) \quad (2.1)$$

The recurrence relation is computed in the network at every time interval. The value of h_t is decided by function f that is characterized by the older state of the network expressed as h_{t-1} , the input vector x_t and weight w at time interval t . Moreover, the loss value from each element is calculated by moving one iteration of forwarding pass through the network. All loss values from each time interval are then

added into a single loss value L that defines the total loss of the network. Now the updated hidden state of each stage in the forward pass can be described by,

$$h_t = \tanh(W_{hh}^T h_{t-1} + W_{hx}^T x_t) \quad (2.2)$$

Where \tanh is the hyperbolic non-linear operation applied with recurrent neural network whose value can be positive or negative, permitting to decrease or increase in events in comparison with sigmoid operation which produces non-negative values. When two separate inputs are loaded, one from the last event and another from the input x_t , two weight matrices are computed representing by W_{hh}^T and W_{hx}^T . The output vector for every time interval is described by,

$$\hat{y}_t = W_{hy}^T h_t \quad (2.3)$$

Where ht is the calculated hidden event and W_{hy}^T is the weight vector among the hidden event and the output element. The training of recurrent neural network needs the updated weight in the network at every time interval that the backpropagation variation is applied called backpropagation through time method, at which the backward propagation of error is produced at every time interval and all-time intervals are passed to the starting in the sequence of data. The computation of gradient on the network relating with cell state 0 contains various iterative multiplications of weighted vector with iterative computation of gradient applying the activation operations. This consequence results in the investigation of gradients at which the gradients occur rising as the constant computation in each step and this network has no ability for optimization becoming to the overall variability of the network as the extreme weight updates. Another issue of recurrent neural network architecture is disappearing gradients, at which gradients results rising in the middle of iterated matrix multiplications becoming to the network with no training ability and optimization after many cycles of epoch.

2.6.2 Long Short-Term Memory

Long Short-Term Memory is a common type of recurrent neural networks proposed as a new gradient-based approach for producing the answer to the issues. They are designed for learning capability to long-time dependencies no loss in short-

time abilities. This is produced with utilizing constant error flow aside investigating gradients as back propagation through time. This possesses a common linear element with self-connecting by value 1. With applying output and input gates, the management of the information flow is done in relating with disappearing gradient issue. The information gate is the principle elements of long short-term memory that the selected addition or removing in information from its cell event. Gates primary contain a pointwise multiplier element and a sigmoid neural network layer. The limitation of information flow across the cell from one and zero is done by the sigmoid layer that critically gates the information flow. The long short-term memory contains three gate elements:

- **Forget Gate:** This gate decides which information flowed through from the cell event. The determination is produced with the sigmoid layer, that at the values of h_{t-1} and x_t to result a number among 1 and 0 for the cell step C_{t-1} . The result describes the degree to which information is maintained. The value of 0 describes completely forget this information and the value of 1 describes keep everything. This gate can be described by,

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2.4)$$

- **Store Gate: Store Gate:** This gate decides which information is kept in the new cell stage. In its two-step procedure, a sigmoid layer expressed as the input gate layer i_t determines which values will be upgrading. The next layer \tanh generates a vector in new component \hat{C}_t which will be appended to the new state. These stages can be described by,

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2.5)$$

$$\hat{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (2.6)$$

The old cell state C_{t-1} into the next cell state C_t is updated according upon the calculation in the last two gates. The old state f_t is multiplied as forgetting the earlier information, after the addition with the information from store gate is performed i.e. the value produced from $i_t * \hat{C}_t$. This stage is described by,

$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t \quad (2.7)$$

- **Output Gate:** This gate decides what information will be resulted at the current cell stage. Applying the gate's sigmoid layer, how much information produced of the cell state is determined. Moreover, the cell state is put across *tanh* unit, that squashes the values among 1 and -1, that is the multiplication with the result of the sigmoid gate. The procedure can be described,

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (2.8)$$

$$h_t = o_t * \tanh(C_t) \quad (2.9)$$

The principle concept of long-short term memory is the creation ability of an uninterrupted gradient flow among many cell states with keeping independence to every cell in the network, that eliminates the issues in disappearing and investigating gradients at simple recurrent neural networks. This has the ability of the network for creating short-term and long-term dependencies no loss in the potential information and filtering the non-potential information.

2.6.3 Convolutional Neural Networks

Convolutional neural network is a type in feed-forward deep learning networks utilized to many text-based issues and visual analysis. The architecture of a convolutional neural network is investigated for the analysis of the neurons in the visual cortex of mammals for comprehending how neurons in visual pathways deduce information from patterns cast on a retina in an eye and convert it on the way to cerebral cortex that implements and recognizes an image. The multilayered artificial neural network contains cascading layers comprising with two components: the C-cell layer and the S-cell layer. The C-cell layers perform the information pooling from the previous simple cells and transfers the output to the associative simple cell layers in a feed-forward manner when the S-cell layers are the main feature extraction elements in neocognitron. The common convolutional neural network is a continuous repetition of neocognitron by the exception of backpropagation for the learning algorithm. The earlier convolutional neural network called LeNet-5 is used for the recognition in hand-written digit applying the MNIST dataset. The convolutional neural network contains various layers in organizing with two core elements, a pooling and

convolution layers that performs the feature extraction at input layer and a fully-connected layer is utilized for prediction the outcomes of the previous feature extraction with a forecasting label result.

CHAPTER 3

DESIGN OF THE SYSTEM

The aim of this system is to develop the intrusion detection system using convolutional neural network model on unbalanced dataset and balanced dataset. In this system, the dataset for network intrusion detection, NSL-KDD data set, is employed. The data set consists of 42 features, 41 features grouped into four categories, such as essential features, content features, time-based, and host-based features [6]. The last feature is about all the data of other features. The system architecture is described in Figure 3.1. Table 3.1 shows different features in dataset. Features under various categories is shown in Table 3.2.

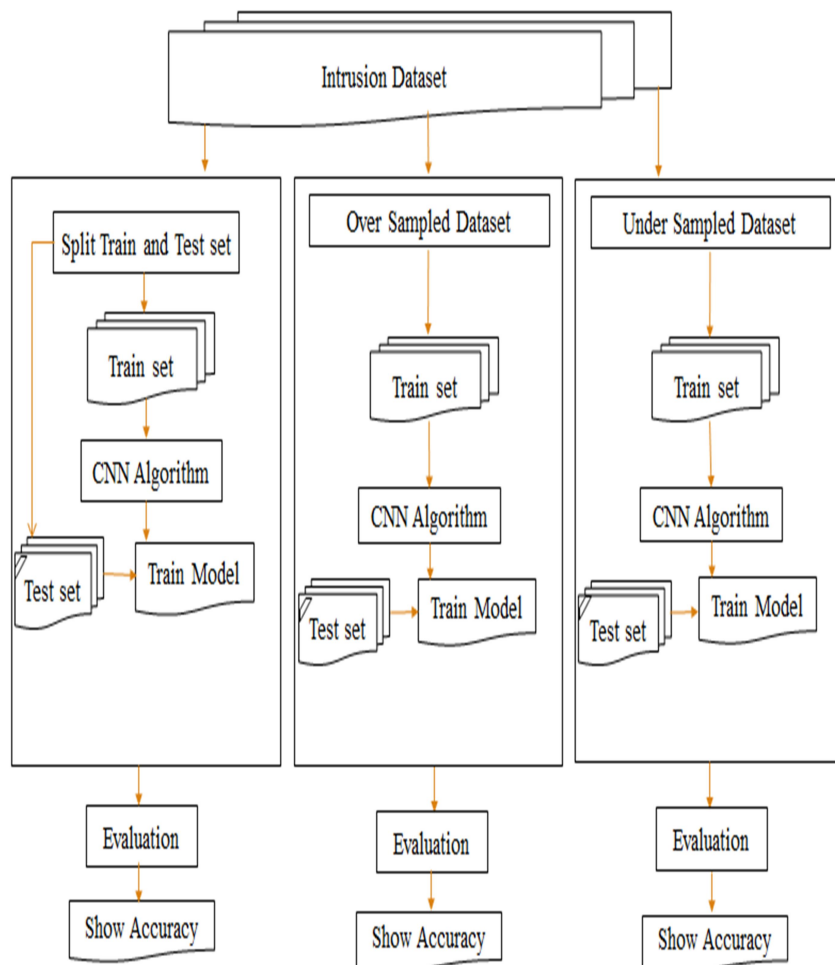


Figure 3.1 System Architecture

3.1 Overview of the Proposed System

An intrusion detection system is a network traffic monitoring system for detecting mistrustful activities and it gives an alarm if the mistrustful activities are found. It is a software application for analysis and monitoring the network system for the detection of harmful activity and brokerage of policy prior to the serious damage in network and the corruption of data assets.

The aim of the intrusion detection system is in order to discover various types of security violations outside the system brokerage and inside the hateful system features and prevalence of data misuse. The rule-based feature matching strategy including normal action features in safe library is utilized. The comparison of this strategy with audited usage features for alerting any abnormal actions. The intrusion detection system can perform the detection various intrusions such as trojan horses, misuse of legitimate users, impersonating attempts, and viruses.

Table 3.1 Different Features

Feature Number	Feature Name
1	Duration
2	Protocol_type
3	Service
4	Flag
5	Src_bytes
6	Dst_bytes
7	Land
8	Wrong_fragment
9	Urgent
10	Hot
11	Num_failed_logins
12	Logged_in
13	Num_compromised
14	Root_shell
15	Su_attempted
16	Num_root

17	Num_file_creations
18	Num_shells
19	Num_access_files
20	Num_outbound_cmds
21	Is_hot_login
22	Is_guest_login
23	Count
24	Srv_count
25	Serror_rate
26	Srv_serror_rate
27	Rerror_rate
28	Srv_rerror_rate
29	Same_srv_rate
30	Diff_srv_rate
31	Srv_diff_host_rate
32	Dst_host_count
33	Dst_host_srv_count
34	Dst_host_same_srv_rate
35	Dst_host_diff_srv_rate
36	Dst_host_same_src_port_rate
37	Dst_host_srv_diff_host_rate
38	Dst_host_serror_rate
39	Dst_host_srv_serror_rate
40	Dst_host_rerror_rate
41	Dst_host_srv_rerror_rate

Table 3.2 Features Under Various Categories

Category	Features
Basic Features	Feature 1 to Feature 10 all
Content Features	Feature 11 to Feature 22 all
Time-based Features	Feature 23 to Feature 31 all
Host-based Features	Feature 32 to Feature 41 all

This system is firstly tested with unbalanced nature of original dataset using convolutional neural network model. Then the system is tested with balanced nature by applying undersampling and oversampling using convolutional neural network model. The key metrics of performance measures (accuracy, recall, f-measure, and precision) are evaluated for this proposed system analysis.

3.1.1 Random Sampling

Random sampling is one kind of probability sampling in that every instance possesses the equality in probability for the selection. A random selected instance is an unbiased description for the total population. If the instance does not represent the population, the variation becomes a sampling error. Random sampling is a method for choosing each participant or a subset of the population for providing the statistical inferences from them and estimation the characteristics of the whole population. It can be utilized as a data reduction method as it allows a large data set to be represented by a much smaller random data instance (or subset).

3.1.2 Undersampling

Undersampling is a balancing approach for asymmetric datasets with maintaining all of the data in the minority class and reducing the size of the majority class. In another words, the deletion of samples from the majority class are performed and this sampling can occur to the invaluable information loss in the model. Majority classes are classes which provides the larger proportion of the dataset [5]. Minority classes are classes which provides the smaller proportion of the dataset. Assume that D is a large data set, and that contains the number of instances, N .

Simple random sample without replacement (SRSWOR) of size s : This creation is done by deleting s from the N instances at D ($s < N$), in where the deletion probability of any instance in D is $1/N$, i.e., all instances are equally likely to be sampled. This sampling is appropriate in such conditions as there is plenty of data for an accurate analysis. All rare instances are utilized however the number of abundant instances is reduced for the creation of two equally sized classes.

3.1.3 Oversampling

Over-sampling is a balancing approach for asymmetric datasets with maintaining all of the data in the majority class and increasing the size of the minority class by adding the instances to it. In another words, the duplication of samples to the minority class in the training dataset and this sampling can occur overfitting to some models as learning algorithms focus on replication of minority instances.

Simple random sample with replacement (SRSWR) of size s : This is similar to SRSWOR, except that each time an instance is chosen from D , it is recorded and then replaced, i.e., after an instance is drawn, the back placement is performed at D so that the choosing may be done again. This sampling is appropriate in such conditions as there is no enough information. One class is the majority, or abundant, and the other class is the minority, or rare. The number of rare instances is increased in this sampling.

3.1.4 Convolutional Neural Networks

A convolutional neural network (CNN) is a type of deep, feedforward neural networks. It applies the multilayer perceptron and it has been developed for the reduction of processing needs [13]. It can be applied for the collection of spatial data or sequential data and it is widely utilized in speech recognition, time series analysis and image processing, etc. It consists of an input layer, an output layer and a hidden layer which contains many pooling layers, convolutional layers, normalization layers, and fully connected layers.

- **Input layer:** Text documents or images are kept by the representation of vector.
- **Convolutional layer:** The output is decided with the calculation of dot products between set of weights at input layer. The aim of this layer is the observation of features. The principle construction block of the convolutional neural network is the convolutional layer that performs the creation of feature activation maps from the input layer applying many related objects called filters, by passing the certain filter across the height and width of the input layer, therefore, for the calculation of the dot product among the input layer and entries in the filter. The convolution computation occurs in the creation of

many two-dimensional activation maps, that are later fed into continuous pooling layers. The times of movement of the filter for each stage is decided by the stride's value, the default is one. The activation function ReLU, is applied by the convolutional layer for the conversion of all the negative values into value zero. Every convolutional layer is specified by various parameters containing kernel size, zero padding, stride, input size, and the map stack. Then, the calculation of input signal is performed with an activation function, Rectified Linear Units (ReLUs). This activation function is used on the input data; therefore, the dimensions of the input and the output are same.

$$f(x) = \max(0, x) \quad (3.1)$$

- **Pooling Layer:** This layer acts as a mediator between many convolutional layers. It performs the reduction of the spatial dimensions of the input data. This is similar to the previous convolutional layer as this layer sweeps the filtering among all input data however this filtering does not possess any weights. Two kinds of pooling operations are maximum pooling, and average pooling. Maximum pooling takes the selection of the pixel by the maximum value for sending to the output array when the movement of filter among the input. Average pooling performs the computation of the average value in the receptive field by sending to the output array when the movement of filter among the input. This layer is applied for the downsampling, that reduces the disperse size of created feature maps with convolution layer with the reducing of their dimension according to a selected criterion. Pooling directs to extract the most significant feature from the feature maps and optimize the overall computation required for data processing. This layer provides many benefits to convolutional neural network whereas there is the information loss at this layer. The reduction of noise features, the efficiency improvement, and the overfitting prevention are provided by this layer.
- **Fully-Connected Layer:** This layer is same with the output layer of multilayer perceptron (MLP). The aggregation of information from the final feature maps and the generation of final classification are performed. The fully connection of all neurons with all neurons in the previous layer is taken. This is the last element of convolutional neural network and is a fully connected layer of the preceding artificial neural networks. The flattened column vector then becomes the input for the fully connected layer for transforming feature

vectors, that is performed by training the network utilizing backpropagation on many epochs. The last element of the fully connected layer applies an activation function like a sigmoid or softmax activation function for the class label forecasting creation, that is also the final result of convolutional neural network. The reduction of data dimension at pooling layer to a single dimension and the connection with every neuron are taken. The classification is performed with activation function, SoftMax.

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad (3.2)$$

- Where, \vec{z} is the input vector, z_i is the elements of the input vector, e^{z_i} is the standard exponential function, and K is the number of classes in the multi-class classifier. The SoftMax function does the testing of result by using training data. Finally, this provides the result which the input image to which the relating class.

3.2 Evaluation of the Performance of Methods

In all learning algorithm, evaluating performance is a fundamental aspect. A measurement is needed to determine the effectiveness of the learning algorithm used for a system. Not only it is important in order to compare competing algorithms, but in much case is an integral part of the learning algorithms itself. Estimating classifier accuracy is important in that it allows one to evaluate how accurately a given classifier will label future data. For determining effectiveness of the algorithm, commonly used measurements include classification accuracy, F-Measure, Precision and Recall. These measurements can be calculated by the classification results commonly tabulated in a matrix format called a Confusion Matrix. Classification accuracy is defined as the percentage of the examples correctly classified by algorithm. Bootstrap, Holdout and Cross-validation methods are common techniques for accessing classifier accuracy, based on randomly sampled partitions of the given data.

3.2.1 Confusion Matrix Performance

In a classic binary classification problem, the classifier labels the items as either positive or negative. A confusion matrix summarizes the outcome of the

algorithm in a matrix format. In our binary example, the confusion matrix would have four outcomes:

True positives (TP): These refer to the positive tuples that were correctly labeled by the classifier. Let TP be the number of true positives.

True negatives (TN): These are the negative tuples that were correctly labeled by the classifier. Let TN be the number of true negatives.

False positives (FP): These are the negative tuples that were incorrectly labeled as positive. Let FP be the number of false positives.

False negatives (FN): These are the positive tuples that were mislabeled as negative. Let FN be the number of false negatives.

These terms are summarized in the confusion matrix of Table 3.3. The confusion matrix is a useful tool for analyzing how well your classifier can recognize tuples of different classes. TP and TN tells the user when the classifier is getting things right, while FP and FN tells the user when the classifier is getting things wrong (i.e., mislabeling). Given m classes (where $m \geq 2$), a confusion matrix is a table of at least size m by m . An entry, CM_{ij} in the first m rows and m columns indicates the number of tuples of class i that were labeled by the classifier as class j .

Table 3.3 Confusion Matrix

Confusion Matrix		Predicted class:		
		Positive	Negative	Total
Actual Class	Positive	TP	FN	P
	Negative	FP	TN	N
	Total	P'	N'	P+N

For a classifier to have good accuracy, ideally most of the tuples would be represented along the diagonal of the confusion matrix, from entry $CM_{1,1}$ to entry

$CM_{m,m}$, with the rest of the entries being zero or close to zero. That is, ideally, FP and FN are around zero.

The simplest performance measure is accuracy. The overall effectiveness of the algorithm is calculated by dividing the correct labeling against all classifications. The accuracy of a classifier on a given test set is the percentage of test set tuples that are correctly classified by the classifier. That is,

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.3)$$

The accuracy determined may not be an adequate performance measure when the number of negative cases is much greater than the number of positive cases.

The *precision* and *recall* measures are also widely used in classification. **Precision** can be thought of as a measure of *exactness* (i.e., what percentage of tuples labeled as positive are actually such), whereas **recall** is a measure of completeness (what percentage of positive tuples are labeled as such). If recall seems familiar, that's because it is the same as sensitivity (or the *true positive rate*). These measures can be computed as

$$\text{Precision} = \frac{TP}{TP + FP} \quad (3.4)$$

$$\text{Recall} = \text{Sensitivity} = \frac{TP}{TP + FN} \quad (3.5)$$

F-Measure (Lewis and Gale, 1994) is one of the popular metrics used as a performance measure. The measure itself is computed using two other performance measures, precision and recall. Based on these definitions F-measure is defined as follows:

$$\text{F - measure} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (3.6)$$

In essence, the F-Measure is the harmonic mean of the recall and precision measures.

CHAPTER 4

IMPLEMENTATION OF THE SYSTEM

The main purpose of the chapter is to describe the experimental environment and implementation procedures of the proposed system. The performance of the proposed system has been evaluated with convolutional neural networks deep learning model on Tensorflow.

4.1 Experimental Setup

This system model is trained in Keras which is based on Tensorflow. The proposed work is done in python 3.7 with libraries of keras, TensorFlow, matplotlib and other mandatory files.

The system specifications are as follows:

- Intel ® Core i7-8550U CPU @ 3.7GHz,
- 8GB Memory,
- 1TB Hard Disk

The software components are as follows:

- Anaconda 4.8.2
- Scikit-learn 1.1.1
- Tensorflow 2.3.0
- Keras 2.4.3
- Pandas 1.1.0
- Numpy 1.23.1
- Matplotlib 3.3.2
- Python 3.7

4.2 Implementation of the System

In this system, NSL-KDD dataset is used. The NSL-KDD dataset contains 1 training set and 2 testing sets:

- 1) KDDTrain+: The full NSL-KDD train set including attack-type labels
- 2) KDDTest+: The full NSL-KDD test set including attack-type labels

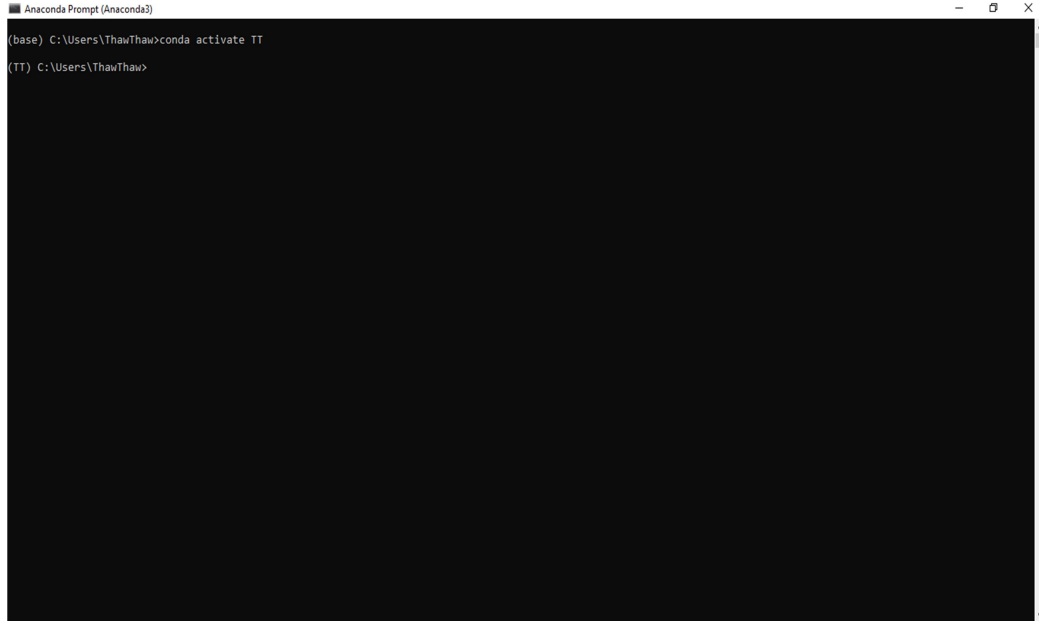
- 3) KDDTest-21: A subset of the KDDTest+ which does not include records with difficulty level of 21 out of 21

From this dataset, 1 training set and 1 testing test: KDDTrain+ and KDDTest+ are utilized. KDDTrain+ dataset contains 125,973 network traffic samples and KDDTest+ has 22,554 network traffic samples. The transformation of NSL-KDD dataset to the 1-dimensional convolution architecture is performed. Moreover, this dataset includes non-numeric and numeric features. As the training input and testing input feeding to the convolutional neural network is in the form of numeric matrix, the conversion to numeric attribute must be performed. In addition, the one-hot encoder is utilized for the conversion of category features in the dataset into numeric matrix as the usage of one-hot encoder can handle the issue in the category conversion to integer. In this system, 32 kernels with 1*3 dimension and 5 convolutional layers are used. For each convolutional layer, maximum pooling and rectified linear unit (ReLU) are used and the pooling size is 4 for only first pooling layer and is 2 for other pooling layers. SoftMax function is used at fully-connected layer. This system model is trained in Keras which is based on Tensorflow. This system is firstly tested with unbalanced nature of original dataset. Then the system is tested with balanced nature by applying undersampling and oversampling. Accuracy, precision, recall, and f-measure are the key metrics of performance evaluation of the proposed system. The NSL-KDD Training dataset is shown in Figure 4.1.

0,	tcp,	ftp_data,SF,	491,0,2,2,0.00,0.00,0.00,0.00,0.00,1.00,0.00,0.00,150,25,0.17,0.03,0.17,0.00,0.00,0.00,0.05,0.00,normal,20
0,	udp,	other,SF,	146,0,13,1,0.00,0.00,0.00,0.00,0.08,0.15,0.00,255,1,0.00,0.60,0.88,0.00,0.00,0.00,0.00,normal,15
0,	tcp,	private,S0,	0,123,6,1.00,1.00,0.00,0.00,0.05,0.07,0.00,255,26,0.10,0.05,0.00,0.00,1.00,1.00,0.00,0.00,neptune,19
0,	tcp,	http,SF,	232,8153,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,5,5,0.20,0.20,0.00,0.00,1.00,0.00,0.00,30,255,1.00,0.00,0.03,0.04,0.03,0.01,0.00,0.01,normal,21
0,	tcp,	http,SF,	199,420,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,30,32,0.00,0.00,0.00,0.00,1.00,0.00,0.09,255,255,1.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,normal,21
0,	tcp,	private,REJ,	0,121,19,0.00,0.00,1.00,1.00,0.16,0.06,0.00,255,19,0.07,0.07,0.00,0.00,0.00,0.00,1.00,1.00,neptune,2
0,	tcp,	private,S0,	0,166,9,1.00,1.00,0.00,0.00,0.05,0.06,0.00,255,9,0.04,0.05,0.00,0.00,1.00,1.00,0.00,0.00,neptune,21
0,	tcp,	private,S0,	0,117,16,1.00,1.00,0.00,0.00,0.14,0.06,0.00,255,15,0.06,0.07,0.00,0.00,1.00,1.00,0.00,0.00,neptune,21
0,	tcp,	remote_job,S0,	0,270,23,1.00,1.00,0.00,0.00,0.09,0.05,0.00,255,23,0.09,0.05,0.00,0.00,1.00,1.00,0.00,0.00,neptune
0,	tcp,	private,S0,	0,133,8,1.00,1.00,0.00,0.00,0.06,0.06,0.00,255,13,0.05,0.06,0.00,0.00,1.00,1.00,0.00,0.00,neptune,21
0,	tcp,	private,REJ,	0,205,12,0.00,0.00,1.00,1.00,0.06,0.06,0.00,255,12,0.05,0.07,0.00,0.00,0.00,0.00,1.00,1.00,neptune,2
0,	tcp,	private,S0,	0,199,3,1.00,1.00,0.00,0.00,0.02,0.06,0.00,255,13,0.05,0.07,0.00,0.00,1.00,1.00,0.00,0.00,neptune,21
0,	tcp,	http,SF,	287,2251,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3,7,0.00,0.00,0.00,0.00,1.00,0.00,0.43,8,219,1.00,0.00,0.12,0.03,0.00,0.00,0.00,0.00,normal,21
0,	tcp,	ftp_data,SF,	334,0,2,2,0.00,0.00,0.00,0.00,1.00,0.00,0.00,2,20,1.00,0.00,1.00,0.20,0.00,0.00,0.00,0.00,warezclient,
0,	tcp,	name,S0,	0,233,1,1.00,1.00,0.00,0.00,0.00,0.00,0.06,0.00,255,1,0.00,0.07,0.00,0.00,1.00,1.00,0.00,0.00,neptune,19
0,	tcp,	netbios_ns,S0,	0,96,16,1.00,1.00,0.00,0.00,0.17,0.05,0.00,255,2,0.01,0.06,0.00,0.00,1.00,1.00,0.00,0.00,neptune,1
0,	tcp,	http,SF,	300,13788,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,8,9,0.00,0.11,0.00,0.00,1.00,0.00,0.22,91,255,1.00,0.00,0.01,0.02,0.00,0.00,0.00,0.00,normal,21
0,	icmp,	eco_i,SF,	18,0,1,1,0.00,0.00,0.00,0.00,1.00,0.00,0.00,1,16,1.00,0.00,1.00,1.00,0.00,0.00,0.00,0.00,ipsweep,18

Figure 4.1 NSL-KDD Training Dataset

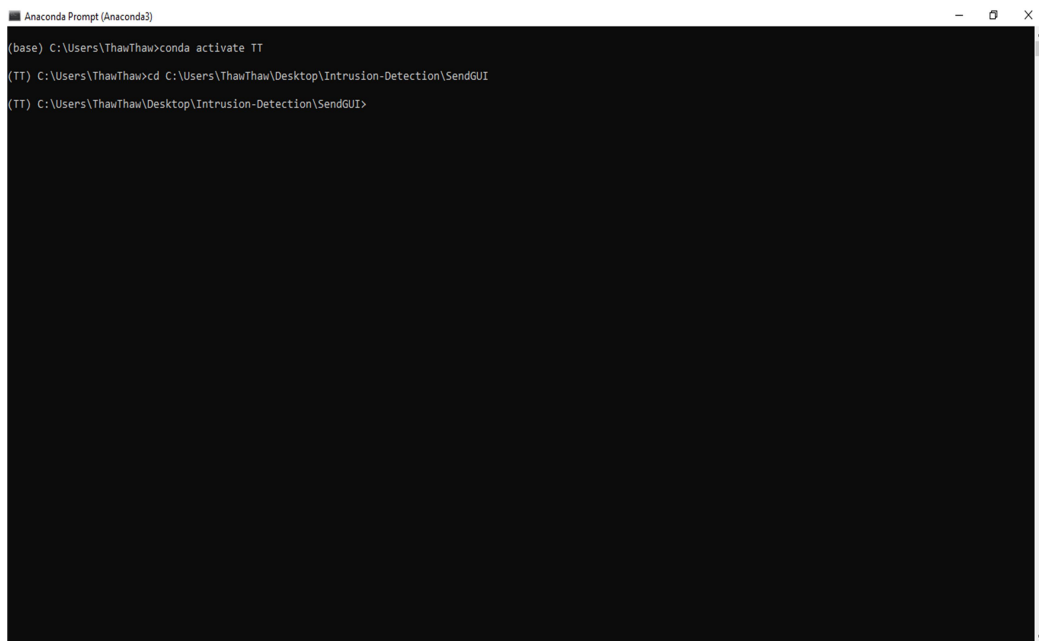
In order to execute intrusion detection system using convolutional neural networks, the actions are performed. Firstly, the anaconda is activated by typing the command: **conda activate TT**.



```
Anaconda Prompt (Anaconda3)
(base) C:\Users\ThawThaw>conda activate TT
(TT) C:\Users\ThawThaw>
```

Figure 4.2 Anaconda Activating

After that, the path is changed to the working directory by typing “**cd C:\Users\ThawThaw\Desktop\Intrusion-Detection\SendGUI**”.



```
Anaconda Prompt (Anaconda3)
(base) C:\Users\ThawThaw>conda activate TT
(TT) C:\Users\ThawThaw>cd C:\Users\ThawThaw\Desktop\Intrusion-Detection\SendGUI
(TT) C:\Users\ThawThaw\Desktop\Intrusion-Detection\SendGUI>
```

Figure 4.3 Changing Directory

To run the program the command is typed by “python main.py”. After that the first page is as shown in Figure 4.5.

```
Anaconda Prompt (Anaconda3) - python main.py
(base) C:\Users\ThawThaw>conda activate TT
(TT) C:\Users\ThawThaw>cd C:\Users\ThawThaw\Desktop\Intrusion-Detection\SendGUI
(TT) C:\Users\ThawThaw\Desktop\Intrusion-Detection\SendGUI>python main.py
2022-09-21 21:42:32.690125: W tensorflow/stream_executor/platform/default/dso_loader.cc:59] Could not load dynamic library 'cudart64_101.dll'; dlerror: cudart64_101.dll not found
2022-09-21 21:42:32.694997: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
```

Figure 4.4 Running Main Program



Figure 4.5 Main Page

By clicking “next” the page is found as following in Figure 4.6.

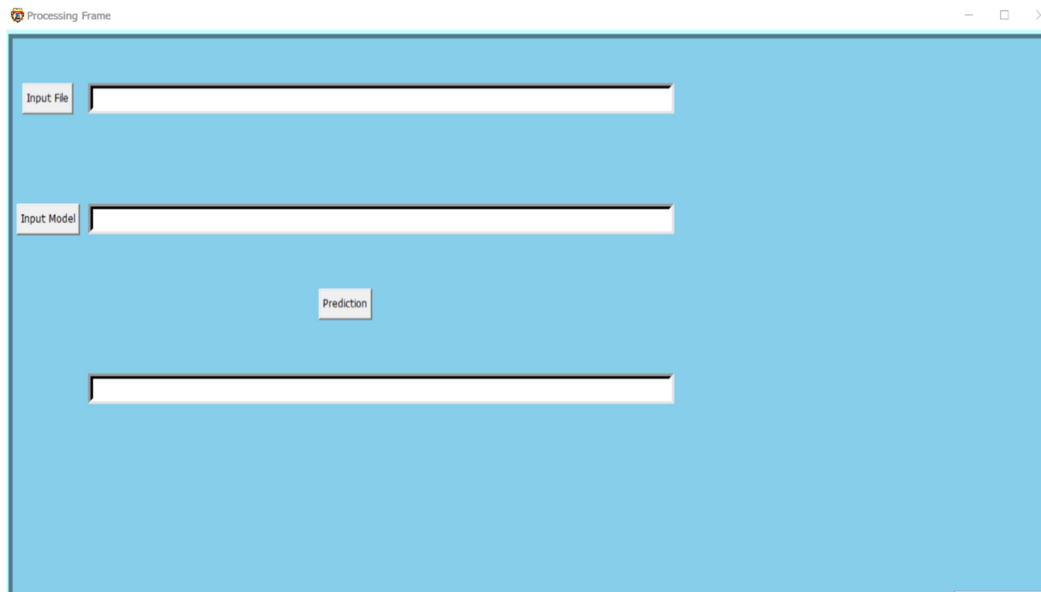


Figure 4.6 Next Page

The testing file is uploaded as shown in Figure 4.7. The input test file and input model: unbalanced model is loaded as described in Figure 4.8.

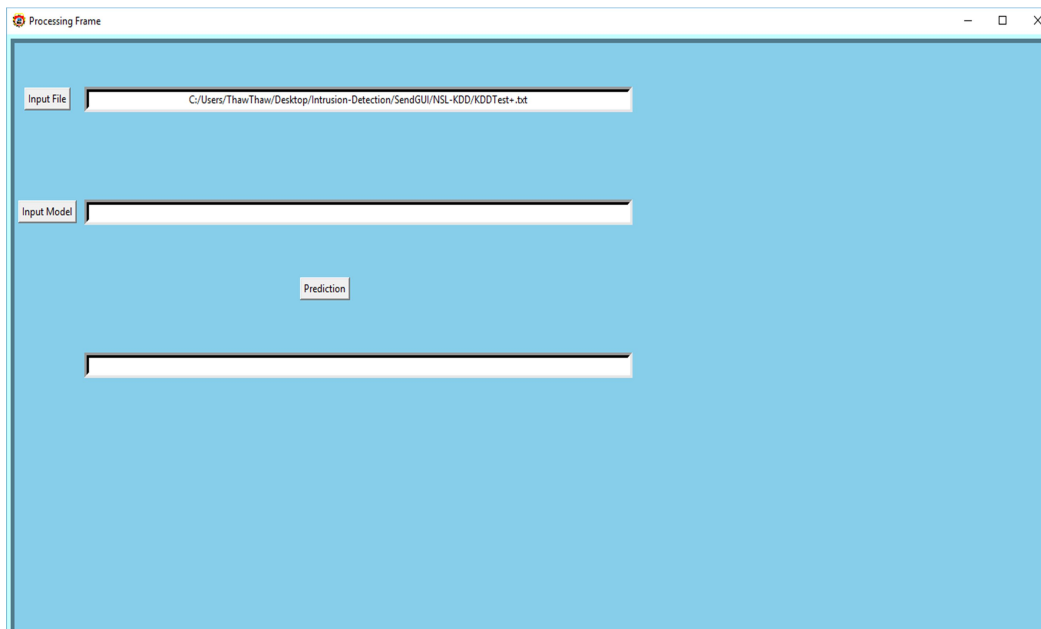


Figure 4.7 Loading Test File

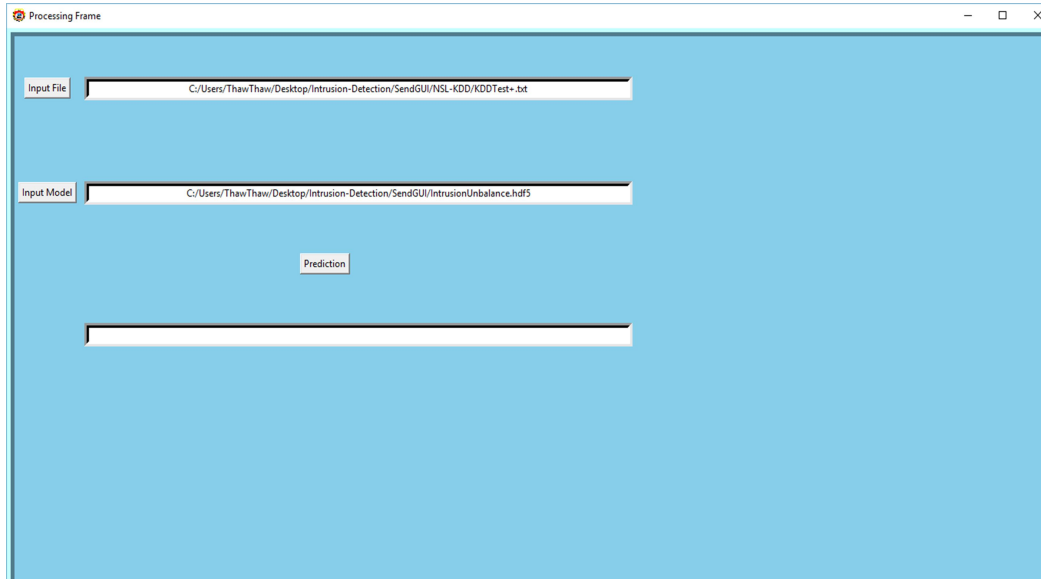


Figure 4.8 Loading Test File and Input Unbalanced Model

Then, the prediction for unbalanced model is performed by clicking the **Prediction** button. The results for the performance of proposed model on unbalanced dataset is shown in Figure 4.9.

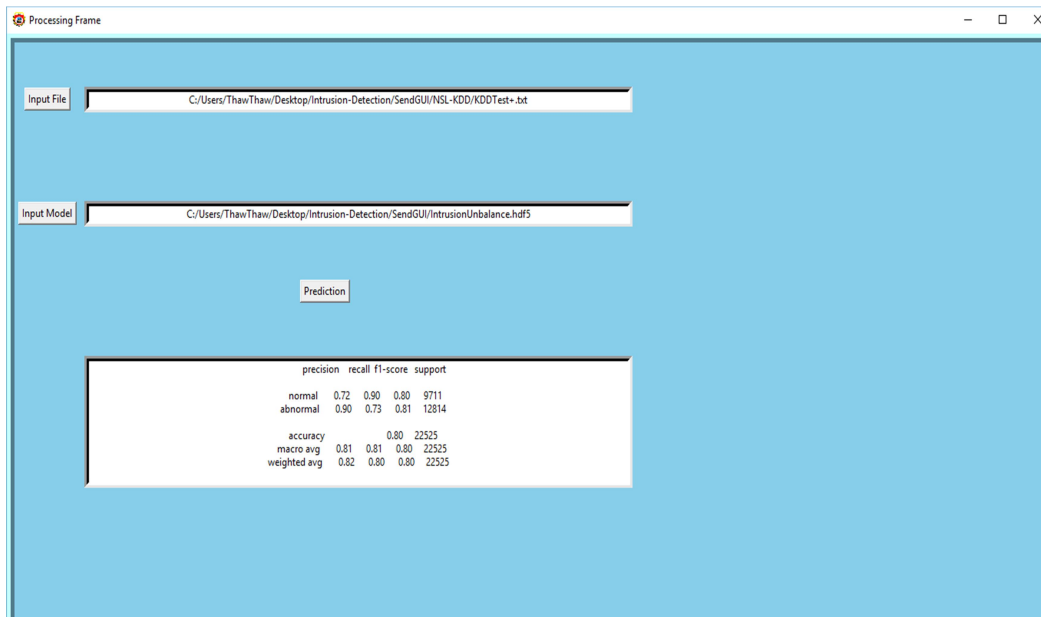


Figure 4.9 Results of Unbalanced Model

Then the system is tested for balanced model by undersampling technique. The loading test file and undersampling model is described in Figure 4.10.

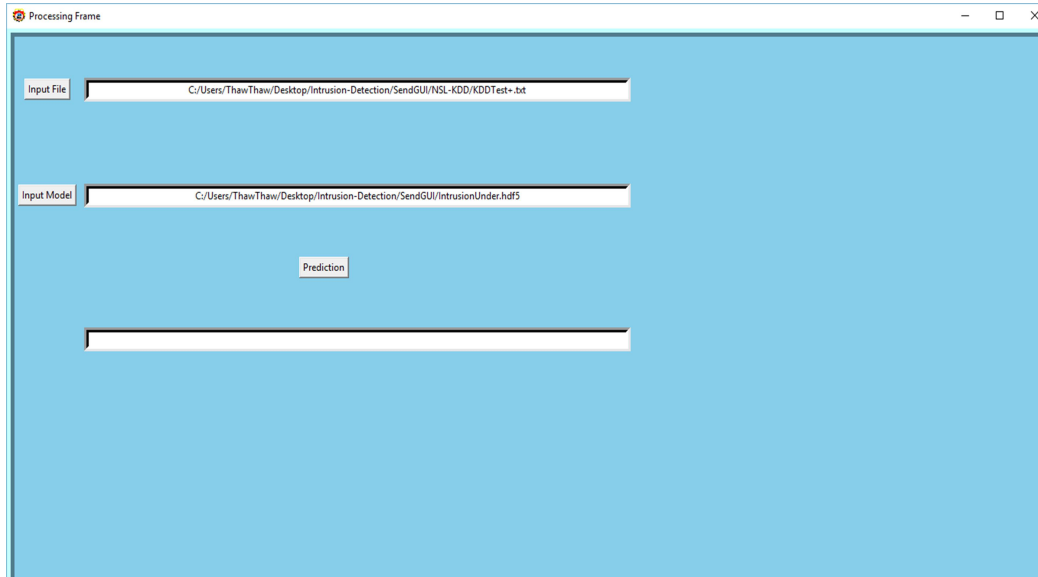


Figure 4.10 Loading Test File and Input Balanced Undersampling Model

Then, the prediction for balanced model by undersampling is performed by clicking the **Prediction** button. The results for the performance of proposed model on balanced undersampling model is shown in Figure 4.11.

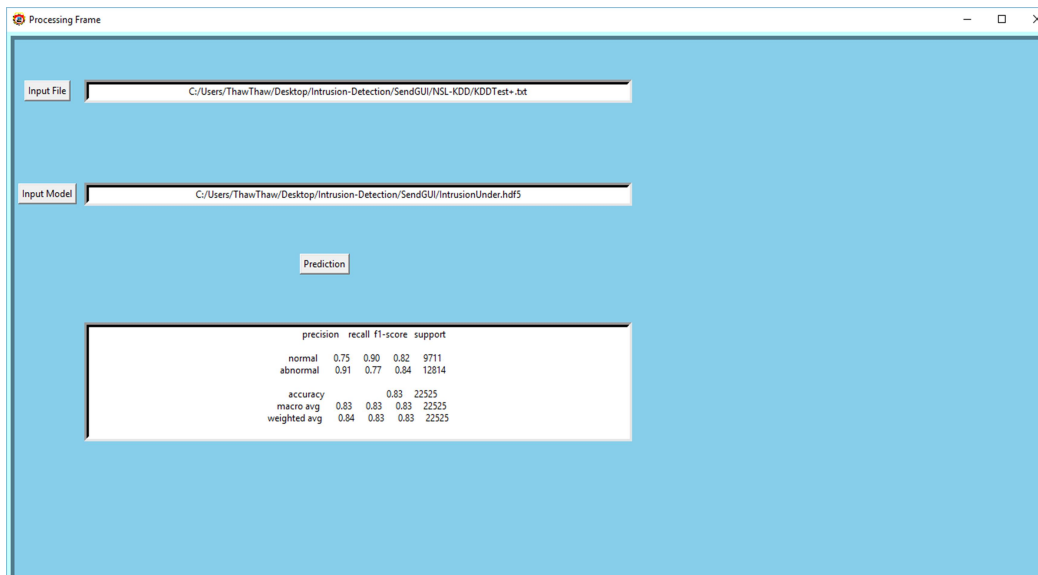


Figure 4.11 Results of Balanced Undersampling Model

Then the system is tested for balanced model by oversampling technique. The loading test file and oversampling model is described in Figure 4.12.

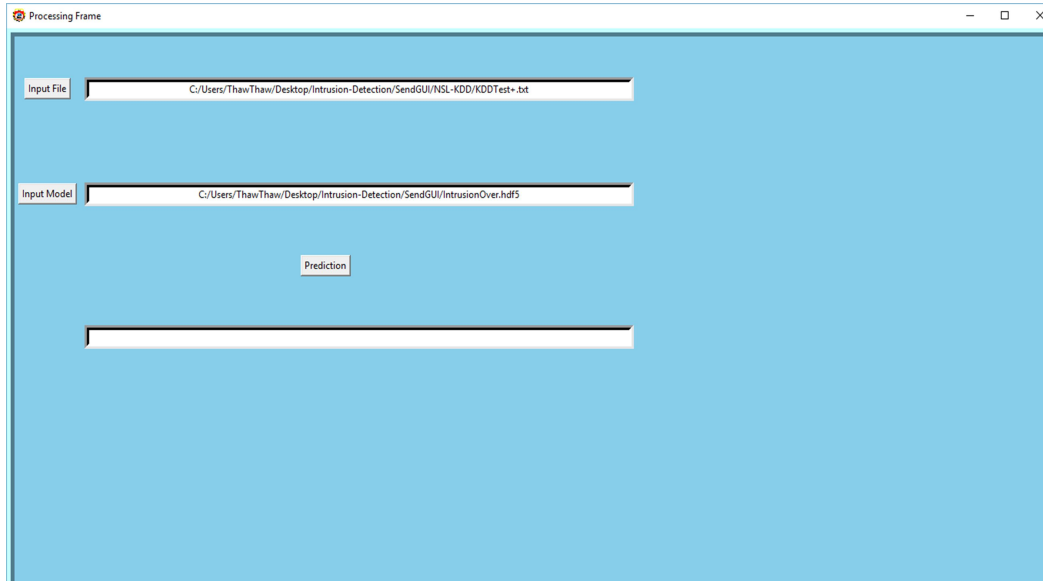


Figure 4.12 Loading Test File and Input Balanced Oversampling Model

Then, the prediction for balanced model by oversampling is performed by clicking the **Prediction** button. The results for the performance of proposed model on balanced oversampling model is shown in Figure 4.13.

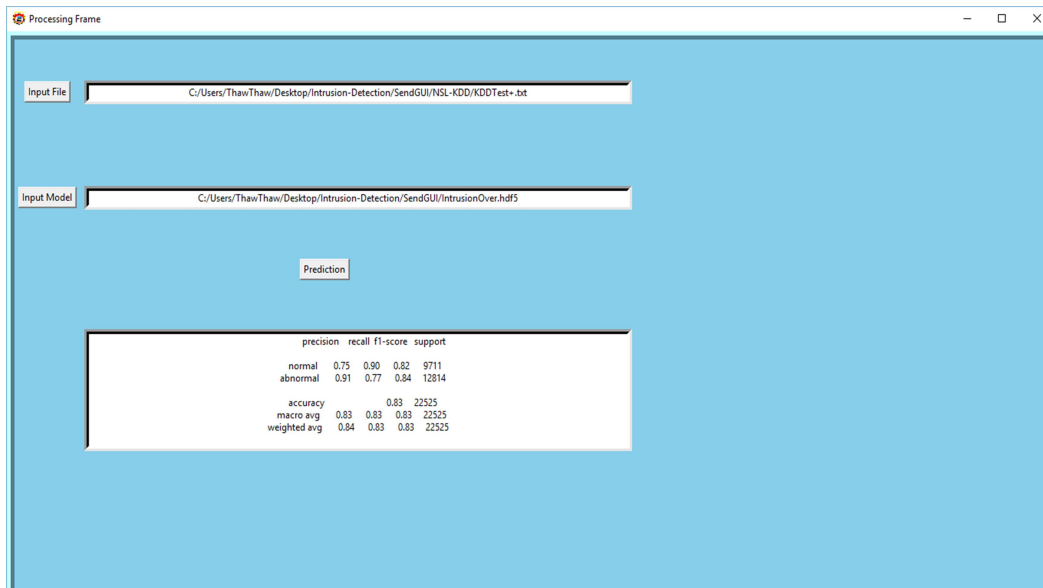


Figure 4.13 Results of Balanced Oversampling Model

4.3 Evaluation of Experimental Results

The popular performance measures (accuracy, recall, f-measure, and precision) are evaluated for this proposed system analysis. Figure 4.14 shows the comparative results for the performance of proposed model for abnormal on unbalanced, balanced by oversampling and undersampling.

	Abnormal		
	Precision	Recall	F-measure
Unbalanced	0.90	0.73	0.81
Oversampling	0.91	0.77	0.84
Undersampling	0.91	0.77	0.84

Figure 4.14 Performance Results for Abnormal

Figure 4.15 shows the comparative results for the performance of proposed model for normal on unbalanced, balanced by oversampling and by undersampling.

	Normal		
	Precision	Recall	F-measure
Unbalanced	0.72	0.90	0.80
Oversampling	0.75	0.90	0.82
Undersampling	0.75	0.90	0.82

Figure 4.15 Performance Results of Normal

Figure 4.16 shows the comparative results for the performance of proposed model for Abnormal type on unbalanced, balanced by oversampling and balanced by undersampling.

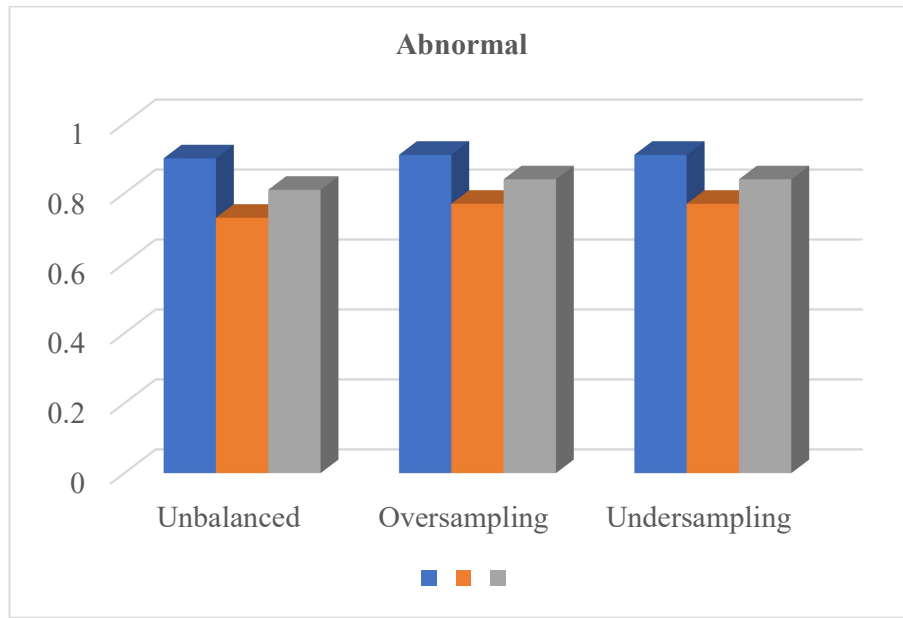


Figure 4.16 Performance Comparison of Unbalanced and Balanced on Abnormal

Figure 4.17 shows the comparative results for the performance of proposed model for Normal type on unbalanced, balanced by oversampling and undersampling.

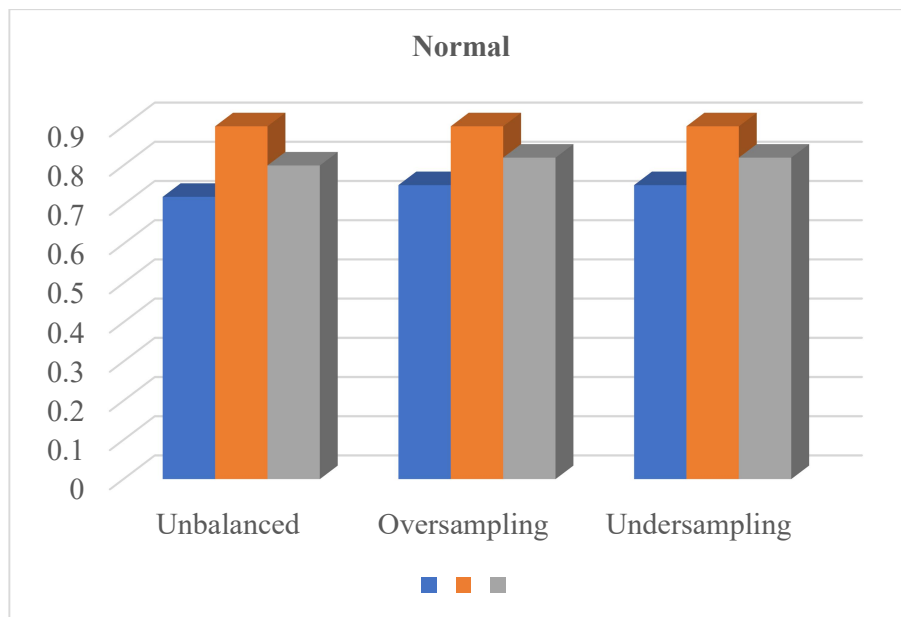


Figure 4.17 Performance Comparison of Unbalanced and Balanced on Normal

Figure 4.18 and Figure 4.19 show the comparative of accuracy of proposed model on unbalanced, balanced by oversampling and undersampling.

Method	Accuracy (%)
Unbalanced	80
Oversampling	83
Undersampling	83

Figure 4.18 Accuracy Results of Unbalanced and Balanced

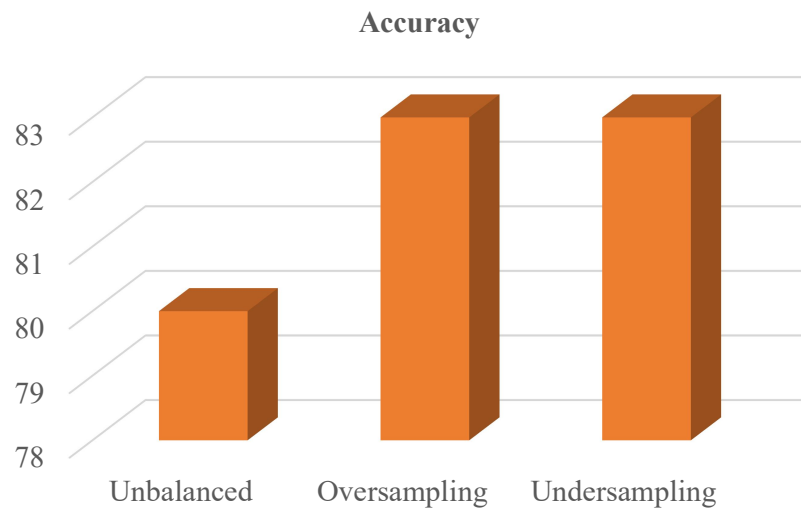


Figure 4.19 Accuracy Comparisons of Unbalanced and Balanced

According to the evaluation results, the improvement of accuracy of CNN is achieved by applying random sampling techniques: oversampling and undersampling.

CHAPTER 5

CONCLUSION

The field of Intrusion detection systems becomes importance because the usage of internet is evolving in current days. There are still many issues in Intrusion Detection systems in which high rates of false positive and false negative are also including. In order to resolve all these issues, there are many techniques in the area of networking, machine learning, and security and so on. This system focuses on the machine learning area to obtained the highest accuracy when classifying the incoming data as normal or abnormal data

Machine learning or deep learning techniques can play in efficient role for conducting intrusion detection systems. These techniques use the collection of training data to create a model by analyzing the relationship between each packet data and define the state of incoming packet data such as normal or abnormal (attack). In real case, the number of packet data is highly emerging for larger networks. This case leads to a time-consuming process when predicting the state of incoming packets data. So, this system uses the effective deep learning algorithm Convolutional Neural Network to improve the work of intrusion detection work over a vast amount of data.

5.1 Advantages

The aim of this proposed system is in order to achieve the improvement in intrusion detection effectiveness as the development of most existing intrusion detection system with the machine learning approaches did not support for the prevention by newly formed attacks using last data. Therefore, convolutional neural network deep learning model is used for developing the intrusion detection system. A deep learning model is implemented to train the model with NSL-KDD data set, namely convolutional neural network. This system achieves accuracy of 80% in unbalanced dataset and 83% in balanced dataset. The experimental outcomes also demonstrate the superiority of the Convolutional Neural Networks Classifier with balanced dataset in terms of accuracy, recall, precision, and F-Measure than Convolutional Neural Networks Classifier with unbalanced dataset.

5.2 Limitations and Further Extensions

The convolutional neural networks can provide accurate results than other deep learning approaches in intrusion detection system when the kernel can be trained for the sufficient reflection in the network features. But, the efficient preprocessing is needed. Two other preprocessing methods: weighted and compressed are recommended. Moreover, the network information is needed for the application of this approaches. But, the direct method is the most intuitive method for field-to-pixel conversion as the accurate reflection of features in an image in the convolutional neural networks.

Real-time intrusion detection system can be proposed in the future. Also aims to apply the model building in this thesis and utilize them to a live network stream for providing our inferences in real-time. A real-time, stream-based IDS architecture can be further implemented on any edge device that applies networking for every operation. Moreover, the data dimension reduction will be considered in preprocessing in computing with huge amount of data. In the future, it is aimed to build the own data sources and test the techniques on various modern network infrastructures.

AUTHOR'S PUBLICATION

- [1] Aye Thawta Sann, Zin Thu Thu Myint, "Improving the Accuracy of CNN by Applying Random Sampling Methods on NSL-KDD Dataset", Parallel and Soft Computing (PSC), UCSY, Yangon, Myanmar, Sept 2022.

REFERENCES

- [1] B. A. Tama, M. Comuzzi, and K. H. Rhee, “TSE-IDS: A Two-Stage Classifier Ensemble for Intelligent Anomaly-based Intrusion Detection System”, IEEE Access, (Volume: 7), IEEE, 11 July, 2019, pp. 94497 – 94507.1
- [2] C. Grosan, and A. Abraham, “Intelligent systems “, Berlin, Springer, 2011.
- [3] I. Abrar, Z. Ayub, F. Masoodi, A. M. Bamhdi, “A Machine Learning Approach for Intrusion Detection System on NSL-KDD Dataset”, Proceedings of the International Conference on Smart Electronics and Communication (ICOSEC 2020), IEEE, Trichy, India, 10-12 September, 2020, pp. 919-924.
- [4] J. Hall, M. Barbeau, and E. Kranakis, “Anomaly-based intrusion detection using mobility profiles of public transportation users”, IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (Wimob'2005), IEEE, Montreal, Que, 03 October, 2005. pp.17-24.
- [5] J. Han, M. Kamber, and J. Pei, “Data Mining: Concepts and Techniques (The Morgan Kaufmann Series in Data Management Systems) 3rd Edition”, Elsevier Science Ltd, USA, 22 June, 2011, pp. 1-703.
- [6] KDD Cup 1999. [Online] (2018, Oct.). Available: <http://kdd.ics.uci.edu/databases/kddcup99/>.
- [7] L. Heng, and T. Weise, “Intrusion Detection System Using Convolutional Neuronal Networks: A Cognitive Computing Approach for Anomaly Detection based on Deep Learning”, 2019 IEEE 18th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC), IEEE, Milan, Italy, 23-25 July, 2019, pp. 34-40.
- [8] N. Fu, N. Kamili, Y. Huang, and J. Shi, “A Novel Deep Intrusion Detection Model Based on a Convolutional Neural Network”, 26th International Conference, ICONIP 2019, Springer, Sydney, NSW, Australia, December 12–15, 2019, pp. 52-59.

- [9] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, “A Deep Learning Approach to Network Intrusion Detection”, IEEE Transactions on Emerging Topics in Computational Intelligence, Volume: 2, Issue: 1, IEEE, February, 2018, pp. 41-50.
- [10] R. Primartha, and B. A. Tama, “Anomaly Detection using Random Forest: A Performance Revisited”, 2017 International Conference on Data and Software Engineering (ICoDSE), IEEE, Palembang, Indonesia, 01-02 November, 2017, pp. 12-17.
- [11] S. Paliwal, and R. Gupta, “Denial-of-service, Probing & Remote to User (R2L) Attack Detection Using Genetic Algorithm”, International Journal of Computer Applications, Volume 60– No.19, December, 2012, pp. 57-62.
- [12] Y. Ding, and Y. Zhai, “Intrusion Detection System for NSL-KDD Dataset Using Convolutional Neural Networks”, 2018 2nd International Conference on Computer Science and Artificial Intelligence (CSAI 2018), Association for Computing Machinery, New York, NY, United States, December, 2018, pp. 81-85.
- [13] Y. LeCun, Y. Bengio, and G. Hinton, “Deep Learning”, vol. 521, Nature, London, 27 May, 2015, pp. 436-444.