GOVERNMENT OF KARNATAKA

DEPARTMENT OF TECHNICAL EDUCATION

GOVERNMENT ENGINEERING COLLEGE
DEVAGIRI, HAVERI-581110

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

A

Project report

On

"INTRUSION DETECTION SYSTEM"

Submitted In the Partial Fulfillment for the Degree of

BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE AND ENGINEERING

Submitted By

LAXMI MAIGUR	**2GO19CS018**
VAISHNAVI G S	**2GO19CS037**
ZEHRAKHATOON	**2GO19CS041**
ANUSHA G S	**2GO19CS042**

Under the Guidance of

Dr. SHIVAPRAKASH

Assistant professor

VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI

[2022-2023]

GOVERNMENT OF KARNATAKA

DEPARTMENT OF TECHNICAL EDUCATION

GOVERNMENT ENGINEERING COLLEGE DEVAGIRI, HAVERI

(Affiliated to Visvesvaraya Technological University)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Certificate

This is to certify that the main project entitled **"**INTRUSION DETECTION SYSTEM**"** carried out by **Vaishnavi G S USN: 2GO19CS037** is bona-fide student of Government Engineering College, Haveri in **partial fulfillment for the award of Bachelor of Engineering in Computer Science and Engineering**of the Visvesvaraya Technological University, Belagavi during the year 2022- 2023. The Project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

Signature of Guide
Dr. SHIVAPRAKASH

Signature of the Coordinator
Prof. D. Chauhan

Signature of HOD
Prof. D. Chauhan

Signature of Principal
Dr. Jagadish Kori

Name of Examiner's:

1. ……

2. ……

ACKNOWLEDGEMENT

The sense of contentment and elation that accomplishes the successful of completion of our task would be incomplete without mentioning the names of the people who helped in accomplishment of this Main Project, whose constant guidance, support and encouragement resulted in its realization.

We would greatly mention the enthusiastic influence provided by **Dr. Shivprakash** Main Project Guide, for their ideas and cooperation showed on us during venture and making this Final Project a great success.

We are greatly thankful to **Prof. D Chauhan,** HOD, Department of computer science and engineering, for his co-operation and encouragement at all moments of our approach.

We take this opportunity to thank the Principal, **Dr.Jagadish Kori,** GEC, HAVERI, for being kind enough to provide us an opportunity to work on a Final Project in this esteemed institution.

We also extend our thanks to all the faculty members of Computer Science Department, GEC Haveri, who have encouraged us throughout the course of bachelor engineering.

VAISHNAVI G S 2GO19CS037

ABSTRACT

Network intrusion detection is an important component of network security. Currently, the popular detection technology used the traditional machine learning algorithms to train the intrusion samples, so as to obtain the intrusion detection model. However, these algorithms have the disadvantage of low detection rate. Deep learning is more advanced technology that automatically extracts features from samples. In view of the fact that the accuracy of intrusion detection is not high in traditional machine learning technology, this paper proposes a network intrusion detection model based on convolutional neural network algorithm. The model can automatically extract the effective features of intrusion samples, so that the intrusion samples can be accurately classified. Experimental results on KDD99 datasets show that the proposed model can greatly improve the accuracy of intrusion detection. Index Terms Network Security, Cyber Security, Intrusion Detection, CNN.

DECLARATION

This is to declare that the dissertation work entitled “INTRUSION DETECTION SYSTEM” is a bonafied work carried out by us at GEC HAVERI in partial fulfilment of the requirement for the award of the degree of BACHELOR OF ENGINEERING ON COMPUTER SCIENCE AND ENGINEERING OF VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI under the guidance of Dr. Shivprakash further it is declared to the best of our knowledge the work reported here in, does not form part of any other thesis or dissertation on the basis of whichany other candidate was conferred a degree or award on earlier occasion.

Place: Haveri
Date:

VAISHNAVI G S

CONTENTS

TITLE	PAGE NO
ACKNOWLEDGEMENT	i.
ABSTRACT	ii
LIST OF FIGURES	iii
LIST OF TABLES	iv
CHAPTER 1 INTRODUCTION	**1**
1.1 Motivation	5
1.2 Problem System	6
1.3 Proposed System	6
1.4 Objectives	6
1.5 Literature Survey	7
CHAPTER 2 REQUIREMENTS	**9**
2.1 Software Requirement Specification	9
2.2 Specific Requirements	10
CHAPTER 3 THEORETICAL BACKGROUND	**11**
CHAPTER 4 MODULES	**14**
4.1 Data Collection	14
4.2 Data Pre-Processing	14
4.3 Data Splitting	15
4.4 Product Function	15
CHAPTER 5 DESIGNS	**16**
5.1 Data flow Diagram	16
5.2 Activity Diagram	17
5.3 Sequence Diagram	17

CHAPTER 6 IMPLEMENTATION **20**

6.1 Recurrent Neural Network 23

6.2 Long Short Term Memory 24

6.3 Decision Tree Algorithm 26

CHAPTER 7 TESTING **28**

7.1 System Testing 28

7.2 Unit Testing 28

7.3 Integration Testing 29

7.4 Acceptance Testing 29

7.5 Test Cases 30

CHAPTER 8 RESULT **32**

CHAPTER 9 ADVANTAGES, DISADVANTAGES AND APPLICATIONS **36**

CONCLUSION **37**

FUTURE ENHANCEMENT **38**

REFERENCE

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
3.1	Software Requirements	13
4.1	Workflow	14
5.1	Data Flow Diagram	16
5.1.1	DFD Level 0	17
5.1.2	DFD Level 1	17
5.2	Activity Diagram	18
5.3	Sequence Diagram	19
5.3.1	Use Case Diagram	19
6.1	Recruitment Neural Network	24
8.1	User Login Page	32
8.2	Display Web Page	33
8.4	Data Input with extracting Features Of Attacks	34
8.6	Detect the Attacks Extracting Features	35

LIST OF TABLES

TABLE NO	TITLE	PAGE NO
7.5	Test Cases	29

CHAPTER 1

INTRODUCTION

Nowadays Machine Learning is play more important role in the business as well as in scientific. Machine learning comes with many technologies like deep learning, which helps classification techniques.it helps in recommendation process easily. In recent years, network attack detection attracts increasing interest in social networking information security as the increasing security threats. With the inexorably profound reconciliation of the Internet and society, the Internet is changing the manner by which individuals live, study and work, yet the different security dangers that we face are turning out to be increasingly genuine. An Intrusion Detection System (IDS), a huge research accomplishment in the data security field, can distinguish an attack, which could be a continuous intrusion or an interruption that has just happened. In this paper we are distinguishing whether system traffic conduct is ordinary or abnormal, or a five-classification arrangement issue, i.e., recognizing whether it is typical or any of the other four assault types: Denial of Service (DOS), User to Root (U2R), Probe (Probing) and Root to Local (R2L). To put it plainly, the primary inspiration of interruption recognition is to improve the exactness of classifiers in adequately distinguishing the meddling conduct.

An Intrusion Detection System (IDS) is a device or software application that monitors a network or systems for malicious activity or policy violations. Any intrusion activity or violation is typically reported either to an administrator or collected centrally using a security information and event management (SIEM) system. A SIEM system combines outputs from multiple sources and uses alarm filtering techniques to distinguish malicious activity from false alarms.

Although Intrusion Detection Systems monitor networks for potentially malicious activity, they are also disposed to false alarms. Hence, organizations need to fine-tune their IDS products when they first install them. It means properly setting up the intrusion detection systems to recognize what normal traffic on the network looks like as compared to malicious activity.

Classification of Intrusion Detection System IDS are classified into 5 types

1. Network Intrusion Detection System (NIDS) Network intrusion detection systems (NIDS) are set up at a planned point within the network to examine traffic from all devices on the network. It performs an observation of passing traffic on the entire subnet and matches the traffic that is passed on the subnets to the collection of known attacks. Once an attack is identified or abnormal behaviour is observed, the alert can be sent to the administrator. An example of an NIDS is installing it on the subnet where firewalls are located in order to see if someone is trying crack the firewall.

2. Host Intrusion Detection System (HIDS)

Host intrusion detection systems (HIDS) run on independent hosts or devices on the network. A HIDS monitors the incoming and outgoing packets from the device only and will alert the administrator if suspicious or malicious activity is detected. It takes a snapshot of existing system files and compares it with the previous snapshot. If the analytical system files 1 were edited or deleted, an alert is sent to the administrator to investigate. An example of HIDS usage can be seen on mission critical machines, which are not expected to change their layout.

3. Protocol-based Intrusion Detection System (PIDS)

Protocol-based intrusion detection system (PIDS) comprises of a system or agent that would consistently resides at the front end of a server, controlling and interpreting the protocol between a user/device and the server. It is trying to secure the web server by regularly monitoring the HTTPS protocol stream and accept the related HTTP protocol. As HTTPS is unencrypted and before instantly entering its web presentation layer then this system would need to reside in this interface, between to use the HTTPS.

4. Application Protocol-based Intrusion Detection System (APIDS)

Application Protocol-based Intrusion Detection System (APIDS) is a system or agent that generally resides within a group of servers. It identifies the intrusions by monitoring and interpreting the communication on application specific protocols. For example, this would monitor the SQL protocol explicit to the middleware as it transacts with the database in the web server.

5. Hybrid Intrusion Detection System

Hybrid intrusion detection system is made by the combination of two or more approaches of the intrusion detection system. In the hybrid intrusion detection system, host agent or system data is combined with network information to develop a complete view of the network system. Hybrid intrusion detection system is more effective in comparison to the other intrusion detection system.

Prelude is an example of Hybrid IDS. Detection Method of IDS

1. Signature-based Method

Signature-based IDS detects the attacks on the basis of the specific patterns such as number of bytes or number of 1's or number of 0's in the network traffic. It also detects on the basis of the already known malicious instruction sequence that is used by the malware. The detected patterns in the IDS are known as signatures.

2. Anomaly-based Method

Anomaly-based IDS was introduced to detect the unknown malware attacks as new malware are developed rapidly. In anomaly-based IDS there is use of machine learning to create a trustful activity model and anything coming is compared with that model and it is declared suspicious if it is not found in model. Machine learning based method has a better generalized property in comparison to signature-based IDS as these models can be trained according to the applications and hardware configurations.

Comparison of IDS with Firewalls

IDS and firewall both are related to the network security but an IDS differs from a firewall as a firewall looks outwardly for intrusions in order to stop them from happening. Firewalls restrict access between networks to prevent intrusion and if an attack is from inside the network it don't signal. An IDS describes a suspected intrusion once it has happened and then signals an alarm.

Need for IDS

Building a reliable network is a very difficult task considering all different possible types of attacks. Nowadays, computer networks and their services are widely used in industry, business,

and all arenas of life. Security personnel and everyone who has a responsibility for providing protection for a network and its users, have serious concerns about intruder attacks.

Network administrators and security officers try to provide a protected environment for user's accounts, network resources, personal files and passwords. Attackers may behave in two ways to carry out their attacks on networks; one of these ways is to make a network service unavailable for users or violating personal information. Denial of service (DoS) is one of the most frequent cases representing attacks on network resources and making network services unavailable for their users. There are many types of DoS attacks, and every type has it is own behaviour on consuming network resources to achieve the intruder's aim, which is to render the network unavailable for its users. Remote to user (R2L) is one type of computer network attacks, in which an intruder sends set of packets to another computer or server over a network where he/she does not have permission to access as a local user. User to root attacks (U2R) is a second type of attack where the intruder tries to access the network resources as a normal user, and after several attempts, the intruder becomes as a full access user. Probing is a third type of attack in 3 which the intruder scans network devices to determine weakness in topology design or some opened ports and then use them in the future for illegal access to personal information. There are many examples that represent probing over a network, such as nmap, portsweep, ipsweep.

IDS becomes an essential part for building computer network to capture these kinds of attacks in early stages, because IDS works against all intruder attacks. IDS uses classification techniques to make decision about every packet pass through the network whether it is a normal packet or an attack (i.e. DOS, U2R, R2L, PROBE) packet. Software to detect network intrusions protects a computer network from unauthorized users, including perhaps insiders. The intrusion detector learning task is to build a predictive model (i.e. a classifier) capable of distinguishing between ``bad'' connections, called intrusions or attacks, and ``good'' normal connections.

Lincoln Labs set up an environment to acquire nine weeks of raw TCP dump data for a local- area network (LAN) simulating a typical U.S. Air Force LAN. They operated the LAN as if it were a true Air Force environment, but peppered it with multiple attacks. The raw training data was about four gigabytes of compressed binary TCP dump data from seven weeks of network traffic. This was processed into about five million connection records. Similarly, the two weeks of test data yielded around two million connection records.

A connection is a sequence of TCP packets starting and ending at some well-defined times, between which data flows to and from a source IP address to a target IP address under some well-defined protocol. Each connection is labelled as either normal, or as an attack, with exactly one specific attack type. Each connection record consists of about 100 bytes.

Different classes of Attacks

Denial of Service (DoS)

An attacker tries to prevent legitimate users from using a service. For example, SYN flood, Smurf and teardrop. **User to Root (U2R)**

An attacker has local access to the victim machine and tries to gain super-user privilege. For example, buffer overflow attacks. **Remote to Local (R2L)**

An attacker tries to gain access to victim machine without having an account on it. For example, password guessing attack.

Probe

An attacker tries to gain information about the target host. For example, port-scan and ping- sweep.

1.1 Motivation

Developing absolutely secure systems is not possible

- Most existing systems have security flaws Abuses by privileged insiders are possible
- Not all kinds of intrusions are known
- Quick detection of intrusions can help to identify intruders and limit damage
- IDS serves as a deterrent

The contributions of proposed work are:

1) Identifying attack class by applying machine learning algorithm,

2) Identifying which algorithm is best suitable for IDS problem to effectively resist insider attack.

Intrusion detection system uses classification techniques to make decision about every packet

pass through the network whether it is a normal packet or an attack. Our objective is to classify the attack into multiple attack types namely DOS, U2R, R2L, PROBE packet.

1.2 PROBLEM STATEMENT

Intrusion detection begins where the firewall ends. Preventing unauthorized entry is best, but not always possible. It is important that the system is reliable and accurate and secure. Intrusion detection is defined as real-time monitoring and analysis of network activity and data for potential Vulnerabilities and attacks in progress.

One major limitation of current intrusion detection system (IDS) technologies is the requirement to filter false alarms. IDS is defined as a system that tries to detect and alert of attempted intrusions into a system or a network. IDSs are classified into two major approaches. Intrusion detection is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of possible incidents, which are violations or imminent threats of violation of computer security policies, acceptable use policies, or standard security practices. Intrusion prevention is the process of performing intrusion detection and attempting to stop detected possible incidents.

1.3 PROPOSED SYSTEM

- The RNN-IDS model not only has a strong modelling ability for intrusion detection, but also has high accuracy in both binary and multiclass classification.
- The model can effectively improve both the accuracy of intrusion detection and the ability to recognize the intrusion type.
- Helps in identifying the network traffic behaviour is normal or anomalous, or a five-category classification problem.
- Finding the network attack types.

1.4 OBJECTIVES

- To achieve an network intrusion detection system based on convolutional neural networks.
- To achieve an intrusion detection system which is designed to detect almost all type of millicious network attacks.

1.5 LITERATURE SURVEY

INTRUSION DETECTION SYSTEM – A STUDY Dr. S.Vijayarani1 and Ms. Maria Sylviaa.S 1Assistant Professor, Department of Computer Science, Bharathiar University, Coimbatore. 2M.Phil Research Scholar,Department of Computer Science, Bharathiar University, Coimbatore. Intrusion Detection System (IDS) is meant to be a software application which monitors the network or system activities and finds if any malicious operations occur. Tremendous growth and usage of internet raises concerns about how to protect and communicate the digital information in a safe manner.

Nowadays, hackers use different types of attacks for getting the valuable information. Many intrusion detection techniques, methods and algorithms help to detect these attacks

This main objective of this paper is to provide a complete study about the definition of intrusion detection, history, life cycle, types of intrusion detection methods, types of attacks, different tools and techniques, research needs, challenges and applications.

In this examination, an artificial insight (AI) interruption recognition framework utilizing a profound neural system (DNN) was explored and tried with the KDD Cup 99 dataset in light of consistently advancing system assaults. To start with, the information were pre-processed through information change and standardization for contribution to the DNN model. The DNN calculation was applied to the information refined through pre-processing to make a learning model, and the whole KDD Cup 99 dataset was utilized to confirm it. At last, the precision, discovery rate, and bogus alert rate were determined to find out the location efficacy of the DNN model, which was found to produce great outcomes for interruption recognition.[In this paper present an examination, routed to security pros, of AI strategies applied to the discoe

[2]In this paper present an examination, routed to security pros, of AI strategies applied to the discovery of interruption, malware, and spam. 2

The objective is twofold: to evaluate the present development of these arrangements and to recognize their primary restrictions that counteract a prompt selection of AI digital discovery plans. Our decisions depend on a broad survey of the writing just as on analyses performed on genuine undertaking frameworks and system traffic. [4] we propose a constant aggregate oddity identification model dependent on neural system learning. Regularly a Long Short-Term Memory

Recurrent Neural Network (LSTM RNN) is prepared distinctly on typical information and it is equipped for anticipating a few time ventures in front of an information. In our methodology, a LSTM RNN is prepared with typical time arrangement information before playing out a live forecast for each time step.[8][10] To achieve high detection rate, data pre- processing, feature abstraction and multi-channel training and detection are seamlessly integrated into an end-to-end detection framework. Data pre-processing provides high-quality data for subsequent processing, then different types of features are extracted from the processed data.

The simulation results show that our simulation system has a good approximation and can be used for intrusion detection in Tor networks. [1] In this paper, model of an interruption discovery framework is investigated dependent on profound learning, and Long Short Term Memory (LSTM) design is applied to a Recurrent Neural Network (RNN) and train the IDS model utilizing KDD Cup 1999 dataset. Through the exhibition test, it is affirmed that the profound neural system is successful for NIDS.[4] THIS paper shows the aftereffects of a writing overview of AI (ML) and information mining (DM) techniques for digital security applications. The ML/DM strategies are portrayed, just as a few utilizations of every strategy to digital interruption location issues.

CHAPTER 2

REQUIREMENTS

2.1 SOFTWARE REQUIREMENT SPECIFICATION

2.1.1 Purpose

The objective of the pre-feasibility study is primarily to facilitate potential entrepreneurs in project identification for investment. The project pre-feasibility may form the basis of an important investment decision and in order to serve this objective, the document/study covers various aspects of project concept development, start-up, and production, marketing, finance and business management.

The purpose of the document is to collect and analyse all assorted ideas that have come up to define the system, its requirements with respect to consumers. Also, we shall predict and sort out how we hope this product will be used in order to gain a better understanding of the project, outline concepts that may be developed later, and document ideas that are being considered, but may be discarded as the product develops.

In short, the purpose of this SRS document is to provide a detailed overview of our software product, its parameters and goals. This document describes the project's target audience and its user interface, hardware and software requirements. It defines how our client, team and audience see the product and its functionality. Nonetheless, it helps any designer and developer to assist in software delivery lifecycle (SDLC) processes.

2.1.2 Scope:

The purpose of this document is to facilitate potential investors in intrusion prediction Service by providing them with a general understanding of the business with the intention of supporting potential investors in crucial investment decisions. The need to come up with pre-feasibility reports for undocumented or minimally documented sectors attains greater imminence as the research that precedes such reports reveal certain thumb rules; best practices developed by existing enterprises by trial and error, and certain industrial norms that become a guiding source regarding various aspects of business set-up and it’s successful management.

2.1.3 Feasibility Study:

This feasibility study analyses the market dynamics and financials of intrusion detection service, which is proposed for bigger cities across the country. The proposed prediction is to make future prediction based on current datasets so the users or the investors can check future prediction and invest.

Operational Feasibility:

The Intrusion Detection provides the good operational feasibility like the application should contain the rich set of operations like functionality that to understand by the user easily.it helps to find the resources very easily and more effective manner.

Economic Feasibility:

The main aim of the ―Intrusion Detection‖ is to make the product in very low cost so that the everyone can understand and invest easily. **Motivational feasibility:**

The motivational feasibility provides the effective user interface and helps the developer to motivate. There are many stakeholders of the system which helps the user to flow of the application according to they need. The stake holders provide the details to the developer in an easy understanding way.

2.2 SPECIFIC REQUIREMENTS

Hardware Requirements

Processor : Intel i5 3.30 GHz.

Hard Disk: 40 GB (min)

Ram : 8GB

Software Requirements

Operating system : Windows 10.

Coding Language: Python.

CHAPTER 3

THEORETICAL BACKGROUND

Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library. Python interpreters are available for many operating systems.

C Python, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations. C Python is managed by the non-profit Python Software Foundation

The objective of the pre-feasibility study is primarily to facilitate potential entrepreneurs in project identification for investment. The project pre-feasibility may form the basis of an important investment decision and in order to serve this objective, the document/study covers various aspects of project concept development, start-up, and production, marketing, finance and business management.

The purpose of the document is to collect and analyse all assorted ideas that have come up to define the system, its requirements with respect to consumers. Also, we shall predict and sort out how we hope this product will be used in order to gain a better understanding of the project, outline concepts that may be developed later, and document ideas that are being considered, but may be discarded as the product develops.

In short, the purpose of this SRS document is to provide a detailed overview of our software product, its parameters and goals. This document describes the project's target audience and its user interface, hardware and software requirements. It defines how our client, team and audience see the product and its functionality. Nonetheless, it helps any designer and developer to assist in software delivery lifecycle (SDLC) processes.

Scikit Learn

Scikit-learn(fomely scikits.learn and also known as sklearn) is a free software machine learning library for the python programming language. It features various classification, regression and clustering algorithms including support vector machine, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy. Scikit-learn is largely written in Python, and uses numpy extensively for high-performance linear algebra and array operations. Furthermore, some core algorithms are written in Cython to improve performance. Support vector machines are implemented by a Cython wrapper around LIBSVM;logistic regression and linear support vector machines by a similar wrapper around LIBLINEAR. In such cases, extending these methods with Python may not be possible.

Tensor flow

Tensor Flow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google. Tensor Flow is Google Brain's second-generation system. Version 1.0.0 was released on February 11, 2017. While the reference implementation runs on single devices, Tensor Flow can run on multiple CPUs and GPUs (with optional CUDA and SYCL extensions for general- purpose computing on graphics processing units). Tensor Flow is available on 64-bit Linux, macOS, Windows, and mobile computing platforms including Android and iOS.

NumPy

NumPy or sometimes /ˈnʌmpi/(*NUM-pee*)) is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices along with a large collection of high-level mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is open-source software and has many contributors.

Software architecture

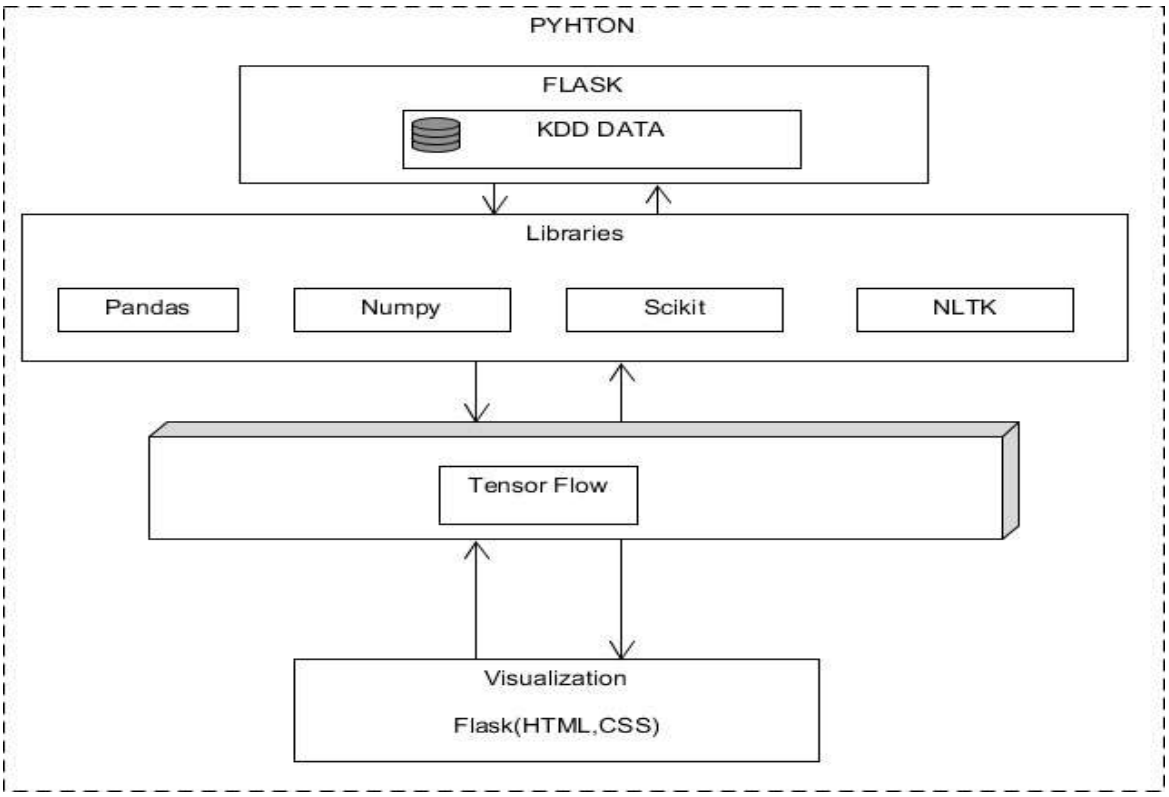


Fig 3.1 Software architecture

CHAPTER 4

MODULES

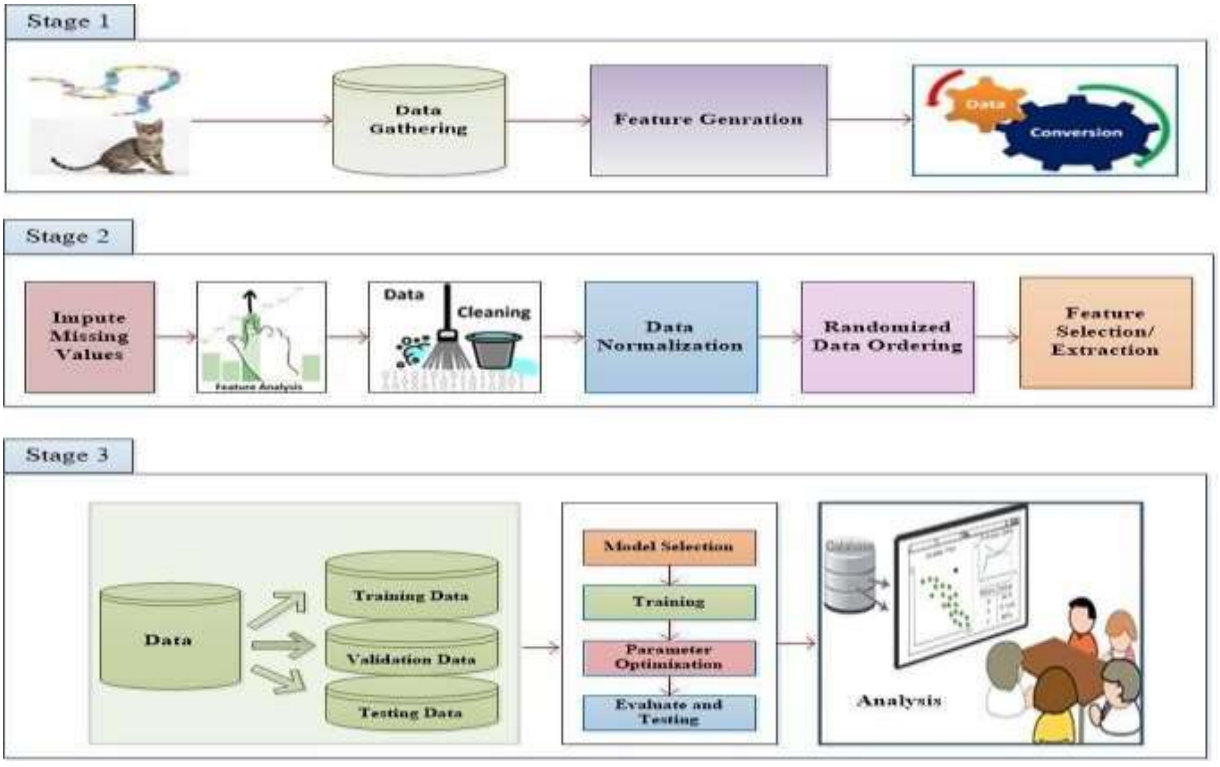


Fig 4.1 Workflow

4.1 Data collection

The data collection process involves the selection of quality data for analysis. Here we used KDD intrusion dataset taken from kaggle for machine learning implementation. The job of a data analyst is to find ways and sources of collecting relevant and comprehensive data, interpreting it, and analyzing results with the help of statistical techniques.

4.2 Data pre-processing

The purpose of pre-processing is to convert raw data into a form that fits machine learning. Structured and clean data allows a data scientist to get more precise results from an applied machine learning model. The technique includes data formatting, cleaning, and sampling.

4.3 Dataset splitting

A dataset used for machine learning should be partitioned into three subsets — training, test, and validation sets.

Training set: A data scientist uses a training set to train a model and define its optimal parameters it has to learn from data.

Test set: A test set is needed for an evaluation of the trained model and its capability for generalization. The latter means a model's ability to identify patterns in new unseen data after having been trained over a training data. It's crucial to use different subsets for training and testing to avoid model over-fitting, which is the incapacity for generalization we mentioned above.

4.4 PRODUCT FUNCTION

- Collected datasets of Intrusion detection from github from KDD datasest
- Pre-processing of obtained datasets
- Select Attributes which helps in predicting the Intrusion detection
- The selected datasets are trained using RNN
- The trained data sets are tested for Accuracy
- The obtained result is showed in the graph

CHAPTER 5

DESIGNS

5.1 DATA FLOW DIAGRAM

A data flow diagram (DFD) is a way of representing a flow of data of a process or a system (usually an information system). The DFD also provides information about the output and inputs of each entity and the process itself. A data flow diagram has no control flow, there are no decision rules and no loops. Specific operations based on the data can be represented by a flowchart. There are several notations for displaying data flow diagrams.

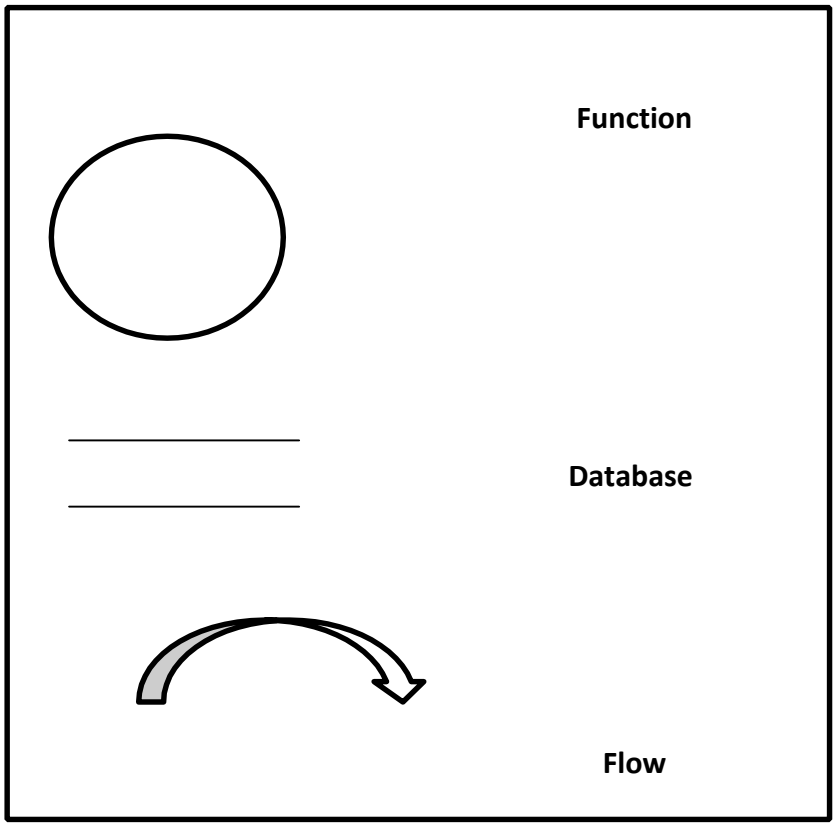


Fig 5.1 Data Flow Diagram

5.1.1 DFD Level 0:

Fig 5.1.1 DFD Level 0

5.1.2 DFD Level 1:

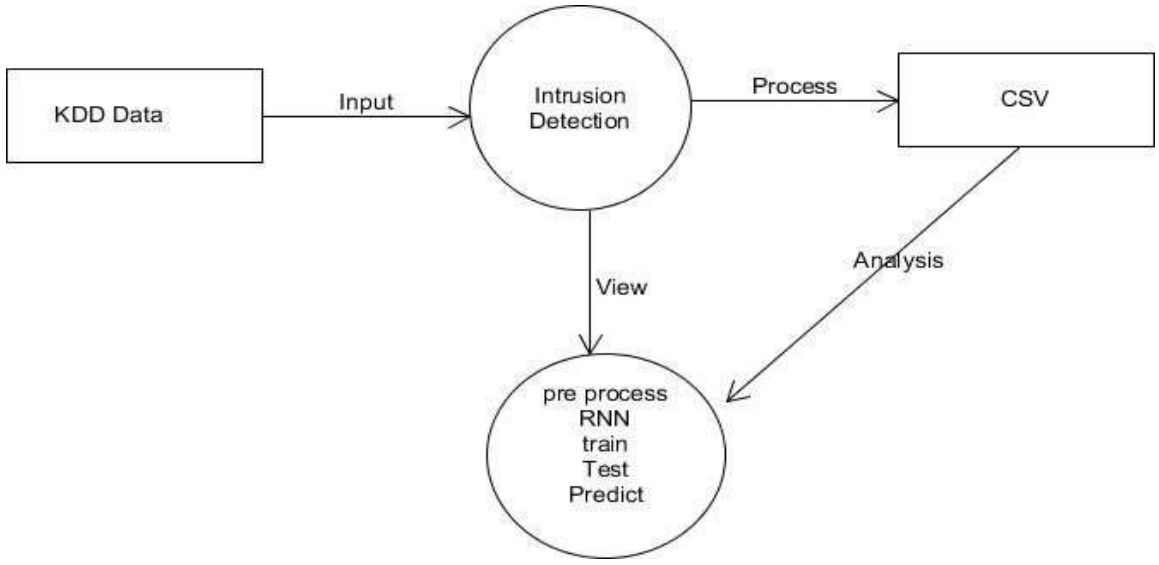


Fig 5.1.2 DFD Level 1

5.2 ACTIVITY DIAGRAM

An activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. Activity diagrams are often used in business process

modelling. They can also describe the steps in a use case diagram. Activities modelled can be sequential and concurrent. In both cases an activity diagram will have a beginning (an initial state) and an end (a final state).

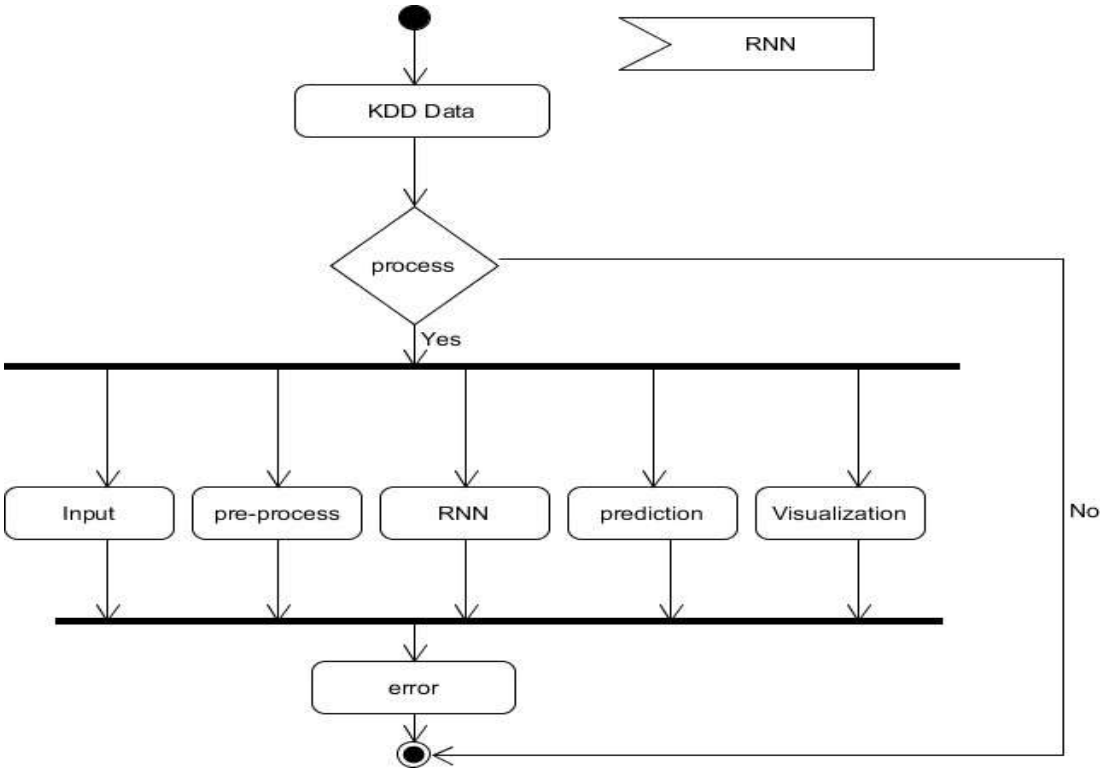


Fig 5.2 Activity Diagram

5.3 SEQUENCE DIAGRAM

Sequence diagrams describe interactions among classes in terms of an exchange of messages over time. They're also called event diagrams. A sequence diagram is a good way to visualize and validate various runtime scenarios. These can help to predict how a system will behave and to discover responsibilities a class may need to have in the process of modelling a new system.

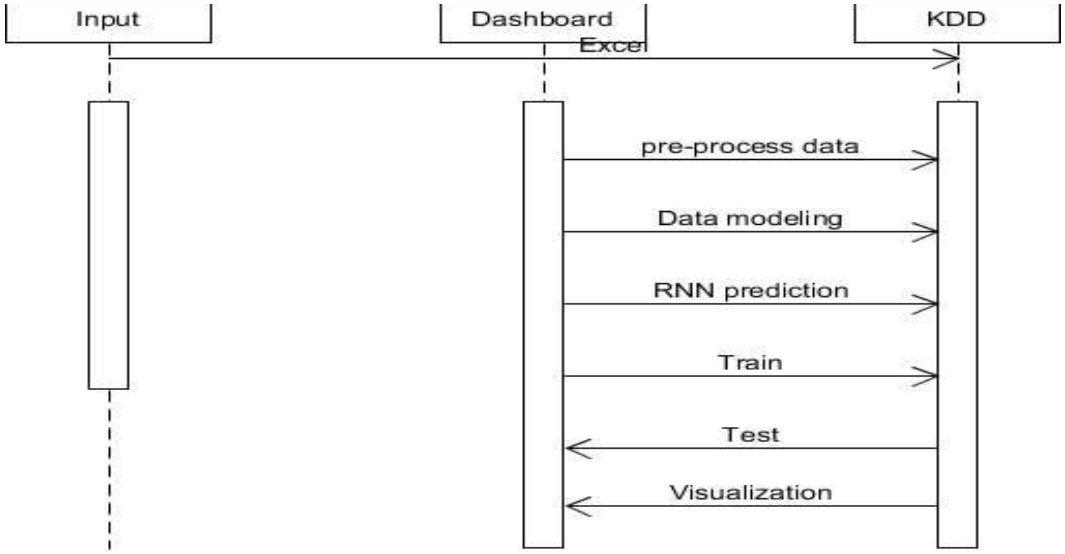


Fig 5.3 Sequence Diagram

5.3.1 USE CASE DIAGRAM

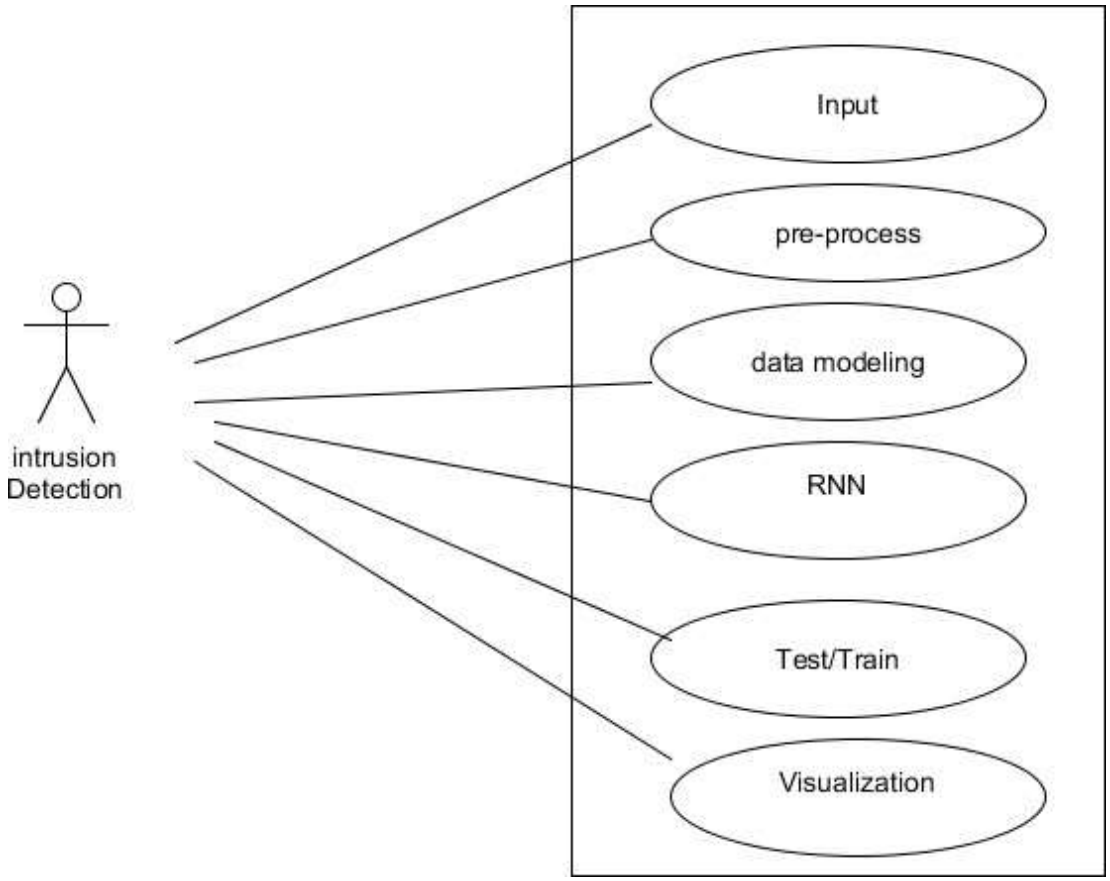


Fig 5.3.1 Use Case Diagram

CHAPTER 6
IMPLEMENTATION

The project is implemented using Python which is an object oriented programming language and procedure oriented programming language. Object oriented programming is an approach that provides a way of modularizing program by creating partitioned memory area of both data and function that can be used as a template for creating copies of such module on demand.

This project is implemented using python programming language. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library. The machine Learning techniques are used in this project.

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended users and the operation of the system. The people are not sure that the software is meant to make their job easier.

- The active user must be aware of the benefits of using the system
- Their confidence in the software built up
- Proper guidance is impaired to the user so that he is comfortable in using the application

Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not running on the server, the actual processes will not take place.

User Training

To achieve the objectives and benefits expected from the proposed system it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for education and training is more and more important. Education is complementary to training. It brings life to formal training by explaining the background to the resources for them. Education involves creating the right atmosphere and motivating user staff. Education information can make training more interesting and more understandable.

Training on the Application Software

After providing the necessary basic training on the computer awareness, the users will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design, type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the data entered. This training may be different across different user groups and across different levels of hierarchy.

Operational Documentation

Once the implementation plan is decided, it is essential that the user of the system is made familiar and comfortable with the environment. A documentation providing the whole operations of the system is being developed. Useful tips and guidance is given inside the application itself to the user. The system is developed user friendly so that the user can work the system from the tips given in the application itself.

System Maintenance

The maintenance phase of the software cycle is the time in which software performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is to make adaptable to the changes in the system environment. There may be social, technical and other environmental changes, which affect a system which is being implemented. Software product enhancements may involve providing new functional capabilities, improving user displays and mode of interaction, upgrading the performance characteristics of the system. So only thru proper system maintenance procedures, the system can be adapted to cope up with these changes. Software maintenance is of course, far more than ―finding mistakes‖.

Corrective Maintenance

The first maintenance activity occurs because it is unreasonable to assume that software testing will uncover all latent errors in a large software system. During the use of any large

program, errors will occur and be reported to the developer. The process that includes the diagnosis and correction of one or more errors is called Corrective Maintenance.

Adaptive Maintenance

The second activity that contributes to a definition of maintenance occurs because of the rapid change that is encountered in every aspect of computing. Therefore, Adaptive maintenance termed as an activity that modifies software to properly interfere with a changing environment is both necessary and commonplace.

Perceptive Maintenance

The third activity that may be applied to a definition of maintenance occurs when a software package is successful. As the software is used, recommendations for new capabilities, modifications to existing functions, and general enhancement are received from users. To satisfy requests in this category, Perceptive maintenance is performed. This activity accounts for the majority of all efforts expended on software maintenance.

Preventive Maintenance

The fourth maintenance activity occurs when software is changed to improve future maintainability or reliability, or to provide a better basis for future enhancements. Often called preventive maintenance, this activity is characterized by reverse engineering and re-engineering techniques.

Machine Learning Vs Deep Learning

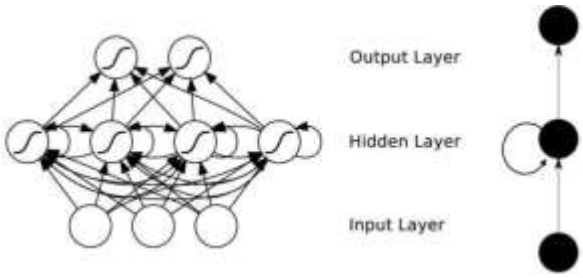
Experts in machine learning and deep learning have not yet reached consensus on these concepts. in this context, almost every day new ideas are being discussed. Machine Learning is an older concept than Deep Learning. Deep learning can also be called a technique that performs machine learning. The differences are listed below;

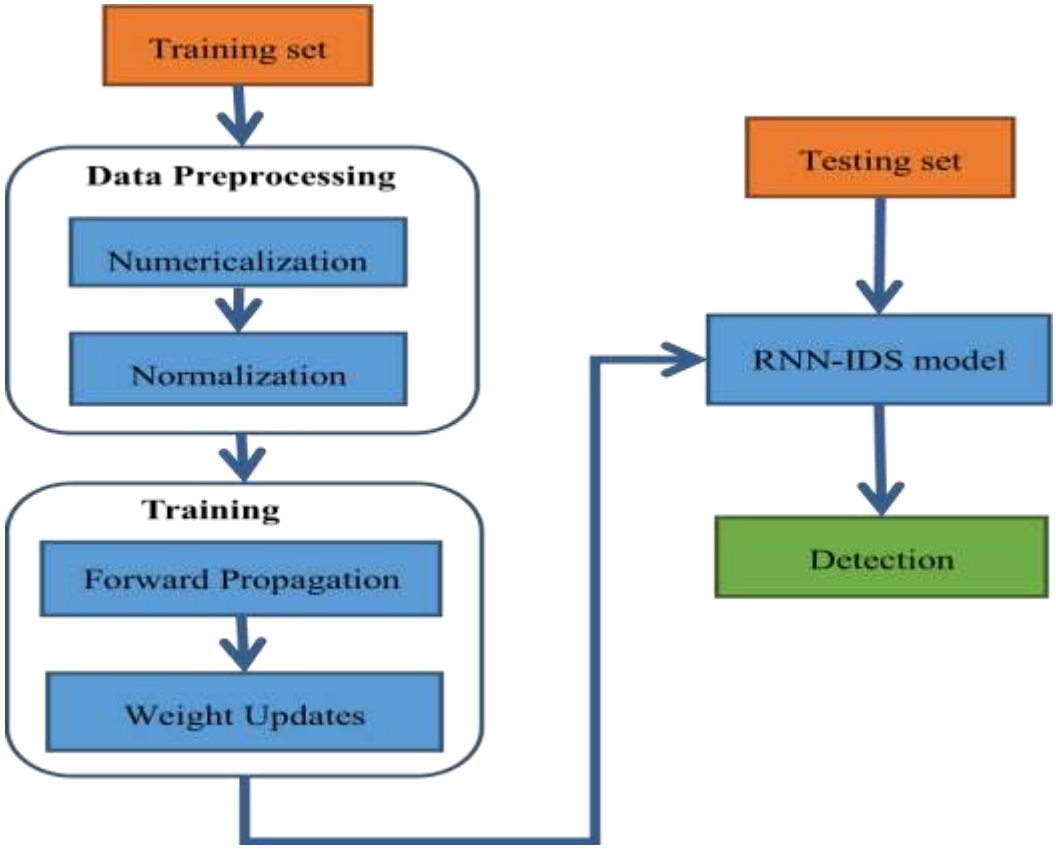
1) In deep learning, too much data is needed to bring the algorithm structure to the ideal.In machine learning, the problem can be solved with much less data because the person gives specific features to the algorithm.

2) Deep learning algorithms try to extract features from data. In machine learning, the features are determined by the expert.
3) While Deep Learning algorithms work on high performance machines, Machine Learning algorithms can work on ordinary CPUs.
4) In machine learning, the problem is usually divided into pieces, these parts are solved one by one and then the solutions are formed as a result of the solutions. In deep learning, the problem is solved end-to-end.
5) It takes a long time to train deep learning algorithms.

6.1 RECURRENT NEURAL NETWORK

Recurrent neural networks include input units, output units and hidden units, and the hidden unit completes the most important work. The RNN model essentially has a one-way flow of information from the input units to the hidden units, and the synthesis of the one-way information flow from the previous temporal concealment unit to the current timing hiding unit is shown in Fig. 1. We can regard hidden units as the storage of the whole network, which remember the endto-end information. When we unfold the RNN, we can find that it embodies the deep learning. A RNNs approach can be used for supervised classification learning.





6.1 RECURRENT NEURAL NETWORK

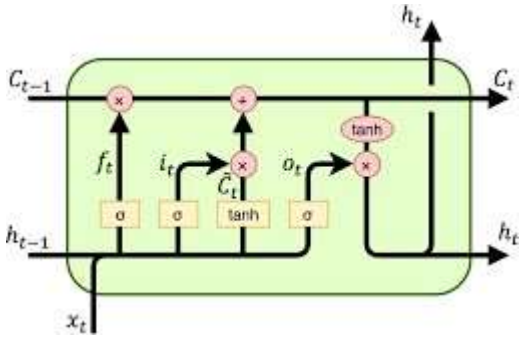
RNN Block Diagram

Recurrent neural networks have introduced a directional loop that can memorize the previous information and apply it to the current output, which is the essential difference from traditional Feed-forward Neural Networks (FNNs). The preceding output is also related to the current output of a sequence, and the nodes between the hidden layers are no longer connectionless; instead, they have connections. Not only the output of the input layer but also the output of the last hidden layer acts on the input of the hidden layer. The step involved in RNN-
IDS

6.2 LONG SHORT TERM MEMORY

Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series.

LSTMs have an edge over conventional feed-forward neural networks and RNN in many ways. This is because of their property of selectively remembering patterns for long durations of time.The purpose of this article is to explain LSTM and enable you to use it in real life problems.



In the case of LSTM architecture, the usual hidden layers are replaced with LSTM cells. The cells are composed of various gates that can control the input flow. An LSTM cell consists of input gate, cell state, forget gate, and output gate. It also consists of sigmoid layer, tanh layer and point wise multiplication operation. The various gates and their functions are as follows

- Input gate: Input gate consists of the input.
- Cell State: Runs through the entire network and has the ability to add or remove information with the help of gates.
- Forget gate layer: Decides the fraction of the information to be allowed.
- Output gate: It consists of the output generated by the LSTM.
- Sigmoid layer generates numbers between zero and one, describing how much of each component should be let through.
- Tanh layer generates a new vector, which will be added to the state.

The cell state is updated based on the outputs form the gates. Mathematically we can represent it using the following equations.

$$f_t = \sigma(W_f.[h_{t-1},x_t] + b_f) \quad (2)$$

$$i_t = \sigma(W_i.[h_{t-1},x_t] + b_i) \quad (3)$$

$$c_t = \tanh(W_c.[h_{t-1},x_t] + b_c) \quad (4)$$

$$ot = \sigma(Wo[ht-1,xt] + bo) \quad (5)$$

$$h_t = o_t * \tanh(c_t) \quad (6)$$

where x_t: input vector, h_t: output vector, c_t: cell state vector, f_t: forget gate vector, i_t: input gate vector, o_t: output gate vector and W,b are the parameter matrix and vector. Convolutional neural networks or CNNs, are a specialized kind of neural network for processing data that has a known, grid-like topology. This include time-series data, which can be thought of as a 1D and image data, which can be thought of as a 2D grid of pixels. The network employs a mathematical operation called convolution and hence known as convolutional neural network. It is a specialized kind of linear operation. Convolutional networks use convolution instead of general matrix multiplication in at least one of their layers. The motivation behind using these three models is to identify whether there is any long term dependency existing in the given data. This can be identified from the performance of the models. RNN and LSTM architectures are capable of identifying long term dependencies and uses them for future prediction. However CNN architectures mainly focuses on the given input sequence and does not use any previous history or information during the learning process. The motivation behind testing the models with data from other companies is to check for interdependencies among the companies and to understand the market dynamics.

The train data was normalized. Test data was also subjected to the same normalization. After obtaining the predicted output, de-normalization was applied and percentage error was calculated using the available true labels. The error percentage was calculated using (7)

$$ep = \frac{abs\left[X^i_{real} - X^i_{predicted}\right]}{X^i_{real}} \times 100 \quad (7)$$

where ep is the error percentage, X_{real}^{i} is the i^{th} real value and $X_{predicted}^{i}$ is the i^{th} predicted value. Error percentage gives the magnitude of error present in the output.

6.3 DECISION TREE ALGORITHM

Decision tree is a type of supervised learning algorithm that is mostly used in classification problems. It works for both categorical and continuous input and output variables. In this technique, we split sample into two or more homogeneous sets (or sub-populations) based on most significant splitter / differentiator in input variables. In decision tree internal node represents a test on the attribute, branch depicts the outcome and leaf represents decision made after computing attribute.

The general motive of using Decision Tree is to create a training model which can be used to predict class or a value of target variables by learning decision rules inferred from prior data (training data).

The understanding level of Decision Tree algorithm is so easy compared with other classification algorithms. The Decision Tree algorithm tries to solve the problem, by using tree representation. Each internal node of the tree corresponds to an attribute, and each leaf node corresponds to a class label.

Decision Tree works in following manner

1. Place the best attribute of the dataset at the root of the tree.
2. Split the training set into subsets. Subsets should be made in such a way that each subset contains data with the same value for an attribute.
3. Repeat step 1 and step 2 on each subset until you find leaf nodes in all the branches of the tree. In decision trees, for predicting a class label for a record we start from the root of the tree. Then compare the values of the root attribute with record's attribute. On the basis of comparison, follow the branch corresponding to that value and jump to the next node.

Decision Tree Classifier is a class capable of performing multi-class classification on a dataset. As with other classifiers, Decision Tree Classifier takes as input two arrays: an array X, sparse or dense, of size [n_samples, n_features] holding the training samples, and an array Y of integer values, size [n_samples], holding the class labels for the training samples.

Pros

- Prone to over fitting
- If you have a lot of features
- Stop growth of tree at the appropriate time You can build bigger classifiers out of this.

Cons

- It involves higher time to train the model.
- It is relatively expensive as complexity and time taken is more.

CHAPTER 7

TESTING

7.1 System Testing

System testing is the stage of implementation, which aimed at ensuring that the system works accurately and efficiently before the live operation commences. Testing is the process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding a yet undiscovered error. A successful test is one that answers a yet undiscovered error.

Testing is vital to the success of the system. System testing makes a logical assumption that if all parts of the system are correct, the goal will be successful achieved. The candidate system is subject to variety of tests-on-line response, Volume Street, recovery and security and usability test. A series of tests are performed before the system is ready for the user acceptance testing. Any engineered product can be tested in one of the following ways. Knowing the specified function that a product has been designed to form, test can be conducted to demonstrate each function is fully operational .Knowing the internal working of a product, tests can be conducted to ensure that ―all gears mesh‖, that is the internal operation of the product performs according to the specification and all internal components have been adequately exercised.

7.2 Unit Testing

Unit testing is the testing of each module and the integration of the overall system is done. Unit testing becomes verification efforts on the smallest unit of software design in the module. This is also known as „module testing". The modules of the system are tested separately. This testing is carried out during the programming itself. In this testing step, each model is found to be working satisfactorily as regard to the expected output from the module. There are some validation checks for the fields. For example, the validation check is done for verifying the data given by the user where both format and validity of the data entered is included. It is very easy to find error and debug the system.

7.3 Integration Testing

Data can be lost across an interface, one module can have an adverse effect on the other sub function, when combined, may not produce the desired major function. Integrated testing is systematic testing that can be done with sample data. The need for the integrated test is to find the overall system performance. There are two types of integration testing, they are:

Top-down integration testing.

Bottom-up integration testing.

7.4 Acceptance Testing

Acceptance testing or User Acceptance Testing (UAT) is a level of the software testing process where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery.

7.5 Test Cases

Test Case	**Test Purpose**	**Test condition**	**Expected outcome**	**Actual result**	**Pass or Fail**
Load Data	Load intrusion data sets In CSV format.	If the data is not in the CSV format, shows a error message.	Load Data sets.	The data is loaded Successfully in CSV format.	Pass

Pre Process data	CSV data	If values are missing, or improper data	Pre-processing is done	As Expected.	Pass
RNN Algorithm	Pre-processed data	KDD datasets Trained for each available data	Training of data is complete	As Expected.	Pass
RNN Algorithm	Pre-processed data	KDD datasets Trained for each available data	Testing of data is complete	As Expected.	Pass
Prediction	Result obtained from RNN	Find networking Attacks	Attacks found	As Expected.	Pass

Failed test Cases

Test Case	**Test Purpose**	**Test condition**	**Expected outcome**	**Actual result**	**Pass or Fail**

Load Data	Load Automobile data sets In CSV format.	If the data is not in the CSV format, shows a error message.	Load Data sets.	The data is not loaded	fail
Pre Process data	CSV data	If values are missing, or improper data	Pre-processing is not done	Result not found	fail
RNN Algorithm	Pre-processed data	KDD datasets Trained for each available data	Training of data is not complete	Error while Training, data not found	Fail
RNN Algorithm	Pre-processed data	KDD datasets Trained for each available data	Testing of data is complete	Error while testing, data not found	Fail
Prediction	Result obtained from RNN	Find networking Attacks	Attacks found	No Attacks or result error	Fail

CHAPTER 8

RESULT

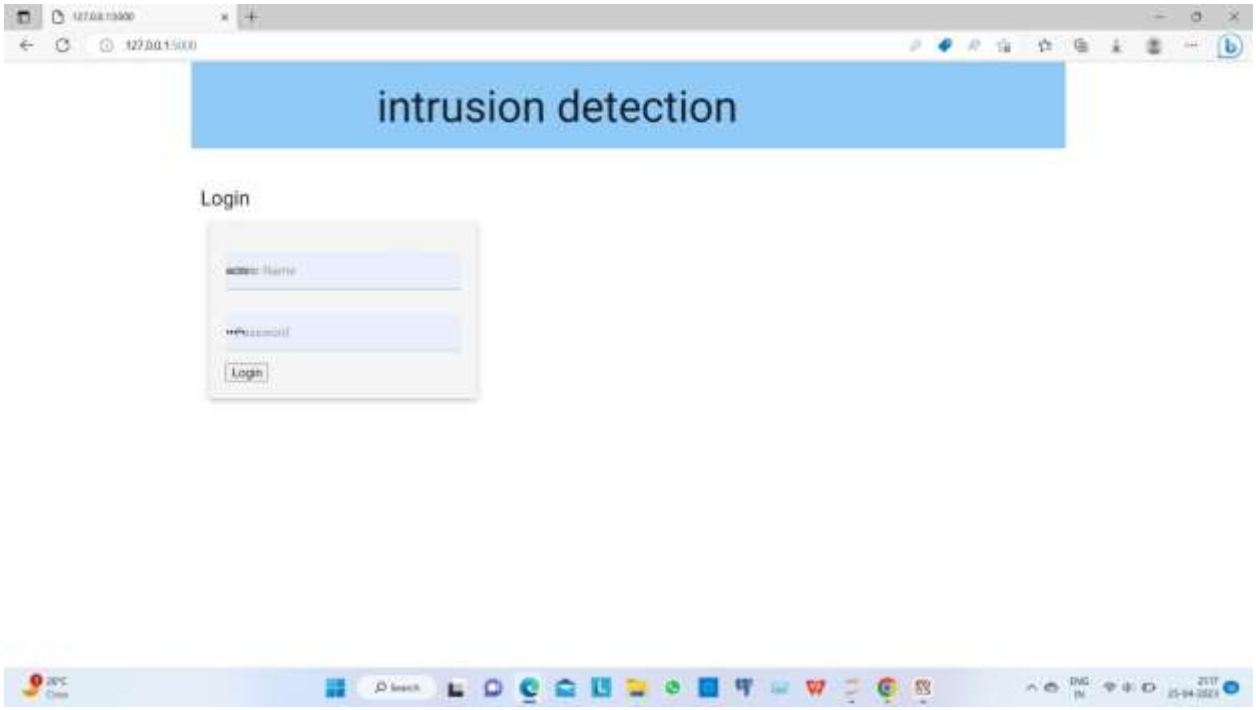


Fig 8.1 User Login Page

Fig 8.2 Display Web Page

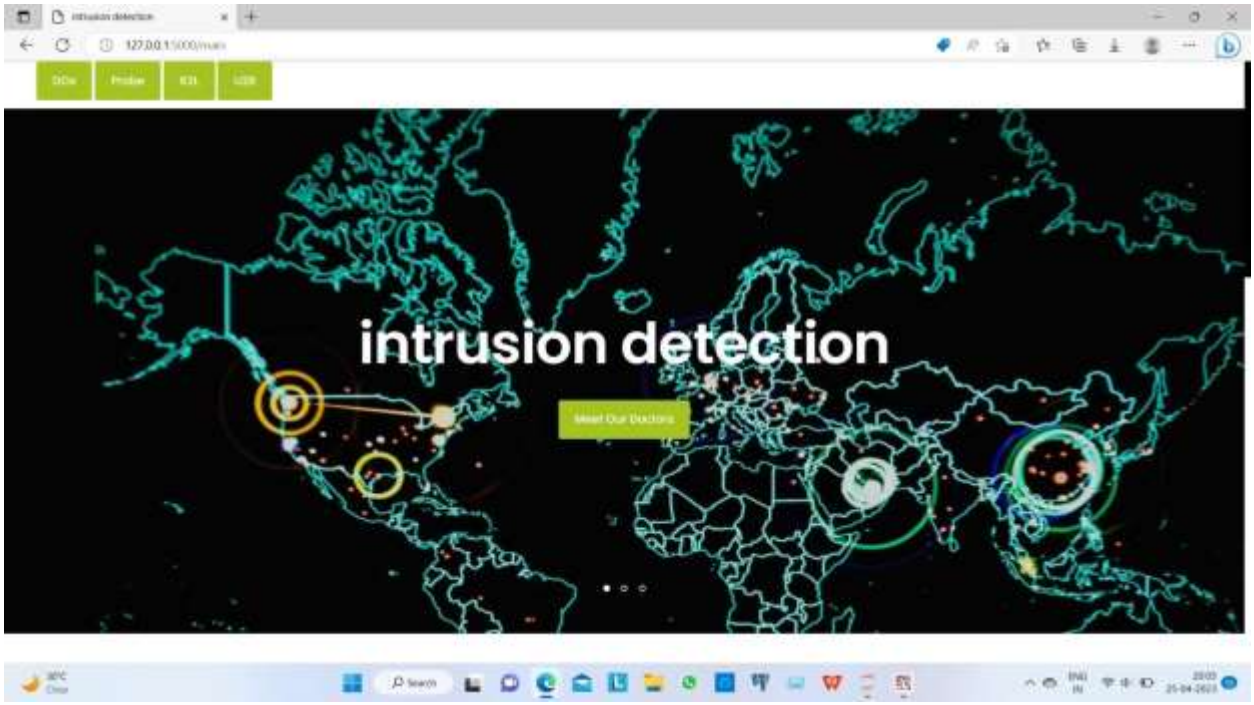


Fig 8.3 Display Web Page with Attacks

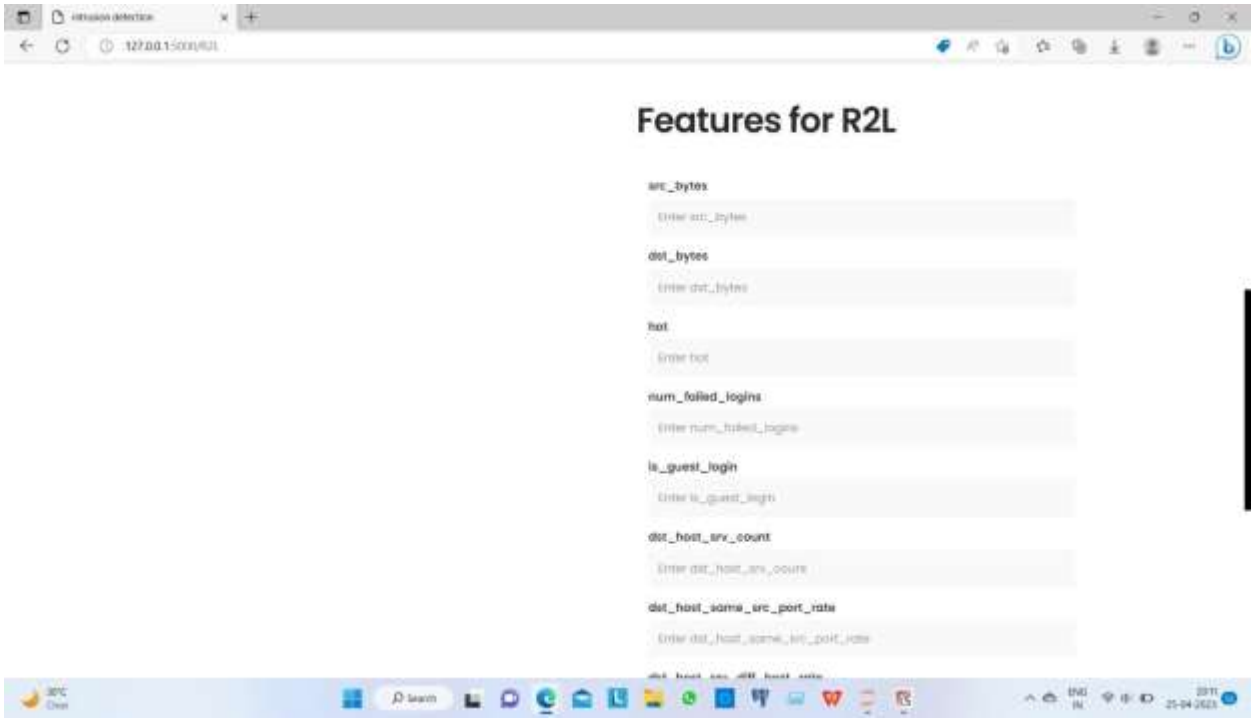


Fig 8.4 Data input with features of Attacks

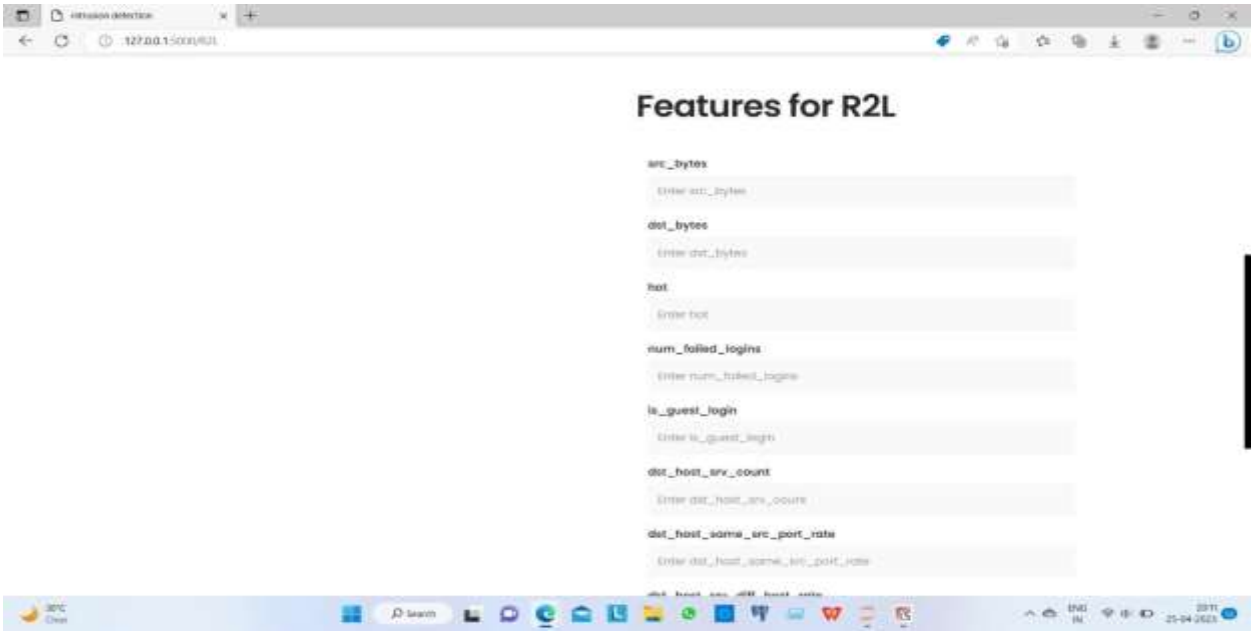


Fig 8.5 Data input with features of Attacks

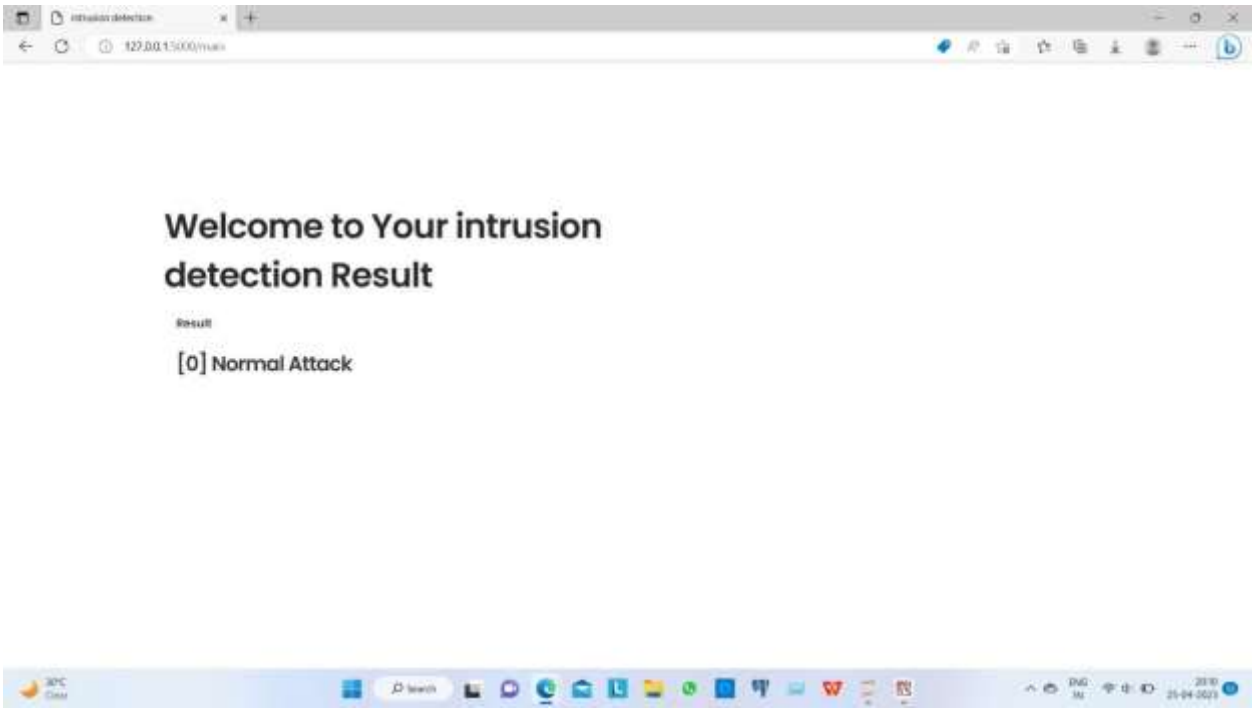


Fig 8.6 Detect the Attacks extracting features

CHAPTER 9

ANDVANTAGES, DISADVANTAGES AND APPLICATIONS

9.1 ADVANTAGES

- It is easy to install and manage single detector.
- End system are unaffected as it does not consume any resources in end systems.
- Offer centralized management for the correlation of the attack.

9.2 DISADVANTAGES

- NIDS cannot monitor encrypted traffic, especially from attackers using private network.
- Its inability to discover network threats against the host.

9.3 APPLICATIONS

- Anomaly detection –based IDS with ML is most suitable to implement.
- Traffic regulation policy traces suspicious traffic across the network, such as an unusually high rate of TCP connections.

CONCLUSION

For preventing attacks to the networks, an intrusion detection system plays a very critical role in the cyber security domain. Its effectiveness directly depends on the used decision engine. To increase the flexibility of the system, instead of signature-based detection, it is required to implement the system as anomaly detection with a learning system. One of the newest training and classification technique, which is executed in this engine, is emerged as deep learning. Therefore, in this project it is aimed to provide a short survey of deep learning-based intrusion detection systems with the overview of various aspects of intrusion detection and deep learning algorithms. Additionally, this work lists and gives details about some publicly available datasets with their characteristics and shortcomings. We believe that this comprehensive survey on deep learning-based IDS could be helpful or researchers in this area. Although most of the researches proposed their system with the older dataset, as future work, it will be helpful to use the newest datasets with alternative deep learning approaches.

FUTURE ENHANCEMENT

In this project we identified 4 types of network attack type with 123 features. In the future enhancement we can apply for other network attack type and we can apply other deep learning algorithm for predicting the better result. In this project detecting all the attacks. And Intrusion Detection System is useful until human uses the system, advances in correlation and alert correlation methods and extended using for business security.

REFERENCES

[1] Wen-Hui Lin, Hsiao-Chung Lin, Ping Wang, Bao-Hua Wu, Jeng-Ying Tsai, ― Using The Convolutional Neural Networks to Network Intrusion Detection for Cyber Threats‖, IEEE ICASI 2018.

[2] Sheraz Naseer, and Yasir Saleem, ―Enhanced Network Intrusion Detection using Deep Convolutional Neural Networks‖, VOL. 12, NO. 10, Oct. 2018

[3] r. u. khan, x. Zheng et r. Kumar, ―Analysis of resnet and google net models for malware detection, Journal of Computer Virology and Hacking Techniques, ‖AUG 2018.

[4] r. u. khan, x. Zheng, r. Kumar et e. o. aboagye, ―Evaluating the performance of resnet model based on image recognition, ‖ in Proceedings of the 2018 International Conference on *C*omputing and Artificial Intelligence, ICCAI 2018, (NEW YORK, NY, USA), P. 86–90, ACM, 2018.

[5] r. Kumar, z. Xiao song, r. u. khan, i. ahead et j. Kumar, ―Malicious code detection based on image processing using deep learning, ‖ in Proceedings of the *2018* International Conference on Computing and Artificial Intelligence, ICCAI 2018, (NEW YORK, NY, USA), P. 81–85, ACM, 2018.

[6] m. g. Raman, n. somu, k. kirthivasan et v. s. sriram, ―A hypergraph and arithmetic residue-based probabilistic neural network for classification in intrusion detection systems‖, Neural Networks, VOL. 92, P. 89–97, 2017.

[7] s. vent cinque et a. amato, ―Smart sensor and big data security and resilience, in Security and Resilience in Intelligent Data- Centric Systems and Communication Networks, ‖ P. 123–141, ELSEVIER, 2018.

[8] s. m. h. bamakan, h. Wang et y. shi, ―Ramp loss k- support vector classification-regression; a robust and sparse multi-class approach to the intrusion detection problem‖, KnowledgeBased Systems, VOL. 126, P. 113–126, 2017.