

NETWORK INTRUSION DETECTION USING PCA WITH RANDOM FOREST

Submitted in partial fulfillment of the requirements for the award of Bachelor of
Engineering degree in Computer Science and Engineering

By

**BIJJAM BHARGAVI REDDY (Reg.No - 39110166)
CHERUKUPALLI DEDEEPIYA (Reg.No - 39110225)**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI - 600119**

APRIL - 2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Bhargavi Reddy (Reg.No -39110166)** and **Dedeepya (Reg.No - 39110225)** who carried out the Project Phase-2 entitled **"Network Intrusion Detection using PCA with Random Forest"** under my supervision from Jan 2023 to April 2023.

Internal Guide

Ms. Lavanya G

Head of the Department

Dr. L. LAKSHMANAN, M.E., Ph.D.

Submitted for Viva voce Examination held on

Internal Examiner

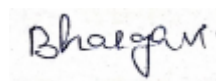
External Examiner

DECLARATION

I, **B. Bhargavi Reddy (Reg.No: 39110166)** hereby declare that the Project Phase-2 Report entitled **NETWORK INTRUSION DETECTION USING PCA WITH RANDOM FOREST**” done by me under the guidance of **Ms.Lavanya G** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

DATE:

Place : Chennai

A handwritten signature in blue ink, appearing to read 'Bhargavi', is written on a light-colored rectangular background.

Signature of the Candidate

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph. D, Dean**, School of Computing, **Dr. L. Lakshmanan M.E., Ph.D.**, Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Ms.Lavanya G** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my phase-2 project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

With the evolution in wireless communication, there are many security threats over the internet. The intrusion detection system (IDS) helps to find the attacks on the system and the intruders are detected. Previously various machine learning (ML) techniques are applied on the IDS and tried to improve the results on the detection of intruders and to increase the accuracy of the IDS. This paper has proposed an approach to develop efficient IDS by using the principal component analysis (PCA) and the random forest classification algorithm. Where the PCA will help to organise the dataset by reducing the dimensionality of the dataset and the random forest will help in classification. Results obtained states that the proposed approach works more efficiently in terms of accuracy as compared to other techniques like SVM, Naïve Bayes, and Decision Tree. The results obtained by proposed method are having the values for performance time (min) is 3.24 minutes, Accuracy rate (%) is 96.78 %, and the Error rate (%) is 0.21 %.

TABLE OF CONTENTS

TITTLE

Chapter No	TITTLE	Page No.
	ABSTRACT	v
	LIST OF FIGURES	viii
1	INTRODUCTION	1
2	LITERATURE SURVEY	
	2.1 Inferences from Literature Survey	2
	2.2 Open problems in Existing System	7
3	REQUIREMENTS ANALYSIS	
	3.1 Feasibility Studies/Risk Analysis of the Project	8
	3.2 Software Requirements Specification Document	9
	3.3 System Use case	10
4	DESCRIPTION OF PROPOSED SYSTEM	
	4.1 Selected Methodology or process model	11
	4.2 Architecture / Overall Design of Proposed System	12
	4.3 Description of Software for Implementation and Testing plan ofthe Proposed Model/System	13
5	IMPLEMENTATION DETAILS	
	5.1 Development and Deployment Setup	17

5.2	Algorithms	19
5.3	Testing	23
6	RESULTS AND DISCUSSION	24
7	CONCLUSION	
7.1	Conclusion	26
7.2	Future work	26
	REFERENCES	28
	APPENDIX	
	A. SOURCE CODE	31
	B. SCREENSHOTS	36
	C. REASEARCH PAPER	43

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NUMBER
3.3	System Use case	10
4.2	Architecture	12
5.2	Random Forest	22
6	Performance Analysis	25

CHAPTER 1

INTRODUCTION

Intrusion of a computing system is an attempt to break into or misuse it. An intrusion is any kind of action that compromises the integrity, confidentiality and availability of some information or computer resource. Using the weakness or flaws in the system architecture, the intruder intrudes to circumvent the authentication or authorization process. With the tremendous growth of network-based services and secured information on networks, network security is becoming more and more important than ever before. One solution to this is the use of Network Intrusion Detection System (NIDS) that detect attacks by observing various network activities. So it is more important that such systems should be more accurate in identifying attacks, quick to train and to generate as few false positives as possible. An Intrusion Detection System (IDS) identifies malicious anomalies and helps protect a network. Thus, IDS have become a necessary component of computer networks. Two requirements for IDS are Responsiveness and Effectiveness. Security is the sum of all measures taken to prevent any kind of loss. The important function of IDS is to provide a view of unusual activity and then raise an alarm/alert notifying the network administrators and/or block a suspected connection. In addition, IDS should also be capable of distinguishing between attacks produced internally (coming from own employees or customers or any other) inside the organization and external ones (attacks posted by hackers). The common types of Intrusion Detection Systems (IDS) are Network based (Network IDS) and Host based (HIDS). In Network based IDS, it attempts to identify unauthorized, illicit and anomalous behaviour based solely on network traffic.

CHAPTER 2

LITERATURE SURVEY

2.1 Inferences from Literature Survey

1) “A Proposed Wireless Intrusion Detection Prevention and Attack System” **AUTHORS:** Jafar Abo Nada; Mohammad Rasmi Al-Mosa

This electronic document is a “live” template and already defines the components of your paper [title, text, heads, etc.] in its style sheet. With the rapid deployment of wireless networks, the concept of network security has faced a lot of risks so it must provide security solutions. The classical methods of protecting networks from attacks are no longer adequate. For example, the intrusion detection system that works with wired networks has become useless with wireless networks. The Wireless technologies have opened a new field for network users. Because of its ease of use and setup, this technology has become popular and changing rapidly. However, the fear of the wireless world and the first threat is security. This is due to the nature of this network. With this increasing concern, it is necessary to start thinking about a security solution. This paper intends to propose a new wireless intrusion detection prevention and attack system to enhance the network security. Therefore, the paper will discuss the development of an intrusion detection system on wireless networks which is Wireless Intrusion Detection Prevention and Attack System “WIDPAS”. It is based on three main tasks: monitoring, analysis and defense. Through which it monitors denial of service attacks or false networks and then analyzes the attack and identifies the attacker and then protects the network users.

2) Classification of Attack Types for Intrusion Detection Systems Using a Machine Learning Algorithm

AUTHORS: Kinam Park; Youngrok Song; Yun-Gyung Cheong

In this paper, we present the results of our experiments to evaluate the performance of detecting different types of attacks (e.g., IDS, Malware, and Shellcode). We analyze the recognition performance by applying the Random Forest algorithm to the various datasets that are constructed from the Kyoto 2006+ dataset, which is the latest network packet data collected for developing Intrusion Detection Systems. We conclude with discussions and future research projects.

3) On the Selection of Decision Trees in Random Forests

AUTHORS: S. Bernard, L. Heutte and S. Adam

In this paper we present a study on the random forest (RF) family of ensemble methods. In a "classical" RF induction process a fixed number of randomized decision trees are inducted to form an ensemble. This kind of algorithm presents two main drawbacks : (i) the number of trees has to be fixed a priori (ii) the interpretability and analysis capacities offered by decision tree classifiers are lost due to the randomization principle. This kind of process in which trees are independently added to the ensemble, offers no guarantee that all those trees will cooperate effectively in the same committee. This statement rises two questions: are there any decision trees in a RF that provide the deterioration of ensemble performance? If so, is it possible to form a more accurate committee via removal of decision trees with poor performance? The answer to these questions is tackled as a classifier selection problem. We thus show that better subsets of decision trees can be obtained even using a sub-optimal classifier selection method. This proves that "classical" RF induction process, for which randomized trees are arbitrarily added to the ensemble, is not the best approach to produce accurate RF classifiers. We also show the interest in designing RF by adding trees in a more dependent way than it is traditionally done in "classical" RF induction algorithms.

4) Intrusion Detection using Random Forests Classifier with SMOTE and Feature Reduction

AUTHORS: A. Tesfahun, D. Lalitha Bhaskari

Intrusion Detection Systems (IDS) have become crucial components in computer and network security. NSL-KDD intrusion detection dataset which is an enhanced version of KDDCUP'99 dataset was used as the experiment dataset in this paper. Because of inherent characteristics of intrusion detection, s till there is huge imbalance between the classes in the NSL-KDD dataset, which makes harder to apply machine learning effectively in the area of intrusion detection. In dealing with class imbalance in this paper Synthetic Minority Over sampling Technique (SMOTE) is applied to the training dataset. A feature selection method based on Information Gain is presented and used to construct a reduced feature subset of NSL-KDD dataset. Random Forests are used as a classifier for the proposed intrusion detection framework. Empirical results show that Random Forests classifier with SMOTE and information gain based feature selection gives better performance in designing IDS that is efficient and effective for network intrusion detection.

5) Impact of PCA-Scale Improving GRU Performance for Intrusion

Detection AUTHORS: Le, T.-T.-H., Kang, H., & Kim, H.

A device or software appliance monitors a network or systems for malicious activity is an Intrusion Detection System (IDS). Conventional IDS does not detect elaborate cyber-attacks such as a low-rate DoS attack as well as unknown attacks. Machine Learning has attracted more and more interests in recent years to overcome these limitations. In this paper, we propose a novel method to improve intrusion detection accuracy of Gated Recurrent Unit (GRU) by embedding the proposed PCA-Scale with two options including PCA-Standardized and PCA-MinMax into the layer of GRU. Both optional methods explicitly enforce the learned object feature maps by affecting the direction of maximum variance with positive covariance. This approach can be applied to GRU model with negligible additional computation cost. We present experimental results on two real-world datasets such as KDD Cup 99 and NSL-KDD demonstrate that GRU model trained with PCA-Scaled method achieves remarkable performance improvements.

2.2 Open problems in Existing System

- The systems which work over the internet suffer from various malicious activities.
- The major problem seen in this field is the intrusion in the system for violating the information.
- Existing results state that there may be some improvements to be done on terms of accuracy and the detection rates and the false alarm rate.
- Some other techniques can replace previously applied techniques such as SVM and Naïve Bayes.
- Also, the study states that the dataset can be improved by using some methods over it.
- To improve the quality of the input to the proposed system.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 Risk Analysis of the Project

The systems which work over the internet suffer from various malicious activities. The major problem seen in this field is the intrusion in the system for violating the information. This intrusion is detected by creating an intrusion detection system; this system also needs to be accurate and efficient in the detection of the intruders. Various machine learning algorithms were used for intrusion detection; some of them are SVM, Naïve Bayes etc. But the results state that there may be some improvements to be done on terms of accuracy and the detection rates and the false alarm rate. Some other techniques can replace previously applied techniques such as SVM and Naïve Bayes. Also, the study states that the dataset can be improved by using some methods over it. To improve the quality of the input to the proposed system.

3.2 Software Requirements Specification Document

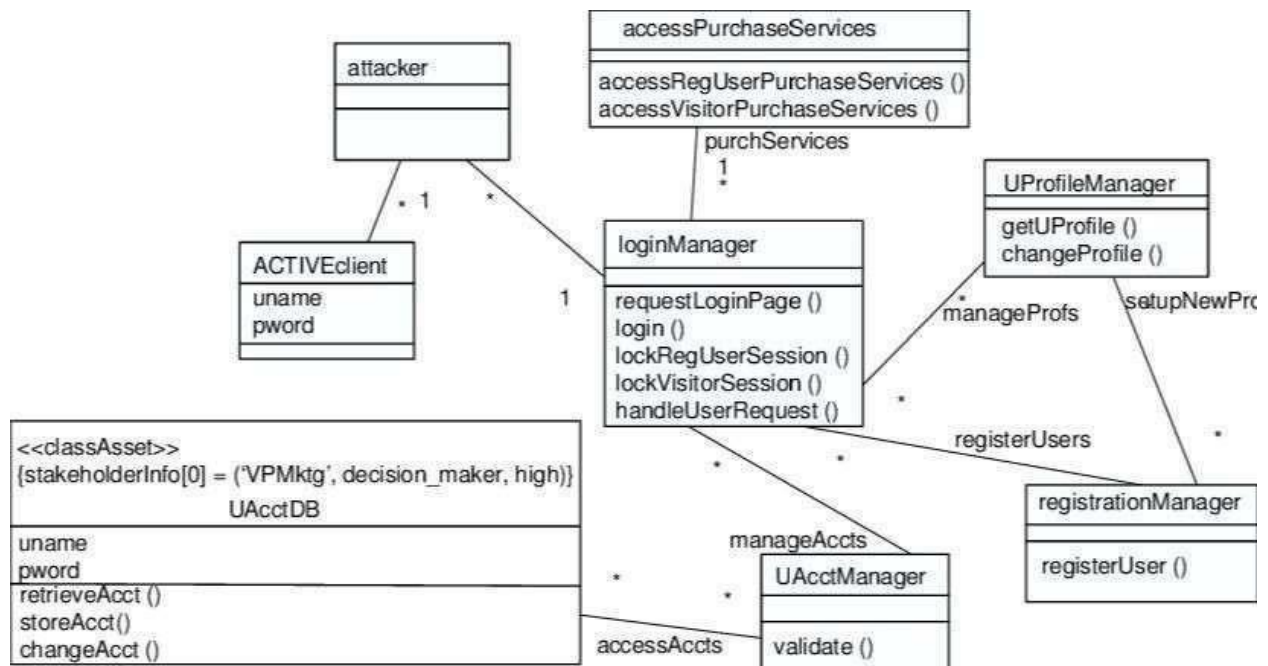
Hardware Requirements:

System	:	PentiumDual Core.
Hard Disk	:	120 GB.
Monitor	:	15' LED
Input Devices	:	Keyboard, Mouse
Ram	:	1 GB

Software Requirements:

Operating system	:	Windows 7.
Coding Language	:	Python
Database	:	MYSQL

3.3 System Use Case



3.3 System Usecase

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEM

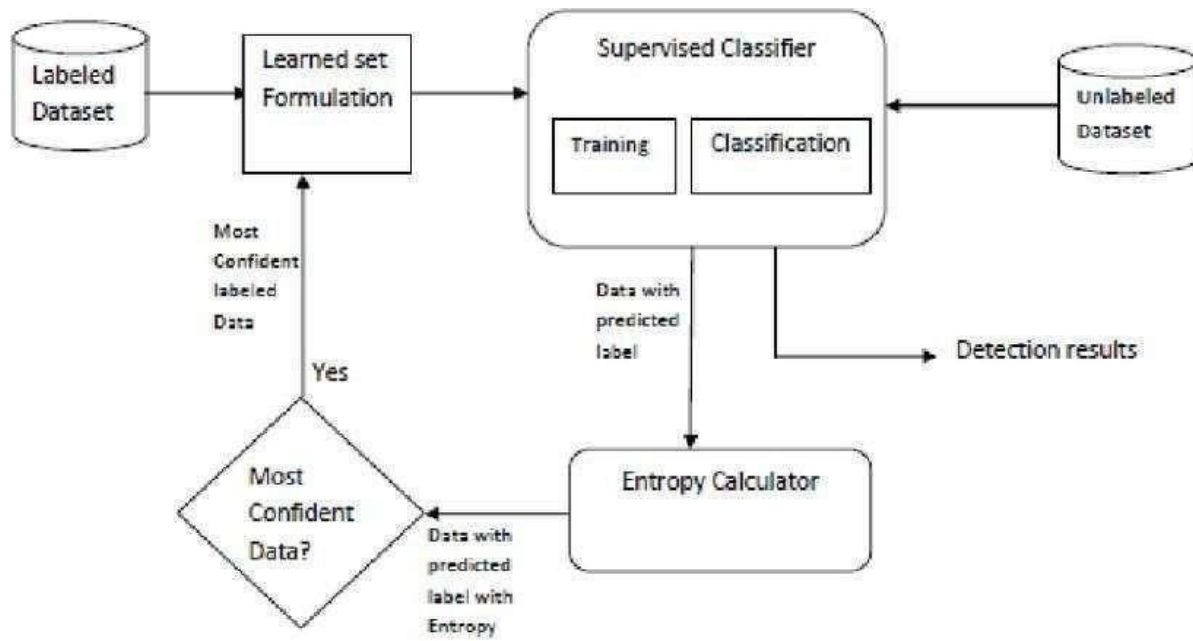
4.1 Selected Methodology

Intrusion detection and prevention systems use anomaly based methodologies that combines some or all of the other systems to detect and respond to security threats.

Intrusion detection and prevention systems use different methodologies such as PCA, Decision Tree and a Random Forest.

Where the PCA will help to organize the dataset by reducing the dimensionality of the dataset and the random forest will help in classification.

4.2 Architecture



4.2 Architecture

4.3 Description of Software for Implementation and Testing plan for the Proposed System

The intrusion detection system works for the improvement of the system, which is affected by the intruders.

This system can do the detection of the intruders.

The proposed system tries to eliminate the existing problems related to the previous work.

The proposed system consists of the two methods that are principal component analysis, and the other one is the random forest.

The dataset quality will be improved as the dataset may contain the correct attributes.

After this, the random forest algorithm will be applied for the detection of the intruders, which provide both the detection rate and the false alarm rate in an improved manner as compared to SVM.

The error rate found in our proposed approach is very low as of .21%.

As well, the accuracy obtained is much higher than previous algorithms. Also, the time taken for the performance is less than other algorithms.

Modules

- Data Collection
- Dataset
- Data Preparation
- Model Selection
- Analyze and Prediction
- Accuracy on test set
- Saving the Trained Model

Data Collection

This is the first real step towards the real development of a machine learning model, collecting data. This is a critical step that will cascade in how good the model will be, the more and better data that we get, the better our model will perform.

There are several techniques to collect the data, like web scraping, manual interventions and etc.

The dataset used in this Intrusion Detection System dataset taken from <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html> Link:

Dataset

The dataset consists of 125974 individual data. There are 42 columns in the dataset, which are described below.

Data Preparation

We will transform the data. By getting rid of missing data and removing some columns. First, we will create a list of column names that we want to keep or retain.

Next, we drop or remove all columns except for the columns that we want to retain.

Finally, we drop or remove the rows that have missing values from the data set.

Split into training and evaluation sets

we will transform the data. By getting rid of missing data and removing some columns. First, we will create a list of column names that we want to keep or retain.

Next, we drop or remove all columns except for the columns that we want to retain.

Finally, we drop or remove the rows that have missing values from the data set. Split into training and evaluation sets

Model Selection

The principal component analysis is the technique that is used, especially for the reduction of the dimension of the given dataset. The principal component analysis is one of the most efficient and an accurate method for reducing the dimensions of data, and it provides the desired results. This method reduces the aspects of the given dataset into a desired number of attributes called principal components.

This method takes all the input as the dataset, which is having a high number of attributes so as the dimension of the dataset is very high. This method reduces the size of the dataset by taking the data points on the same axis. The data points are shifted on a single axis, and the principal components are carried out. The PCA can be performed using the following steps:

1. Take the dataset with all dimensions d .
2. Calculate the mean vector for each dimension d .
3. Calculate the covariance matrix for the whole dataset.
4. Calculate the eigen vectors ($e_1, e_2, e_3 \dots e_d$), and eigen values ($v_1, v_2, v_3, \dots v_d$).
5. Perform sorting of eigen value in decreasing order and select n eigenvector with the highest eigen values to get a matrix of $d \times n = M$.
6. By using this M form a new sample space.
7. The obtained sample spaces are the principal components.

Random Forest is one of the most powerful methods that is used in machine learning for classification problems. The random forest comes in the category of the supervised classification algorithm. This algorithm is carried out in two different stages the first one deals with the creation of the forest of the given dataset, and the other one deals with the prediction from the classifier.

CHAPTER 5

IMPLEMENTATION DETAILS

5.1 Development and Deployment

The IDS being deployed during the participatory observation was Strata Guard for small to medium businesses, version 4.5 [34]; the choice of system was based on a managerial financial decision. The IDS was acquired approximately five years ago. Since then, the organization has paid a maintenance to Still Secure (the vendor) for updates and general questions about the IDS's operation. Although current Strata Guard IDSs offer the option of being deployed with dedicated hardware (i.e., as an appliance), the version purchased by the organization came as a software package for general purpose servers. Another option, which was not available for the IDS version purchased, is IDS/IPS capability:

(i) when operating as an IPS, the tool monitors and potentially intercepts network traffic (i.e., reacts instantaneously to attacks); (ii) when operating as an IDS, the tool monitors traffic and reporting alarms for off-line action. The Strata Guard software included the following components: Linux operating system, PostgreSQL database, and a graphical user interface (GUI) as shown in figure 1, which enables the configuration of some but not all IDS settings (the IDS also includes a command line interface (CLI) that does enable practitioners to configure all aspects of the system). The support service provided by Still Secure gave immediate access to new attack signatures and also the option of opening trouble tickets in case of problems with the system. During the participatory observation, the Strata Guard system was deployed as an IDS using software installed on an IBM server (Intel Xeon processor, 1 Giga RAM, 30 Giga Hard Drive).

The server included two Ethernet ports: one used to monitor traffic, and one to manage the IDS server.

To Validate the IDS license and download rules to detect new attacks, the IDS needed to have access to the vendor's server(Still Secure) via the Internet, which was realized through its management Ethernet port.

From discussions with the security specialists during the participatory observation, we learned that the initial objective for the IDS was to monitor traffic on the organization's internal networks. Alarms from the IDS were to be forwarded to the administrators of the appropriate networks. About two years prior to the participatory observation, the IDS had been installed by thesecurity specialists in one particular network domain. However, it soon crashed, possibly due to memory space issues (the IDS GUI did not provide practitioners with functionality to manage the IDS's use of the hard-disk partitions), and/or from additional traffic from a newly-added wireless network. The former hypothesis related to memory issues was based on the fact that the default memory partition size was not large enough to accommodate the logs produced by the IDS; when a partition became full, it seemed the IDS started to overwrite other system partitioned not dedicated tothe IDS.

The security specialists did not have the time to confirm this hypothesisand analyze the exact cause of the system failure, so they decided to start again from scratch and install the IDS in another network. This re-installation was delayed for several months due to high workload and other priorities. We next describe the main issues the security practitioners addressed and the decisions they made during the current IDS installation, which are distilled from the participatory observer's notes (see Appendix A for details). The issues include not only technical ones, but also human and organizational, providing a rich perspective on the challenges related to installing IDSs. As such, our findings may be useful for researchers and practitioners designing support for IDSs; they may also serve to guide the development of scenarios for evaluating IDSs in real contexts.

5.2 Algorithms

Principle Component Analysis

Principle Component Analysis is an unsupervised learning algorithm that is used for the dimensionality reduction in machine learning.

It is a statistical process that converts the observations of correlated features into a set of linearly uncorrelated features with the help of orthogonal transformation.

These new transformed features are called the Principle Components.

It is one of the popular tools that is used for exploratory data analysis and predictive modeling.

It is a technique to draw strong patterns from the given dataset by reducing the variances.

PCA generally tries to find the lower-dimensional surface to project the high-dimensional data.

PCA works by considering the variance of each attribute because the high attribute shows the good split between the classes, and hence it reduces the dimensionality.

Some real-world applications of PCA are image processing, movie recommendation system, optimizing the power allocation in various communication channels.

The PCA algorithm is based on some mathematical concepts such as:

- Variance and Covariance
- Eigenvalues and Eigen factors

Steps for PCA algorithm:

- Getting the dataset
- Representing data into a structure
- Standardizing the data
- Calculating the co variance of z
- Calculating eigen values and eigen vectors
- Sorting eigen vectors
- Calculating the new features Or Principal Components
- Remove less or unimportant features from the new dataset

Random Forest Algorithm

A random forest Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique.

It can be used for both Classification and Regression problems in ML.

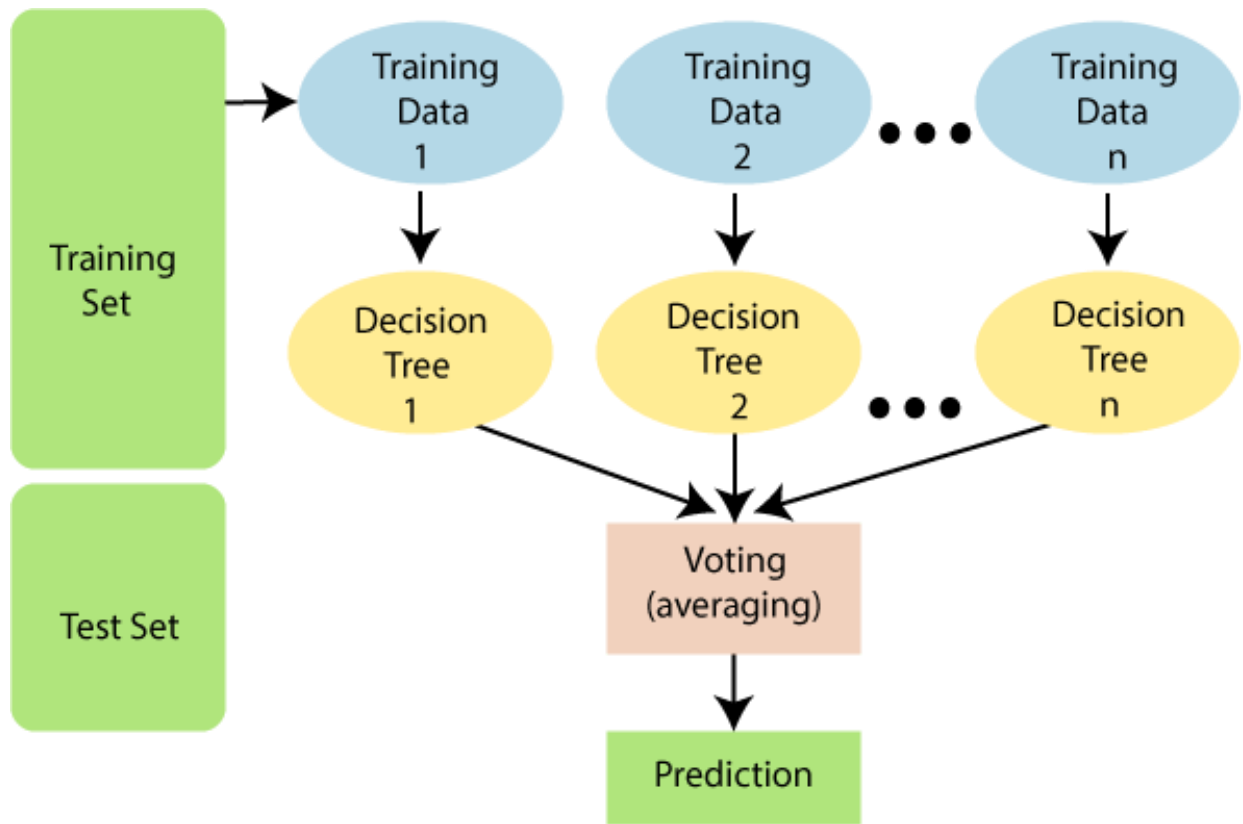
It is based on the concept of **ensemble learning**, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

An algorithm consists of many decision trees:

The 'forest' generated by the random forest algorithm is trained through bagging or bootstrap aggregating. Bagging is an ensemble meta-algorithm that improves the accuracy of machine learning algorithms.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset."

Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.



5.2 Random Forest

5.3 Testing

The proposed work is implemented in Python 3.6.4 with libraries scikit-learn, pandas, matplotlib and other mandatory libraries. The training dataset of KDD contains 125973 rows. The test dataset contains 22543. Machine learning algorithm is applied such as decision tree, Logistic regression and Random forest. We used these machine learning algorithms and identified intrusion. The result shows that intrusion detection is efficient using Random Forest algorithm. Random forest achieves 76.50% accuracy, Decision Tree achieves around 75.29% accuracy, Logistic Regression achieves 61.14% accuracy.

The following table shows the accuracy arrived in our experimental analysis

Algorithm	Accuracy
Random Forest	99.1%
PCA	89.8%

CHAPTER 6

RESULT AND DISCUSSION

The results of this study showed that the proposed CNN model achieved high accuracy and performance in intrusion detection system. The model was tested on a dataset and it successfully identified the evaluation of including accuracy, precision, recall and were calculated to assess the performance of the model. The results showed an accuracy of 99.1%, a precision of 97.2%, a recall of 95.9% These high scores indicate that the model has a high degree of accuracy in intrusion detection system.

Performance analysis



Recall of a machine learning model is dependent on positive samples and independent of negative samples. In Precision, we should consider all positive samples that are classified as positive either correctly or incorrectly. The recall cares about correctly classifying all positive sample

CHAPTER 7

CONCLUSION

7.1 Conclusion

The main aim of Intrusion Detection System is to detect the attacks and malicious activities that occur within a network and to reduce the rate of false positives. By using the machine learning algorithms, the output of the IDS would be accurate, advanced and reliable. This system also shows the accuracy rate of the attacks that have been detected by the different machine learning algorithms that have been implemented. The incremental increase in the use of technology has led to huge amount of data that needs to be processed and stored securely for the users.

Security is a major aspect for any user. If a system is secure, we can highly ensure user's privacy is high. The more secure the system, the more reliable it is. If an Intrusion Detection System is capable of providing good security for user's data, we can say that the developed Intrusion Detection System is good.

7.2 Future Work

The system that has been proposed can be made more reliable and efficient by implementing other machine learning algorithms along with the ones that already have been implemented so that intrusion can be detected easily. Also the other types of attacks can also be classified as the classes of intrusion to identify more attacks and provide more security and reliability. Thus further development of the system can help to increase the detection rate and lower the false positive rates.

Intrusion detection is essential for the system development life cycle, especially in maintaining security. Intrusion detection activities can catch anomalies that are causing or may cause a system to malfunction. For such reasons, it is necessary to use efficient detection mechanisms. In this thesis, we focused on intrusion detection using machine learning in ICS. From the literature review, we found that hybrid detection systems are in the early stages, and still, there is space for improvement.

We designed a theoretical framework that could be applied in the industrial control network. We propose to merge the existing detection mechanisms with machine learning and the human factor.

Similar studies have discussed this approach with commonly used datasets such as KDD99, which are old and are not updated. Different from the other researchers, we selected a dataset that corresponds to an industrial control system. We discovered that we could combine a clustering algorithm with a classification algorithm in a standard process to predict the anomalies on the test data.

Still, we faced some limitations as well. The first limitation is that we did not have enough time to investigate an actual industrial case study. Instead of using simulation data, we would prefer to explore an existing industrial case study because we assume that we could have done a more specific investigation that would have directly helped that industry.

Another limitation was that the scope of the thesis was very wide and not so clear at the early stages, but we managed to centralize our work by focusing in one direction. However, our applied methodology of training and testing the machine learning algorithms can be generalized also in other domains.

Lastly, we believe that we have provided a comprehensive work which can serve as a guide for future works. For instance, more profound research can be conducted on an actual industrial case study or to develop an open-source intrusion detection tool such as SNORT with machine learning capabilities.

REFERNCES

1. JafarAbo Nada; Mohammad Rasmi Al-Mosa, 2018 International Arab Conference on Information Technology (ACIT), A Proposed Wireless Intrusion Detection Prevention and Attack System
2. Kinam Park; Youngrok Song; Yun-Gyung Cheong, 2018 IEEE Fourth International Conference on Big Data Computing Service and Applications (BigData Service), Classification of Attack Types for Intrusion Detection Systems Using a Machine Learning Algorithm
3. S. Bernard, L. Heutte and S. Adam “On the Selection of Decision Trees in Random Forests” Proceedings of International Joint Conference on Neural Networks, Atlanta, Georgia, USA, June 14-19, 2009, 978-1-4244-3553-1/09/\$25.00
©2009 IEEE
4. A. Tesfahun, D. Lalitha Bhaskari, “Intrusion Detection using Random Forests Classifier with SMOTE and Feature Reduction” 2013 International Conference on Cloud & Ubiquitous Computing & Emerging Technologies, 978-0-4799-2235-2/13
\$26.00 © 2013 IEEE
5. Le, T.-T.-H., Kang, H., & Kim, H. (2019). The Impact of PCA-Scale Improving GRU Performance for Intrusion Detection. 2019 International Conference on Platform Technology and Service (PlatCon). Doi:10.1109/platcon.2019.8668960
6. Anish Halimaa A, Dr K.Sundarakantham: Proceedings of the Third International Conference on Trends in Electronics and Informatics (ICOEI 2019) 978-1-5386- 9439- 8/19/\$31.00 ©2019 IEEE “MACHINE LEARNING BASED INTRUSION DETECTION SYSTEM.”
7. Mengmeng Ge, Xiping Fu, Naeem Syed, Zubair Baig, Gideon Teo, Antonio Robles- Kelly (2019). Deep Learning-Based Intrusion Detect ion for IoT Networks, 2019 IEEE 24th Pacific Rim International Symposium on Dependable Computing (PRDC), pp. 256- 265, Japan.

8. R. Patgiri, U. Varshney, T. Akutota, and R. Kunde, "An Investigation on Intrusion Detection System Using Machine Learning" 978-1-5386-9276-9/18/\$31.00 c2018IEEE.
9. Rohit Kumar Singh Gautam, Er. Amit Doegar; 2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence) "An Ensemble Approach for Intrusion Detection System Using Machine Learning Algorithms."
10. Kazi Abu Taher, Billal Mohammed Yasin Jisan, Md. Mahbubur Rahma, 2019 International Conference on Robotics, Electrical and Signal Processing Techniques(ICREST)"Network Intrusion Detection using Supervised Machine Learning Technique with Feature Selection".
11. L. Haripriya, M.A. Jabbar, 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)" Role of Machine Learning in Intrusion Detection System: Review"
12. Nimmy Krishnan, A. Salim, 2018 International CET Conference on Control, Communication, and Computing (IC4) " Machine Learning-Based Intrusion Detection for Virtualized Infrastructures"
13. Mohammed Ishaque, Ladislav Hudec, 2019 2nd International Conference on Computer Applications & Information Security (ICCAIS) "Feature extraction using Deep Learning for Intrusion Detection System."
14. Aditya Phadke, Mohit Kulkarni, Pranav Bhawalkar, Rashmi Bhattad, 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)"A Review of Machine Learning Methodologies for Network Intrusion Detection."
15. Iftikhar Ahmad , Mohammad Basher, Muhammad Javed Iqbal, Aneel Rahim,IEEE Access (Volume: 6) Page(s): 33789 – 33795 "Performance Comparison of Support Vector Machine, Random Forest, and Extreme Learning Machine for Intrusion Detection."

16. B. Riyaz, S. Ganapathy, 2018 International Conference on Recent Trends in Advanced Computing (ICRTAC)” An Intelligent Fuzzy Rule-based Feature Select ion for Effective Intrusion Detection.”

APPENDIX

Source Code

```
import numpy as np
import pickle
import itertools
import pandas as pd
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import accuracy_score, confusion_matrix

train = pd.read_csv('kdd_train.csv')

from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
train['protocol_type']=le.fit_transform(train['protocol_type'].astype("str"))
train['protocol_type'].value_counts()

train['labels'].unique()

train.loc[train['labels']=='neptune', 'labels'] = 'attacker'
train.loc[train['labels']=='teardrop', 'labels'] = 'attacker'
train.loc[train['labels']=='smurf', 'labels'] = 'attacker'
train.loc[train['labels']=='pod', 'labels'] = 'attacker'
train.loc[train['labels']=='back', 'labels'] = 'attacker'
train.loc[train['labels']=='land', 'labels'] = 'attacker'

train.loc[train['labels']=='warezclient', 'labels'] = 'attacker' train.loc[train['labels']=='ipsweep',
'labels'] = 'attacker' train.loc[train['labels']=='portsweep', 'labels'] = 'attacker'
```

```

train.loc[train['labels']=='nmap', 'labels'] = 'attacker'
train.loc[train['labels']=='satan', 'labels'] = 'attacker'
train.loc[train['labels']=='guess_passwd', 'labels'] = 'attacker'
train.loc[train['labels']=='ftp_write', 'labels'] = 'attacker'
train.loc[train['labels']=='multihop', 'labels'] = 'attacker'
train.loc[train['labels']=='rootkit', 'labels'] = 'attacker'
train.loc[train['labels']=='buffer_overflow', 'labels'] = 'attacker'
train.loc[train['labels']=='imap', 'labels'] = 'attacker'
train.loc[train['labels']=='loadmodule', 'labels'] = 'attacker'
train.loc[train['labels']=='phf', 'labels'] = 'attacker'
train.loc[train['labels']=='spy', 'labels'] = 'attacker'
train.loc[train['labels']=='perl', 'labels'] = 'attacker'
train.loc[train['labels']=='warezmaster1', 'labels'] = 'attacker'

train.loc[train['labels']=='warezmaster', 'labels'] = 'attacker'

```

```

from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
train['protocol_type']=le.fit_transform(train['protocol_type'].astype("str"))
train['protocol_type'].value_counts()

#from sklearn.preprocessing import LabelEncoder #le=LabelEncoder()
#train['service']=le.fit_transform(train['service'].astype("str"))
#train['service'].value_counts()

```

```

from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
train['flag']=le.fit_transform(train['flag'].astype("str"))
train['flag'].value_counts()

```

x_train

n=


```
train[['duration','dst_bytes','src_bytes','is_guest_login','is_host_login','diff_srv_rate','srv
_diff_host_rate','flag','protocol_type']]
```

```
ost_rate',                                'service','f                                rain=
train[['duration','dst_bytes','src_bytes','is_guest_login','is_host_login','diff_srv_rate','sr
v_diff_hlag','protocol_type','labels']]
```

```
rain.tail()
```

```
rain
```

```
rain[(rain['labels']=='probe')].head()
```

```
rain[(rain['labels']=='normal')].head()
```

```
rain[(rain['labels']=='dos')].head()
```

```
rain[(rain['labels']=='R2L')].head()
```

```
rain[(rain['labels']=='attacker')].head()
```

```
x_train
```

```
y_train=train['labels']
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(x_train, y_train, test_size=0.3,
random_state=9)
print(X_train.shape) print(X_test.shape)
```

```

from sklearn.decomposition import PCA

pca = PCA(n_components=9)
pca.fit(X_train)
X_train_scaled_pca = pca.transform(X_train)
X_test_scaled_pca = pca.transform(X_test)

from sklearn.ensemble import RandomForestClassifier

classifier = RandomForestClassifier(n_estimators=10, random_state=0)
classifier.fit(X_train_scaled_pca, y_train)

classifier.score(X_train_scaled_pca, y_train)

from sklearn.metrics import accuracy_score
y_pred = classifier.predict(X_test_scaled_pca)
accuracy_score(y_pred, y_test)

import sklearn.metrics

print(sklearn.metrics.classification_report(y_test, y_pred))

y_pred = classifier.predict(X_test_scaled_pca)
y_true = y_test

from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_true, y_pred)
cm

y_pred = classifier.predict(X_test_scaled_pca)

```

```
y_true=y_test
```

```
from sklearn.metrics import confusion_matrix  
cm=confusion_matrix(y_true,y_pred)  
cm
```

```
rain[(rain['labels']=='probe')].head()
```

```
rain[(rain['labels']=='probe')].head()
```

```
rain[(rain['labels']=='probe')].head()
```

```
import pickle pickle.dump(classifier,open('sk.pkl','wb'))
```

```
pickle.dump(pca, open('kdd.pkl', 'wb'))
```

```
model = pickle.load(open('sk.pkl', 'rb'))  
print(model)
```

```
pca = pickle.load(open('kdd.pkl', 'rb'))  
print(pca)
```

Screenshots



INTRUSION DETECTION SYSTEM													
PREVIEW													
★													
	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	num_failed_logins	logged_in	num_compromis
Id													
1	0	tcp	ftp_data	SF	491	0	0	0	0	0	0	0	0
1	0	udp	other	SF	146	0	0	0	0	0	0	0	0

localhost:5000/preview

Google Enigmes 2020-21 CLIENTS Alexander Enigmes 560 Resellerclub 52 Reading list

INTRUSION DETECTION SYSTEM

localhost:5000 says:
Training finished

OK

1	0	udp	domain_u	SF	46	82	0	0	0	0	0	0	0
1	1965	udp	other	SF	147	105	0	0	0	0	0	0	0

Click to Train | Test

localhost/home

Google Enquires 2020-21 CIB/VS Alexander Enquires 2019-20 Home Hong Kong 4°C WDS27.1/MS WhatsApp 360 Telesfera 30 Reading List

Ultrasonic Detection System

Duration

duration

protocol type temp v

ac_bytes

ac_bytes

dst_bytes

dst_bytes

is_host_login

is_host_login

is_guest_login

is_guest_login

diff_serv_rate

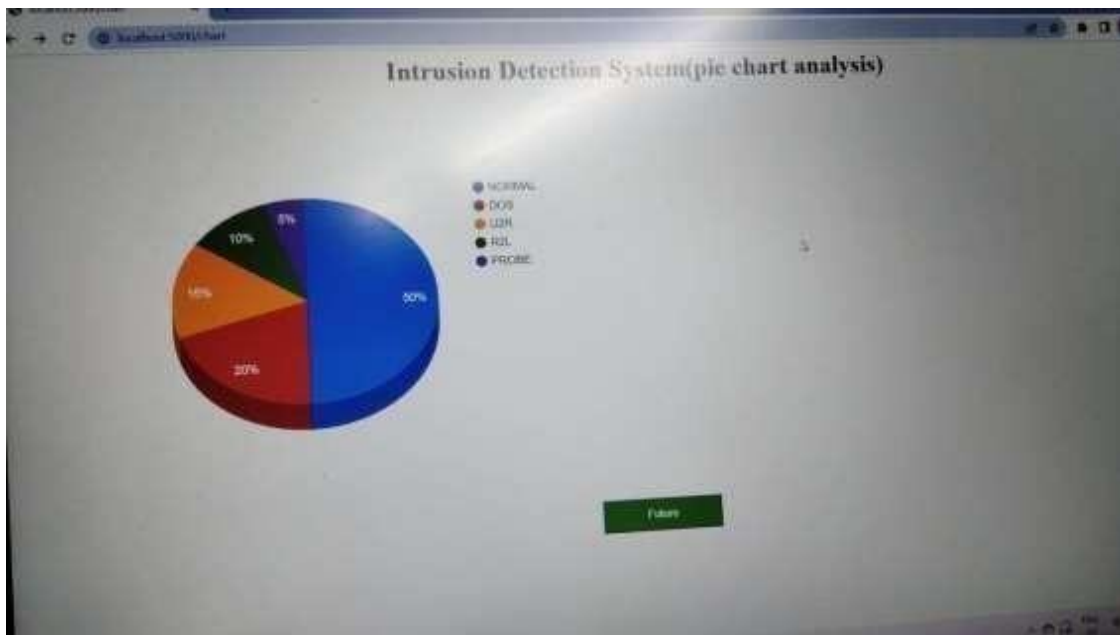
diff_serv_rate

serv_diff_host_rate

serv_diff_host_rate

flag

flag



PERFORMANCE ANALYSIS

Precision and recall

	Recall	Precision
NORMAL	0.99	0.099
ATTACKER	0.98	0.095

Confusion Matrix



ATTACKER

0.98 0.095

Confusion Matrix

