

INVASION IDENTIFICATION SYSTEM USING MACHINE LEARNING APPROACH

Submitted in partial fulfilment of the requirements for the award of
Bachelor of Engineering in Computer Science and Engineering By

MOTURU MOHAN KUMAR (Reg. No. 38110327)
VULLANGULA SAMUEL VIJAY KUMAR (Reg. No. 38110494)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING
SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
JE PPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMILNADU**

MARCH 2022



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



Accredited with Grade “A” by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Professional Training Report is the Bonafide work Of **MOTURU MOHAN KUMAR (Reg no:38110327)** and **VULLANGULA SAMUEL VIJAY KUMAR (Reg no: 38110494)** who out the project entitled **“INVASION IDENTIFICATION SYSTEM USING MACHINE LEARNING APPRAOCH”** under our supervision from October 2022 to March 2022.

Internal Guide

Dr.A.PRAVIN

Head of the Department

DR. VIGNESHWARI S, M.E., Ph. D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

we, **MOTURU MOHAN KUMAR**(38110327) and **VULLANGULA SAMUEL VIJAY KUMAR** (38110494) hereby declare that the Professional Training Report entitled **“INVASION IDENTIFICATION SYSTEM USING MACHINE LEARNING APPROACH”** done by me under the guidance of Dr.A.PRAVIN(Internal), is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge the 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 and **Dr. S. VIGNESHWARI M.E., Ph.D.**, and **Dr. L. Lakshmanan M.E., Ph.D.**, Head of the Department, Dept. 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 sincere and deep sense of gratitude to the 1Project Guide **Dr.A.PRAVIN**, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of the 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.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	Page No
	ABSTRACT	
1.	INTRODUCTION	07
1.1	Proposed Algorithm	08
1.2	Principal components in PCA	09
1.3	Random Forest Algorithm	12
2.	LITERATURE SURVEY	17
2.1	Proposed wireless Invasion detection and attack system	17
2.2	Types of attacks detection	18
2.3	selection of decision trees in random forest	19
3.	SYSTEM REQUIREMENTS	21
3.1	Software Requirements	21
3.2	Hardware Requirements	22
4.	SYSTEM ANALYSIS	25
4.1	purpose	25
4.2	scope	26
4.3	Existing system	26
4.4	proposed system	27
5.	INPUT DESIGN AND OUTPUT DESIGN	27
5.1	Input design	27
5.2	Output design	29

6.	MODULE DESCRIPTION	32
6.1	Data Collection	32
6.2	pre-processing	33
6.3	prediction module	33
7	SYSTEM ARCHITECTURE	
7.1	system architecture	42
8	CONCLUSION	47
9	SOURCE CODE	

ABSTRACT

With the evolution in wireless communication, there are many security threats over the internet. The invasion 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 %.

CHAPTER 1

INTRODUCTION

Invasion of a computing system is an attempt to break into or misuse it. An invasion 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 Invasion 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 Invasion 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 [11]. 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 Invasion Detection Systems (IDS) are Network based (Network IDS) and Host based (HIDS) [8]. In Network based IDS, it attempts to identify unauthorized, illicit and anomalous behaviour based solely on network traffic. Whereas, in HIDS, it attempts to identify unauthorized, illicit and anomalous behaviour on a specific device. In this paper we propose a new approach in Network based Invasion Detection System. Most of the Invasion Detection Systems use one of the two detection techniques: Anomaly based detection technique or Signature based detection technique. Anomaly based IDS identifies the anomalies from “normal” behavior and detects any deviation from it. It learns from the normal data collected when there are no anomalous activities. Whereas in Signature based IDS, it monitors the network traffic and it examines for preconfigured and predetermined attack patterns. A Murali M Rao et. al. [7], presented a survey on invasion detection approaches, in which he lists the various techniques followed for implementing IDS. The techniques are: Data Mining techniques, Statistical Models, Neural Networks, State Transition Analysis, Genetic Algorithm, Immune System Approach, Expert Systems, File System Checking, Pattern Matching, Protocol Analysis and Keystroke Monitoring.

1.1 PROPOSED ALGORITHM

PRINCIPAL COMPONENT ANALYSIS

Principal 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 **Principal 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***. It is a feature extraction technique, so it contains the important variables and drops the least important variable.

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

- Variance and Covariance
- Eigenvalues and Eigen factors

Some common terms used in PCA algorithm:

- **Dimensionality:** It is the number of features or variables present in the given dataset. More easily, it is the number of columns present in the dataset.
- **Correlation:** It signifies that how strongly two variables are related to each other. Such as if one changes, the other variable also gets changed. The

correlation value ranges from -1 to +1. Here, -1 occurs if variables are inversely proportional to each other, and +1 indicates that variables are directly proportional to each other.

- **Orthogonal:** It defines that variables are not correlated to each other, and hence the correlation between the pair of variables is zero.
- **Eigenvectors:** If there is a square matrix M , and a non-zero vector v is given. Then v will be eigenvector if Av is the scalar multiple of v .
- **Covariance Matrix:** A matrix containing the covariance between the pair of variables is called the Covariance Matrix.

1.2 Principal Components in PCA

As described above, the transformed new features or the output of PCA are the Principal Components. The number of these PCs are either equal to or less than the original features present in the dataset. Some properties of these principal components are given below:

- The principal component must be the linear combination of the original features.
- These components are orthogonal, i.e., the correlation between a pair of variables is zero.
- The importance of each component decreases when going to 1 to n , it means the 1 PC has the most importance, and n PC will have the least importance.

Steps for PCA algorithm

1. **Getting the dataset**
Firstly, we need to take the input dataset and divide it into two subparts X and Y , where X is the training set, and Y is the validation set.
2. **Representing data into a structure**
Now we will represent our dataset into a structure. Such as we will represent the two-dimensional matrix of independent variable X . Here each row corresponds to the data items, and the column corresponds to the Features. The number of columns is the dimensions of the dataset.
3. **Standardizing the data**
In this step, we will standardize our dataset. Such as in a particular column, the features with high variance are more important compared to the features

with lower variance. If the importance of features is independent of the variance of the feature, then we will divide each data item in a column with the standard deviation of the column. Here we will name the matrix as Z .

4. Calculating the Covariance of Z

To calculate the covariance of Z , we will take the matrix Z , and will transpose it. After transpose, we will multiply it by Z . The output matrix will be the Covariance matrix of Z .

5. Calculating the Eigen Values and Eigen Vectors

Now we need to calculate the eigenvalues and eigenvectors for the resultant covariance matrix Z . Eigenvectors or the covariance matrix are the directions of the axes with high information. And the coefficients of these eigenvectors are defined as the eigenvalues.

6. Sorting the Eigen Vectors

In this step, we will take all the eigenvalues and will sort them in decreasing order, which means from largest to smallest. And simultaneously sort the eigenvectors accordingly in matrix P of eigenvalues. The resultant matrix will be named as P^* .

7. Calculating the new features Or Principal Components

Here we will calculate the new features. To do this, we will multiply the P^* matrix to the Z . In the resultant matrix Z^* , each observation is the linear combination of original features. Each column of the Z^* matrix is independent of each other.

8. Remove less or unimportant features from the new dataset.

The new feature set has occurred, so we will decide here what to keep and what to remove. It means, we will only keep the relevant or important features in the new dataset, and unimportant features will be removed out.

Applications of Principal Component Analysis

- PCA is mainly used as the dimensionality reduction technique in various AI applications such as **computer vision, image compression, etc.**
- It can also be used for finding hidden patterns if data has high dimensions. Some fields where PCA is used are Finance, data mining, Psychology, etc.

1.2 RANDOM FOREST ALGORITHM

Random forest algorithm can use both for classification and the regression kind of problems. In this you are going to learn, how the **random forest algorithm** works in machine learning for the classification task.

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.

A random forest 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.

The below diagram explains the working of the Random Forest algorithm:

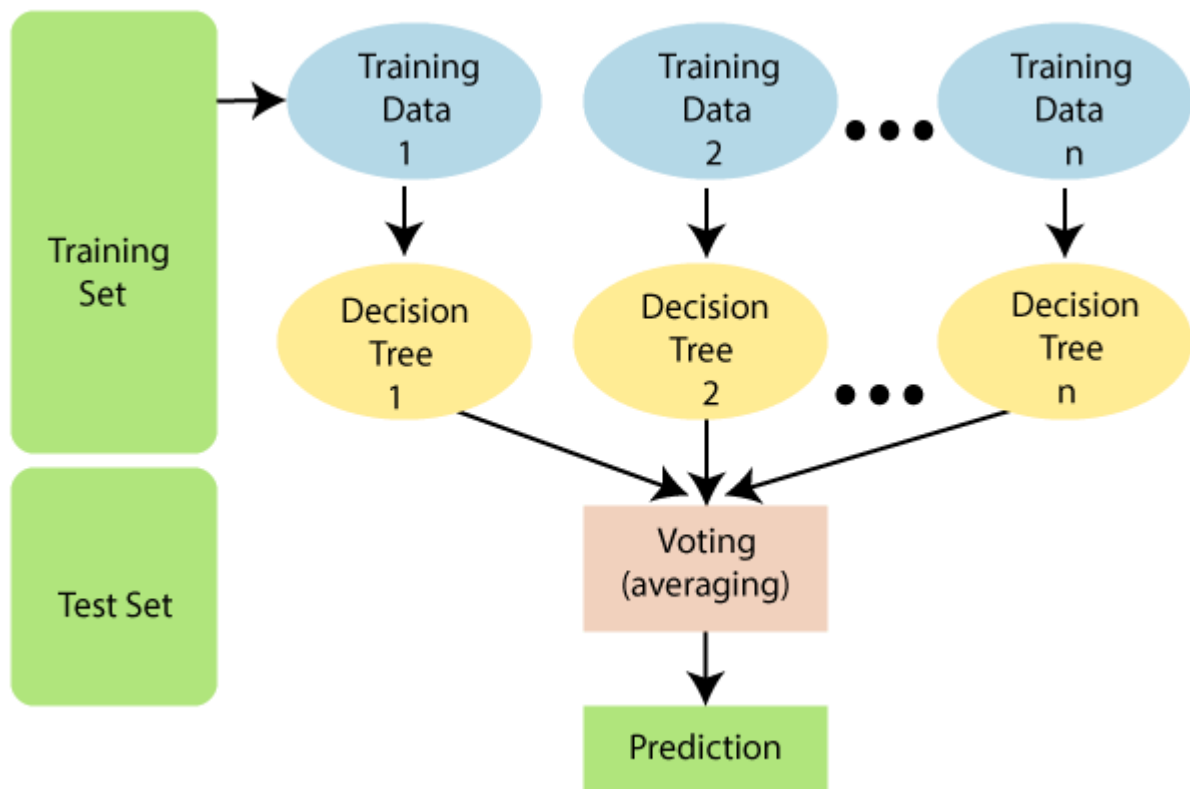


Fig 2: Explaining the working algorithm of the Random Forest algorithm
Below are some points that explain why we should use the Random Forest algorithm:

- It takes less training time as compared to other algorithms.
- It predicts output with high accuracy, even for the large dataset it runs efficiently.
- It can also maintain accuracy when a large proportion of data is missing.

Features of a Random Forest Algorithm

- It's more accurate than the decision tree algorithm.
- It provides an effective way of handling missing data.
- It can produce a reasonable prediction without hyper-parameter tuning.
- It solves the issue of overfitting in decision trees.

- In every random forest tree, a subset of features is selected randomly at the node's splitting point.

Classification in random forests

Classification in random forests employs an ensemble methodology to attain the outcome. The training data is fed to train various decision trees. This dataset consists of observations and features that will be selected randomly during the splitting of nodes.

A rain forest system relies on various decision trees. Every decision tree consists of decision nodes, leaf nodes, and a root node. The leaf node of each tree is the final output produced by that specific decision tree. The selection of the final output follows the majority-voting system. In this case, the output chosen by the majority of the decision trees becomes the final output of the rain forest system. The diagram below shows a simple random forest classifier.

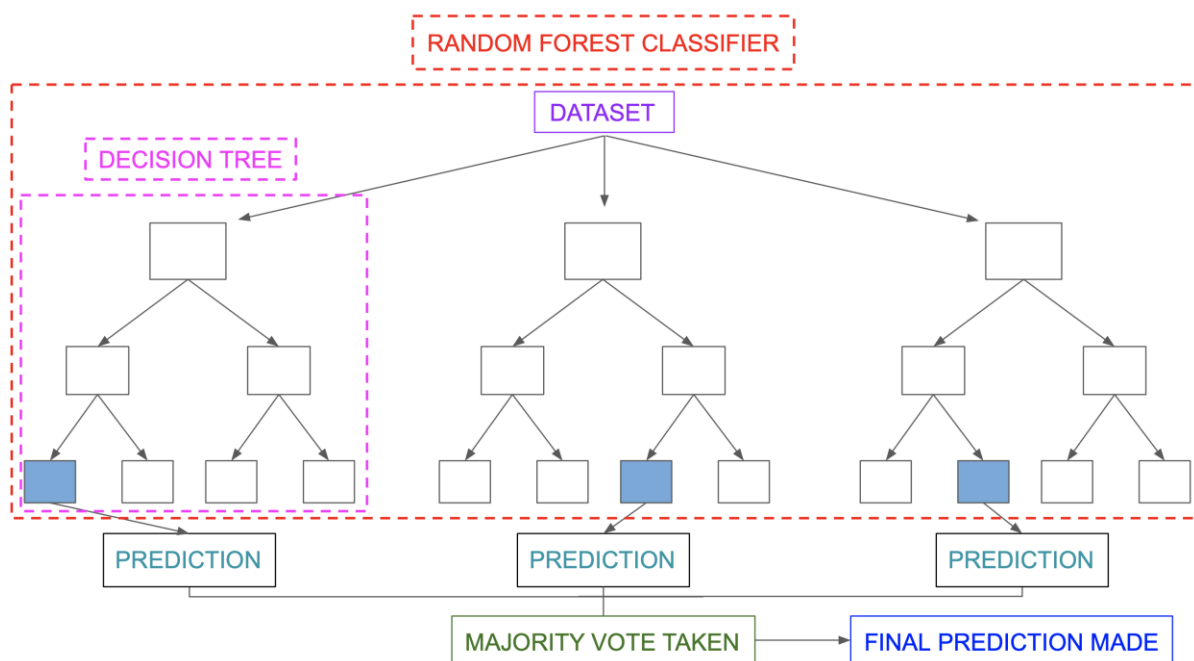


Fig 3: Explaining the Random Forest Classifier

Random Forest Steps

1. Randomly select “k” features from total “m” features.
 1. Where $k \ll m$
2. Among the “k” features, calculate the node “d” using the best split point.
3. Split the node into daughter nodes using the best split.
4. Repeat 1 to 3 steps until “l” number of nodes has been reached.
5. Build forest by repeating steps 1 to 4 for “n” number times to create “n” number of trees.

The beginning of random forest algorithm starts with randomly selecting “k” features out of total “m” features. In the image, you can observe that we are randomly taking features and observations.

The working of the algorithm can be better understood by the below example:

Example: Suppose there is a dataset that contains multiple fruit images. So, this dataset is given to the Random forest classifier. The dataset is divided into subsets and given to each decision tree. During the training phase, each decision tree produces a prediction result, and when a new data point occurs, then based on the majority of results, the Random Forest classifier predicts the final decision. Consider the below image:

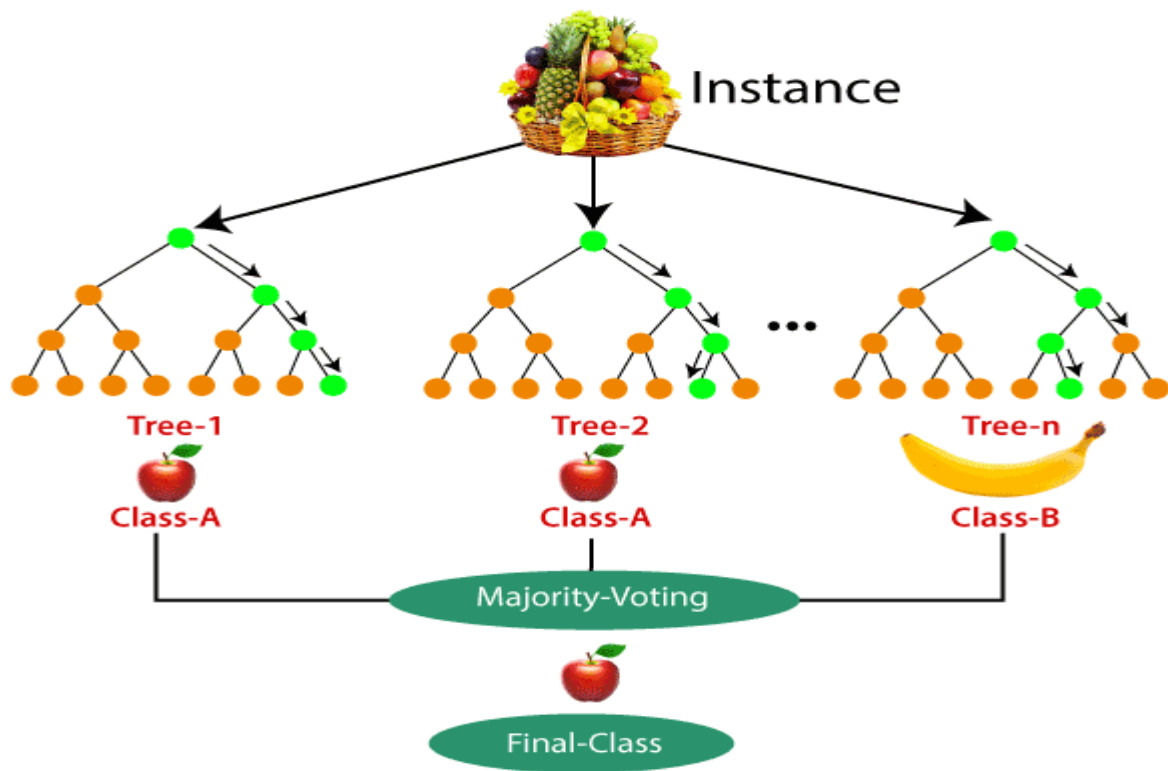


Fig 4: Explaining the Random Forest Classifier algorithm with example

APPLICATIONS OF RANDOM FOREST

There are mainly four sectors where Random forest mostly used:

1. **Banking:** Banking sector mostly uses this algorithm for the identification of loan risk.
2. **Medicine:** With the help of this algorithm, disease trends and risks of the disease can be identified.
3. **Land Use:** We can identify the areas of similar land use by this algorithm.
4. **Marketing:** Marketing trends can be identified using this algorithm.

Advantages of Random Forest

Random Forest is capable of performing both Classification and Regression tasks.

- It is capable of handling large datasets with high dimensionality.
- It enhances the accuracy of the model and prevents the overfitting issue.

Disadvantages of Random Forest

Although random forest can be used for both classification and regression tasks, it is not more suitable for Regression tasks

CHAPTER 2

LITERATURE SURVEY

2.1 A Proposed Wireless Invasion Detection Prevention and Attack System

AUTHORS: JafarAbo 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 invasion 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 invasion detection prevention and attack system to enhance the network security. Therefore, the paper will discuss the development of an invasion detection system on wireless networks which is Wireless Invasion 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.

2.2 Classification of Attack Types for Invasion 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 Invasion Detection Systems. We conclude with discussions and future research projects.

2.3 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 arbitrary 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) Invasion Detection using Random Forests Classifier with SMOTE and Feature Reduction

AUTHORS: A. Tesfahun, D. Lalitha Bhaskari

Invasion Detection Systems (IDS) have become crucial components in computer and network security. NSL-KDD invasion 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 invasion detection, still there is huge imbalance between the classes in the NSL-KDD dataset, which makes harder to apply machine learning effectively in the area of invasion 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 invasion 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 invasion detection.

5) The Impact of PCA-Scale Improving GRU Performance for Invasion 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 Invasion 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 invasion 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.

CHAPTER 3

SYSTEM REQUIREMENTS:

HARDWARE REQUIREMENTS:

- System : Pentium IV 2.4 GHz.
- Hard Disk : 40 GB.
- Floppy Drive : 1.44 Mb.
- Monitor : 15 VGA Colour.
- Mouse : Logitech.
- Ram : 512 Mb.

SOFTWARE REQUIREMENTS:

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

3.3 LANGUAGE SPECIFICATION

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985-1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). This **tutorial** gives enough understanding on **Python programming** language.

3.4. HISTORY OF PYTHON

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

3.5. APPLICATION OF PYTHON

- **Easy-to-learn** – Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.

- **Easy-to-read** – Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** – Python's source code is fairly easy-to-maintain.
- **A broad standard library** – Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** – Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** – Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** – You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** – Python provides interfaces to all major commercial databases.
- **GUI Programming** – Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** – Python provides a better structure and support for large programs than shell scripting.

3.6 FEATURES OF PYTHON

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.

- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

3.7 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

The feasibility study investigates the problem and the information needs of the stakeholders. It seeks to determine the resources required to provide an information systems solution, the cost and benefits of such a solution, and the feasibility of such a solution.

The goal of the feasibility study is to consider alternative information systems solutions, evaluate their feasibility, and propose the alternative most suitable to the organization. The feasibility of a proposed solution is evaluated in terms of its components.

3.7.1 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

3.7.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands

on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

3.7.3 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity.

CHAPTER 4

SYSTEM ANALYSIS

4.1 PURPOSE

The purpose of this document is Invasion detection using pca machine learning algorithms. In detail, this document will provide a general description of our project, including user requirements, product perspective, and overview of requirements, general constraints. In addition, it will also provide the specific

requirements and functionality needed for this project - such as interface, functional requirements and performance requirements.

4.2 SCOPE

The scope of this SRSdocument persists for the entire life cycle of the project. This document defines the final state of the software requirements agreed upon by the customers and designers. Finally at the end of the project execution all the functionalities may be traceable from the SRSto the product. The document describes the functionality, performance, constraints, interface and reliability for the entire life cycle of the project.

4.3 EXISTING SYSTEM:

- ❖ Iftikhar Ahmad et. al, studied various machine learning algorithms for the invasion detection system. They compared some of the techniques like SVM, Extreme learning machine and the random forest. The authors have stated the results as the Extreme machine learning method performs a way better as compared to other algorithms.
- ❖ B. Riyaz et. al., here worked to improve the quality of the dataset to provide it to the invasion detection system. They have used a fuzzy rule-based feature selection technique for the improvement of the dataset. They used the KDD dataset and resulted shown dynamic growth in the result of the IDS.

DISADVANTAGES OF EXISTING SYSTEM:

- ❖ The systems which work over the internet suffer from various malicious activities. The major problem seen in this field is the invasion 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.

4.4 PROPOSED SYSTEM:

- ❖ The invasion 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 principal component analysis is used for the reduction of the dimension of the dataset; by this method, 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.

ADVANTAGES OF PROPOSED SYSTEM:

- ❖ 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.

CHAPTER 5

INPUT DESIGN AND OUTPUT DESIGN

INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing

can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

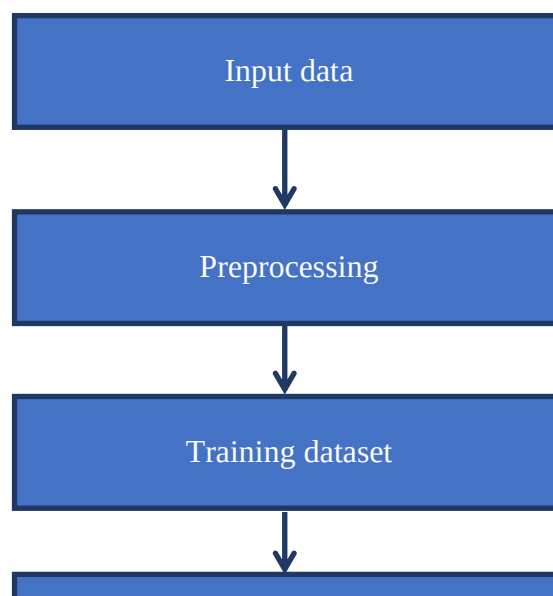
1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.
2. Select methods for presenting information.
3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- ❖ Convey information about past activities, current status or projections of the
- ❖ Future.
- ❖ Signal important events, opportunities, problems, or warnings.
- ❖ Trigger an action.
- ❖ Confirm an action.

DATA FLOW DIAGRAM:

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.



UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

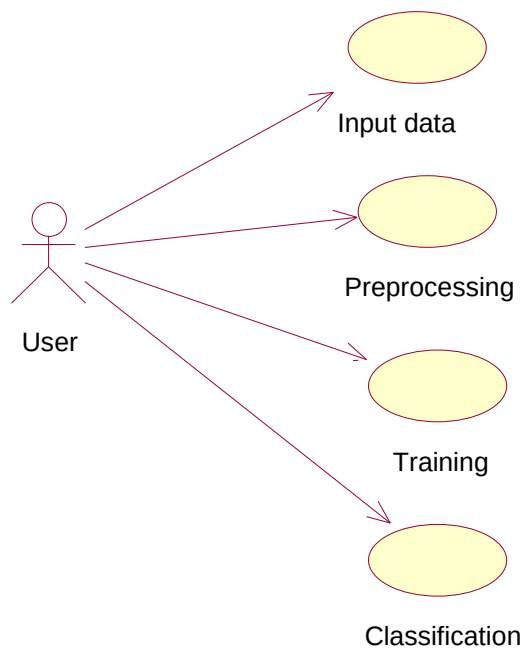
GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

USE CASE DIAGRAM:

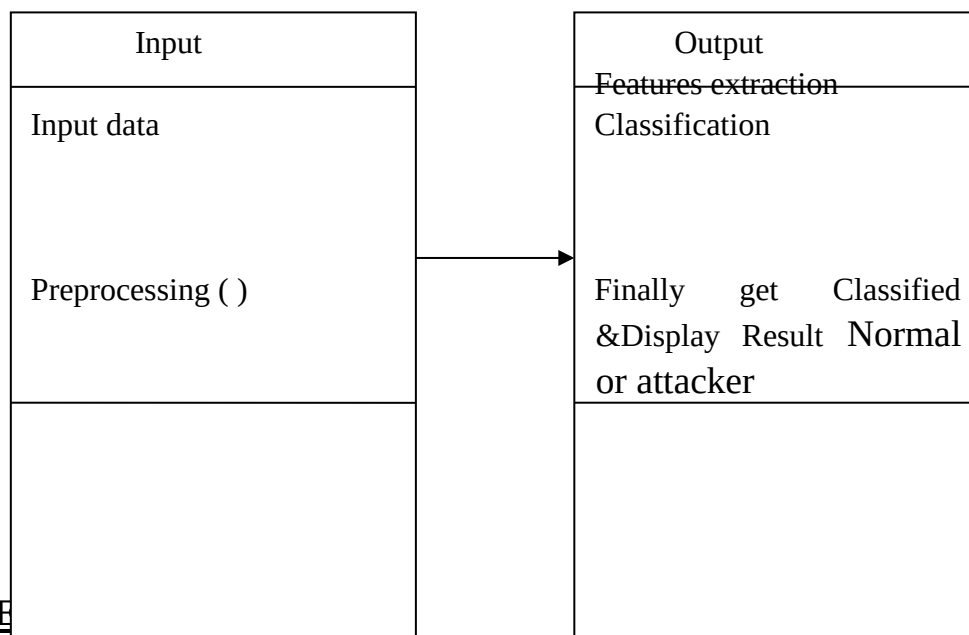
A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



CLASS DIAGRAM:

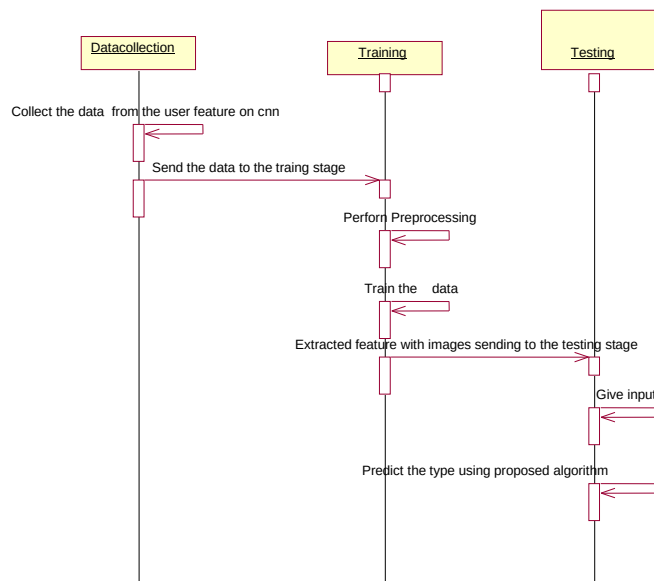
In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system

by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



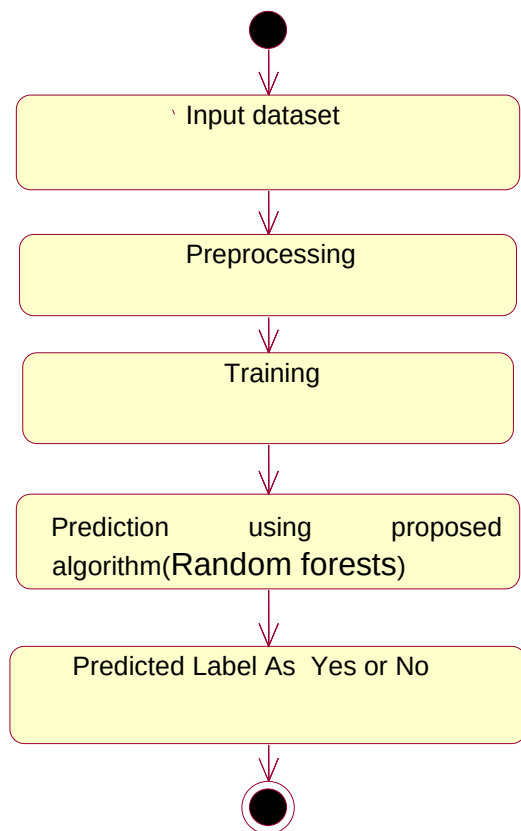
SEQUENCE

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



CHAPTER 6

MODULES

MODULES:

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

MODULES DESCRIPTION:

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 Invasion Detection System dataset taken from kdd
Link: <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>

Dataset:

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

Feature name	Description	Type
Duration	length (number of seconds) of the connection	continuous
Protocol_type	type of the protocol, e.g. tcp, udp, etc.	discrete
Service	network service on the destination, e.g., http, telnet, etc.	discrete
Src_bytes	number of data bytes from source to destination	continuous
Dst_bytes	number of data bytes from destination to source	continuous

Flag	normal or error status of the connection	discrete
Land	1 if connection is from/to the same host/port; 0 otherwise	discrete
Wrong_fragment	number of “wrong” fragments	continuous
Urgent	number of urgent packets	continuous
Hot	number of “hot” indicators	continuous
Num_failed_logins	number of failed login attempts	continuous
Logged_in	1 if successfully logged in; 0 otherwise	discrete
Num_compromised	number of “compromised” conditions	continuous
Root_shell	1 if root shell is obtained; 0 otherwise	discrete
Su_attempted	1 if “su root” command attempted; 0 otherwise	discrete
Num_root	number of “root” accesses	continuous
Num_file_creations	number of file creation operations	continuous
Num_shells	number of shell prompts	continuous
Num_access_files	number of operations on access control files	continuous

Num_outbound_cmds	number of outbound commands in an ftp session	continuous
Is_hot_login	1 if the login belongs to the “hot” list; 0 otherwise	discrete
Is_guest_login	1 if the login is a “guest”login; 0 otherwise	discrete
Error_rate	% of connections that have “SYN” errors	continuous
Error_rate	% of connections that have “REJ” errors	continuous
Same_srv_rate	% of connections to the same service	continuous
Diff_srv_rate	% of connections to different services	continuous
Srv_count	number of connections to the same service as the current connection in the past two seconds	continuous
Srv_error_rate	% of connections that have “SYN” errors	continuous
Srv_rerror_rate	% of connections that have “REJ” errors	continuous
Srv_diff_host_rate	% of connections to different hosts	continuous

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

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 eigenvalue in decreasing order and select n eigenvector with the highest eigenvalues 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.

Analyze and Prediction:

In the actual dataset, we chose only 9 features :

1.Duration	length (number of seconds) of the connection
2.Protocol_type	type of the protocol, e.g. tcp, udp, etc.
3.Src_bytes	number of data bytes from source to destination
4.Dst_bytes	number of data bytes from destination to source
5.Is_hot_login	1 if the login belongs to the “hot” list; 0 otherwise
6.Is_guest_login	1 if the login is a “guest”login; 0 otherwise
7.Diff_srv_rate	% of connections to different services
8.Srv_diff_host_rate	% of connections to different hosts
9.Flag	normal or error status of the connection
10. Labels	Normal or attacker

Accuracy on test set:

We got a accuracy of 99.1% on test set.

Saving the Trained Model:

Once you're confident enough to take your trained and tested model into the production-ready environment, the first step is to save it into a .h5 or .pkl file using a library like `pickle`.

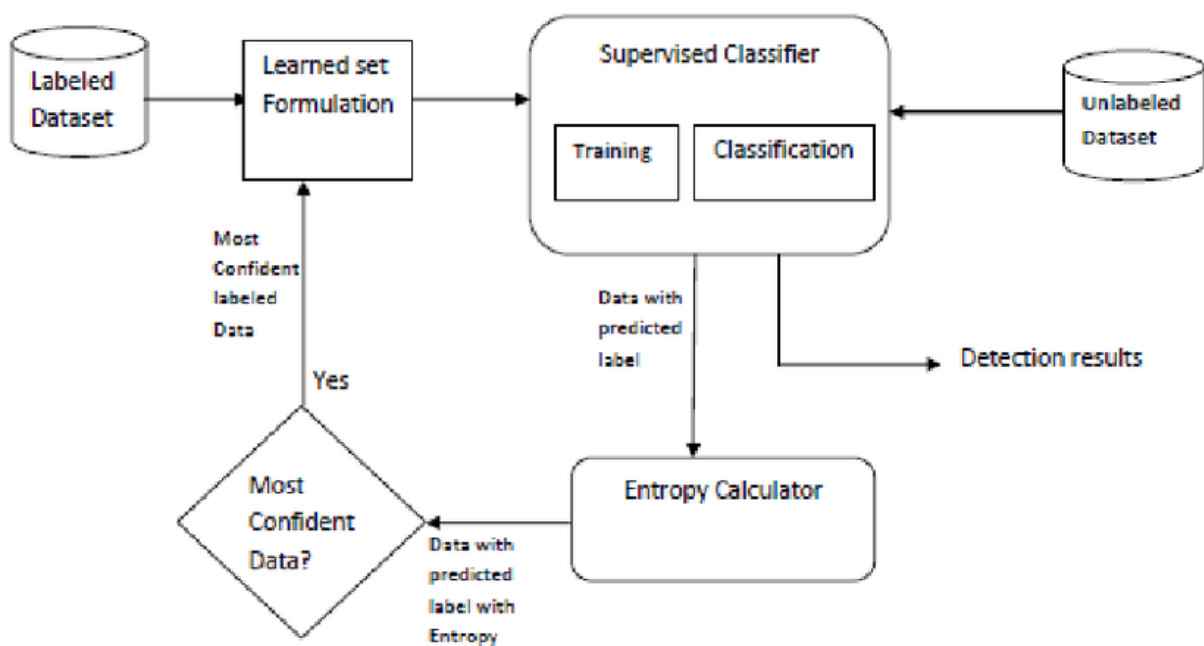
Make sure you have `pickle` installed in your environment.

Next, let's import the module and dump the model into . pkl file

CHAPTER 7

SYSTEM ARCHITECTURE

SYSTEM ARCHITECTURE:



CHAPTER 8

SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

- Valid Input : identified classes of valid input must be accepted.
- Invalid Input : identified classes of invalid input must be rejected.
- Functions : identified functions must be exercised.
- Output : identified classes of application outputs must be exercised.
- Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete,

additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

6.1 Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

6.2 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

6.3 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

8 CONCLUSION

As the involvement of the systems over the internet increasing rapidly, the security concerns have also seen. The proposed approach deals with the detection of intruders over the internet efficiently. The proposed algorithm has performed well as compared to the previously applied algorithms such as SVM, Naïve Bayes, and Decision Tree. The detection rates and the false error rates can be improved at a great extent by the proposed approach. The dataset used here is the knowledge discovery dataset. The results obtained by our proposed method having the values for Performance time (min) is 3.24 minutes, Accuracy rate (%) is 96.78 %, and the Error rate (%) is 0.21 %.

REFERENCES

1. JafarAbo Nada; Mohammad Rasmi Al-Mosa, 2018 International Arab Conference on Information Technology (ACIT), A Proposed Wireless Invasion 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 Invasion 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, “Invasion 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 Invasion 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 INVASION DETECTION SYSTEM.”
7. Mengmeng Ge, Xiping Fu, Naeem Syed, Zubair Baig, Gideon Teo, Antonio Robles-Kelly (2019). Deep Learning-Based Invasion Detection 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 Invasion 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 Invasion 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 Invasion 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 Invasion Detection System: Review”
12. Nimmy Krishnan, A. Salim, 2018 International CET Conference on Control, Communication, and Computing (IC4) “Machine Learning-Based Invasion 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 Invasion 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 Invasion 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 Invasion 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 Invasion Detection.”

Source code

SOURCE CODE :

```
import numpy as np
import pandas as pd
from flask import Flask, request, jsonify, render_template, redirect,
flash, send_file
from sklearn.preprocessing import MinMaxScaler
from werkzeug.utils import secure_filename
import pickle
from sklearn.decomposition import PCA
```

```
app = Flask(__name__) #Initialize the flask App
```

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

```
@app.route('/')
@app.route('/')
@app.route('/index')
def index():
    return render_template('index.html')
@app.route('/login')
```

```

def login():
    return render_template('login.html')
@app.route('/chart')
def chart():
    return render_template('chart.html')
@app.route('/abstract')
def abstract():
    return render_template('abstract.html')
@app.route('/performance')
def performance():
    return render_template('performance.html')
@app.route('/future')
def future():
    return render_template('future.html')
@app.route('/upload')
def upload():
    return render_template('upload.html')
@app.route('/preview',methods=["POST"])
def preview():
    if request.method == 'POST':
        dataset = request.files['datasetfile']
        df = pd.read_csv(dataset,encoding = 'unicode_escape')
        df.set_index('Id', inplace=True)
        return render_template("preview.html",df_view = df)

@app.route('/home')

def home():
    return render_template('test.html')

@app.route('/predict',methods=['POST'])
def predict():

    features = [float(x) for x in request.form.values()]
    final_features = [np.array(features)]
    final_features = pca.transform(final_features)
    prediction = model.predict(final_features)
    output = prediction
    print(output)
    return render_template('test.html', prediction_text= output)

if __name__ == "__main__":
    app.run(debug=True)

TEMPLATES :

ABSTRACT :

```

```

<!DOCTYPE HTML>
<!--
    Road Trip by TEMPLATED
    templated.co @templatedco
    Released for free under the Creative Commons Attribution 3.0
    license (templated.co/license)
-->
<html>
    <head>
        <title>Intrusion Detection System </title>
        <meta charset="utf-8" />
        <meta name="viewport" content="width=device-width,
initial-scale=1" />
        <link rel="stylesheet" href="../static/css/mains.css" />
    </head>
    <body>

        <!-- Banner -->
        <!--
            Note: To show a background image, set the "data-
bg" attribute below
            to the full filename of your image. This is used
in each section to set
            the background image.
        -->

        <!-- One -->
        <section id="one" class="wrapper post bg-img"
data-bg="../images/image.jpg">
            <div class="inner">
                <article class="box">
                    <header>
                        <h2>ABSTRACT</h2>

                    </header>
                    <div class="content">
                        <p>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 %.</p>

</div>

</article>

</div>

</section>

<!-- Two -->

<!-- Scripts -->

<script

src="../../../static/js/jquery.min.js"></script>

<script

src="../../../static/js/jquery.scrolly.min.js"></script>

<script

src="../../../static/js/jquery.scrollex.min.js"></script>

<script src="../../../static/js/skel.min.js"></script>

<script src="../../../static/js/util.js"></script>

<script src="../../../static/js/mains.js"></script>

<input type="button"

onclick="window.location.href='{{url_for('upload')}}';" value="UPLOAD"

/>

</body>

</html>

INDEX :

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<!-- SEO Meta Tags -->

<meta name="description" content="Aria is a business focused HTML landing page template built with Bootstrap to help you create lead generation websites for companies and their services.">

<meta name="author" content="Inovatik">

<!-- OG Meta Tags to improve the way the post looks when you share the page on LinkedIn, Facebook, Google+ -->

<meta property="og:site_name" content="" /> <!-- website name -->

<meta property="og:site" content="" /> <!-- website link -->

<meta property="og:title" content="" /> <!-- title shown in the actual shared post -->

<meta property="og:description" content="" /> <!-- description shown in the actual shared post -->

<meta property="og:image" content="" /> <!-- image link, make sure it's jpg -->

```

        <meta property="og:url" content="" /> <!-- where do you want
your post to link to -->
        <meta property="og:type" content="article" />

<!-- Website Title -->
<title>INTRUSION DETECTION SYSTEM</title>

<!-- Styles -->
<link
href="https://fonts.googleapis.com/css?family=Montserrat:500,700&displa
y=swap&subset=latin-ext" rel="stylesheet">
<link
href="https://fonts.googleapis.com/css?family=Open+Sans:400,400i,600&di
splay=swap&subset=latin-ext" rel="stylesheet">
<link href="../../static/css/bootstrap.css" rel="stylesheet">
<link href="../../static/css/fontawesome-all.css" rel="stylesheet">
<link href="../../static/css/swiper.css" rel="stylesheet">
<link href="../../static/css/magnific-popup.css" rel="stylesheet">
<link href="../../static/css/styles.css" rel="stylesheet">

<!-- Favicon -->

</head>
<body data-spy="scroll" data-target=".fixed-top">

<!-- Preloader -->
<div class="spinner-wrapper">
<div class="spinner">
<div class="bounce1"></div>
<div class="bounce2"></div>
<div class="bounce3"></div>
</div>
</div>
<!-- end of preloader -->

<!-- Navbar -->
<nav class="navbar navbar-expand-md navbar-dark navbar-custom
fixed-top">
<!-- Text Logo - Use this if you don't have a graphic logo -->
<!-- <a class="navbar-brand logo-text page-scroll"
href="index.html">Aria</a> -->

<!-- Mobile Menu Toggle Button -->
<button class="navbar-toggler" type="button" data-
toggle="collapse" data-target="#navbarsExampleDefault" aria-
controls="navbarsExampleDefault" aria-expanded="false" aria-
label="Toggle navigation">
<span class="navbar-toggler-awesome fas fa-bars"></span>
<span class="navbar-toggler-awesome fas fa-times"></span>
</button>
<!-- end of mobile menu toggle button -->

<div class="collapse navbar-collapse"
id="navbarsExampleDefault">
<ul class="navbar-nav ml-auto">

```

```

        <li class="nav-item">
            <a class="nav-link page-scroll" href="#">HOME <span
class="sr-only">(current)</span></a>
        </li>
        <li class="nav-item">
            <a class="nav-link page-scroll" href="{{
url_for('abstract') }}">ABSTRACT</a>
        </li>
        <li class="nav-item">
            <a class="nav-link page-scroll" href="{{
url_for('upload') }}">UPLOAD DATA</a>
        </li>
        <li class="nav-item">
            <a class="nav-link page-scroll" href="{{
url_for('home') }}">PREDICTION</a>
        </li>
        <li class="nav-item">
            <a class="nav-link page-scroll" href="{{
url_for('future') }}">FUTURE</a>
        </li>
        <li class="nav-item">
            <a class="nav-link page-scroll" href="{{
url_for('chart') }}">CHART</a>
        </li>
        <li class="nav-item">
            <a class="nav-link page-scroll" href="{{
url_for('performance') }}">PERFORMANCE ANALYSIS</a>
        </li>

        <!-- Dropdown Menu -->

    </ul>

</div>
</nav> <!-- end of navbar -->
<!-- end of navbar -->

<!-- Header -->
<header id="header" class="header">
    <div class="header-content">
        <div class="container">
            <div class="row">
                <div class="col-lg-12">
                    <div class="text-container">
                        <h1>INVASION IDENTIFICATION SYSTEM<span
id="js-rotating"> </span></h1>
                        <p class="p-heading p-large">Using PCA with
Random Forest Approach</p>
                        <a class="btn-solid-lg page-scroll"
href="{{ url_for('login') }}">LOGIN</a>
                    </div>
                </div> <!-- end of col -->
            </div> <!-- end of row -->
        </div> <!-- end of container -->
    </div> <!-- end of header-content -->
</header> <!-- end of header -->
<!-- end of header -->

```

```

    <!-- Scripts -->
    <script src="../../static/js/jquery.min.js"></script> <!-- jQuery for
Bootstrap's JavaScript plugins -->
    <script src="../../static/js/popper.min.js"></script> <!-- Popper
tooltip library for Bootstrap -->
    <script src="../../static/js/bootstrap.min.js"></script> <!--
Bootstrap framework -->
    <script src="../../static/js/jquery.easing.min.js"></script> <!--
jQuery Easing for smooth scrolling between anchors -->
    <script src="../../static/js/swiper.min.js"></script> <!-- Swiper for
image and text sliders -->
    <script src="../../static/js/jquery.magnific-popup.js"></script> <!--
Magnific Popup for lightboxes -->
    <script src="../../static/js/morphext.min.js"></script> <!-- Morphtext
rotating text in the header -->
    <script src="../../static/js/isotope.pkgd.min.js"></script> <!--
Isotope for filter -->
    <script src="../../static/js/validator.min.js"></script> <!--
Validator.js - Bootstrap plugin that validates forms -->
    <script src="../../static/js/scripts.js"></script> <!-- Custom scripts
-->
</body>
</html>

```

LOGIN :

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-
scale=1.0" />
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <title>Login</title>
    <link rel="preconnect" href="https://fonts.gstatic.com" />
    <link
href="https://fonts.googleapis.com/css2?family=Open+Sans:wght@400;600&d
isplay=swap"
      rel="stylesheet"
    />
    <link
      rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.css"
      integrity="sha512-
5A8nwdMOWrSz20fDsJczgUIdUBR8liPYU+WymTZP1lmY9G6Oc7HlZv156XqnsqNUzTyMefF
TcsFH/tnJE/+xBg=="
      crossorigin="anonymous"

```

```

    />
    <link rel="stylesheet" href="../../static/css/main.css" />
</head>
<body>
    <div id="container">
        <div id="header">
            <div class="logo"><i class="fa fa-weibo" aria-
hidden="true"></i>
            </div>
            <h1>Login Admin</h1>
        </div>
        <form action="{{ url_for('abstract')}}">
            <div id="loginForm">
                <div class="formFields">
                    <input type="email" placeholder="Email Address" />
                    <input type="pass" placeholder="Password" />
                </div>
                <input type="submit" value="Log in" />
            </div>
        </form>

    </div>
</body>
</html>

```

UPLOAD :

```

<!DOCTYPE html>
<html lang="en">

<head>

    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">

    <title>INVASION IDENTIFICATION System</title>

    <!-- Custom fonts for this theme -->
    <link href="../../static/vendor/fontawesome-free/css/all.min.css"
rel="stylesheet" type="text/css">
    <link
href="https://fonts.googleapis.com/css?family=Montserrat:400,700"
rel="stylesheet" type="text/css">
    <link
href="https://fonts.googleapis.com/css?family=Lato:400,700,400italic,70
0italic" rel="stylesheet" type="text/css">

    <!-- Theme CSS -->
    <link href="../../static/css/freelancer.min.css" rel="stylesheet">

</head>

<body id="page-top">

```

```

    <!-- Navigation -->
    <nav class="navbar navbar-expand-lg bg-secondary text-uppercase
fixed-top" id="mainNav">
        <div class="container">
            <a class="navbar-brand js-scroll-trigger" href="#page-
top">Invasion Identification System</a>

        </nav>

    <!-- About Section -->
    <section class="page-section bg-primary text-white mb-0" id="about">
        <div class="container">
            <br>
            <br>
            <!-- About Section Heading -->
            <h2 class="text-center text-uppercase text-white">Upload
Dataset</h2>

            <!-- Icon Divider -->
            <div class="divider-custom divider-light">
                <div class="divider-custom-line"></div>
                <div class="divider-custom-icon">
                    <i class="fas fa-star"></i>
                </div>
                <div class="divider-custom-line"></div>
            </div>

            <!-- About Section Content -->
            <div class="row">
                <div class="col-lg-4 ml-auto" style="margin-right:250px;">
                    <form action="http://localhost:5000/preview" name="fs"
id="fs" method="post" enctype=multipart/form-data>
                        <br/>
                        <input type="file" name="datasetfile" id="file1"
required />
                        <br/>
                        <br/>
                        <input type="submit" style="margin-right:250px"
class="" value="Upload">
                    </form>
                </div>
            </div>

        </section>

        <div class="scroll-to-top d-lg-none position-fixed ">
            <a class="js-scroll-trigger d-block text-center text-white rounded"
href="#page-top">
                <i class="fa fa-chevron-up"></i>
            </a>
        </div>

    <!-- Bootstrap core JavaScript -->
    <script src="../../static/vendor/jquery/jquery.min.js"></script>

```

```

    <script
src="../../static/vendor/bootstrap/js/bootstrap.bundle.min.js"></script>

    <!-- Plugin JavaScript -->
    <script src="../../static/vendor/jquery-
easing/jquery.easing.min.js"></script>

    <!-- Contact Form JavaScript -->
    <script src="../../static/js/jqBootstrapValidation.js"></script>
    <script src="../../static/js/contact_me.js"></script>

    <!-- Custom scripts for this template -->
    <script src="../../static/js/freelancer.min.js"></script>

</body>

```

```

</html>

```

PREVIEW :

```

<!DOCTYPE html>
<html lang="en">

<head>

    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">

    <title>Invasion identification System</title>

    <!-- Custom fonts for this theme -->
    <link href="../../static/vendor/fontawesome-free/css/all.min.css"
rel="stylesheet" type="text/css">
    <link
href="https://fonts.googleapis.com/css?family=Montserrat:400,700"
rel="stylesheet" type="text/css">
    <link
href="https://fonts.googleapis.com/css?family=Lato:400,700,400italic,70
0italic" rel="stylesheet" type="text/css">

    <!-- Theme CSS -->
    <link href="../../static/css/freelancer.min.css" rel="stylesheet">
    <style>
    #loading {
        background: url('../../static/ajax-loader.gif') no-repeat center
center;
        position: absolute;
        top: 0;
        left: 59;
        height: 100%;
    }

```

```

        width: 86%;
        z-index: 9999999;
    }
    </style>

</head>

<body id="page-top">

    <!-- Navigation -->
    <nav class="navbar navbar-expand-lg bg-secondary text-uppercase
fixed-top" id="mainNav">
        <div class="container">
            <a class="navbar-brand js-scroll-trigger" href="#page-
top">Invasion identification System</a>

        </nav>

    <!-- Contact Section -->
    <section class="page-section" id="contact">
        <div class="container">
            <br>
            <br>
            <!-- Contact Section Heading -->
            <h2 class="text-center text-uppercase text-secondary mb-
0">Preview</h2>

            <!-- Icon Divider -->
            <div class="divider-custom">
                <div class="divider-custom-line"></div>
                <div class="divider-custom-icon">
                    <i class="fas fa-star"></i>
                </div>
                <div class="divider-custom-line"></div>
            </div>

            <!-- Contact Section Form -->
            <div class="row" style="margin-left:-350px">
                <div class="col-lg-8 mx-auto">
                    <!-- To configure the contact form email address, go to
mail/contact_me.php and update the email address in the PHP file on
line 19. -->
                    {{ df_view.to_html(classes="table striped",na_rep="-") | safe}}
                </div>

            </div>

        </section>
        <div class="form-group" style="padding:0px 250px 10px
40px;height:200px">
            <input style="margin-left:500px" type="button"
onclick="hideLoader()" class="btn btn-primary" value="Click to Train |
Test" />

            <div id="loading"
style="display:None;margin-top:552950px"></div>
        </div>

```

```

<!-- Copyright Section -->
<section class="copyright py-4 text-center text-white">
  <div class="container">

    </div>
  </section>

  <!-- Scroll to Top Button (Only visible on small and extra-small
screen sizes) -->
  <div class="scroll-to-top d-lg-none position-fixed ">
    <a class="js-scroll-trigger d-block text-center text-white rounded"
href="#page-top">
      <i class="fa fa-chevron-up"></i>
    </a>
  </div>

  <!-- Bootstrap core JavaScript -->
  <script src="../../static/vendor/jquery/jquery.min.js"></script>
  <script
src="../../static/vendor/bootstrap/js/bootstrap.bundle.min.js"></script>

  <!-- Plugin JavaScript -->
  <script src="../../static/vendor/jquery-
easing/jquery.easing.min.js"></script>

  <!-- Contact Form JavaScript -->
  <script src="../../static/js/jqBootstrapValidation.js"></script>
  <script src="../../static/js/contact_me.js"></script>

  <!-- Custom scripts for this template -->
  <script src="../../static/js/freelancer.min.js"></script>

  <script type='text/javascript'
src='https://ajax.googleapis.com/ajax/libs/jquery/2.2.4/jquery.min.js'>
</script>
  <script type='text/javascript'>

function hideLoader() {
$('#loading').show(0).delay(10000).hide(0,function(){
  alert("Training finished!");
  window.location = "{{url_for('home')}}";
});
}

  </script>

</body>

</html>

TEST :

<!DOCTYPE html>

```

```

<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Intrusion Detection System</title>
  <link rel="stylesheet" href="{{ url_for('static',
filename='css/style.css') }}">
</head>
<body class = "background">

  <div class="login">
    <h1>Invasion identification System</h1>
    <form action="{{ url_for('predict') }}"method="post">
      <label for="Duration">Duration:</label>
      <input type="text" name="duration" placeholder="duration"
required="required" />
      <label for="protocol_type">protocol_type:</label>
      <select class="input1"
id="protocol_type" name="protocol_type" required="required"
placeholder="protocol_type">
        <option
value="0">icmp</option>
        <option
value="1">tcp</option>
        <option
value="2">udp</option>
      </select></br>
      </br>
      <label for="src_bytes">src_bytes:</label>
      <input type="text" name="src_bytes" placeholder="src_bytes"
required="required" />

      <label for="dst_bytes">dst_bytes:</label>
      <input type="text" name="dst_bytes" placeholder="dst_bytes"
required="required" />

      <label for="is_host_login">is_host_login:</label>
      <input type="text" name="is_host_login"
placeholder="is_host_login" required="required" />
      <label for="is_guest_login">is_guest_login:</label>
      <input type="text" name="is_guest_login"
placeholder="is_guest_login" required="required" />
      <label
for="diff_srv_rate">diff_srv_rate:</label>
      <input type="text" name="diff_srv_rate"
placeholder="diff_srv_rate" required="required" />
      <label
for="srv_diff_host_rate">srv_diff_host_rate:</label>
      <input type="text" name="srv_diff_host_rate"
placeholder="srv_diff_host_rate" required="required" />
      <label for="flag">flag:</label>
      <input type="text" name="flag" placeholder="flag"
required="required" />
    </form>
  </div>

```

```
        <button type="submit" class="btn btn-primary btn-block btn-
large">Predict label</button>
    </form>
    <br>
    <br>
    class is {{ prediction_text }}
    <br>
    <br>
    <a href="{{ url_for('chart')}}"button type="submit" class="btn btn-
primary btn-block btn-large">Analysis</button></a>
</div>

</body>
</html>
```

