

PREDICTION OF NETWORK ATTACKS USING SUPERVISED MACHINE LEARNING TECHNIQUE

¹B.SAI GIRIDHAR, ²A. Mallikarjuna, ³N. Naresh, ⁴K. Rakesh

^{1,2,3,4}Students,

Bharath institute of higher education and research

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 Naïve Bayes, and Decision Tree.

OBJECTIVE

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

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.

LITERATURE SURVEY

1) A Proposed Wireless Intrusion Detect ion Prevent ion 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 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 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) 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, 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 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) The 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.

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

SYSTEM ANALYSIS

EXISTING SYSTEM:

❖ Iftikhar Ahmad et. al, studied various machine learning algorithms for the intrusion detection system. They compared some of the techniques Extreme learning machine. 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 intrusion 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 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 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.

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 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.

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

- **Python is Interpreted** – Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** – You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** – Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** – Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

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.

Python Features

Python's features include –

- **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.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below –

- 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.

Python is available on a wide variety of platforms including Linux and Mac OS X. Let's understand how to set up our Python environment.

Getting Python

The most up-to-date and current source code, binaries, documentation, news, etc., is available on the official website of Python <https://www.python.org>.

Windows Installation

Here are the steps to install Python on Windows machine.

- Open a Web browser and go to <https://www.python.org/downloads/>.
- Follow the link for the Windows installer python-XYZ.msifile where XYZ is the version you need to install.
- To use this installer python-XYZ.msi, the Windows system must support Microsoft Installer 2.0. Save the installer file to your local machine and then run it to find out if your machine supports MSI.
- Run the downloaded file. This brings up the Python install wizard, which is really easy to use. Just accept the default settings, wait until the install is finished, and you are done.

The Python language has many similarities to Perl, C, and Java. However, there are some definite differences between the languages.

First Python Program

Let us execute programs in different modes of programming.

Interactive Mode Programming

Invoking the interpreter without passing a script file as a parameter brings up the following prompt –

```
$ python
Python2.4.3(#1,Nov112010,13:34:43)
[GCC 4.1.220080704(RedHat4.1.2-48)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Type the following text at the Python prompt and press the Enter –

```
>>>print"Hello, Python!"
```

If you are running new version of Python, then you would need to use print statement with parenthesis as in **print ("Hello, Python!");**. However in Python version 2.4.3, this produces the following result –

```
Hello, Python!
```

Script Mode Programming

Invoking the interpreter with a script parameter begins execution of the script and continues until the script is finished. When the script is finished, the interpreter is no longer active.

Let us write a simple Python program in a script. Python files have extension **.py**. Type the following source code in a test.py file –

```
print"Hello, Python!"
```

We assume that you have Python interpreter set in PATH variable. Now, try to run this program as follows –

```
$ python test.py
```

This produces the following result –

```
Hello, Python!
```

Flask Framework:

Flask is a web application framework written in Python. Armin Ronacher, who leads an international group of Python enthusiasts named Pocco, develops it. Flask is based on Werkzeug WSGI toolkit and Jinja2 template engine. Both are Pocco projects.

Http protocol is the foundation of data communication in world wide web. Different methods of data retrieval from specified URL are defined in this protocol.

The following table summarizes different http methods –

Sr.No	Methods & Description
1	GET Sends data in unencrypted form to the server. Most common method.
2	HEAD Same as GET, but without response body
3	POST Used to send HTML form data to server. Data received by POST method is not cached by server.

4	<p>PUT Replaces all current representations of the target resource with the uploaded content.</p>
5	<p>DELETE Removes all current representations of the target resource given by a URL</p>

By default, the Flask route responds to the **GET** requests. However, this preference can be altered by providing methods argument to **route()** decorator.

In order to demonstrate the use of **POST** method in URL routing, first let us create an HTML form and use the **POST** method to send form data to a URL.

Save the following script as login.html

```
<html>
<body>
<formaction="http://localhost:5000/login"method="post">
<p>Enter Name:</p>
<p><inputtype="text"name="nm"/></p>
<p><inputtype="submit"value="submit"/></p>
</form>
</body>
</html>
```

Now enter the following script in Python shell.

```
from flask import Flask, redirect, url_for, request
app = Flask(__name__)
@app.route('/success/<name>')
def success(name):
return 'welcome %s' % name
@app.route('/login', methods=['POST', 'GET'])
def login():
if request.method == 'POST':
user = request.form['nm']
return redirect(url_for('success', name = user))
else:
user = request.args.get('nm')
return redirect(url_for('success', name = user))
if __name__ == '__main__':
app.run(debug = True)
```

After the development server starts running, open **login.html** in the browser, enter name in the text field and click **Submit**.



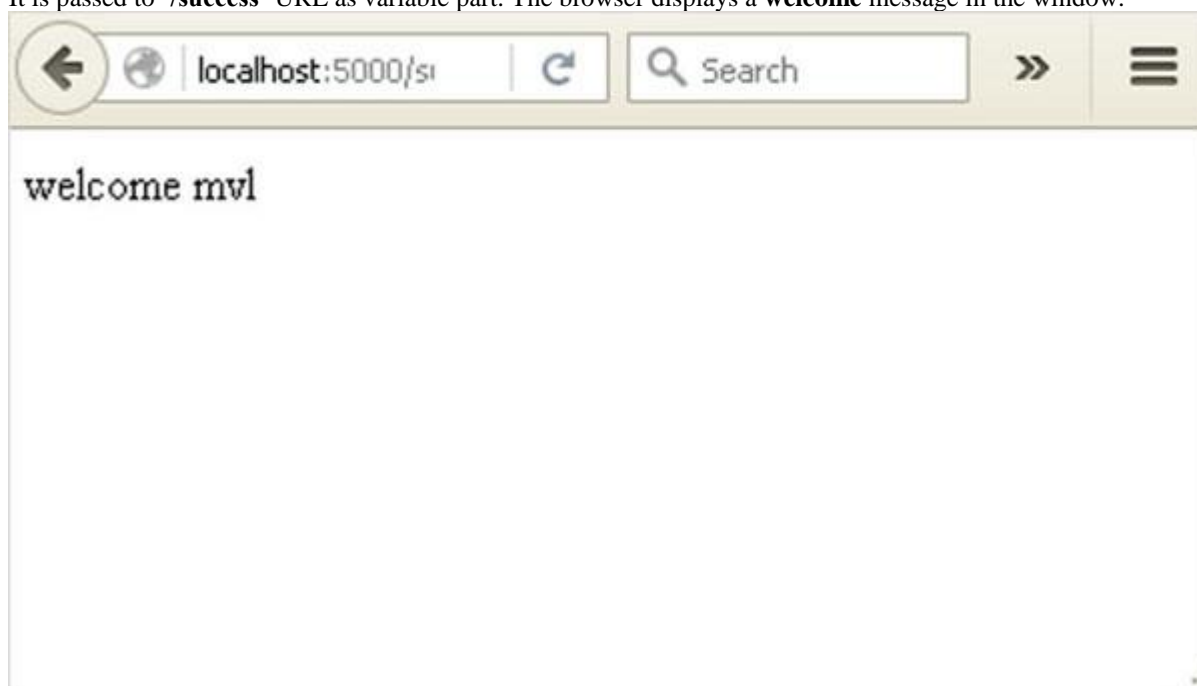
Enter Name:

Form data is POSTed to the URL in action clause of form tag.

http://localhost/login is mapped to the **login()** function. Since the server has received data by **POST** method, value of 'nm' parameter obtained from the form data is obtained by –

```
user = request.form['nm']
```

It is passed to '/success' URL as variable part. The browser displays a **welcome** message in the window.



welcome mvl

Change the method parameter to '**GET**' in **login.html** and open it again in the browser. The data received on server is by the **GET** method. The value of 'nm' parameter is now obtained by –

```
User = request.args.get('nm')
```

Here, **args** is dictionary object containing a list of pairs of form parameter and its corresponding value. The value corresponding to 'nm' parameter is passed on to '/success' URL as before.

What is Python?

Python is a popular programming language. It was created in 1991 by Guido van Rossum.

It is used for:

- web development (server-side),
- software development,
- mathematics,

- system scripting.

What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-orientated way or a functional way.

Good to know

- The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is still quite popular.
- In this tutorial Python will be written in a text editor. It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Netbeans or Eclipse which are particularly useful when managing larger collections of Python files.

Python Syntax compared to other programming languages

- Python was designed to for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

Python Install

Many PCs and Macs will have python already installed.

To check if you have python installed on a Windows PC, search in the start bar for Python or run the following on the Command Line (cmd.exe):

```
C:\Users\Your Name>python --version
```

To check if you have python installed on a Linux or Mac, then on linux open the command line or on Mac open the Terminal and type:

```
python --version
```

If you find that you do not have python installed on your computer, then you can download it for free from the following website: <https://www.python.org/>

Python Quickstart

Python is an interpreted programming language, this means that as a developer you write Python (.py) files in a text editor and then put those files into the python interpreter to be executed.

The way to run a python file is like this on the command line:

```
C:\Users\Your Name>python helloworld.py
```

Where "helloworld.py" is the name of your python file.

Let's write our first Python file, called helloworld.py, which can be done in any text editor.

```
helloworld.py
```

```
print("Hello, World!")
```

Simple as that. Save your file. Open your command line, navigate to the directory where you saved your file, and run:

```
C:\Users\Your Name>python helloworld.py
```

The output should read:

```
Hello, World!
```

Congratulations, you have written and executed your first Python program.

The Python Command Line

To test a short amount of code in python sometimes it is quickest and easiest not to write the code in a file. This is made possible because Python can be run as a command line itself.

Type the following on the Windows, Mac or Linux command line:

```
C:\Users\Your Name>python
```

From there you can write any python, including our hello world example from earlier in the tutorial:

```
C:\Users\Your Name>python
```

```
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)] on win32
```

Type "help", "copyright", "credits" or "license" for more information.

```
>>> print("Hello, World!")
```

Which will write "Hello, World!" in the command line:

```
C:\Users\Your Name>python
```

```
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)] on win32
```

Type "help", "copyright", "credits" or "license" for more information.

```
>>> print("Hello, World!")
```

```
Hello, World!
```

Whenever you are done in the python command line, you can simply type the following to quit the python command line interface:

```
exit()
```

Execute Python Syntax

As we learned in the previous page, Python syntax can be executed by writing directly in the Command Line:

```
>>> print("Hello, World!")
```

```
Hello, World!
```

Or by creating a python file on the server, using the .py file extension, and running it in the Command Line:

```
C:\Users\Your Name>python myfile.py
```

Python Indentations

Where in other programming languages the indentation in code is for readability only, in Python the indentation is very important.

Python uses indentation to indicate a block of code.

Example

```
if 5 > 2:
    print("Five is greater than two!")
```

Python will give you an error if you skip the indentation:

Example

```
if 5 > 2:
print("Five is greater than two!")
```

Comments

Python has commenting capability for the purpose of in-code documentation.

Comments start with a #, and Python will render the rest of the line as a comment:

Example

Comments in Python:

```
#This is a comment.
```

```
print("Hello, World!")
```

Docstrings

Python also has extended documentation capability, called docstrings.

Docstrings can be one line, or multiline.

Python uses triple quotes at the beginning and end of the docstring:

Example

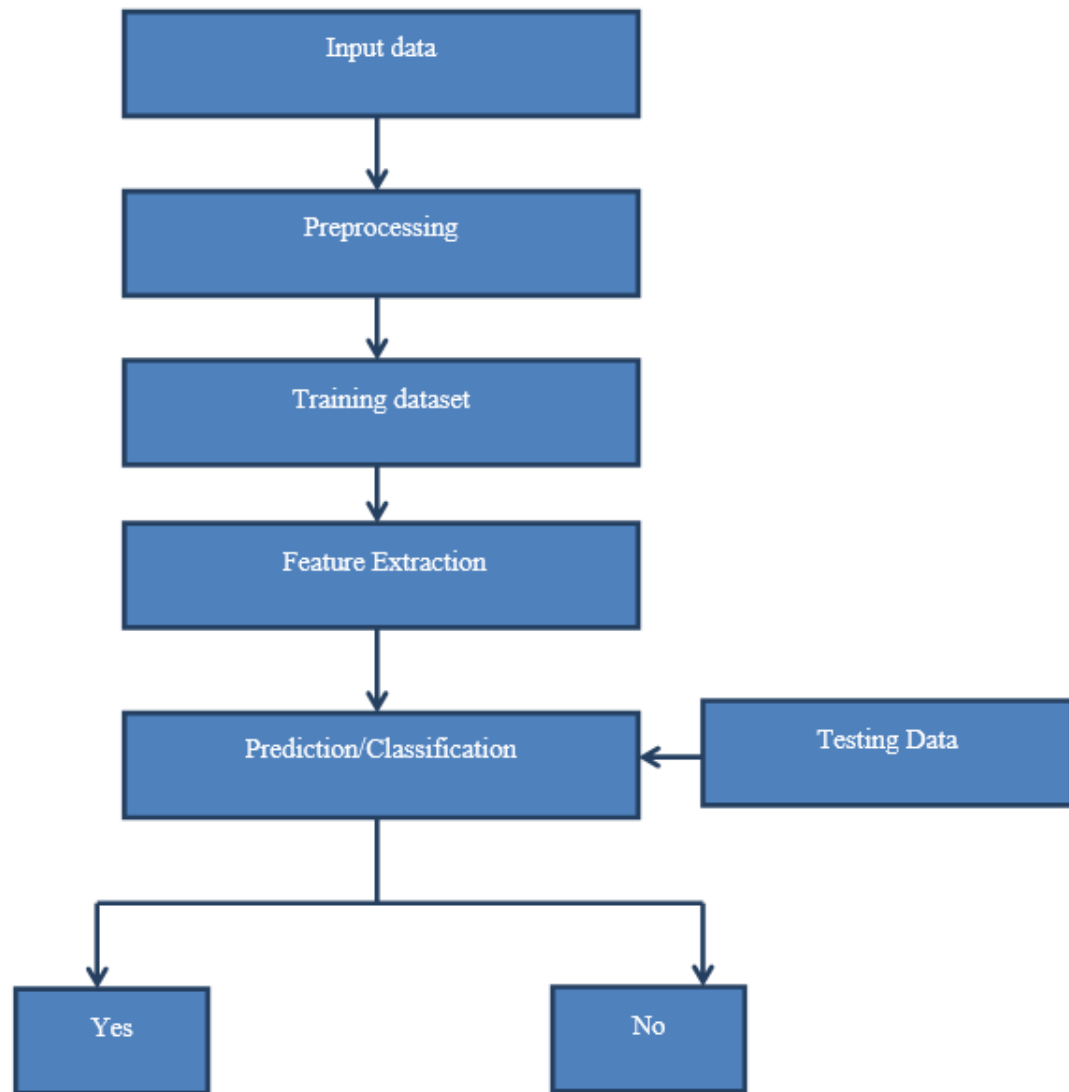
Docstrings are also comments:

```
"""This is a docstring."""
```

```
print("Hello, World!")
```

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:

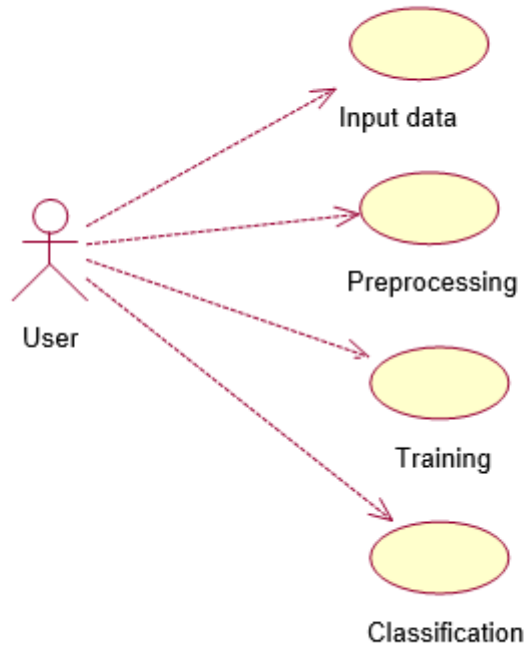
The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extensibility 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.

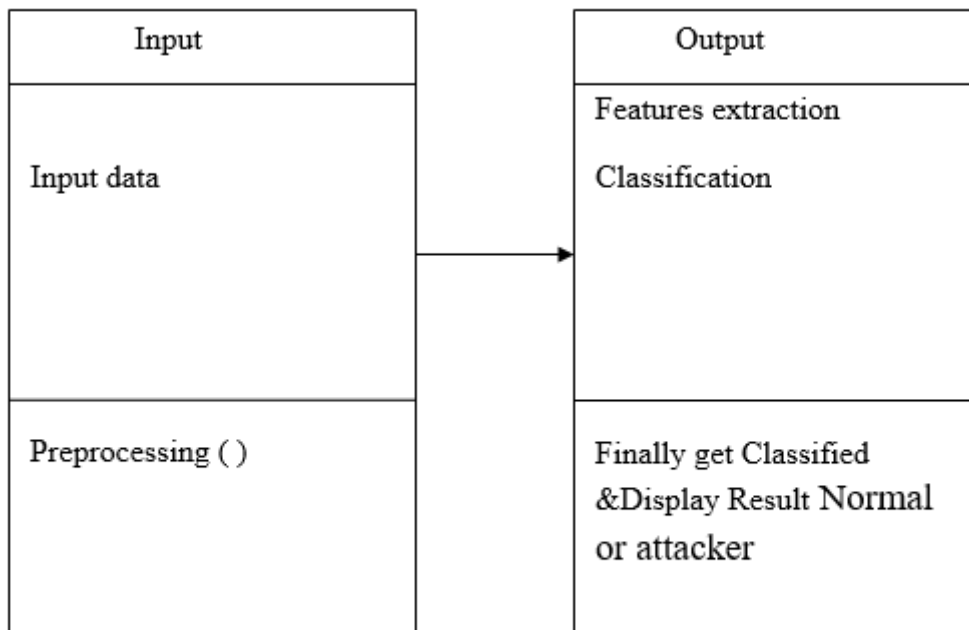
USE CASE DIAGRAM:

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.

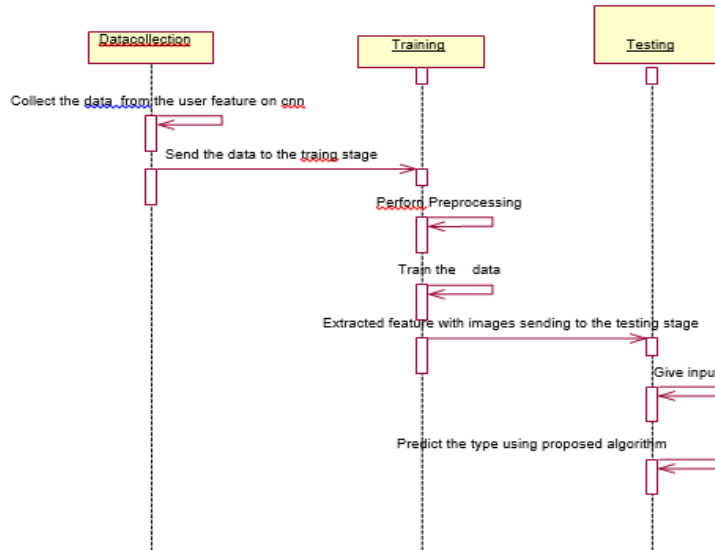


CLASS DIAGRAM:



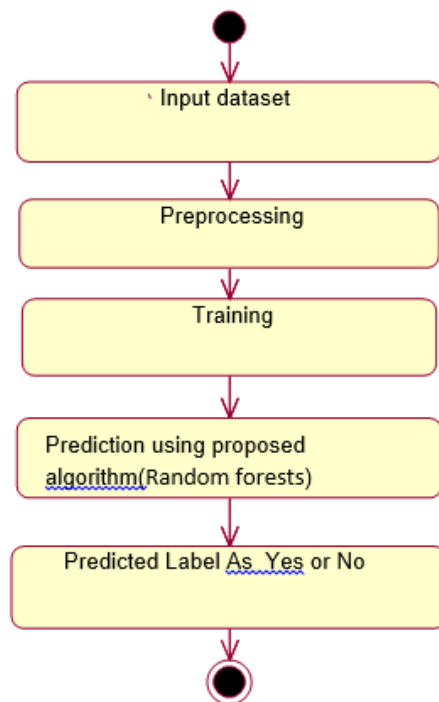
SEQUENCE DIAGRAM:

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.



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.

SYSTEM STUDY**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.

Three key considerations involved in the feasibility analysis are

- ◆ ECONOMICAL FEASIBILITY
- ◆ TECHNICAL FEASIBILITY
- ◆ SOCIAL FEASIBILITY

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.

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.

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. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

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 satisfactory, 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 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.

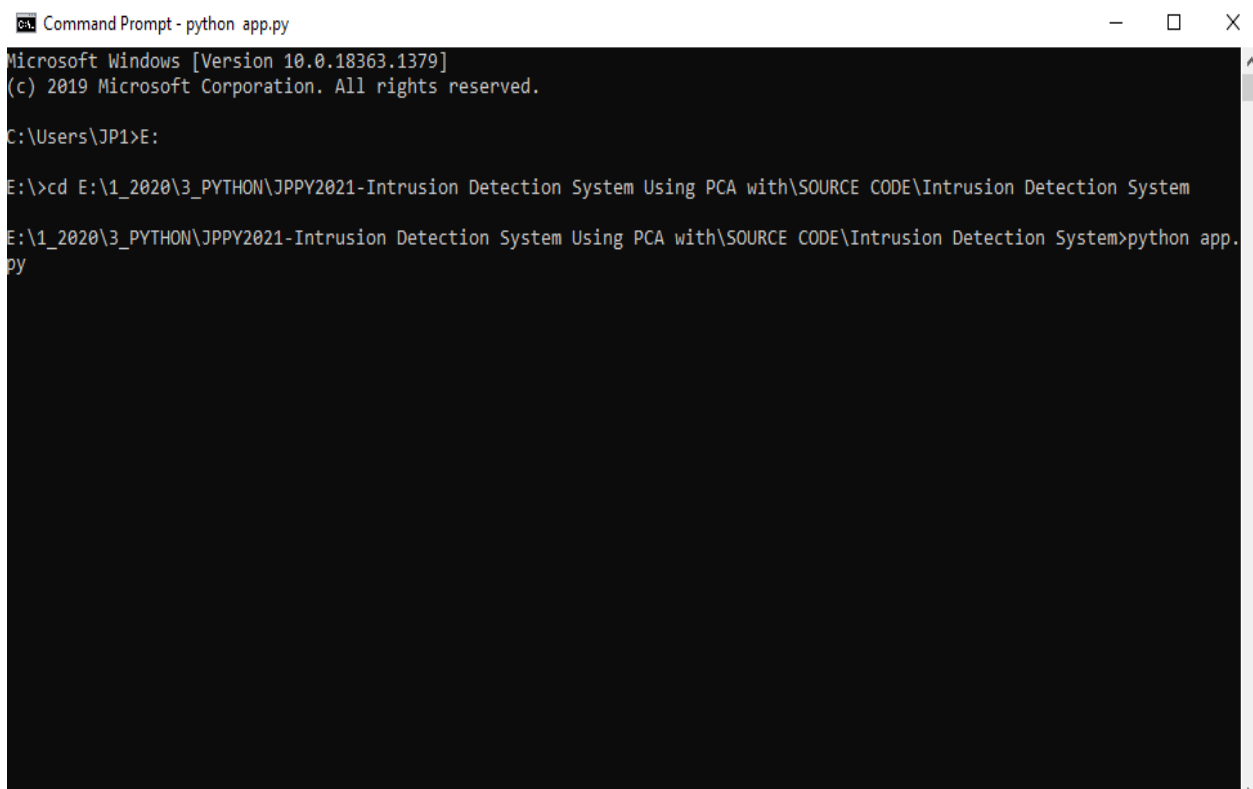
Test Results: All the test cases mentioned above passed successfully. No defects encountered.

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.

SCREENSHOT



```

Command Prompt - python app.py
Microsoft Windows [Version 10.0.18363.1379]
(c) 2019 Microsoft Corporation. All rights reserved.

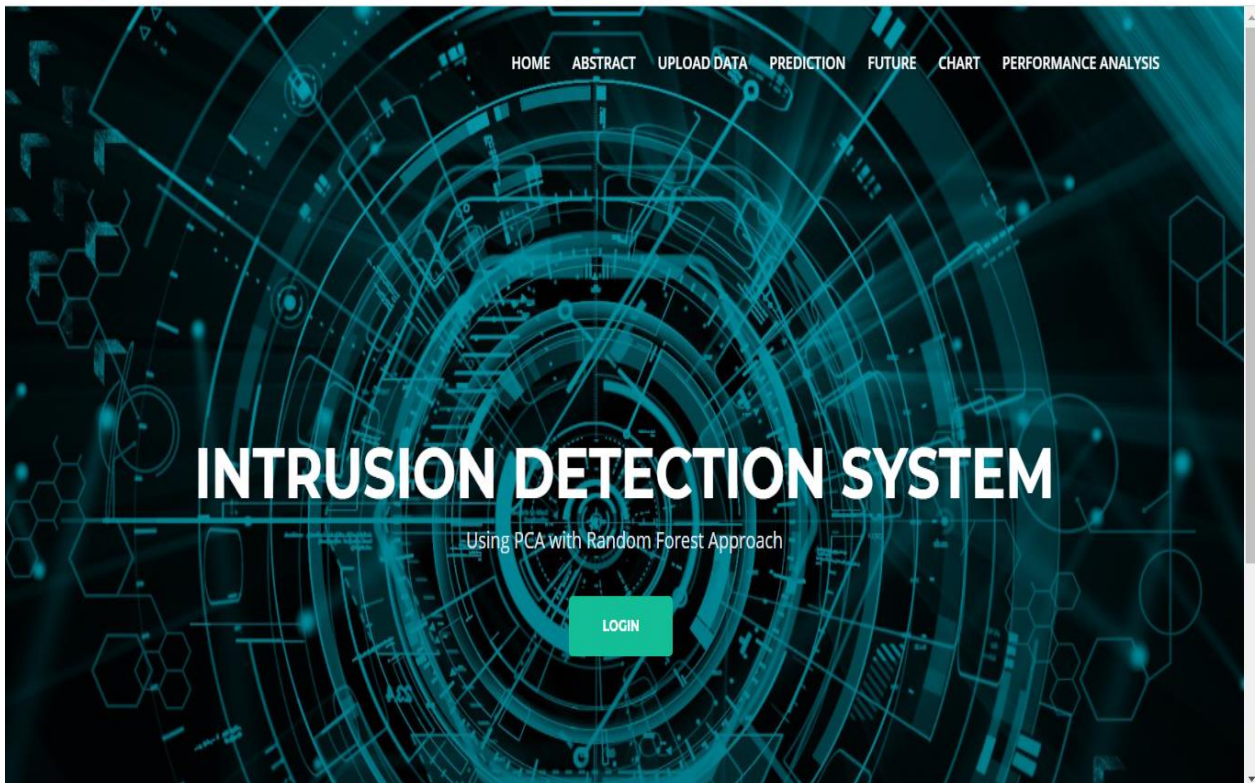
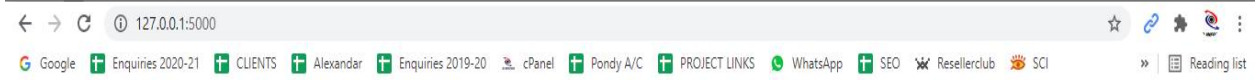
C:\Users\JP1>E:

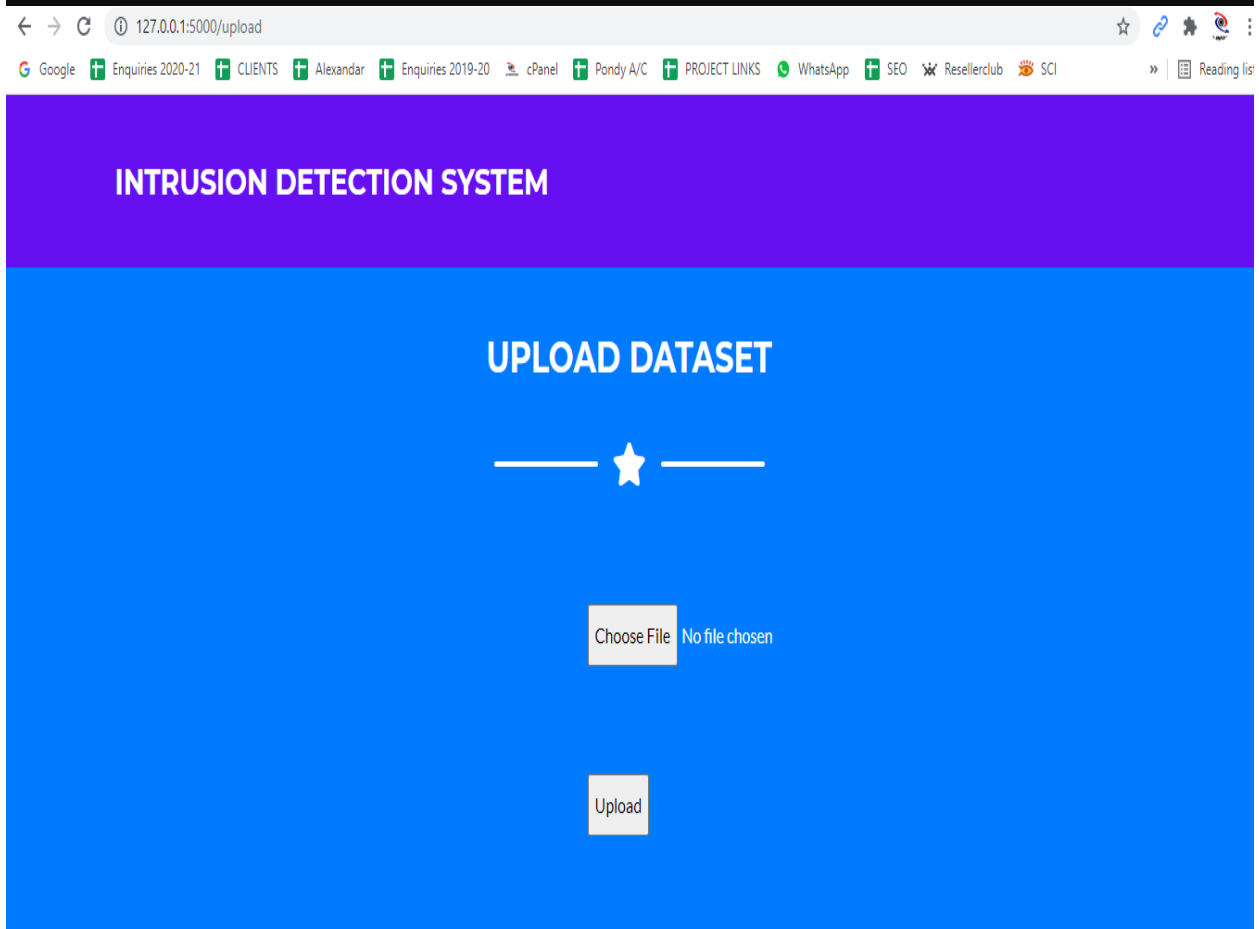
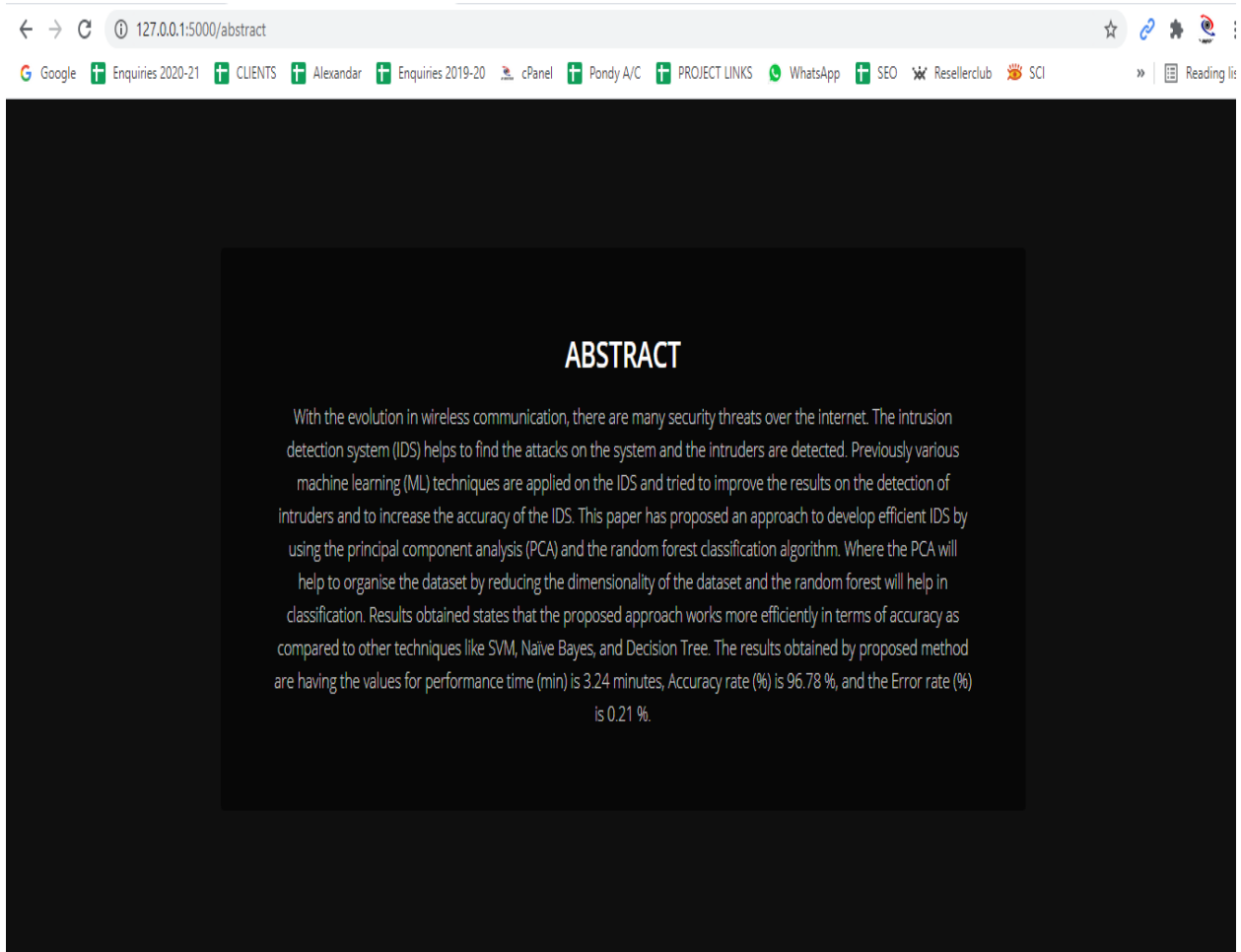
E:\>cd E:\1_2020\3_PYTHON\JPPY2021-Intrusion Detection System Using PCA with\SOURCE CODE\Intrusion Detection System
E:\1_2020\3_PYTHON\JPPY2021-Intrusion Detection System Using PCA with\SOURCE CODE\Intrusion Detection System>python app.py
  
```

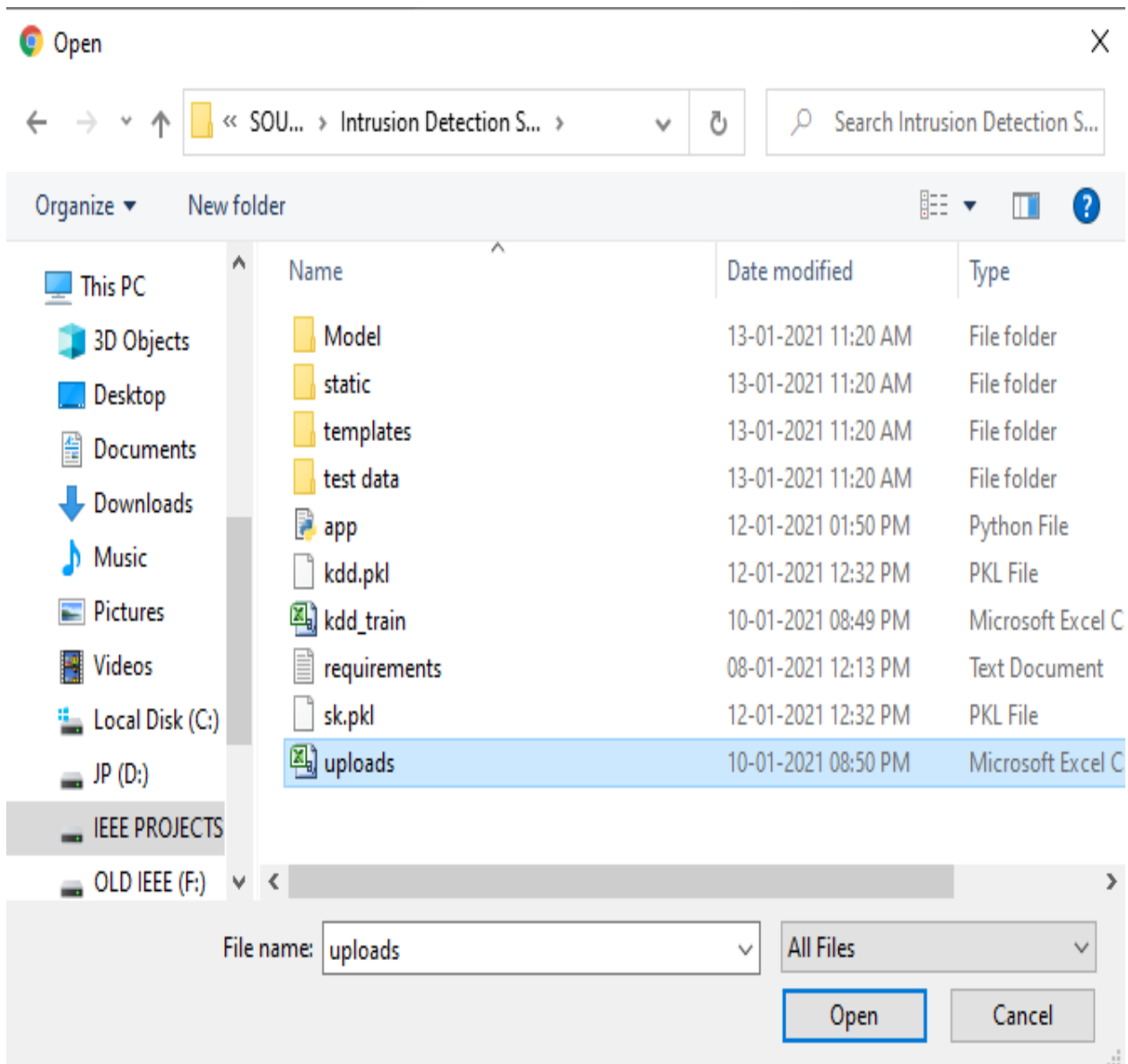
```

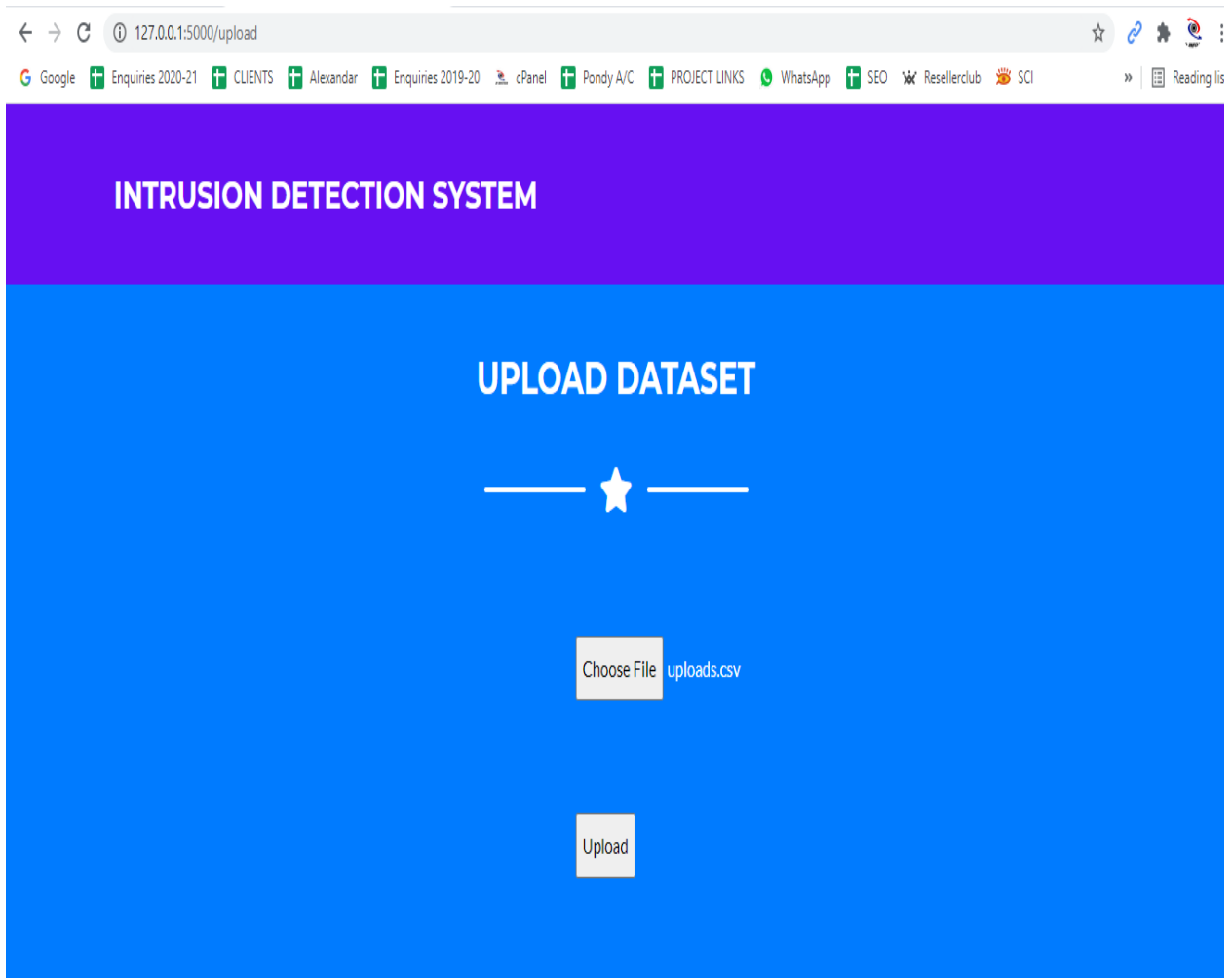
Select Command Prompt - python app.py
C:\Users\JP1\AppData\Roaming\Python\Python38\site-packages\sklearn\base.py:313: UserWarning: Trying to unpickle estimator RandomForestClassifier from version 0.23.2 when using version 0.22.2. This might lead to breaking code or invalid results. Use at your own risk.
  warnings.warn(
C:\Users\JP1\AppData\Roaming\Python\Python38\site-packages\sklearn\base.py:313: UserWarning: Trying to unpickle estimator PCA from version 0.23.2 when using version 0.22.2. This might lead to breaking code or invalid results. Use at your own risk.
  warnings.warn(
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
C:\Users\JP1\AppData\Roaming\Python\Python38\site-packages\sklearn\base.py:313: UserWarning: Trying to unpickle estimator DecisionTreeClassifier from version 0.23.2 when using version 0.22.2. This might lead to breaking code or invalid results. Use at your own risk.
  warnings.warn(
C:\Users\JP1\AppData\Roaming\Python\Python38\site-packages\sklearn\base.py:313: UserWarning: Trying to unpickle estimator RandomForestClassifier from version 0.23.2 when using version 0.22.2. This might lead to breaking code or invalid results. Use at your own risk.
  warnings.warn(
C:\Users\JP1\AppData\Roaming\Python\Python38\site-packages\sklearn\base.py:313: UserWarning: Trying to unpickle estimator PCA from version 0.23.2 when using version 0.22.2. This might lead to breaking code or invalid results. Use at your own risk.
  warnings.warn(
* Debugger is active!
* Debugger PIN: 224-664-831
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)

```









localhost:5000/preview

Google Enquiries 2020-21 CLIENTS Alexandar Enquiries 2019-20 cPanel Pondy A/C PROJECT LINKS WhatsApp SEO Resellerclub SCI Reading list

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

localhost:5000 says
Training finished!
OK

1	0	udp	domain_u	SF	46	82	0	0	0	0	0	0	0
1	1985	udp	other	SF	147	105	0	0	0	0	0	0	0

Click to Train | Test

← → ↻ localhost:5000/home

Google Enquiries 2020-21 CUJENTS Alexandar Enquiries 2019-20 cPanel Pandy A/C PROJECT LINKS WhatsApp SEO Resellerclub SCI » Reading list

Intrusion Detection System

Duration:

protocol_type:

src_bytes:

dst_bytes:

is_host_login:

is_guest_login:

diff_srv_rate:

srv_diff_host_rate:

flag:

REFERENCES:

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 Detect ion System Using Machine Learning Algorithms."
10. Kazi Abu Taher, Billal Mohammed Yasin Jisan, Md. Mahbubur Rahma, 2019 International Conference on Robot ics, Electrical and Signal Processing Techniques (ICREST)"Network Intrusion Detect ion 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 Selection for Effective Intrusion Detection.”