**A Project Report**

**On**

**A DETAILED INVESTIGATION AND ANALYSIS OF USING MACHINE LEARNING TECHNIQUES FOR INTRUSION DETECTION**

*Submitted to*

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR, ANANTHAPURAMU**

*In Partial Fulfilment of the Requirements for the Award of the Degree of*

**BACHELOR OF TECHNOLOGY**

**In**

**Information Technology**

**Submitted By**

**K. Siva Sai Chetan - (17691A1247)**

**D. Pavan Kumar Reddy - (17691A1228)**

**G. Sai Sumith - (17691A1241)**

**Under the Guidance of**

**Mr. S. Thangam**

**Assistant Professor**

**Department of Computer Science & Technology**



**MADANAPALLE INSTITUTE OF TECHNOLGY & SCIENCE**

**(UGC – AUTONOMOUS)**

**(Affiliated to JNTUA, Ananthapuramu)**

**Accredited by NBA, Approved by AICTE, New Delhi)**

**AN ISO 9001:2008 Certified Institution**

**P. B. No: 14, Angallu, Madanapalle – 517325**

**2017-2021**

**DEPARTMENT OF COMPUTER SCIENCE & TECHNOLOGY**

**BONAFIDE CERTIFICATE**

# This is to certify that the project work entitled “A DETAILED INVESTIGATION AND ANALYSIS OF USING MACHINE LEARNING TECHNIQUES FOR INTRUSION DETECTION” is a bonafide work carried out by

**K. SIVA SAI CHETAN - (17691A1247)**

**D. PAVAN KUMAR - (17691A1228)**

**G. SAI SUMITH - (17691A1241)**

Submitted in partial fulfilment of the requirements for the award of degree **Bachelor of Technology** in the stream of **Information Technology** in **Madanapalle Institute of Technology and Science,** **Madanapalle,** affiliated to **Jawaharlal Nehru Technological University Anantapur, Ananthapuramu** during the academic year 2020-2021

Guide Head of the Department

**Mr. S. Thangam, M.E. Dr.** **M. Sreedevi, Ph.D**

**Assistant Professor, Professor and Head,**

**Department of CST Department of CST**

Submitted for the University examination held on:

**Internal Examiner External Examiner**

**Date:**  **Date:**

**ACKNOWLEDGEMENT**

We sincerely thank the **MANAGEMENT** of **Madanapalle Institute of Technology and Science** for providing excellent infrastructure and lab facilities that helped us to complete this project.

We sincerely thank **Dr. C. Yuvaraj, M.E., Ph.D., Principal** for guiding and providing facilities for the successful completion of our project at **Madanapalle Institute of Technology and Science,** Madanapalle.

We express our deep sense of gratitude to **Dr. M Sreedevi, Ph.D, Professor and Head of the Department of CST** for her valuable guidance and constant encouragement given to us during this work.

We express our deep gratitude to our guide **Mr. S. Thangam, M.E.,Assistant Professor, Department of CST** for his guidance and encouragement that helped us to complete this project.

We express our deep sense gratitude to **Dr. Muralidhar Pantula Ph.D,Project Coordinator** for their guidance and encouragement that helped us to complete this project.

We also wish to place on record my gratefulness to other **Faculty of CST Department** and to our friends and our parents for their help and cooperation during our project work.

**DECLARATION**

We hereby declare that the results embodied in this project **“ A DETAILED INVESTIGATION AND ANALYSIS OF USING MACHINE LEARNING TECHNIQUES FOR INTRUSION DETECTION”** by us under the guidance of **Mr. S. Thangam,M.E., Assistant Professor, Dept. of CST** in partial fulfilment of the award of **Bachelor of Technology** in **Information Technology** from **Jawaharlal Nehru Technological University Anantapur, Ananthapuramu** and we have not submitted the same to any other University/institute for award of any other degree.

**Date :**

**Place :**

**PROJECTASSOCIATES**

**K. Siva Sai Chetan**

**D. Pavan Kumar Reddy**

**G. Sai Sumith**

I certify that above statement made by the students is correct to the best of my knowledge.

**Date : Guide**

**CONTENTS**

**SNO DESCRIPTION PAGENO**

**1 INTRODUCTION 1**

**2 LITERATURE SURVEY 1**

**3 SYSTEM ANALYSIS 5**

3.1 EXISTING SYSTEM 5

3.2 DISADVANTAGES 5

3.3 PROPOSED SYSTEM 5

3.4 ADVANTAGES 5 3.5 MODULES 5

**4 SYSTEM STUDY 8**

4.1 FEASIBILITY STUDY 8

**5 SYSTEM SPECIFICATIONS 9**

5.1 BASIC HARDWARE REQUIREMENTS 9

5.2 SOFTWARE REQUIREMENTS 9

**6 SYSTEM DESIGN 10**

6.1 INPUT DESIGN 10

6.2 OUTPUT DESIGN 11

6.3 ARCHITECTURE DIAGRAM 12

6.4 FLOW DIAGRAM 13

6.5 USECASE DIAGRAM 14

6.6 SEQUENCE DIAGRAM 15

6.7 ER DIAGRAM 16

**7 SYSTEM IMPLEMENTATION 17**

7.1 USER TRAINING 17

7.2 TRAINING ON THE APPLICATION

SOFTWARE 17

7.3 OPERATIONAL DOCUMENTATION 18

7.4 SYSTEM MAINTENANCE 18

7.5 CORRECTIVE MAINTENANCE 18

7.6 ADAPTIVE MAINTENANCE 18

7.7 PERSPECTIVE MAINTENANCE 19

7.8 PREVENTIVE MAINTENANCE 19

**8 SYSTEM ENVIRONMENT 20**

8.1 PYTHON 20

8.2 FEATURES OF PYTHON 20

**9 SAMPLE CODE 23**

**10 SYSTEM TESTING 27**

10.1 TESTING OF PRODUCT 27

10.2 UNIT TESTING 27

10.3 INTEGRATION TESTING 27

10.4 WHITE BOX TESTING 28

10.5 BLACK BOX TESTING 28

10.6 VALIDATION TESTING 28

10.7 USER ACCEPTANCE TESTING 29

10.8 OUTPUT TESTING 29

**11 SCREEN SHOTS** **30**

**12 FUTURE SCOPE 39**

**13 CONCLUSION 40**

**14** **REFERENCES 40**

**LIST OF FIGURES**

**FIGURE NO FIGURE DESCRIPTION PAGE NO**

FIGURE 6.3 ARCHITECTURE DIAGRAM 12

FIGURE 6.4 FLOW DIAGRAM 13

FIGURE 6.5 USECASE DIAGRAM 14

FIGURE 6.6 SEQUENCE DIAGRAM 15

FIGURE 6.7 ER DIAGRAM 16

**LIST OF SCREENSHOTS**

**SNO DESCRIPTION PAGE NO**

1. DATASET 30
2. MISSING DATA REMOVED 31
3. X-TEST 32
4. X-TRAIN 33
5. CONFUSION MATRIX 34
6. Y-PRED 35
7. Y-TEST 36
8. Y-TRAIN 37
9. RESULT 38

**ABSTRACT**

Intrusion detection is one of the important security problems in today’s cyber world. A significant number of techniques have been developed which are based on machine learning approaches. So, for identifying the intrusion we have designed the machine learning algorithms. By using the algorithm, we find out intrusion and we can identify the attacker’s details also. IDS are mainly two types: Host based, and Network based.

A Host based Intrusion Detection System (HIDS) monitors individual host or device and sends alerts to the user if suspicious activities such as modifying or deleting a system file, unwanted sequence of system calls, unwanted configuration changes are detected.

A Network based Intrusion Detection System (NIDS) is usually placed at network points such as a gateway and routers to check for intrusions in the network traffic. Here we use SVM (Support Vector machine is used for classification. The basics of Support Vector Machines and how it works are best understood with a simple example. Let’s imagine we have two tags: red and blue, and our data has two features: x and y. We want a classifier that, given a pair of (x, y) coordinates, outputs if it’s either red or blue. We plot our already labelled training data on a plane.

1. **INTRODUCTION**

A detailed investigation and analysis of various machine learning techniques have been carried out for finding the cause of problems associated with various machine learning techniques in detecting intrusive activities. Attack classification and mapping of the attack features is provided corresponding to each attack.

Issues which are related to detecting low-frequency attacks using network attack dataset are also discussed and viable methods are suggested for improvement. Machine learning techniques have been analysed and compared in terms of their detection capability for detecting the various category of attacks.

1. **LITERATURE SURVEY**

**2.1 Title**: The Internet of Things - How the Next Evolution of the Internet is Changing Everything.

**Year**: 2011.

**Author**: D. Evans

**Methodology:**

This research describes the methodology and the development process of creating an IoT platform. This paper also presents the architecture and implementation for the IoT platform. The goal of this research is to develop an analytics engine which can gather sensor data from different devices and provide the ability to gain meaningful information from IoT data and act on it using machine learning algorithms.

**Advantage:**

The proposed system is introducing the use of a messaging system to improve the overall system performance as well as provide easy scalability.

**Disadvantage**

Low cost devices are easily able to connect wirelessly to the Internet, from handhelds to coffee machines, also known as Internet of Things (IoT).

**2.2 Title:** The Internet of Things: A survey

**Year:** 2010.

**Author:** L. Atzori, A. Iera, and G. Morabito

**Methodology:**

The object unique addressing and the representation and storing of the exchanged information become the most challenging issues, bringing directly to a third, ‘‘Semantic oriented”, perspective of IoT.

**Advantage:**

People are informed of the scope and the way in which their movements are tracked by the system (taking people informed about possible leaks of their privacy is essential and required by most legislations).

**Disadvantage:**

The user can set the preferences of the proxy. When sensor networks and RFID systems are included in the network, then the proxy operates between them and the services.

**2.3 Title**: Addressing the Class Imbalance Problem in Medical Datasets

**Year**: 2012.

**Author**: M. Mostafizur Rahman and D. N. Davis

**Methodology:**

A balanced dataset is very important for creating a good training set. They aim to optimize the overall accuracy without considering the relative distribution of each class. Typically, real world data are usually imbalanced, and it is one of the main causes for the decrease of generalization in machine learning algorithms.

**Advantage:**

The aim was to reduce the ratio gap between the majority classes with the minority class. The proposed method is found to be useful for such datasets where the class labels are not certain and can also help to overcome the class imbalance problem of clinical datasets and also for other data domains.

**Disadvantage:**

The outcome labels of most of the clinical datasets are not consistent with the underlying data. The conventional over-sampling and under-sampling technique may not always be appropriate for such datasets.

**2.4 Title:** A survey on cloud computing security

**Year:** 2010.

**Author:** R. Kanday

**Methodology:**

This survey paper provides a general overview on Cloud Computing. The topics that are discussed include characteristics, deployment and service models as well drawbacks.

**Advantage:**

The major part of countermeasures focuses on Intrusion Detection Systems. Moving towards Mobile Cloud Computing and Internet of Things, this survey paper gives a general explanation on the applications and potential that comes with the integration of Cloud Computing with any device that has Internet connectivity as well as the challenges that are before it.

**Disadvantage:**

Several security issues and countermeasures are also discussed to show the major issues and obstacles that Cloud Computing faces as it is being implemented further.

**2.5 Title**: Data Mining: Practical Machine Learning Tool and Technique with Java Implementation

**Year**: 2000.

**Author**: Ian H. Witten and Eibe Frank

**Methodology:**

The convergence of computing and communication has produced a society that feeds on information. Yet most of the information is in its raw form: data. If data is characterized as recorded facts, then information is the set of patterns, or expectations, that underlie the data. There is a huge amount of information locked up in databases—information that is potentially important but has not yet been discovered or articulated. Our mission is to bring it forth. The weather data (Tables 1.2 and 1.3) presents a set of days together with a decision for each as to whether to play the game or not.

**Advantage:**

In these cases, the output took the form of decision trees and classification rules, which are basic knowledge representation styles that many machine learning methods used.

**Disadvantage:**

The weather problem is a tiny dataset that we will use repeatedly to illustrate machine learning methods.

1. **SYSTEM ANALYSIS**

**3.1 EXISTING SYSTEM:**

HACKING incidents are increasing day by day as technology rolls out. A large number of hacking incidents are reported by companies each year. The existing system doesn’t effectively classify and predict the attack which is presented in the network

**3.2 DISADVANTAGES:**

* Doesn’t Efficient for handling large volume of data.
* Theoretical Limits
* Incorrect Classification Results.
* Less Prediction Accuracy.

**3.3 PROPOSED SYSTEM:**

The proposed model is introduced to overcome all the disadvantages that arises in the existing system. This system will increase the accuracy of the classification results by classifying the data based on the social network mental disorders and others using random forest classification algorithm. It enhances the performance of the overall classification results.

**3.4 ADVANTAGES:**

* High performance.
* Provide accurate prediction results.
* It avoids sparsity problems.
* Reduces the information Loss and the bias of the inference due to the multiple estimates.

**3.5 MODULES DESCRIPTION:**

* Data Selection and Loading
* Data Preprocessing
* Splitting Dataset into Train and Test Data
* Feature Extraction
* Classification
* Prediction
* Result Generation

**3.5.1 DATA SELECTION AND LOADING:**

* The data selection is the process of selecting the data for detecting the attacks.
* In this project, the KDDCUP dataset is used for detecting attacks.
* The dataset which contains the information about the duration, flag, service, source bytes, destination bytes and class labels.

**3.5.2 DATA PREPROCESSING**

* as variables with a finite set of label values. That most machine learning algorithms require numerical input and output variables. Data pre-processing is the process of removing the unwanted data from the dataset.
* Missing data removal
* Encoding Categorical data
* **Missing data removal**: In this process, the null values such as missing values are removed using imputer library.
* **Encoding Categorical data**: That categorical data is defined That an integer and one hot encoding is used to convert categorical data to integer data.

**3.5.3 SPLITTING DATASET INTO TRAIN AND TEST DATA:**

* Data splitting is the act of partitioning available data into two portions, usually for cross-validator purposes.
* One Portion of the data is used to develop a predictive model and the other to evaluate the model's performance.
* Separating data into training and testing sets is an important part of evaluating data mining models.
* Typically, when you separate a data set into a training set and testing set, most of the data is used for training, and a smaller portion of the data is used for testing.
  + 1. **FEATURE EXTRACTION:**
* Feature scaling. Feature scaling is a method used to standardize the range of independent variables or features of data. In data processing, it is also known as data normalization and is generally performed during the data pre-processing step.
* Feature Scaling or Standardization: It is a step of Data Pre-processing which is applied to independent variables or features of data. It basically helps to normalise the data within a particular range. Sometimes, it also helps in speeding up the calculations in an algorithm.

**3.5.5 CLASSIFICATION:**

Support Vector Machines are based on the concept of decision planes that define decision boundaries. A decision plane is one that separates between a set of objects having different class memberships.

Support Vector Machine (SVM) is primarily a classier method that performs classification tasks by constructing hyper planes in a multidimensional space that separates cases of different class labels. SVM supports both regression

and classification tasks and can handle multiple continuous and categorical variables. For categorical variables a dummy variable is created with case values as either 0 or 1.

* + 1. **PREDICTION:**
* It’s a process of predicting the attacks in the network from the dataset.
* This project will effectively predict the data from dataset by enhancing the performance of the overall prediction results.

**3.5.7 RESULT GENERATION:**

The Final Result will get generated based on the overall classification and prediction. The performance of this proposed approach is evaluated using some measures like,

* True positive
* True negative
* False positive
* False negative
* Accuracy
* Precision
* Recall
* F1-score

1. **SYSTEM STUDY**

**4.1 FEASIBILITY STUDY:**

The feasibility study is carried out to test whether the proposed system is worth being implemented. The proposed system will be selected if it is best enough in meeting the performance requirements.

The feasibility carried out mainly in three sections namely.

**•** Economic Feasibility

• Technical Feasibility

• Behavioural Feasibility

**4.1.1 Economic Feasibility:**

Economic analysis is the most frequently used method for evaluating effectiveness of the proposed system. More commonly known as cost benefit analysis. This procedure determines the benefits and saving that are expected from the system of the proposed system.

**4.1.2 Technical Feasibility:**

This study centre around the system’s department hardware, software and to what extend it can support the proposed system department is having the required hardware and software there is no question of increasing the cost of implementing the proposed system. The criteria, the proposed system is technically feasible, and the proposed system can be developed with the existing facility.

**4.1.3 Behavioural Feasibility:**

People are inherently resistant to change and need sufficient amount of training, which would result in lot of expenditure for the organization. The proposed system can generate reports with day-to-day information immediately at the user’s request, instead of getting a report, which doesn’t contain much detail.

1. **SYSTEM SPECIFICATIONS**

**5.1 BASIC HARDWARE REQUIREMENTS:**

* System : Intel i3
* Hard Disk : 160 GB
* Monitor : 15 VGA colour
* Mouse : Microsoft.
* Keyboard : 110 keys enhanced
* RAM : 2GB

**5.2 SOFTWARE REQUIREMENTS:**

* OS : Windows 10.
* Language : Python.
* IDE : Spyder.
* Data Base : MySQL.

1. **SYSTEM DESIGN**

**6.1 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.

**6.1.1 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.

**6.2 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.

4. 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.

* 1. **ARCHITECTURE DIAGRAM**

An architecture diagram is a graphical representation of a set of concepts, that are part of an architecture, including their principles, elements, and components. Architecture is a coherent set of concepts for a structure. These concepts are often visualized at four levels of abstraction. These are: Conceptual Level, Logical Level, Physical Level, Implementational Level

DATASET

FORMATED DATASET

TRAIN

TEST

CLASSIFICA-TION

PREDICTION

**FIG. 6.3 ARCHITECTURE DIAGRAM**

**6.4 FLOW DIAGRAM:**

When working on continuous improvement projects and attempting to understand current processes on the floor, these charts can be extremely beneficial in the sense that they allow you to break down a process step by step, allocate time to the steps, and ultimately see the big picture behind a process. Flowcharts are better way of communicating the logic of a system to all concerned or involved. With the help of flowchart, problem can be analysed in more effective way therefore reducing cost and wastage of time. Program flowcharts serve as a good program documentation, which is needed for various purposes, making things more efficient.

SELECT DATSET

CLEANING DATSET

SPLIT TRAIN AND TEST

SVM CLASSIFICATION

PREDICTION

**FIG. 6.4 FLOW DIAGRAM**

**6.5 USE CASE DIAGRAM:**

USER

**FIG. 6.5 USECASE DIAGRAM**

**6.6 SEQUENCE DIAGRAM:**

Clean

Split

Classification

Prediction

Select

Select dataset

 Load dataset

Start

Result Generation

**FIG. 6.6 SEQUENCE DIAGRAM**

**6.7 ER DIAGRAM:**

ER Diagram is so popular in the corporate and business world is because it helps create and define a clear relationship among the entities and the attributes involved. It helps define the logical structure of the database better. ER model is very simple because if we know relationship between entities and attributes, then we can easily draw an ER diagram. Advantages of ER diagram are better visual representation, effective communication tool, highly integrated with relational model, easy conversion to any data model.

**DATA SELECTION & LOAD**

**Split Train and Test**

**CLASSIFICATION**

**Result Generation**

**FIG. 6.7 ER DIAGRAM**

**7. SYSTEM IMPLEMENTATION**

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

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

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

**7.1 User Training:**

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

**7.2 Training on the Application Software:**

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

**7.3 Operational Documentation:**

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

**7.4 System Maintenance:**

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

**7.5 Corrective Maintenance:**

The first maintenance activity occurs because it is unreasonable to assume that software testing will uncover all latent errors in a large software system. During the use of any large program, errors will occur and be reported to the developer. The process that includes the diagnosis and correction of one or more errors is called Corrective Maintenance.

**7.6 Adaptive Maintenance:**

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

**7.7 Perceptive Maintenance:**

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

**7.8 Preventive Maintenance:**

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

**8.SYSTEM ENVIRONMENT**

**8.1 Python:**

Python is one of those rare languages which can claim to be both *simple* and powerful. You will find yourself pleasantly surprised to see how easy it is to concentrate on the solution to the problem rather than the syntax and structure of the language you are programming in. The official introduction to Python is Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming.

Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms. I will discuss most of these features in more detail in the next section.

**8.2 Features of Python:**

**8.2.1 Simple:**

Python is a simple and minimalistic language. Reading a good Python program feels almost like reading English, although very strict English! This pseudo-code nature of Python is one of its greatest strengths. It allows you to concentrate on the solution to the problem rather than the language itself.

**8.2.2 Easy to Learn:**

As you will see, Python is extremely easy to get started with. Python has an extraordinarily simple syntax, as already mentioned.

**8.2.3 Free and Open Source:**

Python is an example of a *FLOSS* (Free/Libré and Open Source Software). In simple terms, you can freely distribute copies of this software, read its source code, make changes to it, and use pieces of it in new free programs.

FLOSS is based on the concept of a community which shares knowledge. This is one of the reasons why Python is so good - it has been created and is constantly improved by a community who just want to see a better Python.

**8.2.4 High-level Language:**

When you write programs in Python, you never need to bother about the low-level details such as managing the memory used by your program, etc.

**8.2.5 Portable:**

Due to its open-source nature, Python has been ported to (i.e. changed to make it work on) many platforms. All your Python programs can work on any of these platforms without requiring any changes at all if you are careful enough to avoid any system-dependent features.

You can use Python on GNU/Linux, Windows, FreeBSD, Macintosh, Solaris, OS/2, Amiga, AROS, AS/400, BeOS, OS/390, z/OS, Palm OS, QNX, VMS, Psion, Acorn RISC OS, VxWorks, PlayStation, Sharp Zaurus, Windows CE and PocketPC!

You can even use a platform like [Kivy](http://kivy.org) to create games for your computer *and* for iPhone, iPad, and Android.

**8.2.6 Interpreted:**

This requires a bit of explanation:

A program written in a compiled language like C or C++ is converted from the source language i.e. C or C++ into a language that is spoken by your computer (binary code i.e. 0s and 1s) using a compiler with various flags and options. When you run the program, the linker/loader software copies the program from hard disk to memory and starts running it.

Python, on the other hand, does not need compilation to binary. You just *run* the program directly from the source code. Internally, Python converts the source code into an intermediate form called bytecodes and then translates this into the native language of your computer and then runs it. All this, actually, makes using Python much easier since you don't have to worry about compiling the program, making sure that the proper libraries are linked and loaded, etc. This also makes your Python programs much more portable, since you can just copy your Python program onto another computer and it just works!

**8.2.7 Object Oriented:**

Python supports procedure-oriented programming as well as object-oriented programming. In *procedure-oriented* languages, the program is built around procedures or functions which are nothing but reusable pieces of programs. In *object-oriented* languages, the program is built around objects which combine data and functionality. Python has a very powerful but simplistic way of doing OOP, especially when compared to big languages like C++ or Java.

**8.2.8 Extensible:**

If you need a critical piece of code to run very fast or want to have some piece of algorithm not to be open, you can code that part of your program in C or C++ and then use it from your Python program.

**8.2.9 Embeddable:**

You can embed Python within your C/C++ programs to give *scripting* capabilities for your program's users.

**8.2.10 Extensive Libraries:**

The Python Standard Library is huge indeed. It can help you do various things involving regular expressions, documentation generation, unit testing, threading, databases, web browsers, CGI, FTP, email, XML, XML-RPC, HTML, WAV files, cryptography, GUI (graphical user interfaces), and other system-dependent stuff. Remember, all this is always available wherever Python is installed. This is called the *Batteries Included* philosophy of Python.

Besides the standard library, there are various other high-quality libraries which you can find at the [Python Package Index](http://pypi.python.org/pypi).

**9.SAMPLE CODE**

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

#import dataset

Dataset=pd.read\_csv('files.csv')

print(Dataset.describe())

x=Dataset.iloc[:,:-1].values

x1=pd.DataFrame(x)

y=Dataset.iloc[:,10].values

y1=pd.DataFrame(y)

for i in range(119):

if y[i]=='anom':

y[i]=0

else:

y[i]=1

type(y)

type(x)

y=y.astype('int')

#Encoding categorical data

from sklearn.preprocessing import LabelEncoder,OneHotEncoder

labelencoder\_x=LabelEncoder()

for i in range(10) :

x[:,i]=labelencoder\_x.fit\_transform(x[:,i])

Y=pd.DataFrame(x[:,i])

for i in range(10) :

#onehotencoder=OneHotEncoder(categorical\_features=[i])

#x=onehotencoder.fit\_transform(x).toarray()

onehotencoder = OneHotEncoder()

X = onehotencoder.fit\_transform(x).toarray()

X = X[:, 1:]

from sklearn.preprocessing import LabelEncoder, OneHotEncoder

#write in file

np.savetxt('encode\_valuse.txt',x)

#Missing Data Removal

from sklearn.impute import SimpleImputer

imputer = SimpleImputer(missing\_values='NaN', strategy='most\_frequent')

imputer = imputer.fit(x[:,:])

x[:,:]=imputer.transform(x[:,:])

Missing\_Data\_Removed=imputer.transform(x[:,:])

#write in file

np.savetxt('Missing\_values.txt',Missing\_Data\_Removed)

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size = 0.20, random\_state = 0)

from sklearn.preprocessing import StandardScaler

sc\_X = StandardScaler()

X\_train = sc\_X.fit\_transform(X\_train)

X\_test = sc\_X.transform(X\_test)

#SVM Apply

from sklearn.svm import SVC

svclassifier = SVC(kernel='linear')

svclassifier.fit(X\_train, y\_train)

#prediction

y\_pred = svclassifier.predict(X\_test)

from sklearn.metrics import classification\_report, confusion\_matrix

print(confusion\_matrix(y\_test, y\_pred))

print(classification\_report(y\_test, y\_pred))

dt = pd.DataFrame({'Actual': y\_test, 'Predicted': y\_pred})

#performance anaylsis

import warnings

warnings.filterwarnings('ignore')

from sklearn.metrics import accuracy\_score

score = accuracy\_score(y\_test, y\_pred)

from sklearn import metrics

print("Accuracy:",accuracy\_score(y\_test,y\_pred)\*100)

print("Accuracy:",metrics.accuracy\_score(y\_test, y\_pred))

from sklearn.metrics import confusion\_matrix

cm = confusion\_matrix(y\_test, y\_pred)

TP = cm[0][0]

FP = cm[0][1]

FN = cm[1][0]

TN = cm[1][1]

Total\_TP\_FP=cm[0][0]+cm[0][1]

Total\_FN\_TN=cm[1][0]+cm[1][1]

#True Positive Calculation

TP1=(((cm[0][0])/Total\_TP\_FP)\*100)

#False Positive Calculation

FP1= (((cm[0][1])/Total\_TP\_FP)\*100)

#False Negative Calculation

FN1=(((cm[1][0])/Total\_FN\_TN)\*100)

#True Negative Calculation

TN1=(((cm[1][1])/Total\_FN\_TN)\*100)

#Total TP,TN,FP,FN

Total=TP1+FP1+FN1+TN1

#Accuracy Calculation

accuracy=((TP1+TN1)/Total)\*100

#Error Rate Calculation

error\_rate=((FP1+FN1)/Total)\*100

#Precision Calculation

precision=TP1/(TP1+FP1)\*100

#Recall Calculation

recall=TP1/(TP1+FN1)\*100

#F1 Score

f1=2\*((precision\*recall)/(precision+recall))

print("\n\n\tResult Generation")

print("\t------------------")

print('\n\tTrue positive = ', TP1, '%')

print('\n\tFalse positive = ', FP1, '%')

print('\n\tFalse negative = ', FN1, '%')

print('\n\tTrue negative = ',TN1 , '%')

print('\n\tAccuracy = ',accuracy , '%')

print('\n\tError Rate = ',error\_rate , '%')

print('\n\tPrecision = ',precision , '%')

print('\n\tRecall = ',recall , '%')

print('\n\tF1-Score = ',f1 , '%')

**10.SYSTEM TESTING**

**10.1 TESTING OF PRODUCT:**

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

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

**10.2 UNIT TESTING:**

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

**10.3 INTEGRATION TESTING:**

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

i)Top-down integration testing.

ii)Bottom-up integration testing.

**10.4 WHITE BOX TESTING:**

White Box testing is a test case design method that uses the control structure of the procedural design to drive cases.  Using the white box testing methods, we derived test cases that guarantee that all independent paths within a module have been exercised at least once.

**10.5 BLACK BOX TESTING:**

* Black box testing is done to find incorrect or missing function
* Interface error
* Errors in external database access
* Performance errors
* Initialization and termination errors

In ‘functional testing’, is performed to validate an application conforms to its specifications of correctly performs all its required functions. So this testing is also called ‘black box testing’.  It tests the external behaviour of the system.  Here the engineered product can be tested knowing the specified function that a product has been designed to perform, tests can be conducted to demonstrate that each function is fully operational.

**10.6 VALIDATION TESTING:**

After the culmination of black box testing, software is completed assembly as a package, interfacing errors have been uncovered and corrected and final series of software validation tests begin validation testing can be defined as many, but a single definition is that validation succeeds when the software functions in a manner that can be reasonably expected by the customer.

**10.7 USER ACCEPTANCE TESTING:**

User acceptance of the system is the key factor for the success of the system.  The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system at the time of developing changes whenever required.

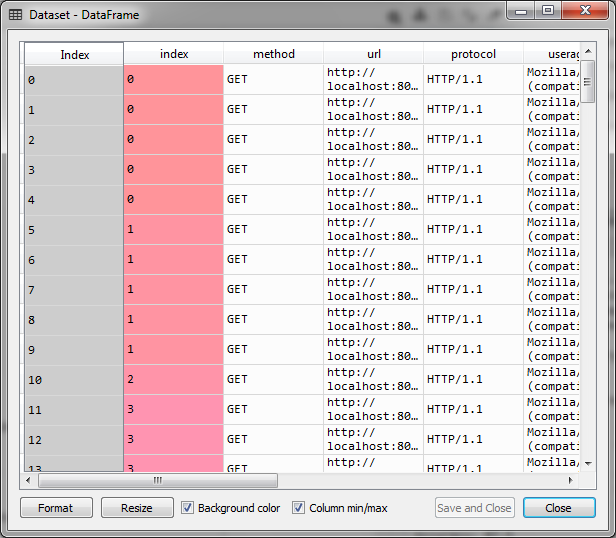
1. **10.8 OUTPUT TESTING:**

After performing the validation testing, the next step is output asking the user about the format required testing of the proposed system, since no system could be useful if it does not produce the required output in the specific format.  The output displayed or generated by the system under consideration.  Here the output format is considered in two ways.  One is screen and the other is printed format.

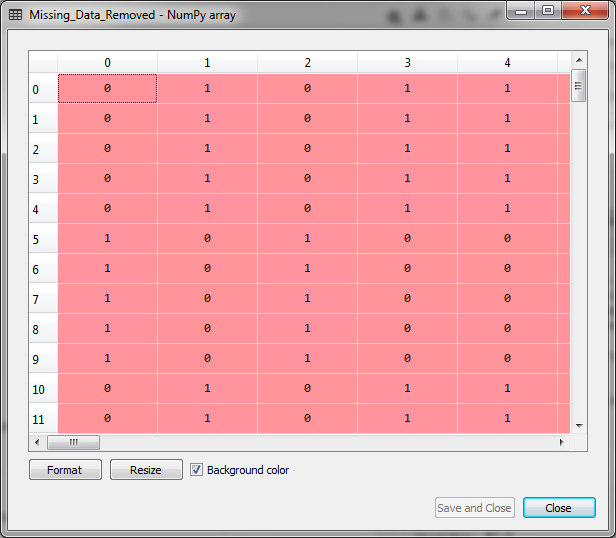
The output format on the screen is found to be correct as the format was designed in the system phase according to the user needs.  For the hard copy also output comes out as the specified requirements by the user. Hence the output testing does not result in any connection in the system.

**11.SCREEN SHOTS**

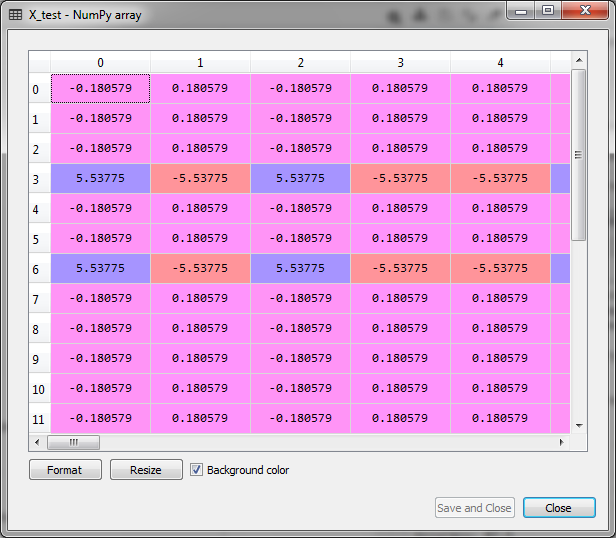
**11.1 Dataset:**



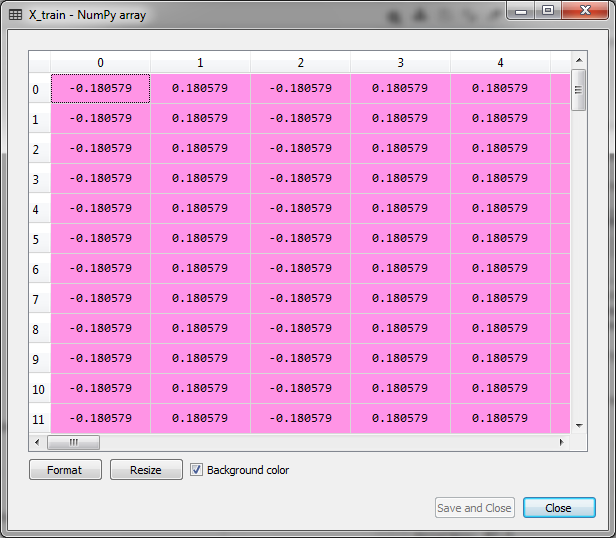
**11.2 Missing Data Removed:**



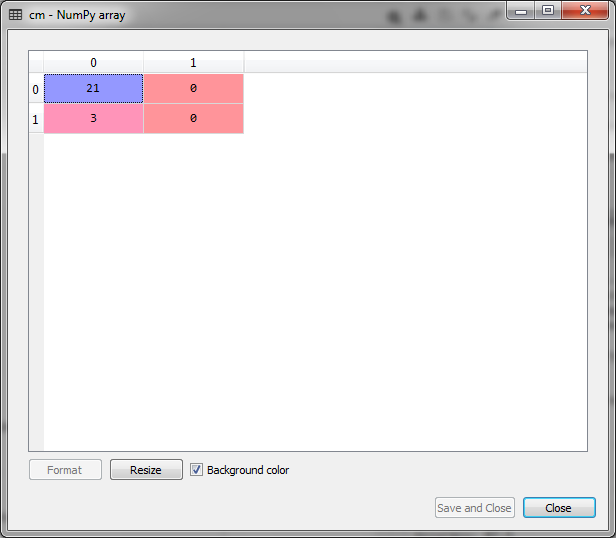
**11.3 X\_test:**



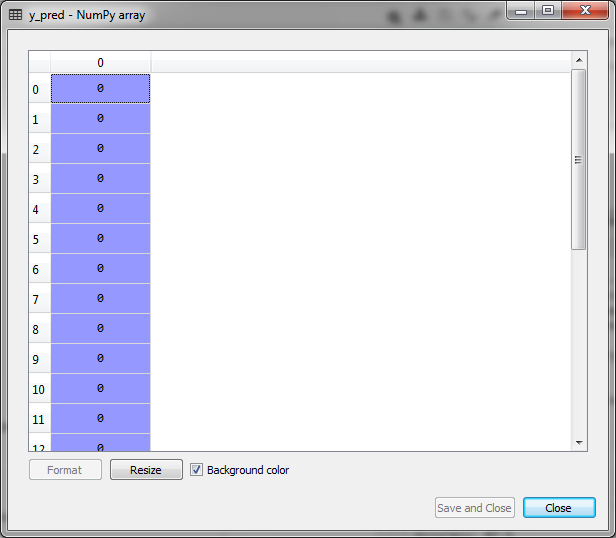
**11.4 X\_train:**



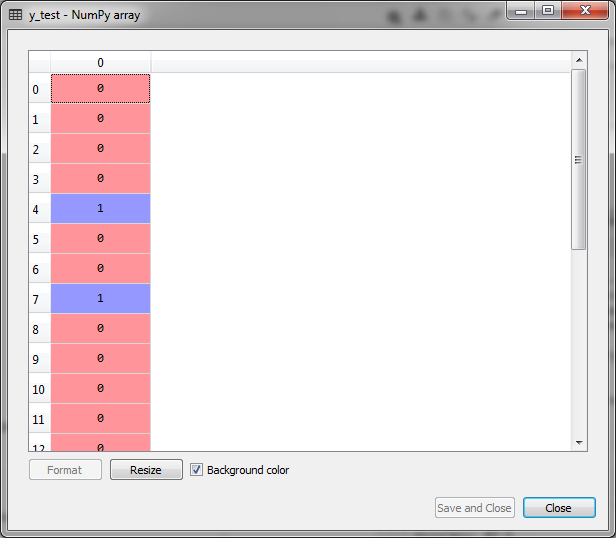
**11.5 Confusion Matrix:**



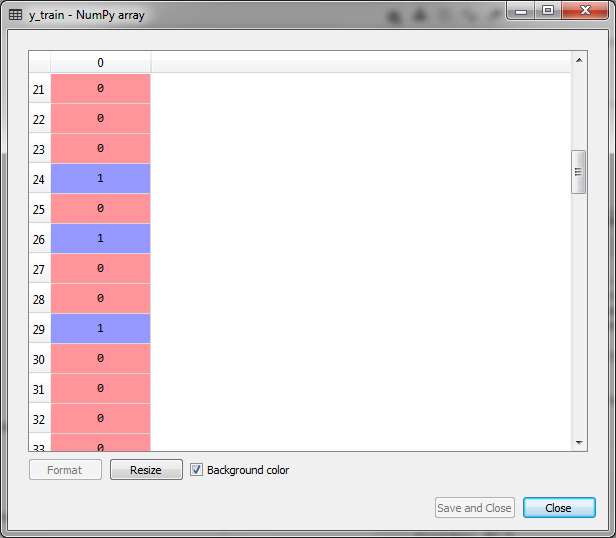
**11.6 Y\_pred:**



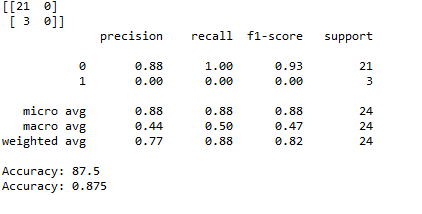
**11.7 Y\_test:**



**11.8 Y\_train:**



**11.9 Result:**



**12.FUTURE SCOPE**

As a refresher, Intrusion Detection System (IDS) identify when someone or something attempts to compromise a system or resource. Detection mechanisms include signature-based methods – comparing a pattern or signature to previous events – and behaviour analysis, which detects anomalous actions. Over the next several years, intrusion detection will evolve in two directions:

**IoT, An expanding attack surface:**

Intrusion detection systems, algorithms and data analysis must take the emerging IoT into the equation. Attackers can breach organizations from multiple points via cameras, automotive or wearable devices. In order to deduce the intruder path, multiple sources of data from all IoT devices in the organization will have to be distilled into a centralized place.

**No more hide and seek:**

It’s time for event detection, Cyber criminals are developing new and innovative attacks that employ evasive and polymorphic techniques to escape detection. These techniques render the old hermetic intrusion detection paradigm useless. Famous for this is anti-forensic malware.

**13. CONCLUSION**

We reviewed several inﬂuential algorithms for intrusion detection based on various machine learning techniques. Characteristics of ML techniques makes it possible to design IDS that have high detection rates and low false positive rates while the system quickly adapts itself to changing malicious behaviours.

We divided these algorithms into two types of ML-based schemes: Artiﬁcial Intelligence (AI) and Computational Intelligence (CI). Although these two categories of algorithms share many similarities, several features of CI-based techniques, such as adaptation, fault tolerance, high computational speed and error resilience in the face of noisy information, conform the requirement of building eﬃcient intrusion detection systems.

**14. REFERENCES**

1. S. Yinbiao and K. Lee, “Internet of Things: Wireless Sensor Networks Executive summary,” 2014. [5] F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci
2. “A survey on sensor networks,” IEEE Commun. Mag., vol. 40, no. 8, pp. 102–105, 2002.
3. X. Chen, K. Makki, K. Yen, and N. Pissinou, “Sensor network security: a survey,” IEEE Commun. Surv. Tutorials, vol. 11, no. 2, pp. 52–73, 2009.
4. A.-S. K. Pathan, H.-W. Lee, and C. S. Hong, “Security in wireless sensor networks: issues and challenges,” 2006 8th Int. Conf. Adv. Commun. Technol., vol. 2, p. 6 pp.-pp.1048, 2006.
5. P. Yi, Y. Jiang, Y. Zhong, and S. Zhang, “Distributed Intrusion Detection for Mobile Ad Hoc Networks,” 2005 Symp. Appl. Internet Work. (SAINT 2005 Work., pp. 94–97, 2005.
6. H. Sedjelmaci and M. Feham, “Novel Hybrid Intrusion Detection System for Clustered Wireless Sensor Network,” Int. J. Netw. Secur. Its Appl. (IJNSA), Vol.3, No.4, July 2011, vol. 3, no. 4, pp. 1–14, 2011.
7. L. Khan, M. Awad, and B. Thuraisingham, “A new intrusion detection system using support vector machines and hierarchical clustering,” VLDB J., vol. 16, no. 4, pp. 507–521, 2007.
8. S. K. Sahu, S. Sarangi, and S. K. Jena, “A detail analysis on intrusion detection datasets,” Souvenir 2014 IEEE Int. Adv. Comput. Conf. IACC 2014, pp. 1348–1353, 2014.