### A PROJECT REPORT ON

### CYBER SPHERE

**“Intrusion Detection and Recommendation System”**

**Abstract**

From the advent of computer networks to the diversification of networks worldwide,

Cyber-Threat has become one of the major concerns of humanity. This Pandemic has only increased the threats of ransomware, data hijack, or Intrusion into our networks. We attempted to make an Intrusion Detection system which can give accurate detections and relevant recommendations as well as be accessible to all in the form of a website. Our study of Machine Learning in the field of Intrusion Detection helped us in building a Random Forest Classifier which can classify with near perfect accuracy. We focus upon securing networks of small to medium sized business and recommending them the best steps they can undertake to secure their systems.

**INDEX**

1. **Introduction 1**
   1. Overview . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .2
      1. Objectives……………………………………………………...2
      2. Scope…………………………………………………………..3
      3. Motivation……………………………………………………..3
      4. Application…………………………………………………….3
2. **Literature Survey 4**
3. **Project Plan 7**
   1. Project Schedule . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . 8
      1. Project task set………………………………………………….8
      2. Timeline chart…………………………………………………..8
      3. Project Architecture…………………………………………….9
      4. Sequence Diagram……………………………………………...10
4. **System Requirement and Specifications 11**
   1. Scope . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . 12
   2. Functional Requirements . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12
      1. Login………………………………………………………...…..12
      2. Data Input …………………………………………………...…..13
      3. Output Results..……………………………………………...…..14
      4. Recommendation System..…………………………………..…..14
      5. Contacts…………………………………………………...……..15
   3. Non-Functional Requirements. . . . . . . . . . . . . . . . . … . . . . . . . . . . .. . . 15
      1. Security…………………………………………………………..15
      2. Reliability.………………………………………………………..16
   4. Constraints . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 16
      1. Software Requirements…………………………………………..16
      2. Hardware Requirements………………………………………….16
      3. Design Constraints……………………………………………….17
   5. Usage Scenario. .. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 17
      1. User Profiles.…………………………………………………….17
      2. System…………………………………………………………….18
5. **Detailed Design Document 19**
   1. Background. . . . . . . . . . . . . . . . . . . . . . . .. .. . . . . . . . . . . . . . . . . . . . . . . 20
   2. Theory. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20
      1. Detection Techniques……………………………………………..20
      2. Intrusion Detection System……………………………………….20
      3. Random Forest Classification…………………………………….22
6. **Approach 24**
   1. Training the Algorithm . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .25
   2. Uploading the Network Traffic File . . . . . . . . . . . . . . . . . . . . . . . . . . . . .26
   3. Prediction Using Random Forest Classifier. . . . . . . . . . . . . . . . . . . . . . . .27
   4. Displaying Results and Recommendations………. . . . . . . . . . . . . . . . . . .28
   5. History and its Functions.. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .28
7. **Prototype 30**
8. **Conclusion 33**
   1. Conclusion .. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . . . . . . . .34
   2. Future Scope . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . . . . . . . . . . . 35
9. **References 36**

**LIST OF FIGURES**

3.1: Timeline chart……………………………………………………………………8

3.2: Architecture of project………………………………………………………… ..9

3.3: Sequence Diagram……………………………………………………………….10

4.1: Use Case Diagram……………………………………………………………….18

5.1: Random Forest…………………………………………………………………...22

5.2: Prediction using Random Forest…………………………………………………23

6.1: The KDD’99 Data Distribution…………………………………………………..25

6.2: Training and Testing our Algorithm……………………………………………..26

6.3: Uploading file to the server……………………………………………………………27

6.4: Displaying the Results………………………………………………………………….28

6.5: History……………………………………………………………………………………29

7.1: Our Website…………………………………………….………………………………..31

7.2: User Login………………………………………………………………………………..31

7.3: Uploading the file………………………………………………………………………..32

7.4: Getting Results and Recommendations……………………………………………32

**CHAPTER 1**

**INTRODUCTION**

### OVERVIEW 2

### To distinguish the activities of the network traffic between intrusion and

### normal is very difficult and much time consuming. Therefore, we need a

### tool that can detect network intrusion and give us relevant preventive and

### security measures. Many organizations, especially small and midsize

### businesses, do not have the knowledge, time, or resources to detect and

### respond to cyberthreats. And hackers know this. Smaller businesses are

### the low-hanging fruit. They have weaker security systems and smaller IT

### teams than the larger businesses do. Therefore, they are easier to infiltrate.

### Plus, they are usually connected to a larger organization.

### Objectives

1. A Web Application that uses Machine Learning Techniques to

correctly detect possible intrusion on the network and recommend

solutions based on the detection.

1. User will upload a Network Traffic or TCP dump file in .csv format.
2. The trained model will classify the potential threat into a specific category,

search for the best possible solutions and display the category and solutions

as part of the recommendation option.

1. The main categories include:

* DOS: denial-of-service, e.g., syn flood.
* R2L: unauthorized access from a remote machine, e.g., guessing

password.

* U2R:  unauthorized access to local superuser (root) privileges, e.g.,

various ``buffer overflow'' attacks.

* Probing: surveillance and other probing, e.g., port scanning.

1. The Application will act as a link between the users and the Professionals

who can be contacted once a threat is identified.

### Scope 3

A complete end to end website which will give accurate detection of the attacks and then recommend steps to be undertaken. There will also be a history option that can be used by our clients to analyze their networks at different times. The Model will try to connect the clients with the correct people which can help them securing their networks. Further to extend our project we will try to find the source and addresses of attackers from the data using some of the Machine Learning Techniques.

### Motivation

These IDS (Intrusion Detection System) usually have high cost if they are providing

fool proof and accurate results. There is software which are free of cost if they must

be used in a single PC. But for a big network a better solution is required. Small

businesses do not pose such high resources and time; hence they require an easy

to use solution that is accurate and efficient.

### Applications

* + - 1. Our Project will be useful to mid-range businesses. Making their networks secure and their system vulnerability free.
      2. Tedious task and cost of installing Intrusion detection systems will not be

required.

* + - 1. With Further advancements, the model will be able to continuously scan the

Network traffic and can be used as an antivirus on our PCs.

**CHAPTER 2**

**LITERATURE SURVEY**

1. **A New Hybrid Machine Learning for Cyber security Threat Detection** 5

**based on Adaptive Boosting.**

This paper provides New hybrid classifier is the combination of k-NN,

C4.5, MLP, SVM and LDA based on adaptive boosting. Correlation-

based feature selection method is applied to all datasets to

reduce redundant features. This algorithm is very efficient in detecting

tor and non-tor intrusions. High efficiency in classifying types of tor

intrusions Multiclass classifier.

1. **OPEM: A Hybrid Static-Dynamic Approach for Machine-Learning based**

**Malware Detection.**

This paper gives an algorithm which uses a combination of Static and Dynamic

Approaches In malware detection. This algorithm dramatically increases

efficiency of malware detection improved over static or dynamic approaches

when run separately.

1. **Anomaly detection of IOT Cyber Attacks in mart city using Machine learning**

This paper gives us an idea of using the Random Forest Machine Learning

Algorithm. It has the following advantages: Intelligent system to detect IOT

Cyberattacks or unusual activities in IOT network traffics from the fog networks

distributed over the smart city. The only requirement is it needs large amount of

data for efficient functioning.

1. **Unsupervised Machine Learning-based Detection of Covert Data integrity**

**Assault (CDIA) in Smart Grid Networks (SGs) utilizing Isolation Forest (iForest).**

This Paper uses Unsupervised Learning-Based Isolation Forest Algorithm. Forest

based Detection scheme detects the anomaly/outlier which usually pass through

BDD (Bad Data Detector) of the SMART GRID NETWORK (SG) . The Low

Computational complexity of the Forest enables the identification of outliers or

anomalies in a short time.

6

1. **Using Imbalanced Triangle Synthetic Data for Machine Learning**

**Anomaly Detection.**

This paper does calculations on the basis of Imbalanced Triangle

Synthetic Data Method. It synthesizes new outlier data by using imbalanced

triangle method. It produces more Robust datasets overcoming the

limitation of (SMOTE) techniques of working on a line segment.

1. **Anomaly Detection with Generative Adversarial Networks for**

**Multivariate Time Series.**

In this paper the author uses the GAN based anomaly detector is being used in

this technique. It consists of one Generator(G) and one Discriminator(D).

Long Short Term-Recurrent Neural Networks (LSTM-RNN) are used to

make these. This technique uses Multivariate analysis using unsupervised

Machine Learning. Accuracy and precision is greatly increased when multiple

variables are used. It uses PCA based Mapping to reduce huge dimensions

of data into smaller matrices which helps in basic training.

1. **Scalable Cyber-security analytics with a new summary-based**

**Approximate Query Engine**

This paper uses a new summary-based approach is used to speed up the

process of decision making. It provides quick insights to the specialist who

can further refine the discovery of anomaly. It provides quick query responses

and speeds up the process of detection. There is a standoff between

accuracy and decision time.

1. **Evaluating Shallow and Deep Neural Networks for Network Intrusion**

**Detection Systems in Cyber Security.**

This Paper uses of DNNs (Deep Neural Networks) has been shown. Neural network

with 41 node input layers and 4 hidden layers were used. Relu Classifier is used

instead of normal Sigmoid function to improve the accuracy. Through Statistical

analysis it was shown that DNNs outperform every other ML Algorithm.

**CHAPTER 3**

**PROJECT PLAN**

### PROJECT SCHEDULE 8

* + 1. **Project task set**

Major Tasks in the Project stages are:

* Task 1: Training the Random Forest Classifier
* Task 2: Input the file
* Task 3: Reading and processing the file
* Task 4: Output Accurate Results
* Task 5: Recommending steps based on Results
* Task 6: Show and Update History
* Task 7: Integration

### Timeline Chart

The timeline chart is made by grouping the project tasks into three groups.

Phase 1 Phase 2 Phase 3

Building Databases for results, recommendations, and history.

Writing functions codes for use-case.

Combine backend

With front-end code via Django

Integrating everything on the website.

Training, testing, and saving the Random Forest Classifier.

Saving test data in the form of attack scenarios.

Figure 3.1: Timeline chart

### PROJECT ARCHITECTURE 9

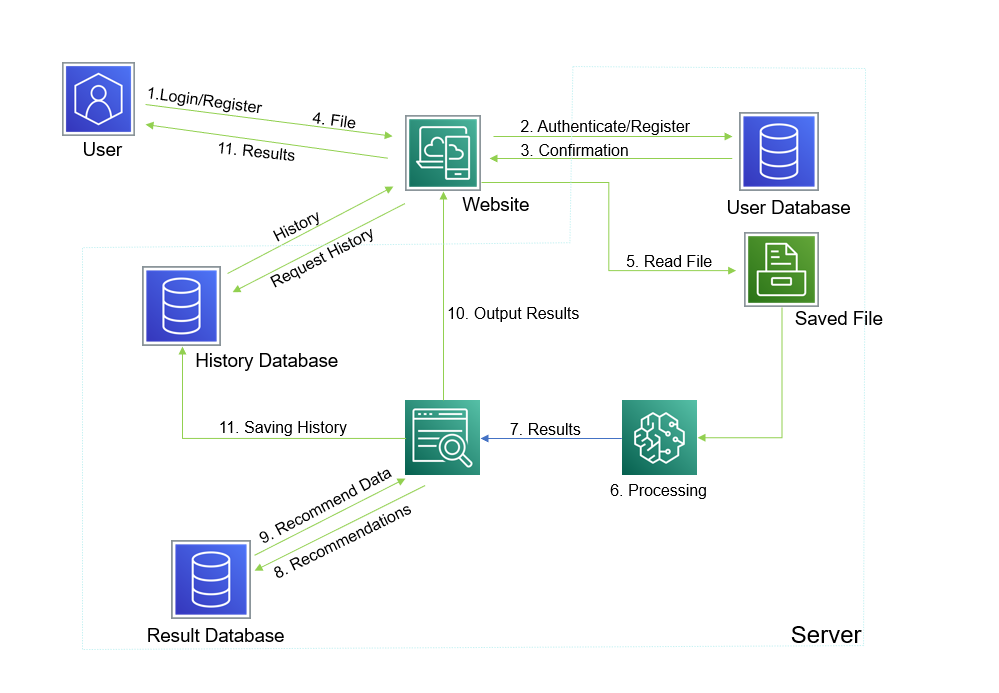
This is the architecture of project which gives details about all various operations involved in the Intrusion detection. 

Figure 3.2: Architecture of project

### SEQUENCE DIAGRAM 10

### 

### Diagram, engineering drawing Description automatically generated

Figure 3.3: Sequence Diagram

**CHAPTER 4**

**SYSTEM REQUIREMENT SPECIFICATIONS**

4.1. **SCOPE:** 12

1. The Application inputs the network data that is accessible to the user in a particular format, Processes data and verifies the security of the Network.
2. Using the Machine Learning approach any potential new threat that might be hidden in the data can be identified. The model uses Pattern based detection contrary to Signature based detection used by modern Anti-Viruses which gives it an advantage of detecting not known or new type of attacks.
3. The Objective of this project is to accurately identify an intrusion however small or diverse it may be. Big institutions depend on these networks as their backbone, characterized by data flow in bulk, these networks however secure need to be continuously tested for new attacks. Applications such as ours can help detect these attacks and even provide a possible solution to deal with the problem.
4. The final goals of the project are to build an application that is accurate, reliable, and scalable in detecting intrusions on the Network, to help cater the modern security demands of commercial firms.

4.2. **FUNCTIONAL REQUIREMENTS**

4.2.1. **Login**

1. Description

An Interface that will have options to Login or Sign Up as a new user.

1. Input
2. Login:

User ID known to the user in order to enter the application.

Account Password: A Strong password known to user given at the time of Sign Up.

1. Sign Up:

Registration details of the New User.

Creation of a new User ID and Password.

13

1. Output
2. Login:

Successfully login the user and enter the website.

1. Sign Up:

Upon successful registration redirect to the login page.

Enter the User ID and password to enter.

4.2.2 **Data Input**

1. Description

An interface that will allow the user to upload the data to be tested on the website. The data should be in .csv format, if not it should be first converted. The necessary fields to be included will be displayed on the screen.

1. Input

Upload file option that will allow user to upload the dataset from his device onto the website.

1. Output

Upon successful loading of the dataset the “Uploaded Successfully” message will be displayed, and the data will be processed for analysis of the security of Network.

The Results will be displayed under the “RESULTS” tab, the processing might take some time during which the tab will show “PROCESSING”.

14

4.2.3. **Output Results**

1. Description

It will be a read-only interface that will display details of the processed data.

1. Output

The Detected output will be displayed in decreasing order of magnitude attacks.

A Button named “COMPROMISED” will turned red if some anomalies found in the dataset. The display window will categorize the different detected attacks and how serious they are.

If no anomalies found the “COMPROMISED” button will remain green and no threats will be displayed.

4.2.4. **Recommendation System**

1. Description

The button will be situated to the bottom of results section,

The user can choose to view the recommendation based on the results of his data.

1. Input

The button to be clicked to view the recommendations provided by the website.

1. Output

The window will contain two sections:

Immediate: The following actions will be recommended to enforce as soon as possible as the network is compromised.

15

Long term: To provide complete solution to the problem certain measures will be displayed.

These recommendations will be based on the results; they should be enforced on different networks uniquely.

4.2.5. **Contacts**

1. Description

This tab will contain contacts to the specialists of the Networking Problems. The results and recommendations can be analyzed to contact the right person. Thus, this channel of discussion will be opened between the Problem Solver and the User.

These contacts will be verified individuals or organizations which provide unique solutions to networking problems.

4.3. **NON-FUNCTIONAL REQUIREMENTS**

4.3.1. **Security**

1. Private Networking data of companies need to be protected.
2. To improve security of Website we will ensure our website is up to date in all areas. Regular scans will help rectify loopholes.
3. ‘WebARX’ platform can provide us great features to enhance our protection. Some features to be used can be:
4. ‘WebARX’ web application firewall.
5. Logs and stats on the cloud-based dashboard for regular checking up.
6. Uptime, Defacement, and blacklist monitoring.

16

4.3.2 **Reliability**

1. Efficient training will lead to the accuracy and hence reliability of our application.
2. We will ensure our model uses the best technique in terms of performance, time, and scalability (ex. SVMs and NNs have proved to be very efficient).
3. Our training will include large datasets so that we get best approximation of coefficients and variables involved, and finally an efficient model.

4.4. **CONSTRAINTS**

4.4.1. **Software Requirements**

1. Operating system—Windows, Linux, iOS and other supporting browser installation.
2. Software – Chrome version 44.
3. 64-bit processing

4.4.2. **Hardware Requirements**

1. Processor – i5/i7/i9 series or AMD series.
2. RAM – Minimum 8 GB
3. Memory – SSD preferred
4. Graphics/GPU – Preferable dedicated Graphics like GTX Nvidia (2 GB)

4.4.3. **Design Constraints** 17

1. Privacy – The Networking data can be at risk if it contains much classified information about the network, a mirror of the original data after the filtration is preferred. Security of data is of utmost priority.
2. Versatility – The model should be able to handle diverse range of attacks, testing on large datasets will be preferred.
3. Fault Tolerance—should handle power cut and internet disruptions by insuring no data loss due to any failure.

### 4.5 USAGE SCENARIO

### User profiles

1. **User**: He is the end-user who is using the service. He has two important actions:
   1. Login or Signup
   2. Input the Network Traffic file in .csv format

18

Diagram

Description automatically generated

Figure 4.1: Use Case Diagram

1. **System**: This is the system which hosts the service which the end-user uses. It has the following important action:
   1. Handle the authorization of user.
   2. Process the File and generated recommendations
   3. Upload new data on the databased so that the website remains up to date.

**CHAPTER 5**

**DETAILED DESIGN DOCUMENT**

20

### BACKGROUND

This section aims to give an overview of the theory related to the approach towards building an Intrusion Detection and recommendation system.

**5.2 THEORY**

**5.2.1 Detection Techniques:**

There are Majorly two types of detection techniques: -

**1. Signature-based**

Signature-based IDS refers to the detection of attacks by looking for specific patterns, such as byte sequences in network traffic, or known malicious instruction sequences used by malware. This terminology originates from anti-virus software, which refers to these detected patterns as signatures. Although signature-based IDS can easily detect known attacks, it is difficult to detect new attacks, for which no pattern is available.

#### **2. Anomaly-based**

#### Anomaly-based intrusion detection systems were primarily introduced to detect unknown attacks, in part due to the rapid development of malware. The basic approach is to use machine learning to create a model of trustworthy activity, and then compare new behavior against this model. Since these models can be trained according to the applications and hardware configurations, machine learning based.

**5.2.2. Intrusion Detection System**

An Intrusion Detection System (IDS) is a device or software application that monitors a network or systems for malicious activity or policy violations. Any intrusion activity or violation is typically reported either to an administrator or collected centrally using a security information and event management (SIEM) system.

21

An IDS differs from a firewall in that a traditional network firewall (distinct from a Next-Generation Firewall) uses a static set of rules to permit or deny network connections. It implicitly prevents intrusions, assuming an appropriate set of rules have been defined. Essentially, firewalls limit access between networks to prevent intrusion and do not signal an attack from inside the network. An IDS describes a suspected intrusion once it has taken place and signals an alarm. An IDS also watches for attacks that originate from within a system. This is traditionally achieved by examining network communications, identifying heuristics and patterns (often known as signatures) of common computer attacks, and taking action to alert operators. A system that terminates connections is called an intrusion prevention system and performs access control like an application layer firewall.

Anomaly-based IDS uses Machine Learning techniques to classify the network

Traffic into ‘normal’ or ‘attack’, thus making this a classification problem.

In our attempt to build an online IDS, we try classifying the network traffic

Present in the uploaded file of client with the help **Random Forest Classification**.

22

**5.2.3 Random Forest Classification**

Random forest, like its name implies, consists of many individual decision trees that operate together. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model’s prediction (see figure below).

A picture containing diagram

Description automatically generated

Fig. 5.1: Random Forest

**Creation of Random Forest**

1. Randomly select “K” features from total “m” features where

k << m.

1. Among the “K” features, calculate the node “d” using the best split point.
2. Split the node into daughter nodes using the best split.
3. Repeat the a to c steps until “l” number of nodes has been reached.

23

1. Build forest by repeating steps a to d for “n” number times to create “n” number of trees

**Prediction using Random Forest**

1. Takes the test features and use the rules of each randomly created decision tree to predict the outcome and stores the predicted outcome (target)
2. Calculate the votes for each predicted target
3. Consider the high voted predicted target as the final prediction from the random forest algorithm.

Diagram, schematic

Description automatically generated

Fig. 5.2: Prediction using Random Forest

**CHAPTER 6**

**APPROACH**

25

This section aims to give an overview of the approach towards building an online Intrusion Detection System.

**6.1 TRAINING THE ALGORITHM**

1. We used the KDD’99 Cup Dataset which is considered as the benchmark for Intrusion Detection Research. It was made by setting up an environment to acquire nine weeks of raw TCP dump data for a local-area network (LAN) simulating a typical U.S. Air Force LAN. The dataset consists of 41 Parameters like Protocol\_type, server\_error, etc. Each row is classified as an ‘attack’ or a ‘normal’ target.
2. The Processes to train our Algorithm were:
   * + 1. Data Cleaning and Correlation analysis
       2. Scaling the data
       3. Fitting the Algorithm with data

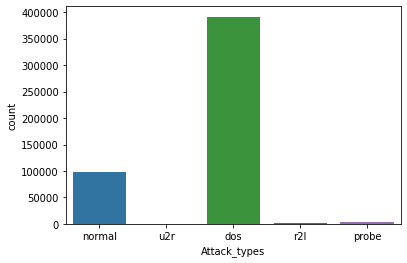


Fig 6.1: The KDD’99 Data Distribution

26

Graphical user interface, text, application, email

Description automatically generated

Fig 6.2: Training and Testing our Algorithm

**6.2 UPLOADING THE NETWORK TRAFFIC FILE**

Capturing of Network Traffic may not be easy, it requires domain knowledge. All

the network traffic can be collected via the TCP dump feature in our PCs. The network data capturing tools such as Wireshark can be very helpful. Wireshark can collect and present data beautifully to be seen and analyzed. This data does not contain the server-side variables, but they can also be collected if the server is internal to the organization. Wireshark tool also gives the flexibility of converting the .pcap capture files to .csv format making them ready to be uploaded.

To upload these files, we build a media directory in our server which stores the uploaded files. The URL patterns are rooted with this media path and whenever a file is uploaded it gets saved and ready to be opened.

27

Graphical user interface, application

Description automatically generated Text

Description automatically generated

*Front-end Back-end*

Fig 6.3: Uploading file to the server

**6.3 PREDICTION USING RANDOM FOREST CLASSIFIER**

1. The previously trained algorithm is saved by using the **Joblib Library** of python. The trained model and the scaler is saved using the command

‘joblib.dump()’ and loaded in the server using ‘joblib.load()’, specifying path in the brackets.

1. The uploaded file is read using the **csv library of Django** in the server and predictions for each row in the file is made.
2. The scaler and model is just used to predict for the given row. As they were giving 99.99% accuracy before, they correctly classify the attacks in the file.

**6.4 DISPLAYING RESULTS AND RECOMMENDATIONS**

28

Based on the predictions given by the Random forest classifier the server searches the databases for the appropriate results and recommendations and displays them on the website.

Fig 6.4: Displaying the Results

**6.5 HISTORY AND ITS FUNCTION**

1. History is a very important feature of our model; it is an analysis tool. The client who is logged in our system can see his detection history using this option.
2. His Detection are saved when he/she selects the ‘**Save Changes’** provided by us after every successful detection. Along with results the time also gets saved. It is helpful in analyzing the network status at different intervals of time.
3. Our Model uses **SQLite database** which is in-built in Django. The history table made in SQLite contains a ‘**Foreign Key**’ which connects it to our user table. The key being the field ‘Username’, it is helpful in mapping history to a specific user.

29

Graphical user interface, text, application, email

Description automatically generated

Fig 6.5: History

**CHAPTER 7**

**PROTOTYPE**

31

**A picture containing text, electronics, circuit

Description automatically generated**

Fig 7.1: Our Website

**Graphical user interface

Description automatically generated**

Fig 7.2: User Login

32

Graphical user interface, application

Description automatically generated

Fig 7.3: Uploading the file

A picture containing table

Description automatically generated

Fig 7.4: Getting Results and Recommendations

**CHAPTER 8**

**CONCLUSION**

### 34

### 8.1 CONCLUSION

* + - From Literature survey, it is concluded that Machine Learning Algorithms gave the best accuracy in Intrusion Detection systems.
    - Through our project we are trying to solve the problem of Un-accessibility of IDS to small and mid-range businesses. We are using the Random Forest Classification as our algorithm,
    - Using the KDD’99 Dataset complete training of our algorithm takes place. The dataset contains ~40 million rows of data, the algorithm gives an accuracy of 99%, hence making is near perfect and efficient.
    - We used Django for our backend, HTML and CSS for out front-end design. SQLite was used as a database using which we created history table which is a major feature of our project.
    - Moreover, since our recommendations are research based we are confident that our solutions will help any organization in securing their system and making their networks vulnerability free.

35

* 1. **FUTURE SCOPE**
* For detection of attacks on different type of servers, we can train different models and give accurate results based on the trained model.
* To further enhance our website, we can also identify the user’s / attacker’s IP address and can help identify from where the attack has happened.
* At the last we will upgrade our User Interface and add better recommendations to our Databases so that our clients receive the Complete Solution.

**CHAPTER 9**

**REFERENCES**

37

[1.] A New Hybrid Machine Learning for Cyber security Threat Detection Based on Adaptive Boosting *by Ployphan Sornsuwit and Saichon Jaiyen, Published dated Mar 1 , 2019*.

[2.] Anomaly detection of IOT Cyber Attacks in smart city using Machine learning *by I Alrashdi and Ali Alqazzaz, Published dated Sep 2019.*

[3.] Network Intrusion detection for Cyber Security using Unsupervised Deep Learning Approaches *by Md Zahangir Alom and Tarek M. Taha dated 2018*.

[4.] Sci-Hub – ISEE Papers and Journals for Literature Survey

[5.] Youtube, Stack Overflow – Providing Django tutorials, Code snippets etc

[6.] Wikipedia – Cyber Security, Cyber Threats etc.

[7.] KDD’99 Cup Archive – 4 Million rows containing database for our model training.