## INNOVATIVE PROJECT DEVELOPMENT REPORT

## AN ENHANCED NETWORK SECURITY USING MACHINE LEARNING AND BEHAVIORAL ANALYSIS

### ***Submitted by***

**M.ASRA BEGUM M.SUMAIYA M.SHAINI P.ABHINAYA**

**22RH1A05F3 22RH1A05G0 22RH1A05G3 22RH1A05J3**

***Under the Esteemed Guidance of***

## MS.B.SRAVANI

#### Assistant Professor

***In partial fulfillment of the Academic Requirements for the Degree of***

## BACHELOR OF TECHNOLOGY

Computer Science & Engineering



**MALLA REDDY ENGINEERING COLLEGE FOR WOMEN**

#### (Autonomous Institution-UGC, Govt. of India)

**Accredited by NAAC with ‘A+’ Grade, UGC, Govt. of India | Programmes Accredited by NBA National Ranking by NIRF Innovation-Rank band(151-300),MHRD, Govt. of India**

Approved by AICTE, Affiliated to JNTUH,ISO 9001-2015 Certified Institution

**Maisammaguda, Dhulapally, Secunderabad, Kompally-500 100.**

[**www.mallareddyecw.com**](http://www.mallareddyecw.com)

**MALLA REDDY ENGINEERING COLLEGE FOR WOMEN**

(Autonomous Institution-UGC, Govt. of India)

**Accredited by NAAC with ‘A+’ Grade, UGC, Govt. of India | Programmes Accredited by NBA National**

**Ranking by NIRF Innovation-Rank band(151-300),MHRD, Govt. of India**

Approved by AICTE, Affiliated to JNTUH,ISO 9001-2015 Certified Institution

**Maisammaguda, Dhulapally, Secunderabad, Kompally-500 100.**

2023-24

***DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING***

## CERTIFICATE

This is to certify that the Project work entitled “AN ENHANCED NETWORK SECUIRTY USING MACHINE LEARNING AND BEHAVIORAL ANALYSIS**”** is carried out by M.ASRA BEGUM **(22RH1A05F3)**, **M.SUMAIYA AMRIN (22RH1A05G0), M.SHAINI (22RH1A05G3), P.ABHINAYA (22RH1A05J3)** in partial fulfillment for the award of degree of **BACHELOR OF TECHNOLOGY** in Computer Science and Engineering, Malla Reddy Engineering Collage For Women (Autonomous), Hyderabad during the academic year 2023-2024.

#### Supervisor’s Signature Head of the Department

**Ms.B.Sravani Dr. Geetha Reddy**

Assistant Professor Professor and HOD

**EXTERNAL EXAMINER**

# ACKNOWLEDGEMENT

We feel ourselves honored and privileged to place our warm salutation to our college **Malla Reddy Engineering College for Women** and Department of **Computer Science and Engineering** which gave us the opportunity to have expertise in engineering and profound technical knowledge.

We wish to convey gratitude to our Principal **Dr. Y. Madhavee Latha**, for providing us with the environment and mean to enrich our skills and motivating us in our endeavour and helping us to realize our full potential.

We would like to thank Prof. **A. Radha Rani**, Director of Computer Science and Engineering & Information Technology for encouraging us to take up a project on this subject and motivating us towards the Project Work.

We express my sincere gratitude to **Dr. Geetha Reddy**, Head of the Department of Computer Science and Engineering for inspiring us to take up a project to this subject and successfully guiding us towards its completion.

We would like to thank our internal guide **Ms.B.Sravani**, Assistant Professor and all the Faculty members for their valuable guidance and encouragement towards the completion ofour project work.

With regards and gratitude,

###### M.ASRA (22RH1A05F3)

###### M.SUMAIYA (22RH1A05G0)

###### M.SHAINI (22RH1A05G3)

###### P.ABHINAYA (22RH1A05J3)

We hereby declare that our project entitled **“AN ENHANCED NETWORK SECURITY USING MACHINE LEARNING AND BEHAVORIAL ANALYSIS”** submitted to **Malla Reddy Engineering College for Women, Hyderabad** for the award of the Degree of Bachelor of Technology in **Computer Science and Engineering** is a result of original research work done by us.

It is declared that the project report or any part thereof has not been previously submitted to any University or Institute for the award of Degree.

###### M.ASRA (22RH1A05F3)

###### M.SUMAIYA (22RH1A05G0)

###### M.SHAINI (22RH1A05G3)

###### P.ABHINAYA (22RH1A05J3)

Feature selection methods for classification are crucial for intrusion detection techniques using machine learning. The high-dimensional features in intrusion detection data affect computational complexity, consume more used resources and more time for data analysis, and the irrelevant and redundant features among them often hinder the performance of classifiers and mislead the classification task. Therefore, it is challenging to select more relevant features from intrusion detection data containing many such features. Therefore, we proposes an efficient feature selection algorithm that considers the correlation between features and the redundancy of pairs of features with respect to class labels based on an improved Pearson correlation coefficient. The evaluation function is later improved based on conditional mutual information to obtain a final subset of features with the goal of improving the classification rate and accuracy. The proposed feature selection method is compared with three existing feature selection methods on the frequently studied public benchmark intrusion detection dataset NSL-KDD, and the experimental results show that the features selected by the proposed method lead to a significant reduction in execution time while resulting in higher classification accuracy.

# INDEX

|  |  |
| --- | --- |
| **TITLE** | **PAGE NO.** |
| **1. INTRODUCTION** | **1** |
| Introduction about the project | 1 |
| Existing System | 2 |
| Proposed System | 2 |
| **2. SYSTEM STUDY** | **3** |
| Economical Feasibility | 3 |
| Technical Feasibility  Social Feasibilty | 3  3 |
| **3. REQUIREMENT ANALYSIS** | **4** |
| Requirement Specification | 4 |
| Hardware and software | 4 |
| **4. SYSTEM DESIGN** | **6** |
| Architecture Diagram | 6 |
| Use case Diagram | 6 |
| Class Diagram | 7 |
| Activity Diagram | 7 |
| Sequence Diagram | 8 |
| **5. MODULES** | **9** |
| **6. ALGORITHMS** | **12** |
| Random Forest | 11 |
| Decision Tree | 11 |
| Support Vector Machine | 13 |
| Source Code | 14 |
| **7. SYSTEM TEST** | **17** |
| Types of Tests | 17 |
| **8. INPUT AND OUTPUT DESIGN** | **20** |
| Input Design | 20 |
| Output Design | 20 |
| Output Screens | 21 |
| **9. CONCLUSION** | **27** |
| **10. FUTURE SCOPE** | **28** |
| **11. REFERENCE** | **29** |

## CHAPTER 1

### INTRODUCTION ABOUT THE PROJECT

Cybercrimes come in many different forms, and they are continually expanding because to the remarkable growth of the web. For instance, theft of expertise, fishing, carding, viruses, economic fraud, intrusions, and attacks are a few instances of various different types of cybercrimes that make investigating them challenging. Therefore, managing and controlling different network business traffic through efficient technical means, detecting network intrusion, differentiating services,have all become challenges in the field of network operation and maintenance management.This attack is cross site scripting attacker community is use this to attack on websites for that they use malicious code embedded on the web applications. This malicious code is nothing but JavaScript. To prevent this first we have to find out that malicious script. This XSS attack is widely used to the thief important information. This attack ranks 3rd in Open-Source Foundation for Application Security (OWASP). To find out that script we build the model to detect XSS attacks. As the attacks on the online applications is enhanced at an terrible rate, therefore this method puts forward a sophisticated XSS detection technique by introducing a new classification system for privilege levels and vulnerability levels of the content rendered within the browser.

### EXISTING SYSTEM

In literature they used machine learning techniques to detect cyberbullying in text data collected from the

website Formspring.me. They used language patterns used by bullies and their victims to develop rules to automatically detect cyberbullying content. They collected data from Formspring.me, which is a question-and-answer formatted website that contains a high percentage of bullying content. The data was labeled using a web service, Amazon's Mechanical Turk. They used the labeled data, in conjunction with machine learning techniques provided by the Weka tool kit, to train a computer to recognize bullying content.Another research focuses on the application of deep learning techniques for various use cases of Cyber security such as Android Malware Classification, Incident Detection, and Fraud Detection. They conducted experiments to choose the best performing neural network architecture for different network parameters and structures. Their RNN architecture consists of an input layer with six hidden layers and an output layer, with fully connected neurons in input to hidden layer and hidden to output layer.Their Recurrent Network is composed of recurrent layers, fully-connected layers, batch normalization layers, and dropout layers.

### PROPOSED SYSTEM:

We proposes a novel supervised machine learning system that classifies network traffic as either malicious or benign. The proposed model uses a combination of supervised learning algorithms and feature selection methods to achieve the best detection success rate. The performance of the model is evaluated using the NSL-KDD dataset to classify network traffic using SVM and ANN supervised machine learning techniques.And compared with other existing models.

#### Extension :-

As an extension we applied an ensemble method combining the predictions of multiple individual models to produce a more robust and accurate final prediction.

However, we can further enhance the performance by exploring other ensemble techniques such as Voting Classifier RF + AB which got 100% of accuracy,

As an extension we can build the front end using the flask framework for user testing and with user authentication.

## CHAPTER 2

###### 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.

## CHAPTER 3

### REQUIREMENT ANALYSIS

The project involved analyzing the design of few applications so as to make the application more users friendly.

**REQUIREMENT SPECIFICATIONS**

Software Requirements

Software requirements deal with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or prerequisites are generally not included in the software installation package and need to be installed separately before the software is installed.

**Platform –** In computing, a platform describes some sort of framework, either in hardware or software, which allows software to run. Typical platforms include a computer’s architecture, operating system, or programming languages and their runtime libraries.

Operating system is one of the first requirements mentioned when defining system requirements (software). Software may not be compatible with different versions of same line of operating systems, although some measure of backward compatibility is often maintained. For example, most software designed for Microsoft Windows XP does not run on Microsoft Windows 98, although the converse is not always true. Similarly, software designed using newer features of Linux Kernel v2.6 generally does not run or compile properly (or at all) on Linux distributions using Kernel v2.2 or v2.4.

1. **Software : Anaconda**
2. **Primary Language : Python**
3. **Frontend Framework : Flask**
4. **Back-end Framework : Jupyter Notebook**
5. **Database : Sqlite3 Front-End Technologies : HTML,CSS,JavaScript and Bootstrap4**

**HARDWARE REQUIREMENTS**

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware, A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatible, and sometimes incompatible hardware devices for a particular operating system or application. The following sub-sections discuss the various aspects of hardware requirements.

**1) Operating System : Windows Only**

**2) Processor : i5 and above**

**3) Ram : 8gb and above**

**4) Hard Disk : 25 GB in local drive**

## CHAPTER 5

### MODULES

MODULES:

* + Data exploration: using this module we will load data into system
  + Processing: Using the module we will read data for processing
  + Splitting data into train & test: using this module data will be divided into train & test
  + Model generation: Model building- SVM, Naive Bayes, Random Forest, ANN, Voting Classifier (RF + AB)
  + User signup & login: Using this module will get registration and login
  + User input: Using this module will give input for prediction
  + Prediction: final predicted displayed

**Note:** As an extension we applied an ensemble method combining the predictions of multiple individual models to produce a more robust and accurate final prediction.

However, we can further enhance the performance by exploring other ensemble techniques such as Voting Classifier RF + AB which got 100% of accuracy.

## CHAPTER 6

### ALGORITHM

SVM: Support Vector Machines (SVMs) are a type of supervised learning algorithm that can be used for classification or regression tasks. The main idea behind SVMs is to find a hyperplane that maximally separates the different classes in the training data. This is done by finding the hyperplane that has the largest margin, which is defined as the distance between the hyperplane and the closest data points from each class. Once the hyperplane is determined, new data can be classified by determining on which side of the hyperplane it falls. SVMs are

particularly useful when the data has many features, and/or when there is a clear margin of separation in the data.

Naive Bayes: Naïve Bayes algorithm is comprised of two words Naïve and Bayes, Which can be described as: Naïve: It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features.

Random Forest: Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

ANN: Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the brain. ANNs, like people, learn by examples. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning largely involves adjustments to the synaptic connections that exist between the neurons.

Voting Classifier: Voting Classifier is a machine learning model that trains on an ensemble of numerous models and predicts an output (class) based on their highest probability of chosen class as the output.

## CHAPTER 7

### SYSTEM TEST

System testing, also referred to as system-level tests or system-integration testing, is the process in which a quality assurance (QA) team evaluates how the various components of an application interact together in the full, integrated system or application. System testing verifies that an application performs tasks as designed. This step, a kind of black box testing, focuses on the functionality of an application. System testing, for example, might check that every kind of user input produces the intended output across the application.

Phases of system testing:

A video tutorial about this test level. System testing examines every component of an application to make sure that they work as a complete and unified whole. A QA team typically conducts system testing after it checks individual modules with functional or user-story testing and then each component through integration testing.

If a software build achieves the desired results in system testing, it gets a final check via acceptance testing before it goes to production, where users consume the software. An app-dev team logs all defects, and establishes what kinds and amount of defects are tolerable.

**Software Testing Strategies:**

Optimization of the approach to testing in software engineering is the best way to make it effective. A software testing strategy defines what, when, and how to do whatever is necessary to make an end-product of high quality. Usually, the following software testing strategies and their combinations are used to achieve this major objective:

Static Testing:

The early-stage testing strategy is static testing: it is performed without actually running the developing product. Basically, such desk-checking is required to detect bugs and issues that are present in the code itself. Such a check-up is important at the pre-deployment stage as it helps avoid problems caused by errors in the code and software structure deficits.

**Structural Testing:**

It is not possible to effectively test software without running it. Structural testing, also known as white-box testing, is required to detect and fix bugs and errors emerging during the pre-production stage of the software development process. At this stage, unit testing based on the software structure is performed using regression testing. In most cases, it is an automated process working within the test automation framework to speed up the development process at this stage. Developers and QA engineers have full access to the software’s structure and data flows (data flows testing), so they could track any changes (mutation testing) in the system’s behavior by comparing the tests’ outcomes with the results of previous iterations (control flow testing).

**Behavioral Testing:**

The final stage of testing focuses on the software’s reactions to various activities rather than on the mechanisms behind these reactions. In other words, behavioral testing, also known as black-box testing, presupposes running numerous tests, mostly manual, to see the product from the user’s point of view. QA engineers usually have some specific information about a business or other purposes of the software (‘the black box’) to run usability tests, for example, and react to bugs as regular users of the product will do. Behavioral testing also may include automation (regression tests) to eliminate human error if repetitive activities are required. For example, you may need to fill 100 registration forms on the website to see how the product copes with such an activity, so the automation of this test is preferable.

**TEST CASES:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **INPUT** | **If available** | **If not available** |
| 1 | User signup | User get registered into the application | There is no process |
| 2 | User sign in | User get login into the application | There is no process |
| 3 | Enter input for prediction | Prediction result displayed | There is no process |

## CHAPTER 8

### INPUT AND OUTPUT DESIGN

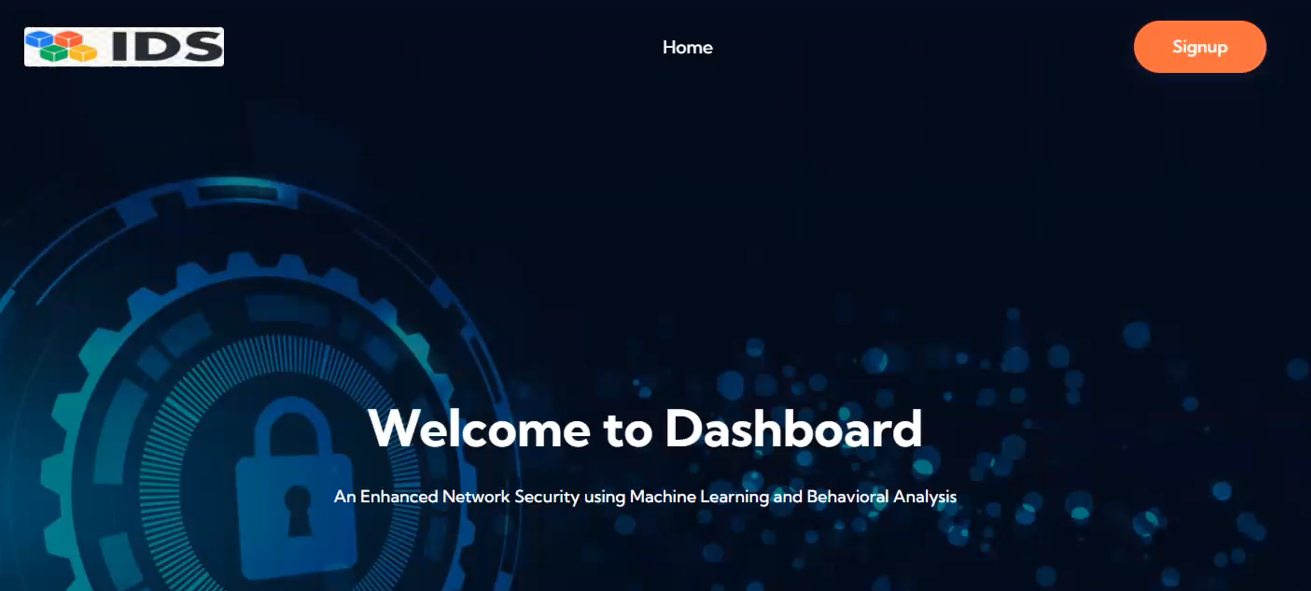
**DATA FLOW DIAGRAM:**

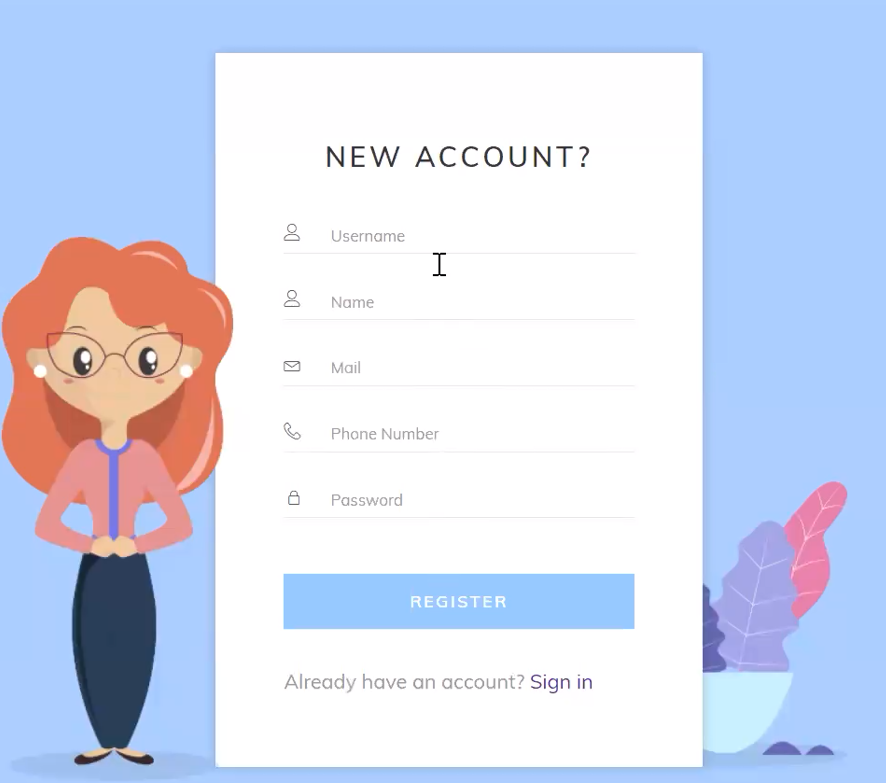
1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important 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.

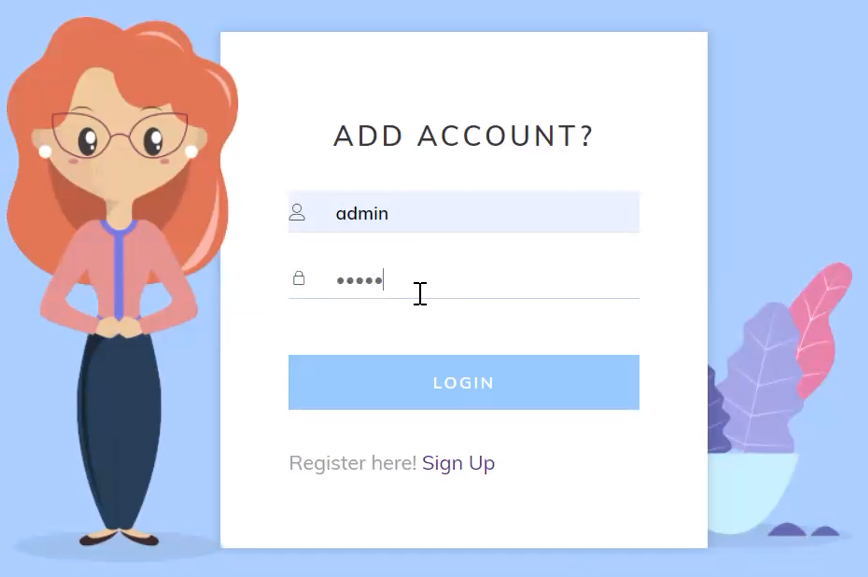
DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that

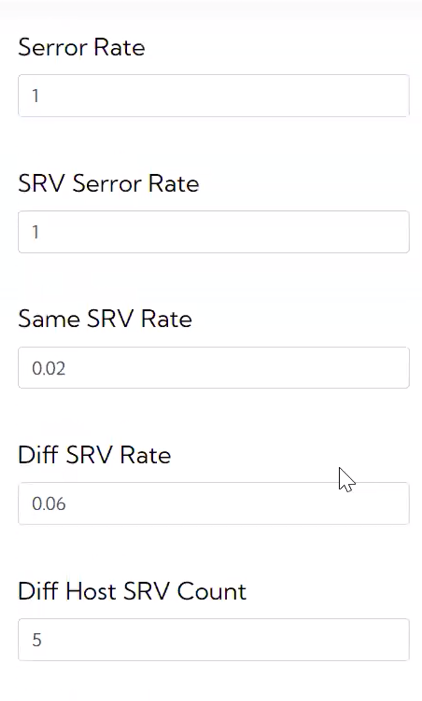
1. depicts information flow and the transformations that are applied as data moves from input to output.
2. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

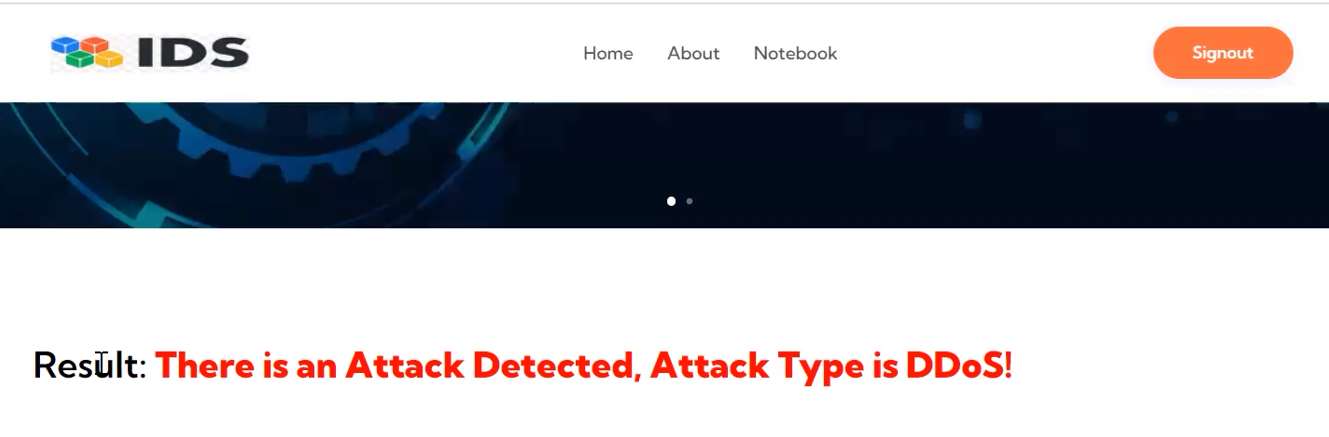
## OUTPUT SCREENS

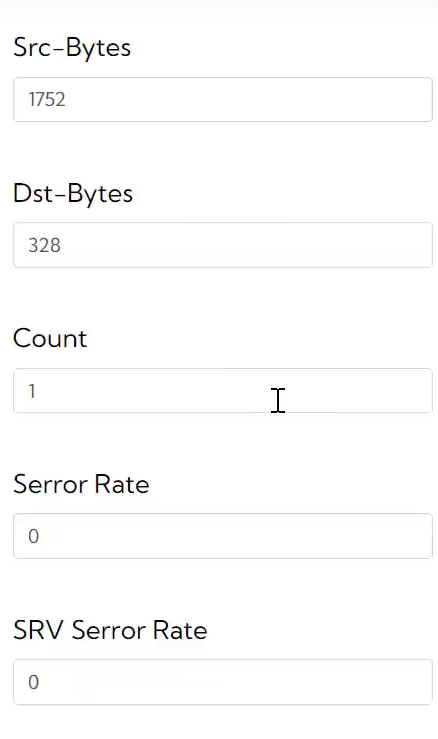


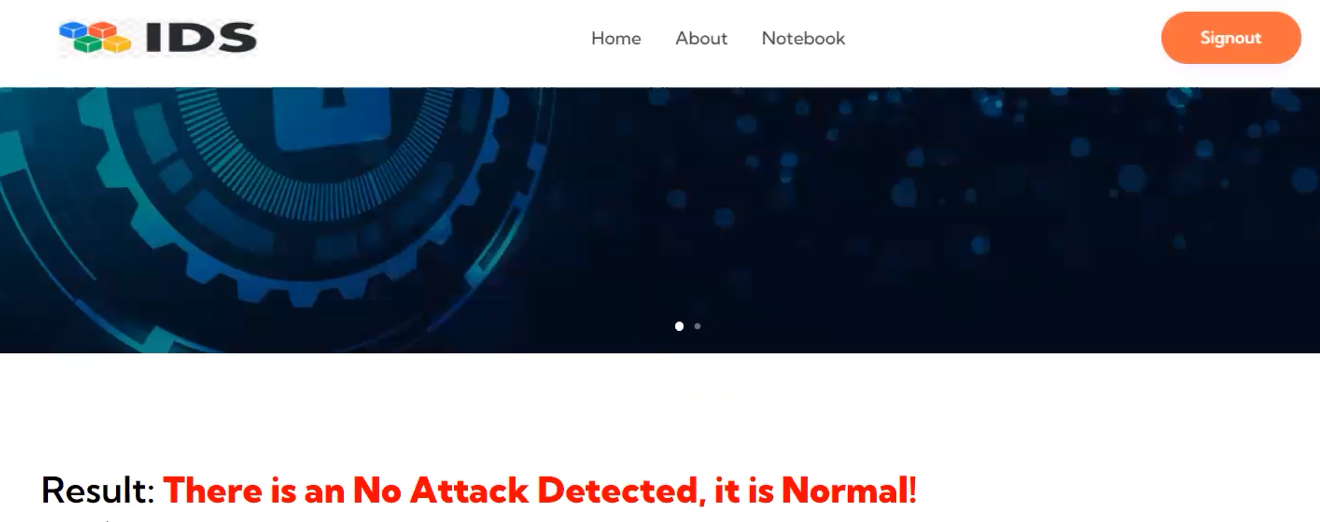


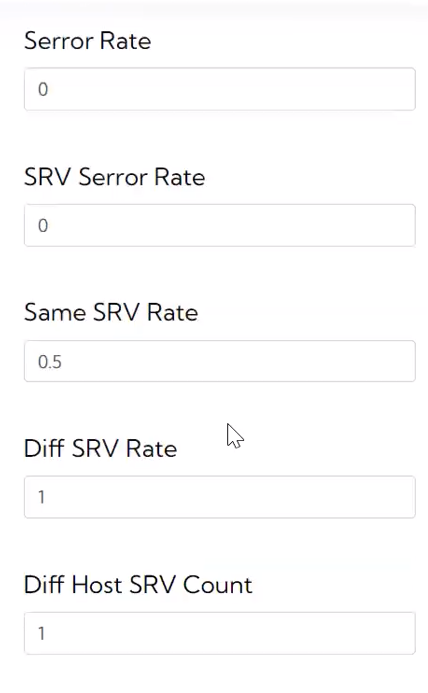


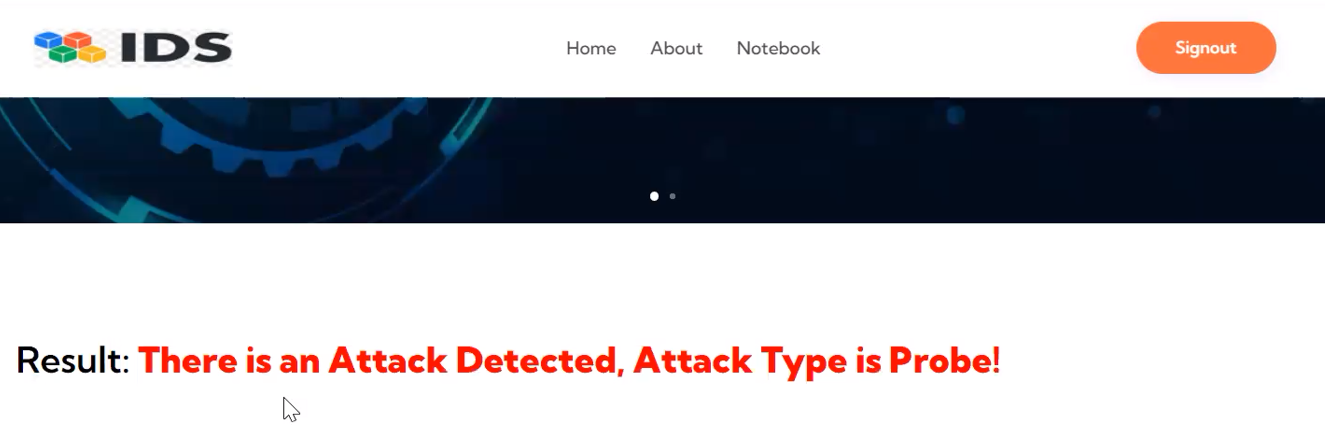












**CHAPTER 9**

### CONCLUSION

We proposes a novel supervised machine learning system that classifies network traffic as either malicious or benign. We uses a combination of supervised learning algorithms and feature selection methods to achieve the best detection success rate. The performance of the model is evaluated using the NSL-KDD dataset to classify network traffic using SVM and ANN supervised machine learning techniques.The results showing that the different machine learning technique is the most satisfactory technique to detect intrusion from network traffic. Using Artificial neural network (ANN) we are able to select and extract the features of network packets; Feature selection facilitates visualization of data & data understanding. And using SVM we classify the dataset and detect the intrusion from network.

**CHAPTER 10**

**FUTURE SCOPE**

The integration of ML techniques in network security has witnessed significant advancements and innovations, paving the way for more effective and adaptive security solutions . Deep learning techniques, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), are increasingly being applied to network intrusion detection systems (NIDS) for detecting and mitigating sophisticated cyber threats. Deep learning models can effectively analyze network traffic patterns, identify anomalous behaviors, and classify malicious activities with high accuracy and efficiency.

**CHAPTER 11 REFERENCES**

[1] Tchakoucht TA, Ezziyyani M. Building a fast intrusion detection system for highspeed-networks: probe and DoS attacks detection. Procedia Comput Sci. 2018;127:521–30.

[2] R Ramasamy, V Rajavel, M Vasim Babu, C N S Vinoth Kumar, S Parthiban, “Design and Analysis of Multiband Bloom Shaped Patch Antenna for IoT Applications”, Turkish Journal of Computer and Mathematics Education, Vol.12 No.3(2021), 4578-4585, April 2021. <https://doi.org/10.17762/turcomat.v12i3.1848>

[3] Aakriti nag, Rohit Ranjan, C.N.S.Vinoth Kumar, “An Approach on Cyber Crime Prediction Using Prophet Time Series”, 2022 IEEE 7th International conference for Convergence in Technology (I2CT), IEEE Xplore ISBN:978-1-6654-2168-3. DOI: 10.1109/I2CT54291.2022.9825386. April 2022.

[4] Zuech R, Khoshgoftaar TM, Wald R. Intrusion detection and big heterogeneous data: a survey. J Big Data. 2015;2:3.