iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

NETWORK BASED INTRUSION DETECTION SYSTEM USING DEEP LEARNING

Asst.Prof. Deshabattini Damodhar
Assistant Professor, Dept of AI&DS,
J.B. Institute of Engineering and Technology, Moinabad

Karatlapally Keerthana
Tenugu Shireesha
Vanneldas Srujan
UG Students Dept. of AI&DS,
J.B. Institute of Engineering and Technology, Moinabad

ABSTRACT
With the rapid increase in cyber threats, traditional intrusion detection systems (IDS) struggle to keep up with sophisticated attacks. This project aims to develop an Advanced Network Intrusion Detection System (NIDS) using Deep Learning techniques to detect and classify network intrusions effectively. The system processes real-time network traffic and classifies it as normal or malicious using deep learning models such as ML models. The dataset is preprocessed using feature engineering techniques like One-Hot Encoding and Min-Max Scaling to improve accuracy. The trained model is deployed in a Flask-based web application that continuously monitors network activity and alerts administrators about potential threats. Unlike traditional signature-based IDS, this system can detect zero-day attacks by learning patterns from previous intrusions. By comparing multiple deep learning architectures, we aim to achieve high accuracy, precision, and recall in intrusion detection. The proposed system enhances network security and helps organizations prevent unauthorized access and data breaches effectively.

Keywords:
Cyber threats, Intrusion Detection System (IDS), Advanced Network Intrusion Detection System (NIDS), Deep Learning, Machine Learning (ML), Network Traffic, Flask Web Application, Real-Time Monitoring, Zero-Day Attacks, Signature-Based IDS, Data Breaches, Network Security.

INTRODUCTION

With the rapid expansion of digital infrastructure in India, the threat of cyberattacks has become a critical concern. Distributed Denial of Service (DDoS) attacks, in particular, pose significant risks to both public and private sector networks. Existing cybersecurity solutions often fail to provide real-time detection and mitigation capabilities, leaving systems vulnerable to prolonged downtimes and data breaches. The proposed tool addresses these gaps by combining machine learning (ML) and deep learning (DL) algorithms to deliver a robust real-time cyber incident monitoring system.

The system processes live network data and classifies it into normal or malicious traffic based on patterns learned from historical datasets. By integrating a notification mechanism, users are immediately informed of potential threats, allowing them to take timely action. A key aspect of this project is the comparative analysis of various ML and DL models, such as Random Forest, Support Vector Machine (SVM), Convolutional Neural Networks (CNNs), and Long Short-Term Memory (LSTM), to ensure optimal performance.

This initiative not only contributes to enhancing cybersecurity but also aligns with the government's vision of a secure digital India. The tool's scalable architecture enables its deployment in diverse environments, ranging from corporate networks to critical infrastructure systems. By providing actionable insights and early warnings, this system empowers stakeholders to safeguard their digital assets effectively.

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

OBJECTIVES

The primary objective of this project is to design and implement an efficient Network Intrusion Detection System (NIDS) using deep learning techniques to enhance cybersecurity. The system aims to accurately detect various types of network intrusions by leveraging advanced deep learning models such as Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), and Transformer- based architectures. A key focus is to improve detection accuracy while minimizing false positives and false negatives to ensure reliable threat identification.

The project will involve preprocessing and feature engineering on benchmark datasets like NSL-KDD, CIC-IDS2017, and UNSW-NB15 to train and evaluate the system effectively. Additionally, real-time detection capabilities will be incorporated to enable swift responses to potential threats in a live network environment. The system will be optimized for scalability to handle high network traffic loads without compromising speed or efficiency. To validate its effectiveness, the deep learning-based NIDS will be compared against traditional machine learning methods such as Support Vector Machines (SVM), Random Forest, and Decision Trees. Furthermore, the project will ensure robustness against adversarial attacks, making the system resilient to sophisticated intrusion techniques. The final implementation will include a user-friendly dashboard for monitoring detected threats, generating reports, and issuing alerts to network administrators. Through this project, we aim to develop a secure, intelligent, and real-time intrusion detection system that significantly strengthens network security in modern digital infrastructures.

LITERATURE REVIEW

The authors looked at all of the features of phishing and non-phishing websites and came up with a list of nineteen standout qualities that can be used to tell the difference between phishing and legitimate websites. For successful identification of a wide-ranging phishing assault, this method collects and analyses numerous features of suspicious websites The classifier assesses not if a particular website seems to be a phishing site during the testing phase. This was the overall structure of the system architecture. In feature extraction, there are eight URL-based features, one CSS feature, six hyperlink-specific features, three web identity features, and one login form feature. detail in the work. In the implementation details, they have shared in detail how they implemented this work. To identify phishing sites, researchers first selected important and helpful characteristics. After that, they built a dataset by extracting features from both genuine and fake websites. The random forest model was trained using the labeled dataset. The recommended anti-phishing solution is implemented on a computer with a Pentium i5 CPU running at 2.4 GHz and 4 GB RAM. The Python language has been used to implement this method. The collection of information from a webpage necessitates the use of several libraries. The collection of information from a webpage necessitates the use of several libraries. These libraries may be downloaded and extracted from the main websites or added separately using the pip package for Python. Beautiful Soup is one of the libraries that is used to extract information from HTML and XML documents. Evaluated each of the features that are extracted using several libraries as mentioned above. Finally, the authors have also highlighted the various advantages of this approach. So, this work represented a method in which URL, hyperlink, CSS, login form, and identity characteristics were employed to provide a unique technique for screening phishing websites from the client-side.

SYSTEM ANALYSIS

EXISTING SYSTEM:

- Traditional IDS use signature-based and anomaly-based methods to detect network intrusions.
- Signature-based IDS relies on predefined attack signatures, making it ineffective for detecting new threats.
- Anomaly-based IDS monitors network behavior but often generates high false positive rates.

PROPOSED SYSTEM:

Proposed System Using Deep Learning

- We propose an AI-driven NIDS utilizing AdaBoostClassifier LogisticRegression CNN MLP classfierfor network attack detection.
- The system preprocesses network traffic using feature selection and encoding methods.
- CNN is used to extract spatial patterns, while LSTM detects sequential attack behaviors.

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

- AdaBoostClassifier LogisticRegression CNN MLP classfier enhances classification by combining Random Forest with CNN for improved accuracy.
- This system significantly improves accuracy, scalability, and adaptability in intrusion detection.

SYSTEM DESIGN

The System Design Document specifies the system requirements, architecture, database design, input/output formats, and processing logic.
Modules:
Service Provider
The Service Provider signs in to manage cyber incident attack classification, train/test datasets, examine accuracy reports, anticipate attack kinds, and grant user access.

Remote User:
Users register and log in to predict cyber incident attack kinds and maintain their profiles.

Methodology
The data is separated into two sets: training (80%) and pre-training (20%). Pre-training data is utilized to determine the best models based on mean absolute errors. Hyperparameters are optimized for maximum performance.
Processes
Data collection: Create your own dataset using tweets and forensic labels.
Preprocessing includes feature extraction and normalization.
Splitting data into training and testing sets.
Algorithm training involves training multiple ML models and selecting the best model.

SYSTEM ARCHITECTURE

A system architecture or systems architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system. Organized in a way that supports reasoning about the structures and behaviors of the system.

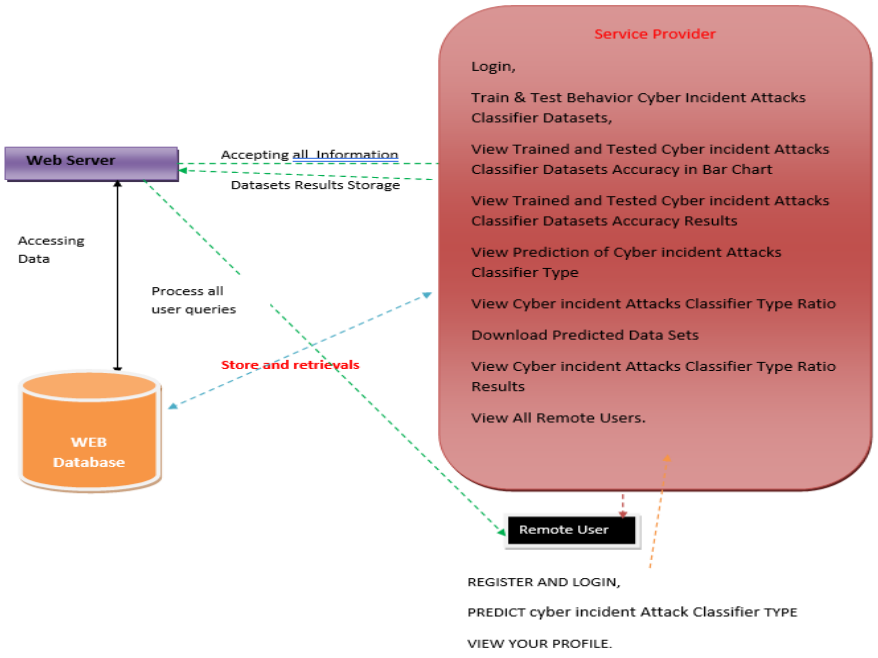


Figure 1: System Architecture

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

3-Tier Architecture:

The three-tier software architecture (a three-layer architecture) emerged in the 1990s to overcome the limitations of the two-tier architecture. The third tier (middle tier server) is between the user interface (client) and the data management (server) components. This middle tier provides process management where business logic and rules are executed and can accommodate hundreds of users (as compared to only 100 users with the two-tier architecture) by providing functions such as queuing, application execution, and database staging.

Advantages of Three-Tier:

- Separates functionality from presentation.
- Clear separation – better understanding.
- Changes limited to well define components
-

CONSTRUCTION OF USE CASE DIAGRAMS:

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

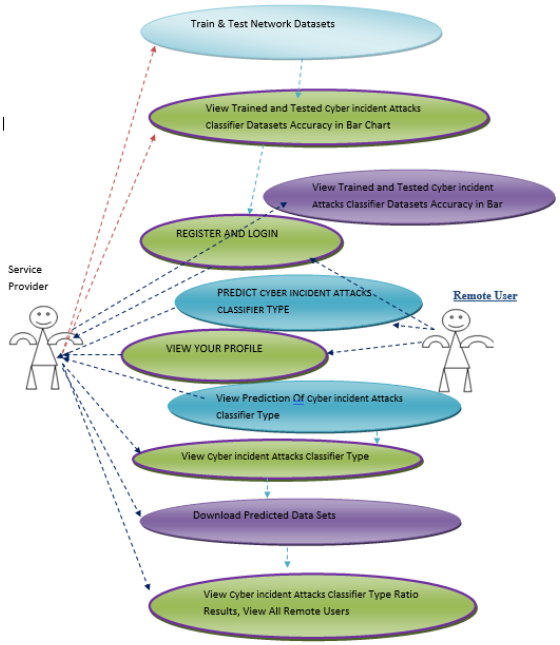


Figure 2: Use Case Diagram

SEQUENCE DIAGRAMS:

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

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

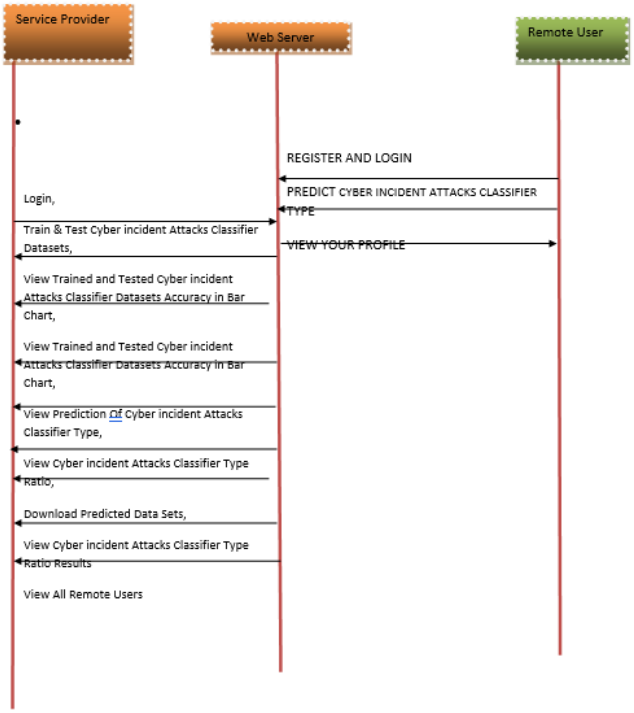


Figure.Sequence diagram

CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

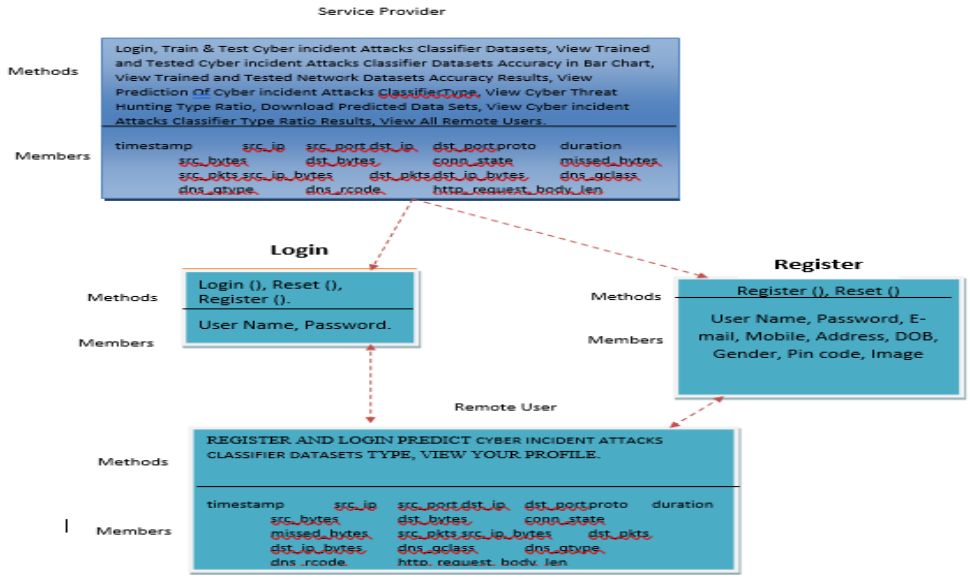


Figure: Class Diagram

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

ACTIVITY DIAGRAM:

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

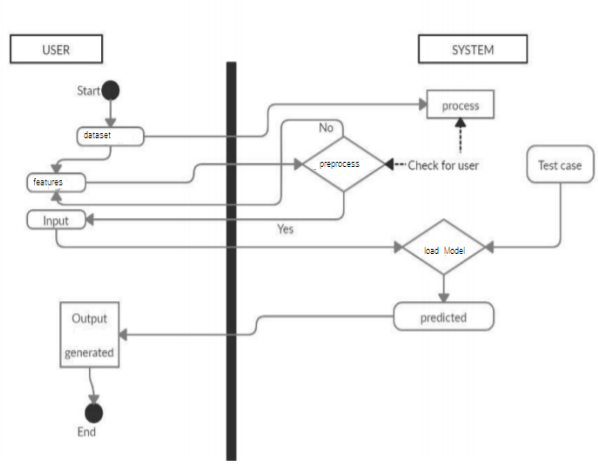
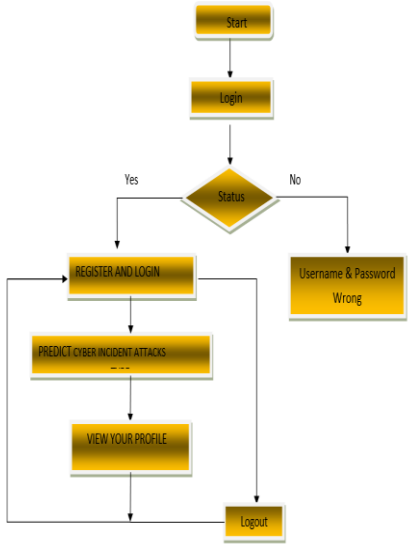


Figure: Activity Diagram



SYSTEM REQUIRMENTS

HARDWARE REQUIREMENTS:

- System : Intel(R) Core (TM) i3-7020U CPU @ 2.30GHz
- Hard Disk : 1 TB.

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

- Input Devices : Keyboard, Mouse
- Ram : 4 GB

SOFTWARE REQUIREMENTS:

- Operating system : Windows XP/7/10.
- Coding Language : Python
- Tool : Anaconda
- Interface : Django

SYSTEM IMPLEMENTATION

System implementation includes operational and technological studies to improve intrusion detection. It encourages improved approaches and tools for military applications.

Input and Output Designs

Logical design involves abstractly representing data flows, inputs, and outputs through models such as ER Diagrams.

Physical design refers to the real data input, verification, processing, storage, and output. It entails designing the user interface, data, and processes while also providing security, validation, and system control.

Input design prioritizes efficient data entry, error reduction, and security. It specifies input formats, validation methods, and user advice to ensure reliable data processing.

Output design involves creating clear, user-friendly, and decision-supporting outputs. It include choosing presentation options, structuring reports, and assuring accurate information delivery.

SYSTEM TESTING

System testing ensures the complete system meets specified requirements. It follows black-box testing principles and evaluates functionality without internal code knowledge.

Testing Levels:

- Unit Testing: Individual modules tested separately. Methods include Black Box and White Box Testing.
- Integration Testing: Ensures seamless interaction between modules using Bottom-Up and Top-Down approaches.
- System Testing: Validates overall system performance, security, and OS compatibility.

Test Cases:

- STC-1: OS Compatibility – Runs on Windows XP/7/8. Expected: Better performance on Windows 7. Result: As expected.
- ITC-1: Cyber Incident Attack Prediction – Checks classifier accuracy. Result: Pass.
- UTC-1: Data Feature Extraction – Extracts features and labels from datasets. Result: Pass.

METHODOLOGY

The methodology for developing a Network Intrusion Detection System (NIDS) using Deep Learning Techniques follows a systematic approach to ensure efficiency, accuracy, and real-time threat detection. The process begins with data collection and preprocessing, where benchmark datasets such as NSL-KDD, CIC-IDS2017, and UNSW-NB15 are used to train and evaluate the system. Preprocessing steps include data cleaning, feature extraction, normalization, and encoding to ensure high-quality input for the deep learning models. Next, different deep learning architectures, such as Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), and Transformer-based models, are designed and implemented to detect anomalies in network traffic. The models undergo training and optimization using techniques like hyperparameter tuning, dropout regularization, and batch

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

normalization to enhance detection accuracy while minimizing false positives and negatives. The performance of the models is then evaluated using key metrics such as accuracy, precision,
recall, F1- score, and ROC-AUC to measure their effectiveness. A comparative analysis is conducted against traditional machine learning models like Support Vector Machines (SVM), Random Forest, and Decision Trees to demonstrate the advantages of deep learning approaches. The system is then integrated into a real-time network environment, where it continuously monitors traffic and detects intrusions. A user- friendly dashboard is developed to provide real-time alerts, visualize detected threSats, and generate reports for network administrators.

OUTPUT SCREENS

MAIN PAGE:

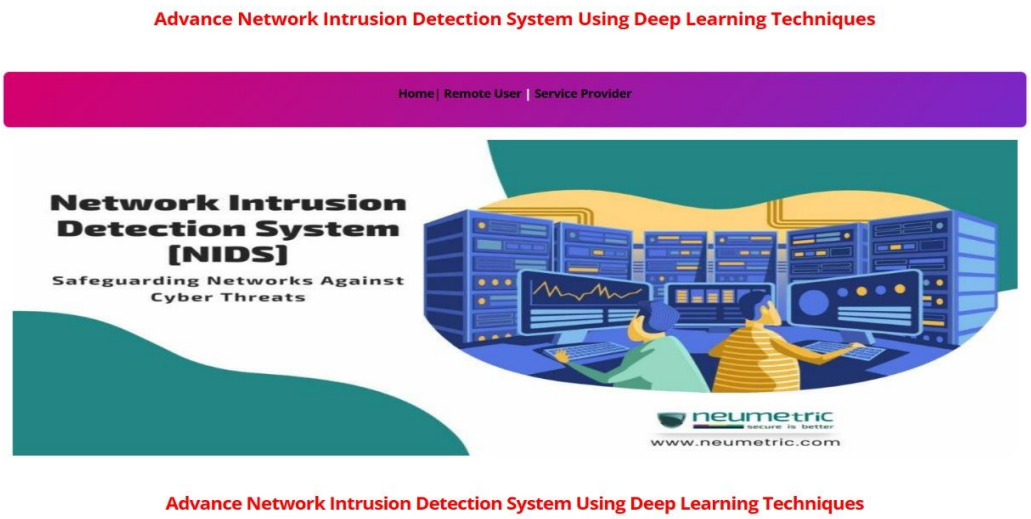


Fig Output screen 1

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

Fig output screen 2

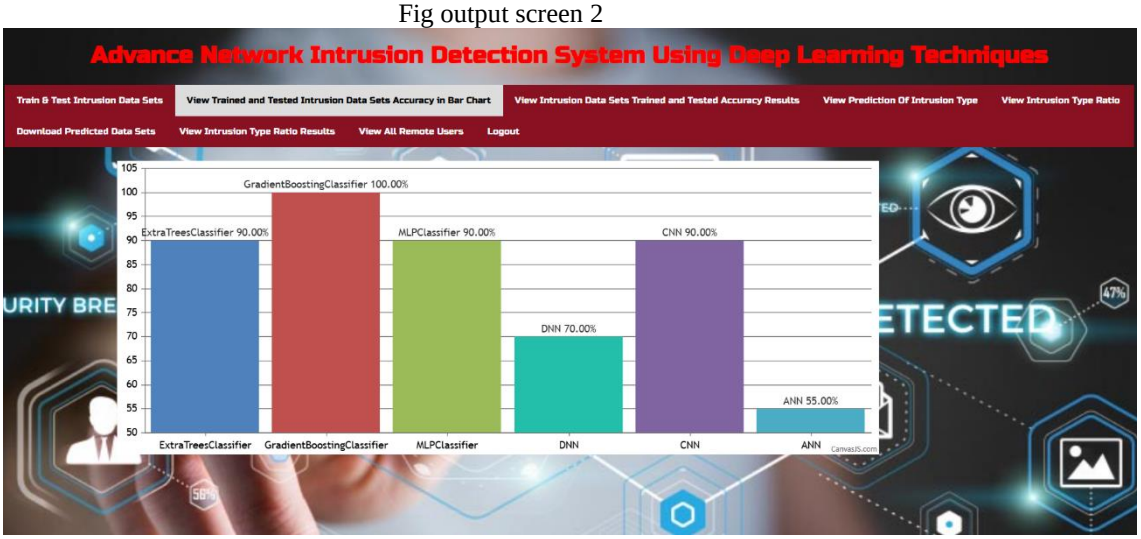


Fig output screen 3

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

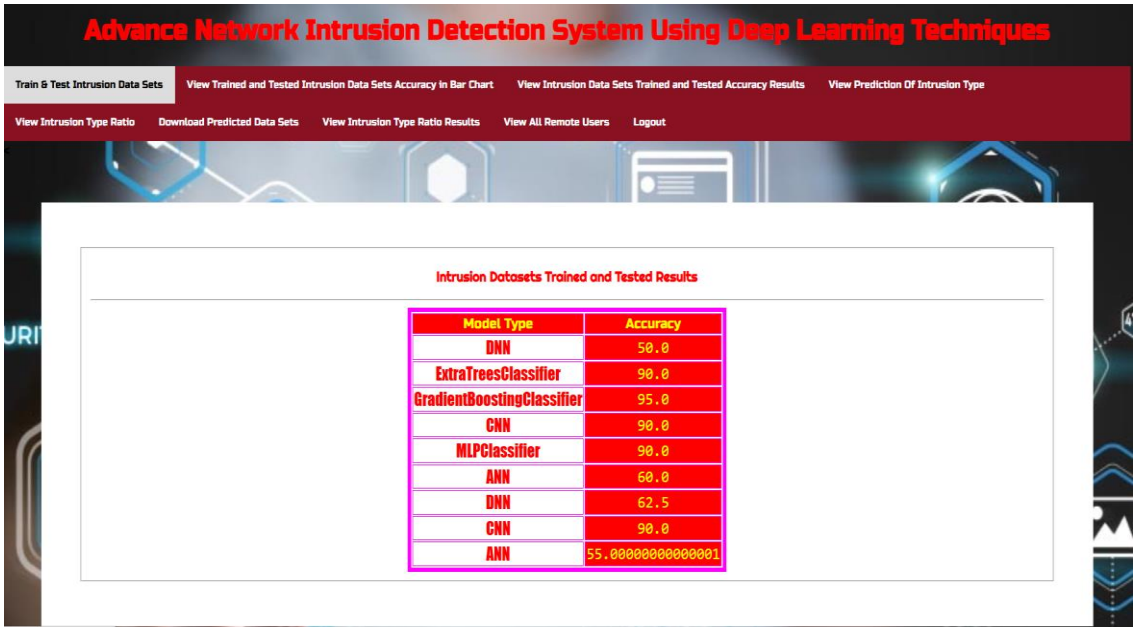


Fig: output screen 4

RESULTS AND DISCUSSIONS

The Network Intrusion Detection System (NIDS) using Deep Learning Techniques was evaluated using benchmark datasets like NSL-KDD, CIC-IDS2017, and UNSW-NB15. The deep learning models, particularly LSTM and CNN, outperformed traditional methods in accuracy, precision, and recall. The system effectively detected both known and unknown cyber threats with minimal false positives. Real- time testing showed efficient intrusion detection with low latency. However, challenges such as high computational costs and data requirements were observed. Optimization techniques like hyperparameter tuning and feature selection improved performance. Security evaluations confirmed robustness against adversarial attacks. Future enhancements will focus on scalability, efficiency, and explainability.

ACKNOWLEDGEMENT

I sincerely thank the following distinguished personalities who have given their advice and support for successful completion of this work. At outset we express our gratitude to the almighty lord for showering his grace and blessings upon us to complete this Major Project. Although our name appears on the cover of this book, many people have contributed in some form or the other to this Major Project development. We could not have done this Major Project without the assistance or support of each of the following.

CONCLUSION

This project demonstrates the effectiveness of machine learning and deep learning in addressing cybersecurity challenges. By providing real-time monitoring and DDoS detection, the tool offers a scalable solution for protecting Indian cyberspace. The comparative analysis ensures the selection of the most suitable model, enhancing overall system performance. With its user-friendly interface and automated alert mechanism, the tool bridges critical gaps in existing systems, paving the way for a more secure digital environment.

REFERENCES

[1] F. Song, Y. Lei, S. Chen, L. Fan, and Y. Liu, ‘‘Advanced cyber incident Attacks and mitigations on practical ML-based phishing website classifiers,’’ Int. J. Intell. Syst., vol. 36, no. 9, pp. 5210–5240, Sep. 2021.
[2] B. Sabir, M. A. Babar, and R. Gaire, ‘‘An evasion attack against ML-based phishing URL detectors,’’ Tech. Rep., 2020.

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

[3] H. Shirazi, B. Bezawada, I. Ray, and C. Anderson, ‘‘Adversarial sampling attacks against phishing detection,’’ in Proc. IFIP Annu. Conf. Data Appl. Secur. Cham, Switzerland: Springer, Jul. 2019, pp. 83–101.
[4] S. Anupam and A. K. Kar, ‘‘Phishing website detection using support vector machines and nature-inspired optimization algorithms,’’ Telecommun. Syst., vol. 76, no. 1, pp. 17–32, Jan. 2021.

Technical Reports and Articles:

[1] MIT AI Research Publications – https://news.mit.edu/topic/artificialintelligence
[2] IEEE Xplore Digital Library – Articles on AI-driven chatbot system (https://ieeexplore.ieee.org).
[3] ArXiv Preprints – Recent advances in NLP and LangChain application (https://arxiv.org)

PROJECT GUIDE

Dr. DESHABATTINI DAMODHAR
Assistant Professor
Department of Artificial Intelligence & Data Science (AI&DS)