

Network Intrusion Detection System

Dr C Murugamani, D. Sree Lakshmi Niharika, Sri Mahalaxmi Mummadi, Sudeeksha Gowlikar

¹Professor, Department of Information Technology, Bhoj Reddy Engineering College for Women.

^{2,3,4}B.tech students, Department of Information Technology, Bhoj Reddy Engineering College for Women.
niharikadaruri1731@gmail.com

ABSTRACT

Intrusions in computing environments represent a significant and ongoing challenge in the realm of cybersecurity. As technology continues to evolve, so do the techniques used by malicious actors to exploit vulnerabilities in digital systems. Despite decades of security advancements, the increasing dependency on interconnected networks demands more intelligent and adaptive security solutions. This project addresses the growing need for proactive threat detection by developing a Machine Learning-based Intrusion Detection System (IDS) that leverages the Random Forest algorithm. Trained on a comprehensive dataset of network traffic, the system is capable of identifying complex patterns and accurately classifying various types of cyberattacks, including denial-of-service (DoS), probing, and unauthorized access attempts. By automating the detection process and enhancing accuracy through data-driven insights, this IDS provides an efficient, scalable, and robust layer of defense to protect critical computing infrastructure from evolving cyber threats.

Keywords: Intrusion Detection System (IDS), denial-of-service (DoS).

1-INTRODUCTION

The evolution of malicious software such as malware poses a critical challenge to the design of Intrusion Detection Systems (IDS). Over time, malicious attacks have become more sophisticated, and the foremost challenge is identifying unknown malware, as malware authors employ various techniques to conceal information and evade detection by IDS. Furthermore, there has been a significant increase in security threats, including zero-day attacks that specifically target internet users. As a result, computer security has become essential, especially since Information Technology (IT) has become an integral part of our day-to-day lives. With the continuous evolution of wireless communication, the internet has become more vulnerable to various security threats. IDS play a crucial role in identifying attacks on systems and detecting intrusions. In recent years, various machine learning techniques have been applied to IDS to enhance their ability to detect intruders and improve overall detection accuracy.

Existing system:

Traditional Network Intrusion Detection Systems (TNIDS) evolved from rule-based engines like Snort, which detect only known threats using predefined signatures. While effective against previously encountered attacks, these systems fail to identify novel or zero-day threats. To overcome this

limitation, modern NIDS incorporate machine learning techniques, leveraging labeled datasets such as KDD 99 and NSL-KDD to classify network traffic as either normal or malicious. This approach enables the detection of previously unseen attack patterns and enhances overall system adaptability. These systems provides both transparency and ease of deployment while still maintaining a high level of detection accuracy.

Proposed System:

Our proposed system is a Network-based Intrusion Detection System (NIDS) specifically designed to detect and classify a wide range of cyberattacks, including Denial of Service (DoS), Probe, Remote to Local (R2L), and User to Root (U2R) attacks. It functions by analyzing network traffic in real-time, focusing on packet header information and selected statistical features to identify abnormal or potentially malicious activity. The system leverages a machine learning model—specifically the Random Forest algorithm to learn from historical attack patterns and accurately classify network behavior as benign or malicious. By training on benchmark datasets like KDD Cup 99, the model gains the ability to generalize well to new, unseen threats, thereby enhancing detection accuracy and reducing false positives.

2-RELATED WORK

Over the years, a wide range of machine learning and hybrid approaches have been explored to strengthen the effectiveness of Network Intrusion Detection Systems (NIDS). The pioneering work by J.P.Anderson [1] introduced one of the first conceptual frameworks for threat monitoring and surveillance, laying a solid foundation for future research in intrusion detection. With the rapid evolution of cyber threats and network complexity, researchers began utilizing data mining and machine learning techniques to detect and classify intrusions. G.V. Nadiammai, S. Krishnaveni, and M. Hemalatha [2] conducted a detailed study on intrusion detection systems using data mining techniques, emphasizing the importance of feature selection and accurate classification for anomaly detection. A major advancement came with the development of the Random Forest algorithm by Leo Breiman [3], which provided a highly effective ensemble method capable of handling high-dimensional data with greater accuracy. Building on this, Arif Jamal Malik, Waseem Shahzad, and Farrukh Aslam Khan [4] proposed a hybrid model that integrated Binary Particle Swarm Optimization (PSO) with Random Forests to enhance detection performance through

intelligent feature selection. Another approach using ensemble methods was developed by P. Natesan and P. Balasubramanie [5], who implemented a multi-stage filter using an enhanced version of Adaboost to reduce false alarms while maintaining high accuracy in intrusion detection. Similarly, Mrutyunjaya Panda, Ajith Abraham, and Manas Ranjan Patra [6] proposed a hybrid intelligent method that combined different AI techniques for real-time detection of network attacks. Md. Al Mehedi Hasan, Mohammed Nasser, Biprodip, and Shamim Ahmad [7] focused on using both Support Vector Machine and Random Forest models to effectively model intrusion detection systems, demonstrating high precision in attack classification. Additionally, Ujwala Ravale, Nilesh Marathe, and Puja Padiya [8] introduced a feature selection-based hybrid anomaly detection model that used K-Means clustering along with a Radial Basis Function (RBF) kernel to identify and isolate unusual network behaviour. The architectural design and conceptual modelling of IDS were examined by D. Powell and R. Stroud [9], who emphasized the layered approach and the need for scalable, modular designs to handle dynamic network environments. Moreover, resources like IEEE Xplore [10] and Research Gate [11] have become essential platforms for accessing the latest advancements in IDS research.

3. REQUIREMENT ANALYSIS

Functional Requirements:

These define the core operations that a machine-learning-based Network Intrusion Detection System (NIDS) must perform to ensure accurate detection, robust performance, and secure handling of network data.

- Data Collection System
- Feature Extraction Module
- Data Preprocessing Module
- ML Model Training Engine
- Intrusion Classification Module
- Evaluation and Metrics Dashboard
- Model Storage and Reusability
- Security and Access Control

3.2 Non-Functional Requirements:

Secure Non-Functional requirements define the quality attributes and performance expectations of the system. For the ML-based NIDS project, the key Non-Functional Requirements include:

4.1 Architecture:

4.1.1 System Architecture:

The system architecture of the proposed Machine Learning-based Network Intrusion Detection System (NIDS) is designed to facilitate the efficient detection of cyber threats through a series of integrated components working in a sequential pipeline. The process begins with the user, typically a network administrator or security analyst, who

3.2.1 Availability

- The system should maintain high availability during both training and classification phases.
- It must be accessible during specified operational hours or provide uptime of at least 99% in production environments.
- Backup and recovery mechanisms should be in place to restore the model and data in case of failure.

3.2.2 Security

- All network traffic data must be securely handled, stored, and transmitted.
- Access to sensitive components like datasets and trained models must be role-based and password-protected.

3.2.3 Scalability

- The system should support increasing volumes of network traffic data without major performance degradation.
- It must allow easy integration with cloud-based solutions for future scalability (e.g., AWS, GCP, or Azure).

Software Requirements

Operating System	:
Windows	
Frontend	:
HTML, CSS, JavaScript	
Dataset	:
KDD CUP 99	
Backend	:
Python	
Web Development Framework	:
Django	

3.3.2 Hardware Requirements:

Processor	:
Intel i5	
RAM	:
8GB	
Storage	:
250GB SSD	
System-type	:
64-bit/32-bit OS	

4. DESIGN

initiates interaction with the system by logging into the detection tool via a secure interface. This login ensures that only authorized users can access and operate the detection environment, maintaining data confidentiality and system integrity.

Once authenticated, the user is prompted to provide necessary input data required for detection. This data can either be pre-recorded network traffic, such as datasets like KDD Cup 99, or live traffic captured

from a network interface. The provided input typically includes various network parameters such as duration, protocol type, source and destination ports, and service types. After the data is submitted, the user triggers the detection process. At this stage, the system begins preprocessing the data — performing steps like cleaning, normalization, and encoding — to prepare it for analysis.

Subsequently, the pre-processed data is passed into the machine learning pipeline, where a pre-trained Random Forest classifier is employed. This model, trained on labelled intrusion datasets, analyzes the input features and classifies the traffic either as normal or as one of several predefined attack

categories including DoS (Denial of Service), Probe, R2L (Remote to Local), and U2R (User to Root). The model leverages its trained decision trees to make high-accuracy predictions based on learned patterns of malicious behaviour. Once classification is complete, the system displays the results to the user, including the predicted class of the network traffic and key evaluation metrics such as accuracy, precision, recall, and confusion matrix. This feedback allows users to interpret detection results effectively and take necessary action to secure the network. Overall, this architecture offers a robust and user-friendly environment for efficient and accurate network intrusion detection using machine learning.

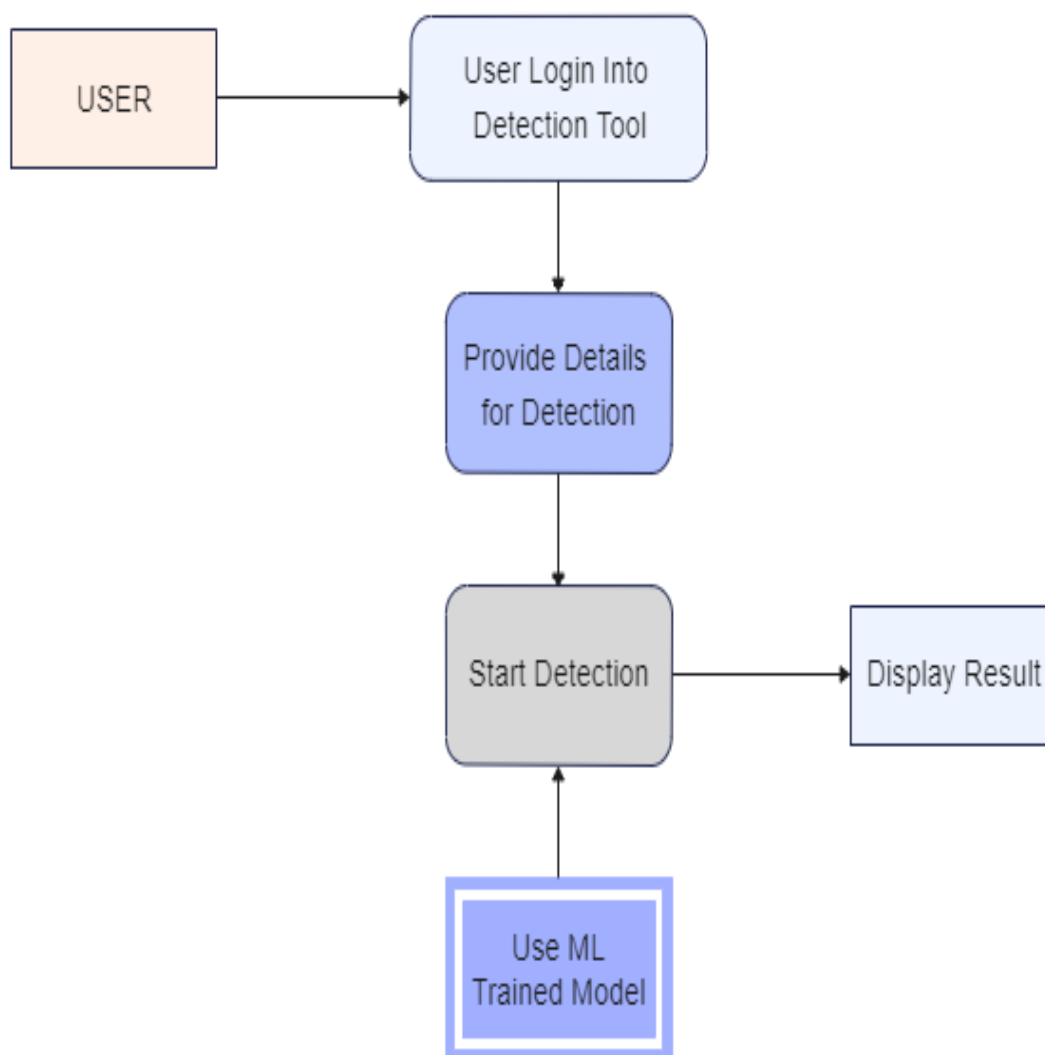


Fig. 4.1.1.1 System Architecture

4

1.2 Technical Architecture:

The Machine Learning-based Network Intrusion Detection System (NIDS) follows a layered, modular architecture for scalability and performance. At the top is the User Interface, built using HTML, CSS, and JavaScript, allowing users to upload network traffic data and view results. This front end communicates with the Web Server, developed using Django, which processes user requests, handles logic execution, and returns detection results. The core Backend Logic Layer, primarily written in Python, carries out feature processing, model training, and prediction using libraries like pandas, NumPy, scikit-learn, and joblib. The system uses datasets like KDD

Cup 99 to train machine learning models such as Random Forests for detecting potential intrusions. The backend also connects with databases such as MySQL and MongoDB to manage user data, detection logs, and model parameters. MySQL suits structured relational data like user details, while MongoDB handles semi-structured logs. All components run on the Operating System (typically Windows), which provides the necessary resources for smooth execution. This architecture ensures efficient real-time detection, robust data handling, and secure interactions, enabling effective deployment of the system in real-world network environments.

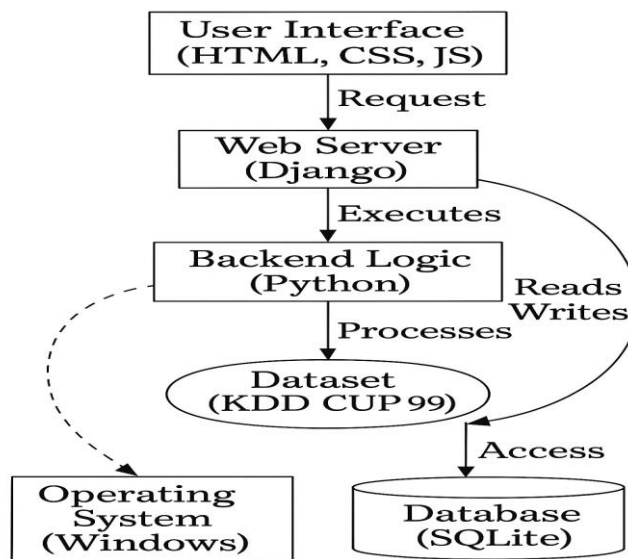


Fig. 4.1.2.1 Technical Architecture

5. IMPLEMENTATION

5.1 Libraries

5.1.1 joblib

joblib is a Python library used to efficiently save and load machine learning models and large data structures. In your project, it is used to load the pre-trained Random Forest model (model.h5) so that the intrusion detection system can classify new data without retraining the model each time. It is especially useful when working with large datasets or complex models due to its optimized performance compared to traditional pickle.

5.1.2 numpy

NumPy (Numerical Python) is a fundamental library for scientific computing that provides support for arrays and numerical operations. In this project, it is used for handling feature vectors, performing array-based computations, and feeding input data to the ML model. Its high-speed performance and array manipulation features make it essential for machine learning tasks.

5.1.3 pandas

pandas is a powerful data analysis library that offers data structures like Series and DataFrame for managing structured data. In your intrusion detection project, it is used to load and preprocess the KDD Cup 99 dataset, clean the data, select relevant columns, and prepare it for model input. It simplifies handling large volumes of tabular data for analysis and training.

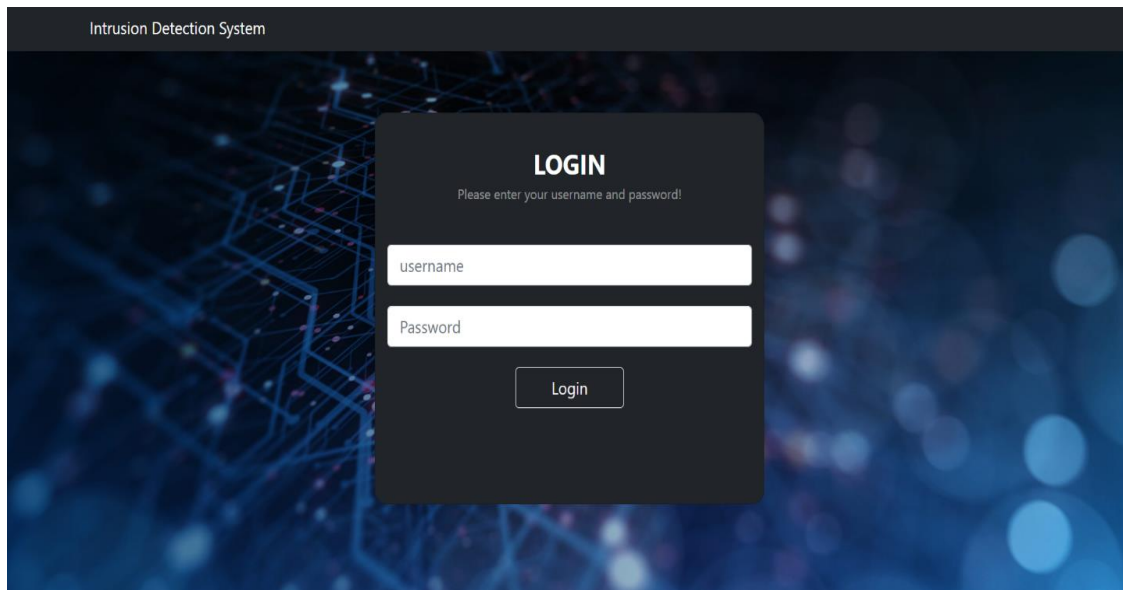
5.1.4 matplotlib.pyplot

matplotlib.pyplot is a data visualization library in Python that allows for the creation of static, animated, and interactive plots. In your project, it is used to generate plots such as bar graphs or accuracy charts, helping to visualize model performance, distribution of attack types, and evaluation metrics. These visual outputs make it easier to interpret results.

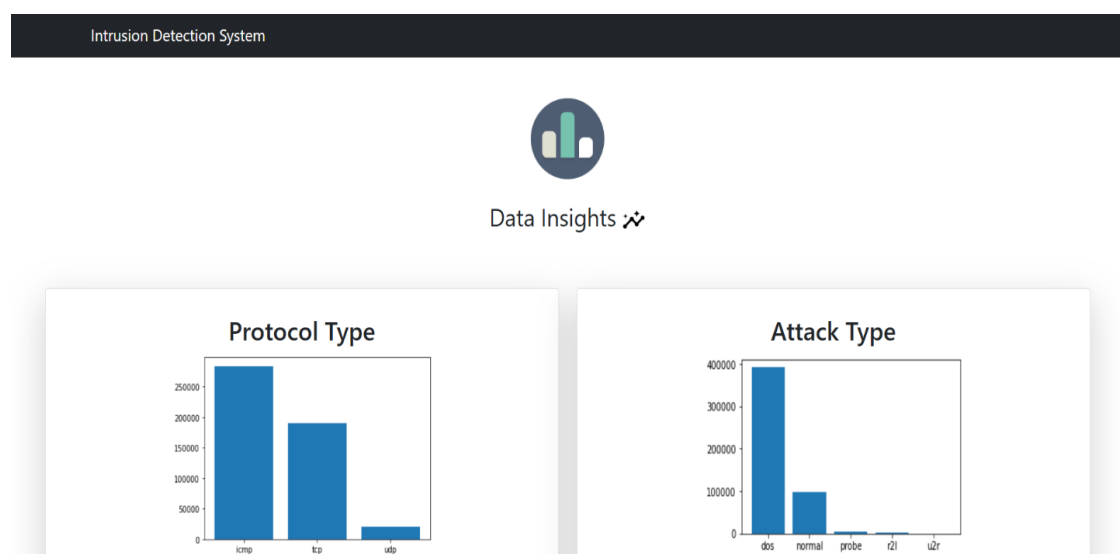
5.1.5 seaborn

seaborn is a statistical data visualization library built on top of Matplotlib, designed to create attractive and informative graphics. It is used in your project to visualize confusion matrices and other evaluation results in a more aesthetically pleasing and readable format.

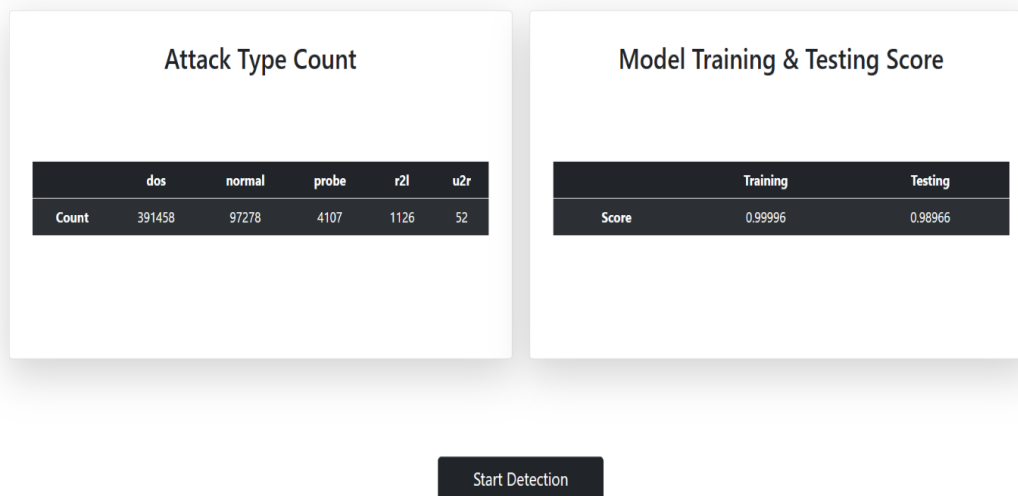
6. SCREENSHOTS



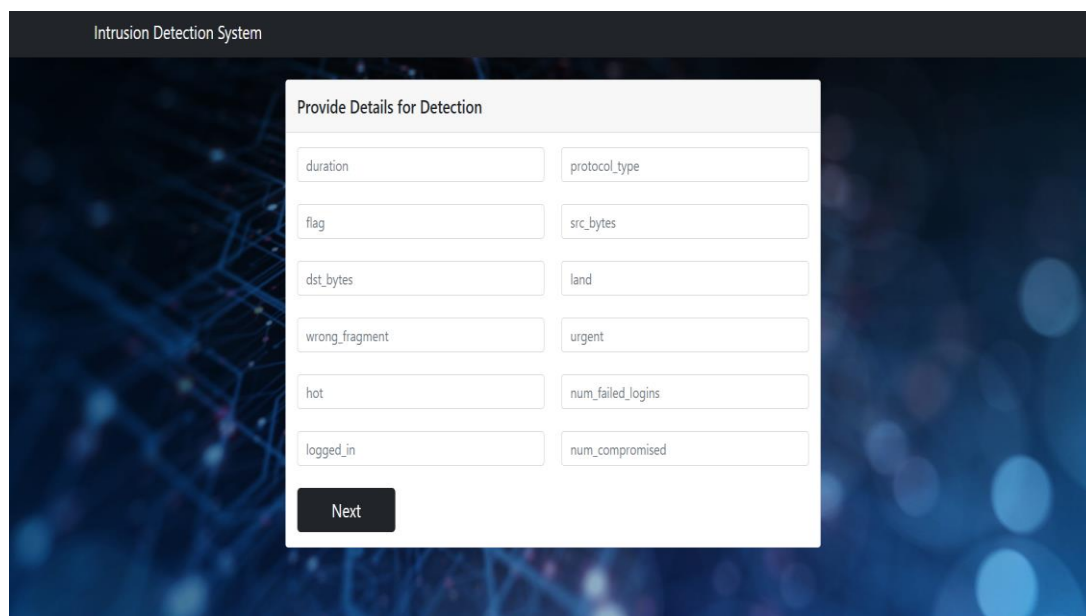
Screenshot 6.1 Login Page



Screenshot 6.2 Data Insights



Screenshot 6.3 Evaluation Page



The screenshot shows a web interface for an 'Intrusion Detection System'. It features a dark header with the title. Below the header is a large, semi-transparent blue network diagram. Overlaid on this is a white form titled 'Provide Details for Detection'. The form contains ten input fields arranged in two columns, each with a label: 'duration', 'protocol_type', 'flag', 'src_bytes', 'dst_bytes', 'land', 'wrong_fragment', 'urgent', 'hot', 'num_failed_logins', 'logged_in', and 'num_compromised'. A dark 'Next' button is located at the bottom of the form.

Intrusion Detection System

Provide Details for Detection

duration protocol_type

flag src_bytes

dst_bytes land

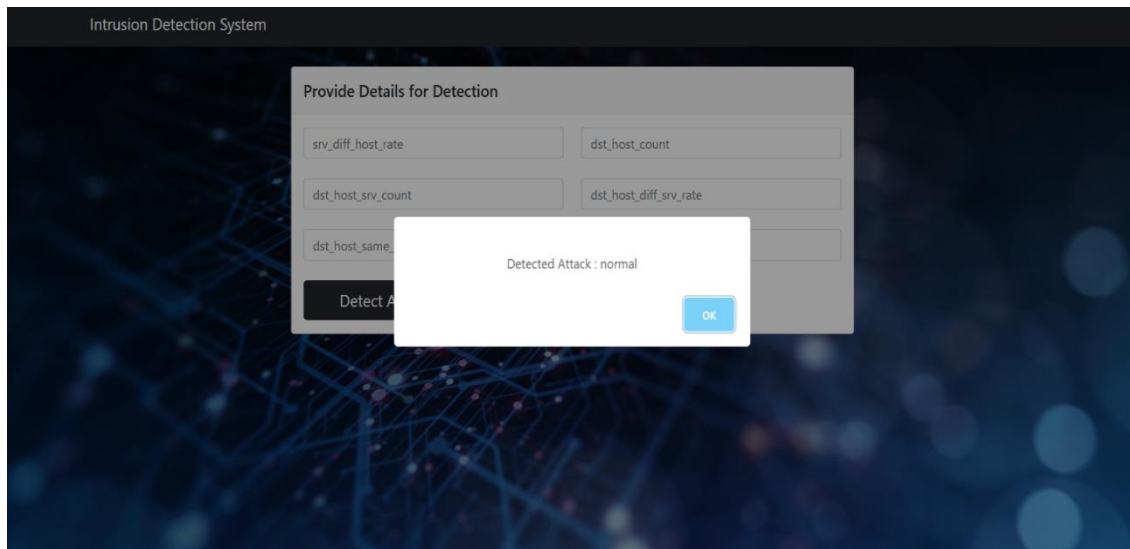
wrong_fragment urgent

hot num_failed_logins

logged_in num_compromised

Next

Screenshot 6.4 Input page



Screenshot 6.5 Final Output

7. CONCLUSION

In conclusion, this Intrusion Detection System (IDS) project serves as an effective solution for identifying and classifying various types of network-based attacks. By leveraging the KDD Cup 99 dataset and implementing the Random Forest algorithm, our system achieves reliable and accurate detection of five major intrusion categories: DoS, Probe, R2L, U2R, and Normal traffic. The architecture is structured to handle Pre-processed offline data and provides meaningful insights into network behaviour. The user interface is designed to be intuitive, responsive, and user-friendly, making the system accessible to both technical and non-technical users. Overall, this project not only demonstrates the practical application of machine learning in the domain of network security but also lays a robust foundation for further enhancement. This IDS represents a significant step toward developing efficient, data-driven security mechanisms in today's digitally connected world.

REFERENCES

- [1] J. P. Anderson, Computer Security Threat Monitoring and Surveillance, Technical Report, James Anderson Report, Pennsylvania, (1980).
- [2] G. V. Nadiammal, S. Krishnaveni and M. Hemalatha, A Comprehensive Analysis and Study in IDS Using Data Mining Techniques, IJCA, vol. 35, pp. 51–56, November–December (2011).
- [3] L. Breiman, Random Forests, Machine Learning, vol. 45, no. 1, pp. 5–32, (2001).
- [4] Arif Jamal Malik, Waseem Shahzad and Farrukh Aslam Khan, Network Intrusion Detection Using Hybrid Binary PSO and Random Forests Algorithm, Security and Communication Networks, (2012).
- [5] P. Natesan and P. Balasubramanie, Multi Stage Filter Using Enhanced Adaboost for Network IDS, International Journal of Network Security and its Applications, vol. 4, no. 3, (2012).
- [6] Mrutyunjaya Panda, Ajith Abraham and Manas Ranjan Patra, A Hybrid Intelligent Approach for Network Intrusion Detection, UCCTSD, pp. 1–9, (2012).
- [7] Md. Al Mehedi Hasan, Mohammed Nasser, Biprodip and Shamim Ahmad, Support Vector Machine and Random Forest Modeling for IDS, JILSA, pp. 45–52, (2014).
- [8] Ujwala Ravale, Nilesh Marathe and Puja Padiya, Feature Selection Based Hybrid Anomaly Intrusion Detection System Using K Means and RBF Kernel Function, ICACTA, pp. 428–435, (2015).
- [9] D. Powell and R. Stroud, Conceptual Model and Architecture, IBM Zurich Laboratory Report RZ 3377, November (2001).
- [10] IEEE Xplore – Intrusion Detection Research Papers <https://ieeexplore.ieee.org/>
- [11] ResearchGate –Intrusion Detection Using Machine Learning <https://www.researchgate.net/>