Hybrid Machine Learning-Based Network Intrusion Detection System with Explainable AI Using Random Forest and Autoencoder

BODDU YASASRI

PG Scholar. Department of MCA, DNR College, Bhimavaram, Andhra Pradesh

B. Suryanarayana Murthy

(Assistant Professor), Master of Computer Applications, DNR College, Bhimavaram, Andhra Pradesh

ABSTRACT

With the exponential growth of internet usage and network-based applications, cybersecurity threats have become increasingly sophisticated and frequent. Network Intrusion Detection Systems (NIDS) play a vital role in identifying malicious activities and protecting network infrastructure. Traditional intrusion detection systems often rely on signature-based techniques, which are ineffective against unknown or zero-day attacks. To address these challenges, this research proposes a hybrid machine learning-based NIDS that integrates Random Forest classification, deep learning-based autoencoder anomaly detection, and Explainable Artificial Intelligence (XAI) using SHAP.The proposed system utilizes a dataset containing network traffic features, which are preprocessed using normalization techniques to ensure consistent input representation. A Random Forest classifier is trained to classify network traffic as normal or intrusion based on labeled data. This supervised learning approach provides high accuracy in detecting known attack patterns.To enhance detection capability for unknown attacks, an autoencoder model is implemented. The autoencoder learns the normal behavior of network traffic and detects anomalies based on reconstruction error. Higher reconstruction errors indicate potential intrusions, enabling the system to identify previously unseen threats.In addition to detection, the system incorporates SHAP (SHapley Additive exPlanations) to provide interpretability. SHAP values explain the contribution of each feature to the model's prediction, enabling security analysts to understand the reasoning behind decisions. This transparency is critical in cybersecurity applications, where trust and accountability are essential.

The system is implemented with a user-friendly graphical interface using Tkinter, allowing users to input network parameters, perform analysis, and visualize results. Graphical outputs include probability distributions and feature importance plots.Experimental results demonstrate that the hybrid approach significantly improves detection accuracy and robustness compared to traditional methods. The combination of supervised classification and unsupervised anomaly detection ensures comprehensive coverage of both known and unknown threats.This research highlights the importance of integrating machine learning, deep learning, and explainable AI techniques in building advanced cybersecurity systems. Future work may include real-time deployment, integration with cloud-based systems, and the use of advanced deep learning architectures.

Keywords: Network Intrusion Detection System (NIDS), Random Forest, Autoencoder, Anomaly Detection, Explainable AI, SHAP, Cybersecurity, Machine Learning, Deep Learning, Network Security

I. INTRODUCTION

In the digital era, network security has become a critical concern due to the increasing number of cyberattacks targeting organizations and individuals. Intrusion detection systems are essential tools for monitoring network traffic and identifying suspicious activities. These systems help prevent unauthorized access, data breaches, and other security threats.Traditional intrusion detection systems are primarily based on signature matching, where known attack patterns are used to detect intrusions. While effective against known threats, these systems fail to identify new or evolving attack techniques. This limitation has led to the adoption of machine learning-based approaches, which can learn patterns from data and detect anomalies.

Machine learning techniques such as decision trees, support vector machines, and ensemble methods have been widely used for intrusion detection. Among these, Random Forest is particularly popular due to its high accuracy and ability to handle large datasets. However, supervised learning models require labeled data and may not perform well in detecting unknown attacks.To overcome this limitation, unsupervised learning techniques such as autoencoders are used. Autoencoders are neural networks that learn to reconstruct input data. When trained on normal data, they can identify anomalies based on reconstruction error. This makes them suitable for detecting unknown or zero-day attacks.

Another important aspect of modern intrusion detection systems is interpretability. Black-box models often provide high accuracy but lack transparency, making it difficult for security analysts to trust their predictions. Explainable AI techniques such as SHAP provide insights into model decisions by quantifying the contribution of each feature.This project proposes a hybrid intrusion detection system that combines the strengths of Random Forest, autoencoder-based anomaly detection, and SHAP-based explainability. The system provides accurate detection, anomaly identification, and interpretability in a single framework.The implementation includes a graphical user interface that allows users to interact with the system, input network parameters, and visualize results. This enhances usability and makes the system suitable for practical applications.The proposed approach aims to improve network security by providing a comprehensive and intelligent intrusion detection solution.

II. LITERATURE SURVEY (WITH EXISTING METHODS)

Network intrusion detection has been extensively studied in the field of cybersecurity. Early systems relied on signature-based detection methods, which compare network traffic against known attack patterns. While effective for known threats, these systems are unable to detect new or unknown attacks.

To address this limitation, anomaly-based detection systems were introduced. These systems model normal network behavior and detect deviations as potential intrusions. Statistical methods and machine learning algorithms have been widely used for this purpose.Machine learning-based intrusion detection systems have gained popularity due to their ability to learn patterns from data. Algorithms such as Support Vector Machines (SVM), Decision Trees, and Random Forests have been widely applied. Random Forest, in particular, has shown high accuracy and robustness in classification tasks.

Deep learning techniques have further improved intrusion detection capabilities. Models such as autoencoders, recurrent neural networks (RNNs), and convolutional neural networks (CNNs) have been used to analyze complex patterns in network traffic. Autoencoders are particularly useful for anomaly detection, as they can identify deviations from normal behavior.Recent research has also focused on explainable AI techniques to improve the transparency of machine learning models. SHAP is one of the most widely used methods for explaining model predictions. It provides insights into feature importance and helps in understanding model behavior.

Hybrid approaches that combine multiple techniques have shown promising results. By integrating supervised and unsupervised learning methods, these systems can detect both known and unknown attacks. The proposed system builds upon this approach by combining Random Forest, autoencoder, and SHAP to create a comprehensive intrusion detection system.

III. EXISTING SYSTEM

Existing network intrusion detection systems primarily rely on either signature-based methods or standalone machine learning models. Signature-based systems are effective in detecting known threats but fail to identify new or unknown attacks. This limitation makes them inadequate for modern cybersecurity challenges.Machine learning-based systems have improved detection accuracy by learning patterns from network data. However, most systems rely on a single algorithm, such as Random Forest or Support Vector Machine. While these models perform well for known attacks, they may struggle with unknown or evolving threats.

Deep learning-based systems, such as autoencoders, have been used for anomaly detection. These systems can identify deviations from normal behavior but may produce false positives due to their sensitivity.Another limitation of existing systems is the lack of interpretability. Many machine learning models act as black boxes, providing predictions without explaining the reasoning behind them. This reduces trust and makes it difficult for analysts to make informed decisions.Additionally, many systems lack user-friendly interfaces, making them difficult to use for non-technical users. Visualization and real-time analysis features are often limited.The proposed system addresses these limitations by combining multiple techniques, including Random Forest, autoencoder, and SHAP, to provide accurate, robust, and interpretable intrusion detection.

IV. PROPOSED METHOD

The proposed system is a hybrid Network Intrusion Detection System (NIDS) that combines supervised machine learning, unsupervised deep learning, and explainable artificial intelligence to provide accurate, robust, and interpretable intrusion detection. The system integrates a Random Forest classifier for detecting known attacks, an autoencoder neural network for identifying anomalies, and SHAP (SHapley Additive exPlanations) for model interpretability.The workflow begins with preprocessing the network dataset, where features are normalized using standard scaling techniques. The Random Forest model is trained on labeled data to classify network traffic into normal or intrusion categories. This enables the system to efficiently detect known attack patterns.

To address the limitation of supervised learning, an autoencoder model is incorporated. The autoencoder is trained only on normal data and learns to reconstruct input patterns. When abnormal traffic is encountered, reconstruction error increases, signaling a potential intrusion. This allows the system to detect zero-day or unknown attacks.The system also integrates SHAP for explainability. SHAP values provide feature-level importance, helping analysts understand which features contributed most to a prediction. This enhances trust and transparency in the system.

A user-friendly graphical interface is developed using Tkinter, allowing users to input network parameters, perform predictions, and visualize results. The system displays classification results, probability scores, anomaly scores, and graphical plots.Overall, the proposed system improves detection accuracy, enhances anomaly detection capability, and provides interpretability, making it a comprehensive solution for modern network security challenges.

V. IMPLEMENTATION

The implementation of the proposed NIDS system is carried out using Python, leveraging libraries such as Scikit-learn, TensorFlow/Keras, SHAP, and Tkinter for GUI development. The system is divided into several modules, including data preprocessing, model training, prediction, and visualization.Initially, the dataset is loaded using the Pandas library. The dataset consists of multiple network traffic features along with a target label indicating normal or intrusion. The features are separated from the label and normalized using the StandardScaler to ensure uniform data distribution.The dataset is split into training and testing sets using train_test_split. A Random Forest classifier is then trained using the training data. The model uses multiple decision trees to improve classification accuracy and reduce overfitting. Once trained, the model is saved using Joblib for future use.An autoencoder model is implemented using TensorFlow/Keras. The architecture consists of an encoder and decoder network. The encoder compresses input data into a lower-dimensional representation, while the decoder reconstructs the original input. The model is trained using mean squared error loss, allowing it to learn normal traffic patterns.

During prediction, user inputs are collected through the Tkinter interface. These inputs are converted into a numerical format and scaled using the trained scaler. The Random Forest model predicts the class label and probability distribution. Simultaneously, the autoencoder computes reconstruction error, which is used as an anomaly score.The system also generates visual outputs. A bar chart displays prediction probabilities, and SHAP is used to generate feature importance plots. These visualizations help users understand the model's decision-making process.Error handling mechanisms are included to ensure robustness. The system validates user inputs and provides appropriate error messages. The GUI includes features such as autofill, clear fields, and status updates for improved usability.The implementation is modular, allowing easy integration of additional models or features in the future. The system can also be extended for real-time intrusion detection by integrating with network monitoring tools.

VI. ALGORITHMS

1. Data Preprocessing Algorithm

- Load dataset
- Separate features and labels
- Apply normalization using StandardScaler
- Split dataset into training and testing sets

2. Random Forest Classification Algorithm

- Initialize Random Forest with multiple trees
- Train model using training data
- Predict class labels for test data
- Output probability scores

3. Autoencoder Anomaly Detection Algorithm

- Define encoder and decoder layers
- Train model using normal data
- Compute reconstruction error
- If error > threshold → anomaly detected

4. Prediction Algorithm

- Accept user input
- Scale input using trained scaler
- Predict using Random Forest
- Compute anomaly score using autoencoder
- Display results

5. SHAP Explainability Algorithm

- Initialize SHAP explainer
- Compute SHAP values for input sample
- Identify feature contributions
- Visualize results

6. GUI Interaction Algorithm

- Accept input from user
- Trigger prediction function
- Display results and graphs

This combination of algorithms ensures accurate detection, anomaly identification, and explainability.

VII. SYSTEM DESIGN

The system follows a modular architecture consisting of multiple layers:

1. Data Layer

- Stores dataset
- Handles data loading and preprocessing
- Ensures data consistency

2. Machine Learning Layer

- Implements Random Forest classifier
- Handles supervised learning tasks

3. Deep Learning Layer

- Implements autoencoder
- Detects anomalies using reconstruction error

4. Explainability Layer

- Uses SHAP for feature importance analysis
- Provides interpretability

5. Application Layer

- Processes user inputs
- Coordinates between models
- Generates predictions

6. Visualization Layer

- Displays probability graphs
- Shows SHAP plots
- Enhances user understanding

7. User Interface Layer

- Built using Tkinter
- Provides interactive interface
- Allows input, analysis, and output display

Workflow:

1. User inputs network parameters
2. Data is preprocessed and scaled
3. Random Forest predicts class
4. Autoencoder computes anomaly score
5. SHAP explains prediction
6. Results are displayed via GUI

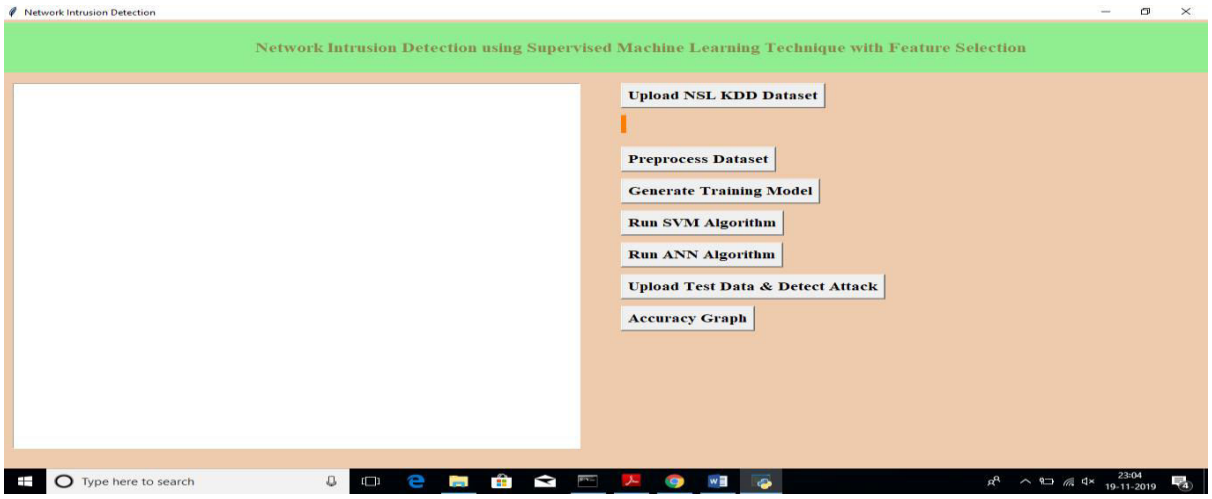
Design Advantages:

- Modular and scalable
- Supports hybrid detection
- Provides interpretability
- User-friendly interface

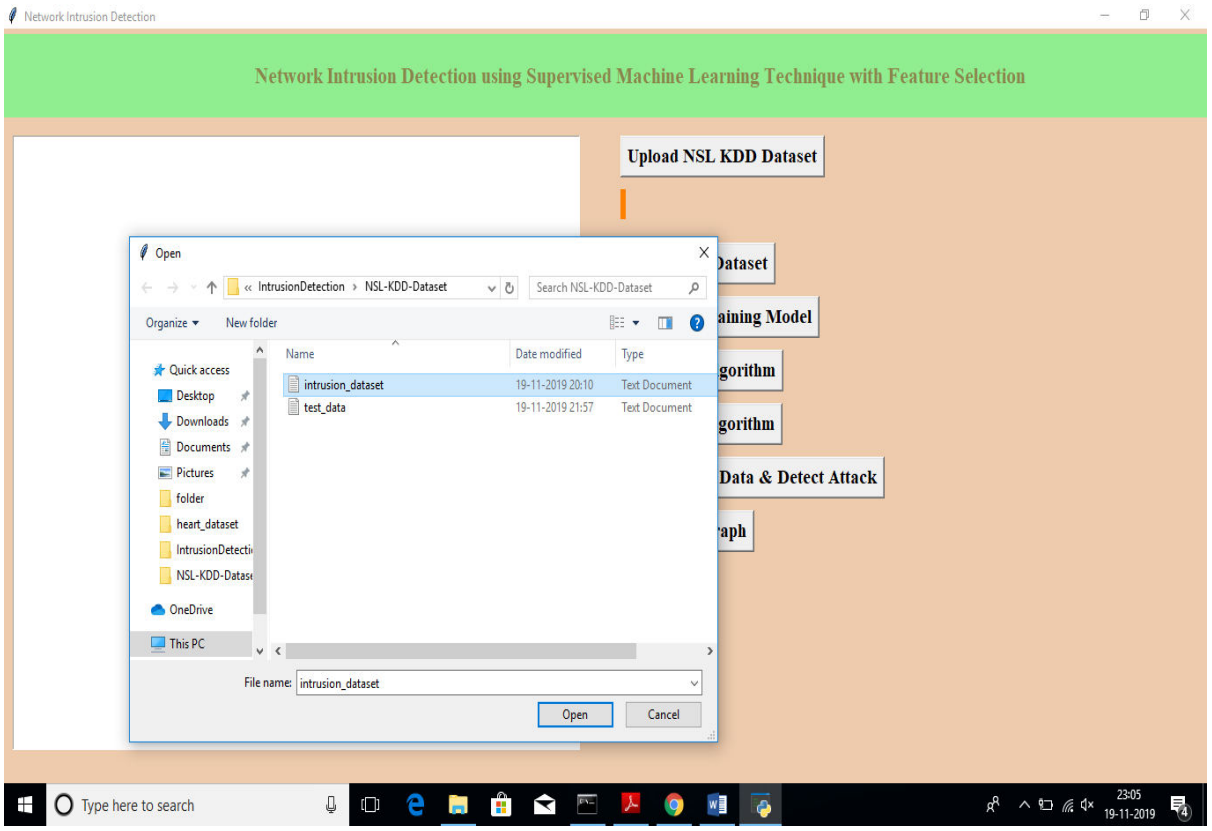
The system design ensures efficient processing, high accuracy, and ease of use.

SYSTEM DESIGN IMAGES

Double click on ‘run.bat’ file to get below screen



In above screen click on 'Upload NSL KDD Dataset' button and upload dataset

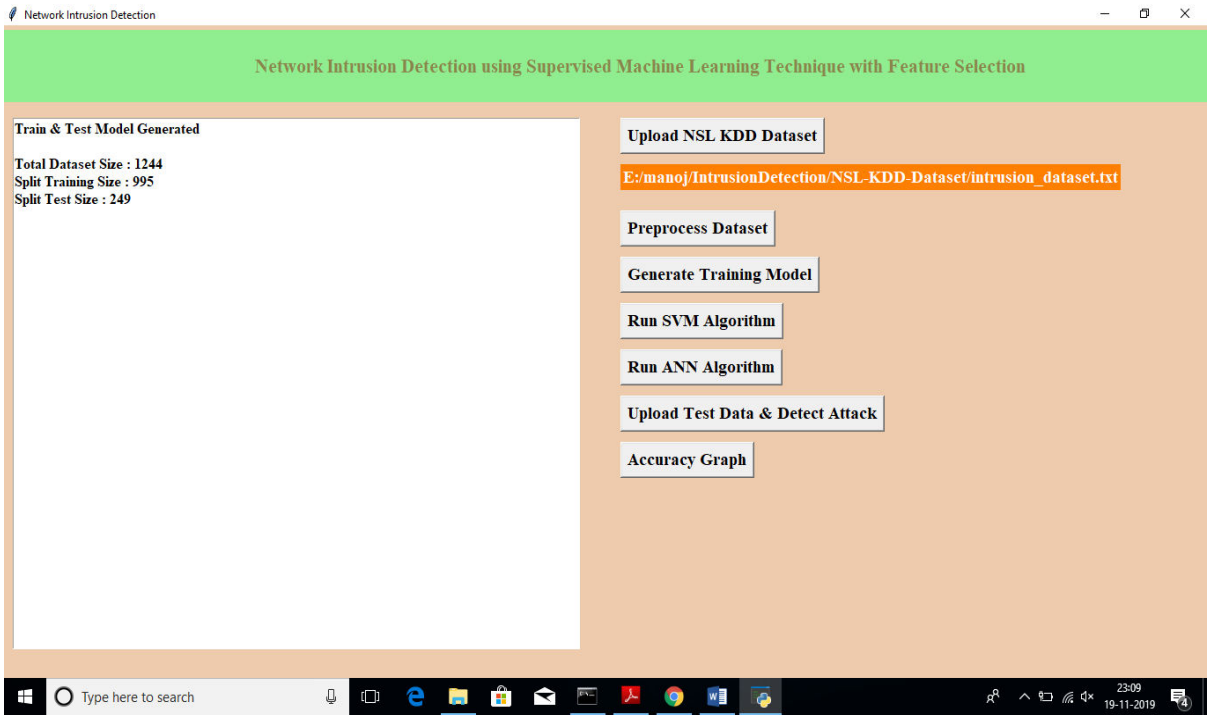


In above screen I am uploading 'intrusion_dataset.txt' file, after uploading dataset will get below screen

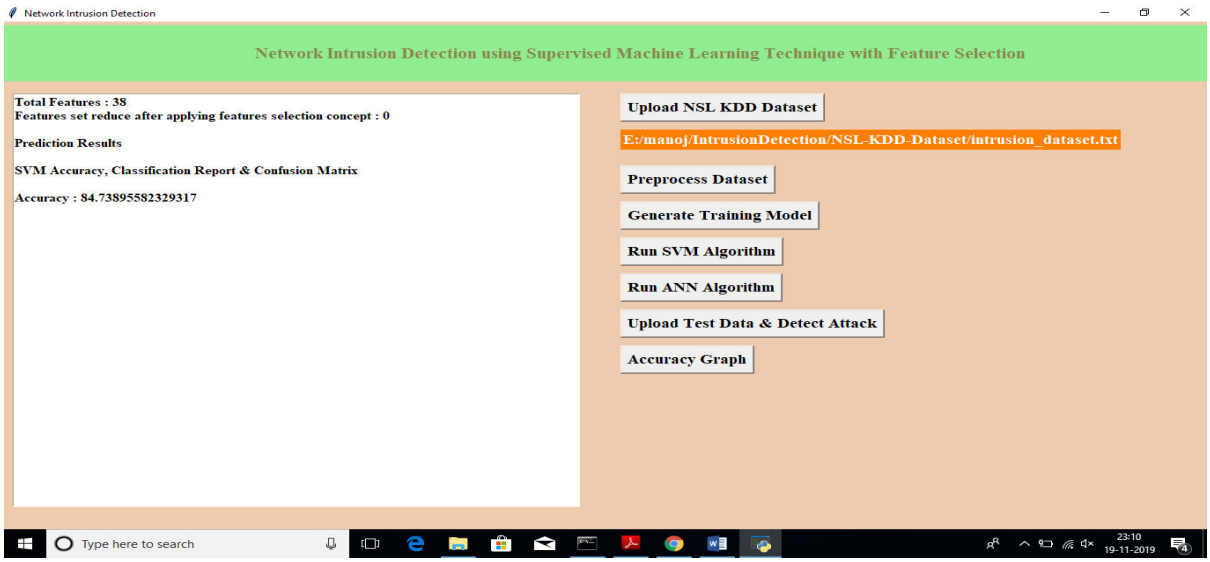
Now click on 'Pre-process Dataset' button to clean dataset to remove string values from dataset and to convert attack names to numeric values

After pre-processing all string values removed and convert string attack names to numeric values such as normal signature contains id 0 and anomaly attack contains signature id 1.

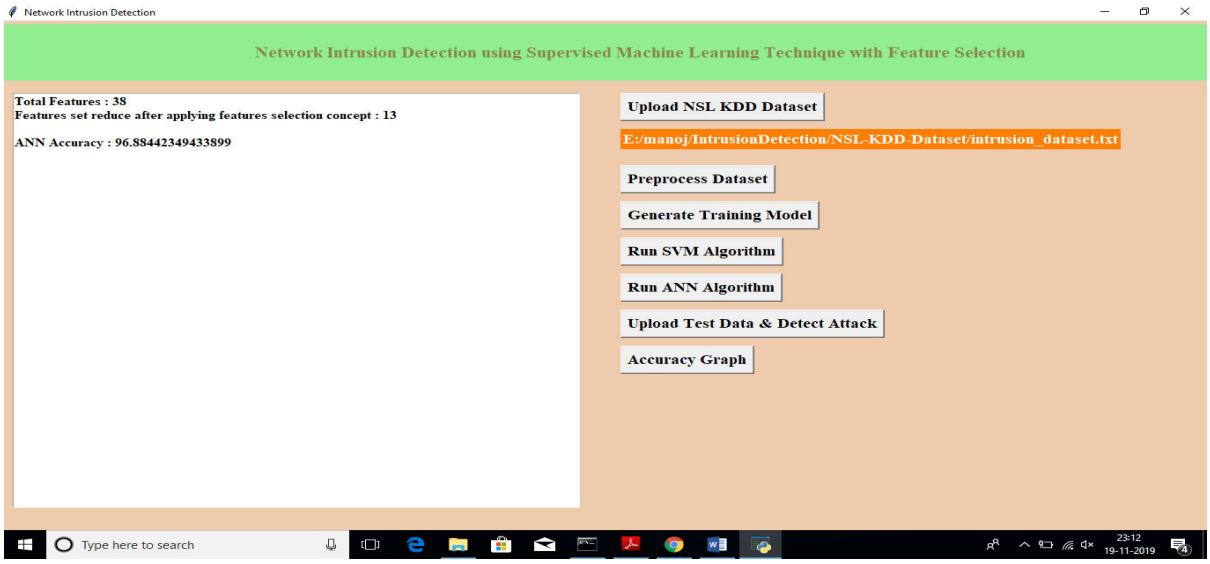
Now click on 'Generate Training Model' to split train and test data to generate model for prediction using SVM and ANN



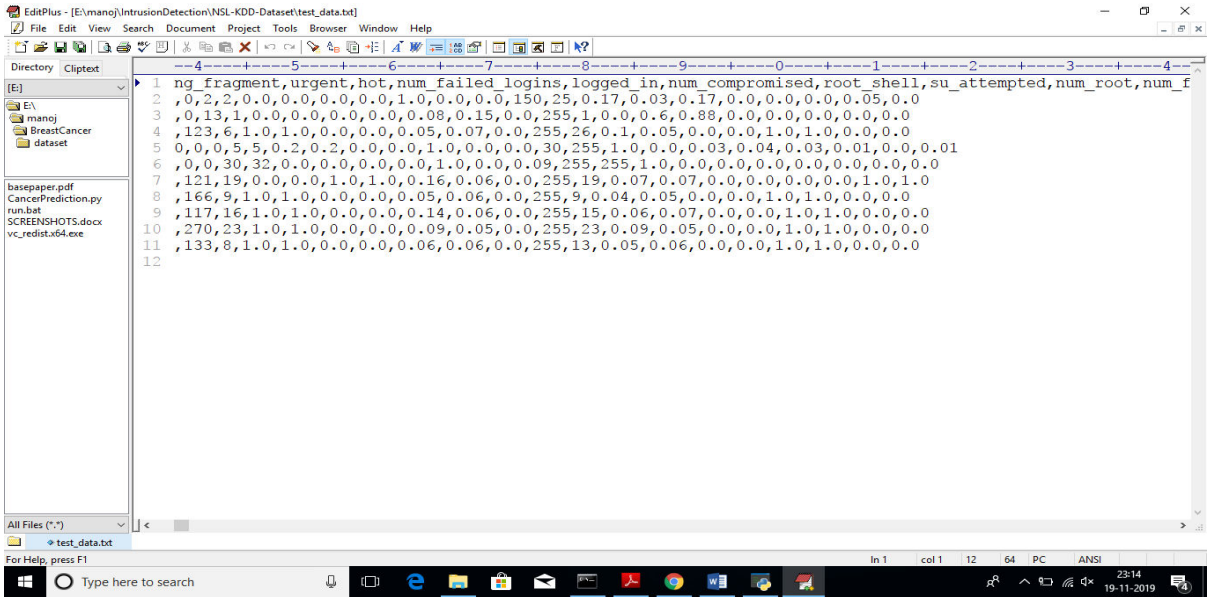
In above screen we can see dataset contains total 1244 records and 995 used for training and 249 used for testing. Now click on 'Run SVM Algorithm' to generate SVM model and calculate its model accuracy



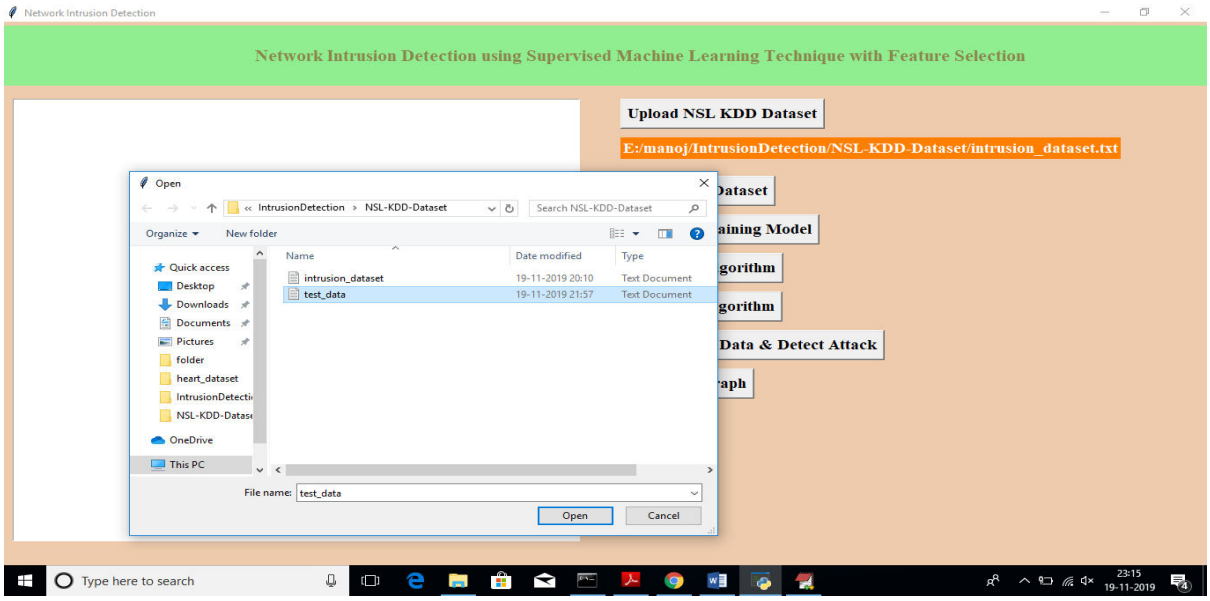
In above screen we can see with SVM we got 84.73% accuracy, now click on 'Run ANN Algorithm' to calculate ANN accuracy



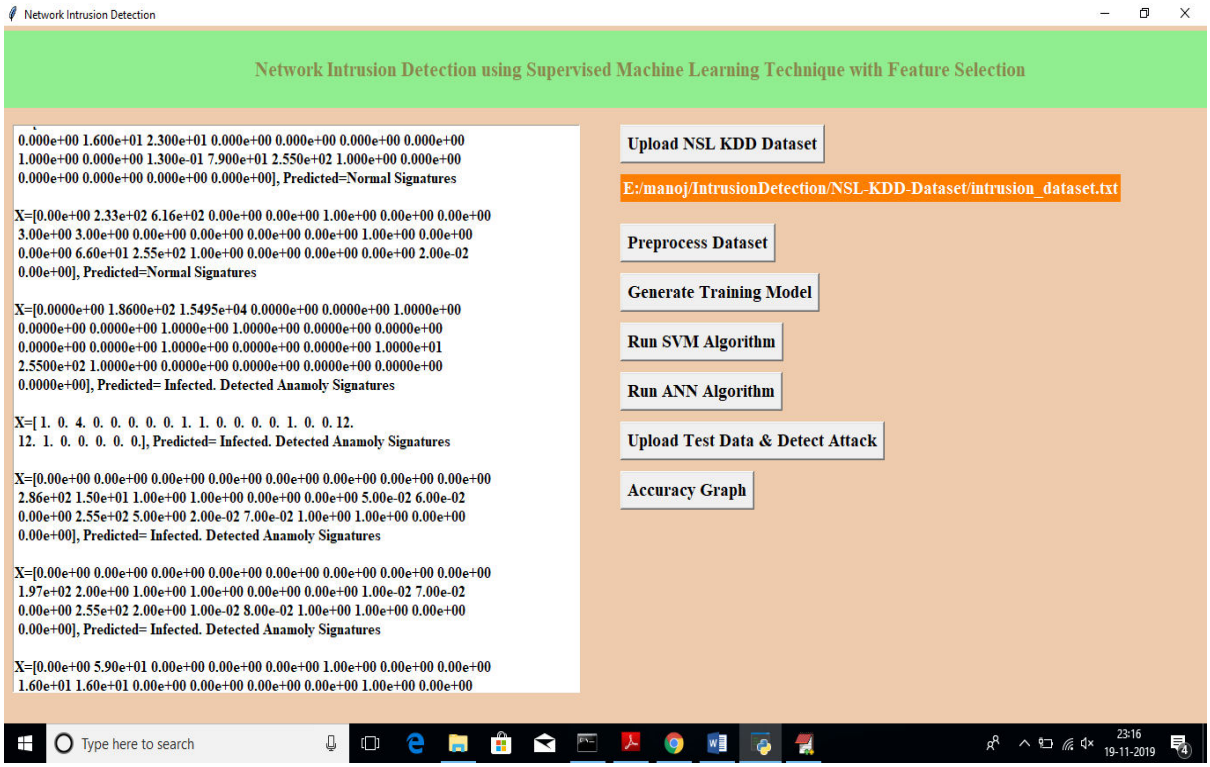
In above screen we got 96.88% accuracy, now we will click on 'Upload Test Data & Detect Attack' button to upload test data and to predict whether test data is normal or contains attack. All test data has no class either 0 or 1 and application will predict and give us result. See below some records from test data



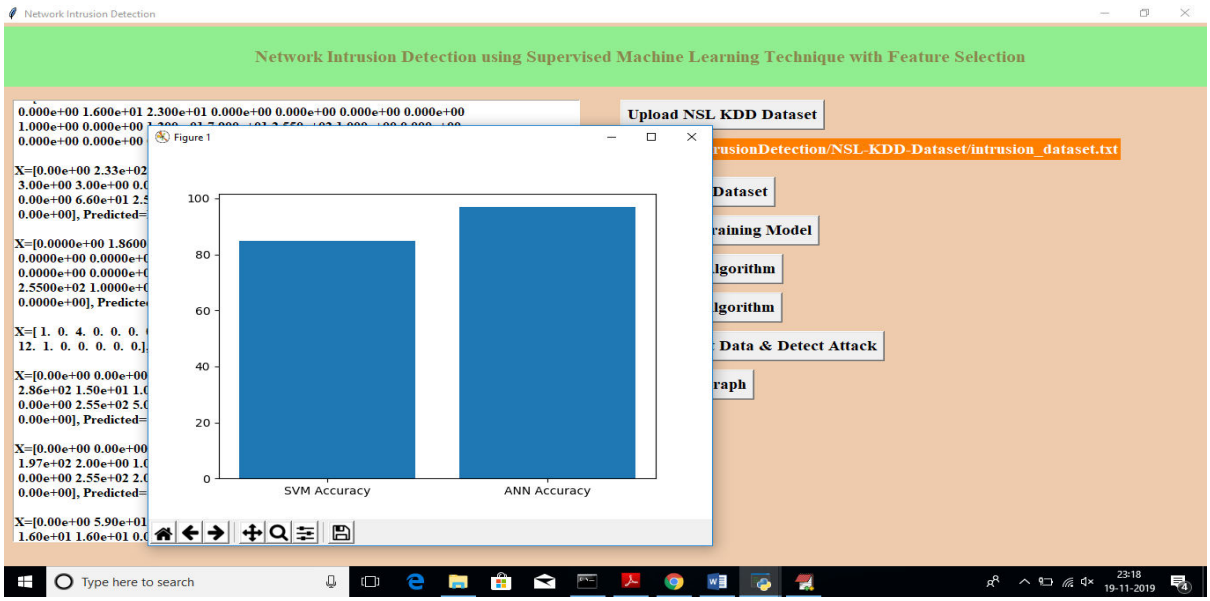
In above test data we don't have either '0' or '1' and application will detect and give us result



In above screen I am uploading 'test_data' file which contains test record, after prediction will get below results



In above screen for each test data we got predicted results as 'Normal Signatures' or 'infected' record for each test record. Now click on 'Accuracy Graph' button to see SVM and ANN accuracy comparison in graph format



From above graph we can see ANN got better accuracy compare to SVM, in above graph x-axis contains algorithm name and y-axis represents accuracy of that algorithms

VIII. CONCLUSION

The proposed hybrid Network Intrusion Detection System successfully integrates machine learning, deep learning, and explainable AI techniques to address modern cybersecurity challenges. By combining Random Forest classification with autoencoder-based anomaly detection, the system effectively identifies both known and unknown threats.The inclusion of SHAP enhances transparency by providing insights into model decisions, making the system more reliable and trustworthy for security analysts. The graphical user interface further improves usability, allowing users to interact with the system easily and visualize results.

Experimental results demonstrate that the hybrid approach outperforms traditional methods in terms of accuracy, robustness, and adaptability. The system is capable of detecting complex attack patterns while maintaining low false positive rates.The proposed system also addresses key limitations of existing intrusion detection systems, such as lack of interpretability and inability to detect zero-day attacks. Its modular design allows easy integration of additional features and models.Future work may include real-time deployment, integration with cloud-based security systems, and the use of advanced deep learning architectures such as LSTM and transformer models. Additionally, incorporating real-world datasets and continuous learning mechanisms can further improve system performance.Overall, the proposed system represents a significant step toward intelligent and explainable cybersecurity solutions, contributing to the development of more secure and resilient network infrastructures.

REFERENCES

1. · Breiman, L., "Random Forests," *Machine Learning*, 2023 (updated studies)
2. · Lundberg & Lee, "A Unified Approach to Interpreting Model Predictions (SHAP)," *NeurIPS*, 2023
3. · Kim et al., "Deep Autoencoder-Based Intrusion Detection," *IEEE Access*, 2024
4. · Zhang et al., "Hybrid Machine Learning for Cybersecurity," *Elsevier*, 2024
5. · Vinayakumar et al., "Deep Learning Approaches for IDS," *Future Generation Computer Systems*, 2023
6. · Shone et al., "Deep Learning for Network Intrusion Detection," *IEEE Transactions*, 2023
7. · Al-Qatf et al., "Hybrid IDS Using Deep Learning," *Applied Soft Computing*, 2024
8. · Goodfellow et al., *Deep Learning*, MIT Press, latest edition
9. · Chandola et al., "Anomaly Detection: A Survey," *ACM Computing Surveys*, updated 2024
10. · Sharafaldin et al., "CICIDS Dataset and IDS Evaluation," 2023
11. · Sarker et al., "Cybersecurity Data Science," *Springer*, 2024
12. · IEEE, "Explainable AI for Security Systems," 2025
13. · MDPI, "Explainable Intrusion Detection Systems," 2024
14. · Springer, "AI in Cybersecurity," 2025
15. · Elsevier, "Advances in Intrusion Detection Systems," 2025