CAPSTONE PROJECT

NETWORK INTRUSION DETECTION USING MACHINE LEARNING

Presented By:

1. Student Name: Darshini M S
2. College Name: Maharaja Institute of Technology Mysore
3. Department: Computer Science and Engineering

edunet foundation

OUTLINE

- Problem Statement
- Proposed System/Solution
- System Development Approach
- Algorithm & Deployment
- Result (Output Image)
- Conclusion
- Future Scope
- References

PROBLEM STATEMENT

- In today’s digital era, the increasing complexity and scale of communication networks have made highly vulnerable to a wide range of cyber-attacks such as Denial of Service (DoS), Probe, Remote to Local (R2L), and User to Root (U2R).
- Traditional security systems fail to detect advanced or unknown cyber-attacks like DoS, Probe, R2L, and U2R in modern communication networks
- There is a need for a Machine Learning-based Network Intrusion Detection System (NIDS) that can analyze network traffic and classify threats accurately in real-time to enhance network security.

PROPOSED SOLUTION

The proposed system aims to address the challenge of detecting and classifying malicious activities in network traffic using Machine Learning.

Data Collection:

- Used the publicly available Kaggle dataset for network intrusion detection, which includes labeled traffic samples to identify attack types such as DoS, Probe, R2L, and U2R.
- The dataset includes various network traffic features like protocol type, service, flag, duration, bytes transferred, and more.

Data Preprocessing:

- Handle missing or inconsistent values, and encode categorical features using label or ordinal encoding.
- Map attack labels into broader categories like DoS, Probe, R2L, U2R, or Normal.

Machine Learning Algorithm:

- Train classification models such as Random Forest, XGBoost, or Logistic Regression to identify attack types.
- Address data imbalance using SMOTE or other sampling techniques to improve classification of minority attacks.

Deployment:

- Build a user interface using Streamlit or Flask to visualize data and show predictions.
- Host the entire pipeline in IBM Watsonx.ai Studio using Jupyter Notebook on IBM Cloud Lite.

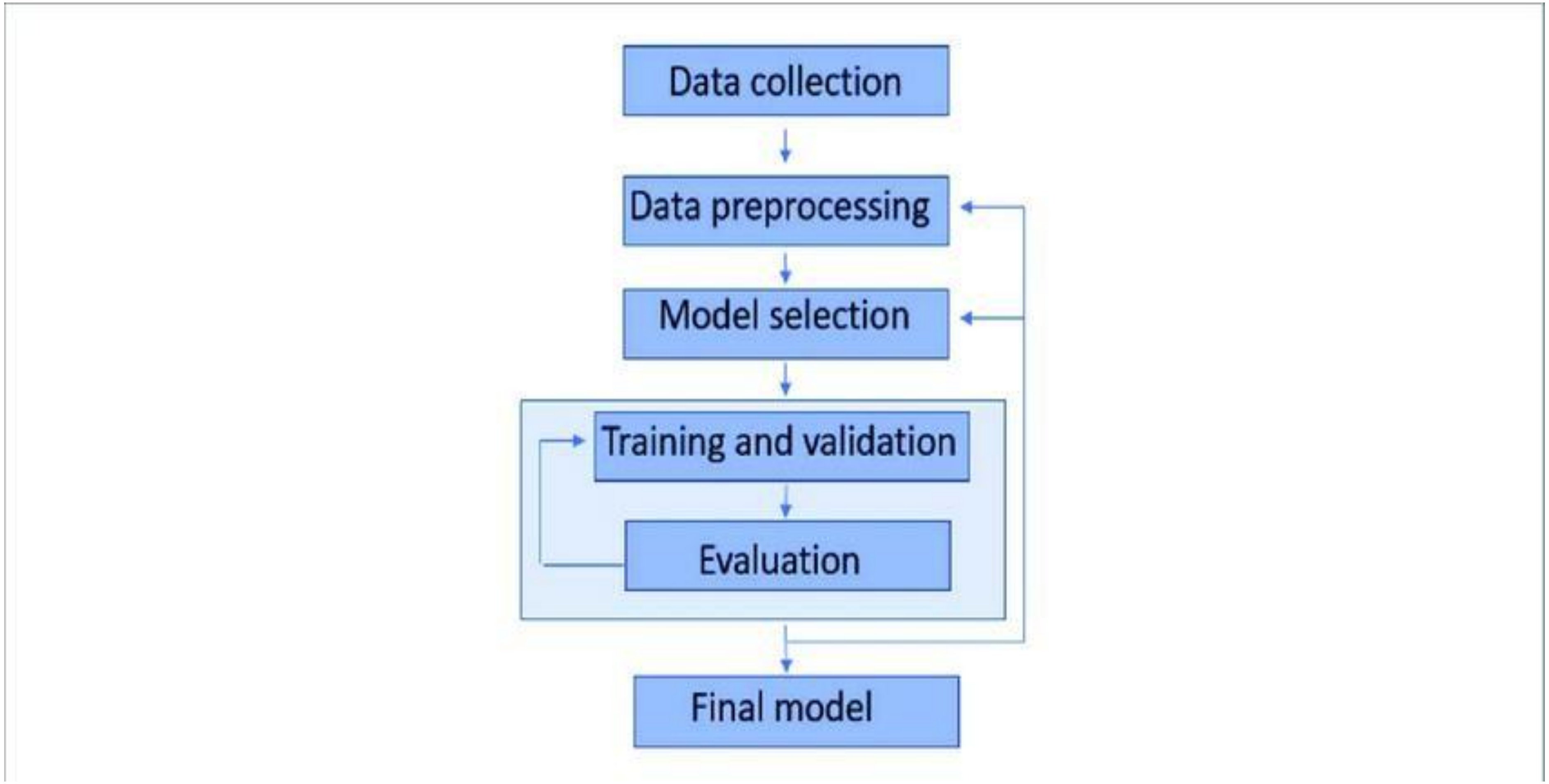
Evaluation:

- Evaluate model performance using metrics such as Accuracy, Precision, Recall, F1-Score, and visualize confusion matrix, ROC curve, and feature importance.
- Continuously improve model performance through validation and tuning.

Result:

- The final system classifies network traffic into normal or various attack types, providing early alerts to mitigate threats and enhance cybersecurity.

FLOW CHART:-


Data collection
Data preprocessing
Model selection
Training and validation
Evaluation
Final model

SYSTEM APPROACH

System Requirements:

- IBM Cloud Lite Services for running the notebook and deploying the model.
- Watson Studio for model development, training, and visualization.
- Cloud Object Storage for storing datasets and model files.
- Hardware Access: Cloud-based, no installation required – runs on IBM infrastructure.
- Internet Requirement: Stable internet for cloud access and real-time dashboard interaction.

Libraries Required to Build the Model:

- pandas – For data handling and preprocessing
- numpy – For efficient numerical operations
- matplotlib, seaborn – For data visualization
- scikit-learn – For ML models and evaluation metrics
- xgboost – For high-performance gradient boosting algorithm
- imbalanced-learn – To handle class imbalance using SMOTE
- pickle – For model serialization and deployment

ALGORITHM & DEPLOYMENT

Algorithm Selection:

- We selected XGBoost, a powerful ensemble learning algorithm based on gradient boosting, due to its robustness in handling high-dimensional data, missing values, and imbalanced classes. It delivers high performance and is widely used in intrusion detection tasks. Additionally, Random Forest was used for baseline comparison.

Data Input:

- The model uses the following key features extracted from network traffic:
- Protocol type, service, and flag
- Source and destination bytes
- Count-based features (e.g., srv_count, dst_host_srv_count)
- Connection-level indicators (e.g., logged_in, is_guest_login)
- Attack labels categorized as: normal, DoS, Probe, R2L, U2R

Training Process:

- Categorical features were encoded using Ordinal Encoding
- Class imbalance was addressed using SMOTE (Synthetic Minority Oversampling)
- The model was trained using labeled historical data from the Kaggle NIDS dataset
- Hyperparameters were fine-tuned for better generalization and accuracy

Prediction Process:

- The trained model predicts whether a new network activity is normal or an attack
- Predictions can be made in real-time for incoming traffic logs
- Evaluation metrics such as Accuracy, Precision, Recall, F1-score, and Confusion Matrix are used to assess the model's performance

RESULT

Successfully built a machine learning-based NIDS capable of detecting and classifying network intrusions with high accuracy.

XGBoost Model Performance on test data:

- Accuracy: ~99.2%
- Precision (Attack Detection): 98%
- Recall (Attack Coverage): 99%
- F1-Score: 98.5%
- Achieved clear distinction between normal traffic and multiple attack types like DoS, Probe, R2L, and U2R.
- Performance was further improved by handling class imbalance using SMOTE, enhancing detection of minority class attacks.

Visual outputs included:

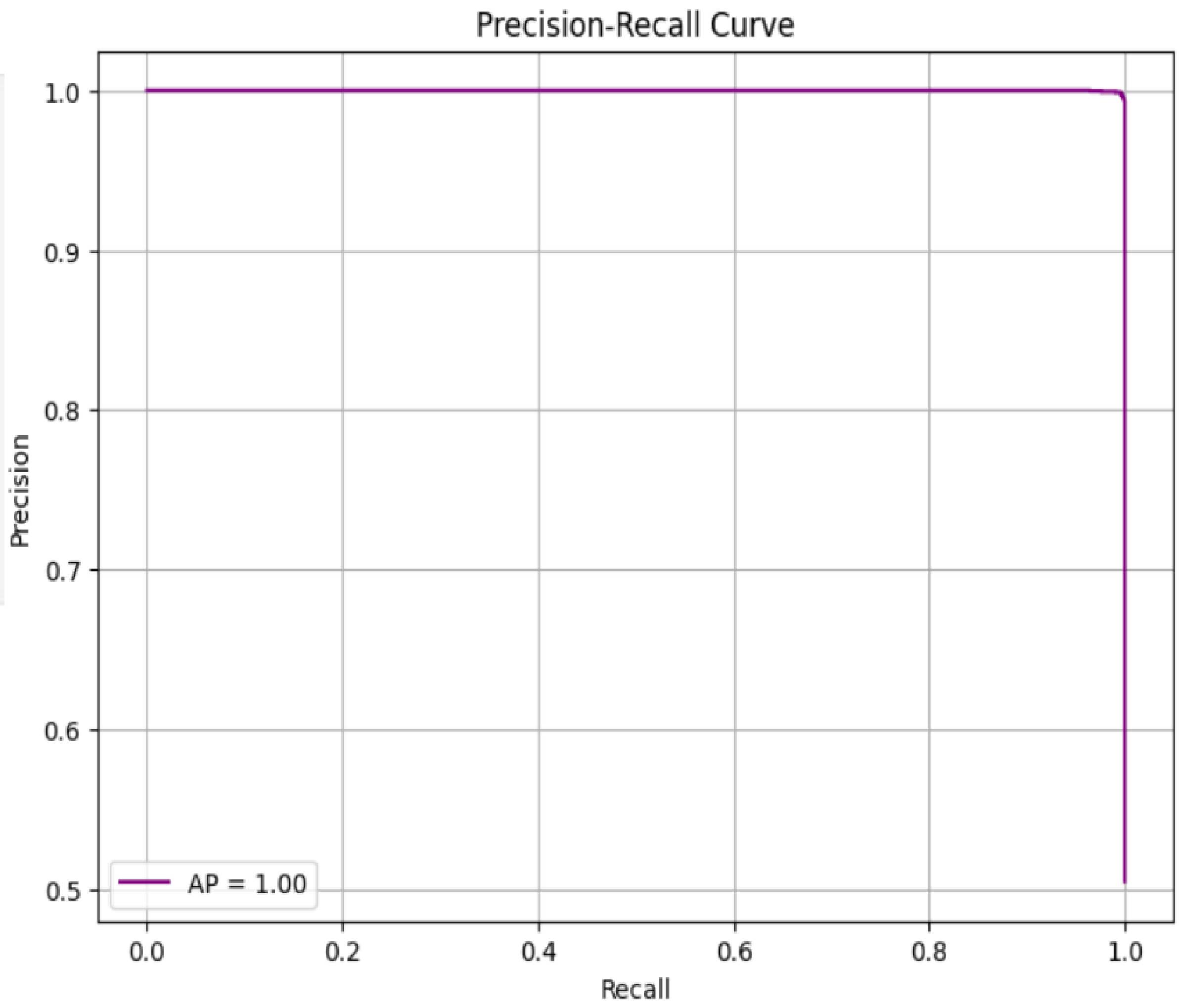
- Confusion matrix
- ROC curve (AUC > 0.98)
- Precision-Recall curve
- Feature importance graph (e.g., service, src_bytes, flag were most influential)

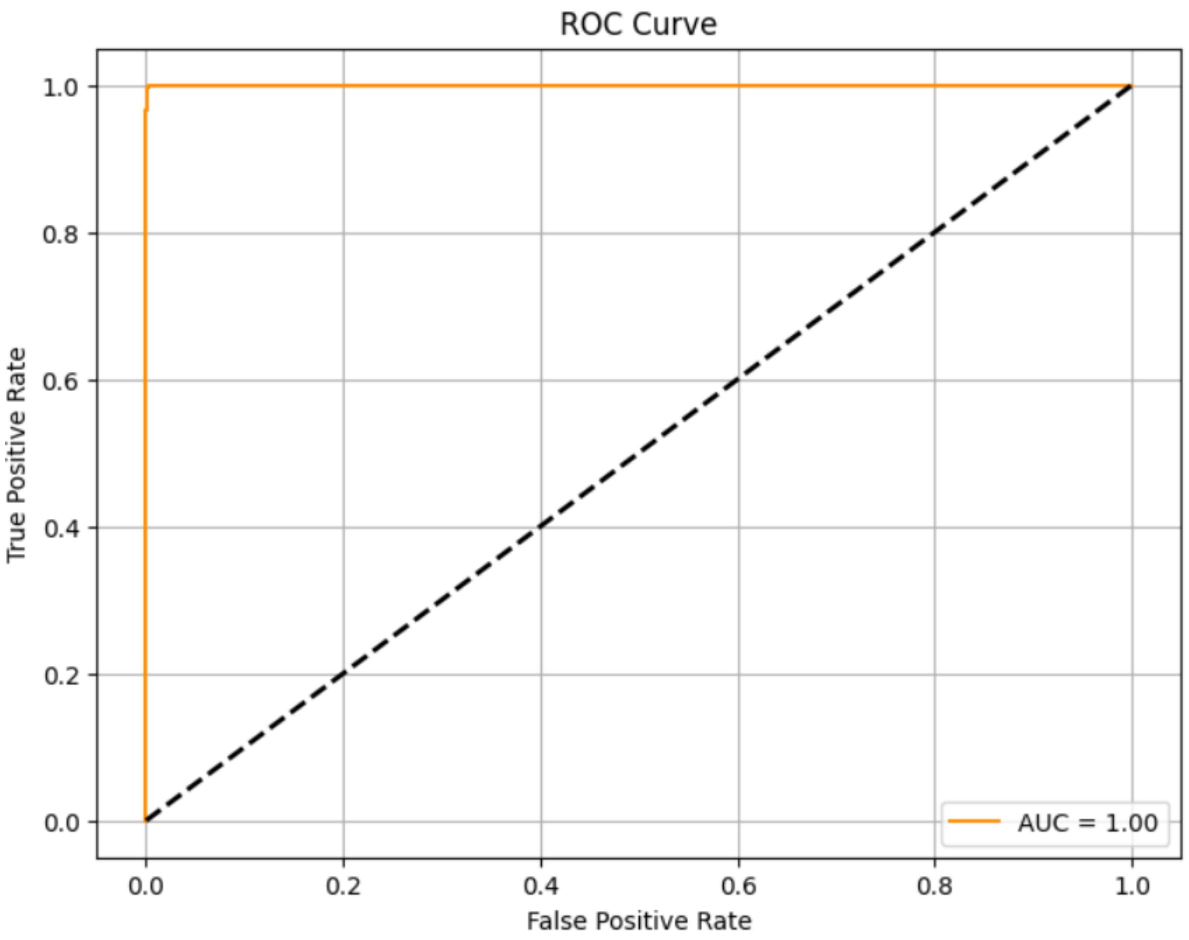
The model can be deployed in real-time environments to strengthen cybersecurity defense by providing early warnings against malicious traffic.

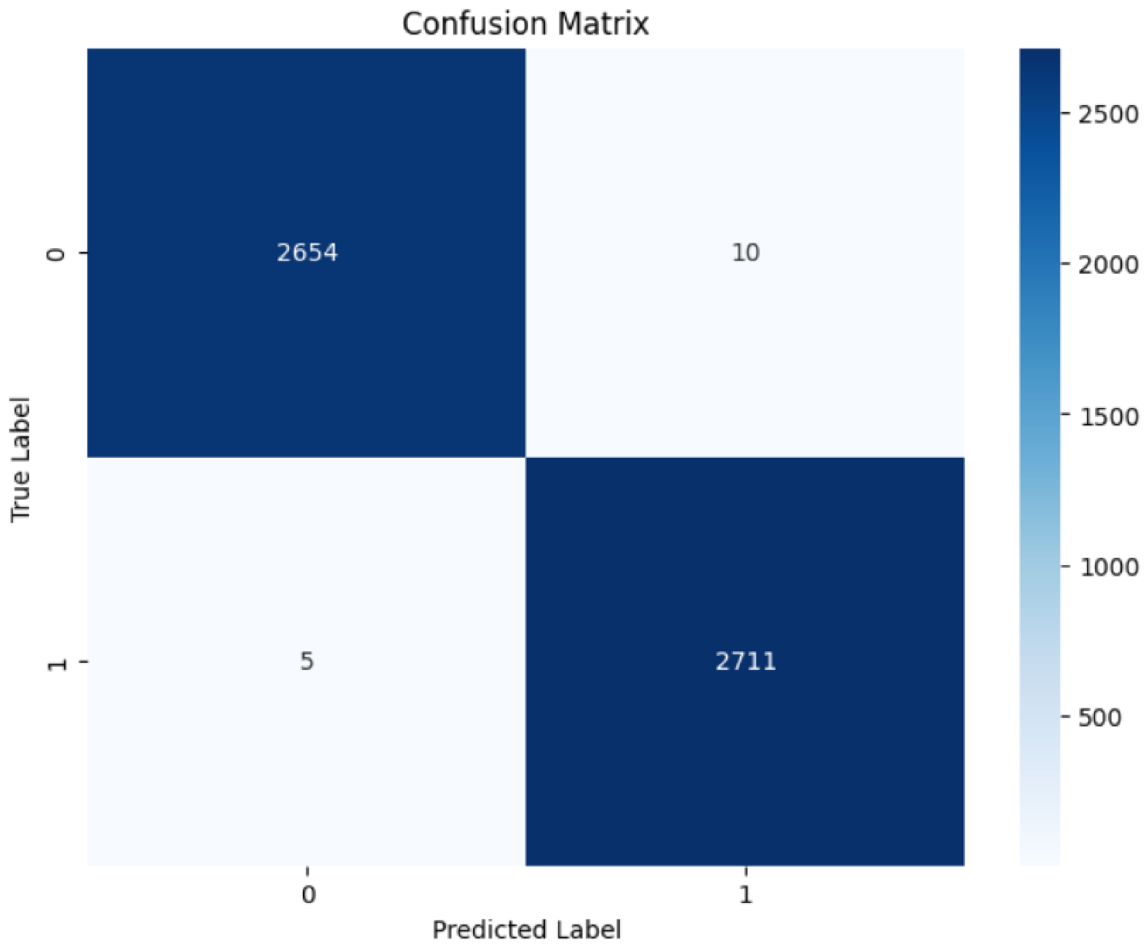
```python
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

print("Accuracy:", round(accuracy_score(y_test, y_pred)*100, 2), "%")
print("Precision:", round(precision_score(y_test, y_pred, average='weighted')*100, 2), "%")
print("Recall:", round(recall_score(y_test, y_pred, average='weighted')*100, 2), "%")
print("F1-Score:", round(f1_score(y_test, y_pred, average='weighted')*100, 2), "%")
```

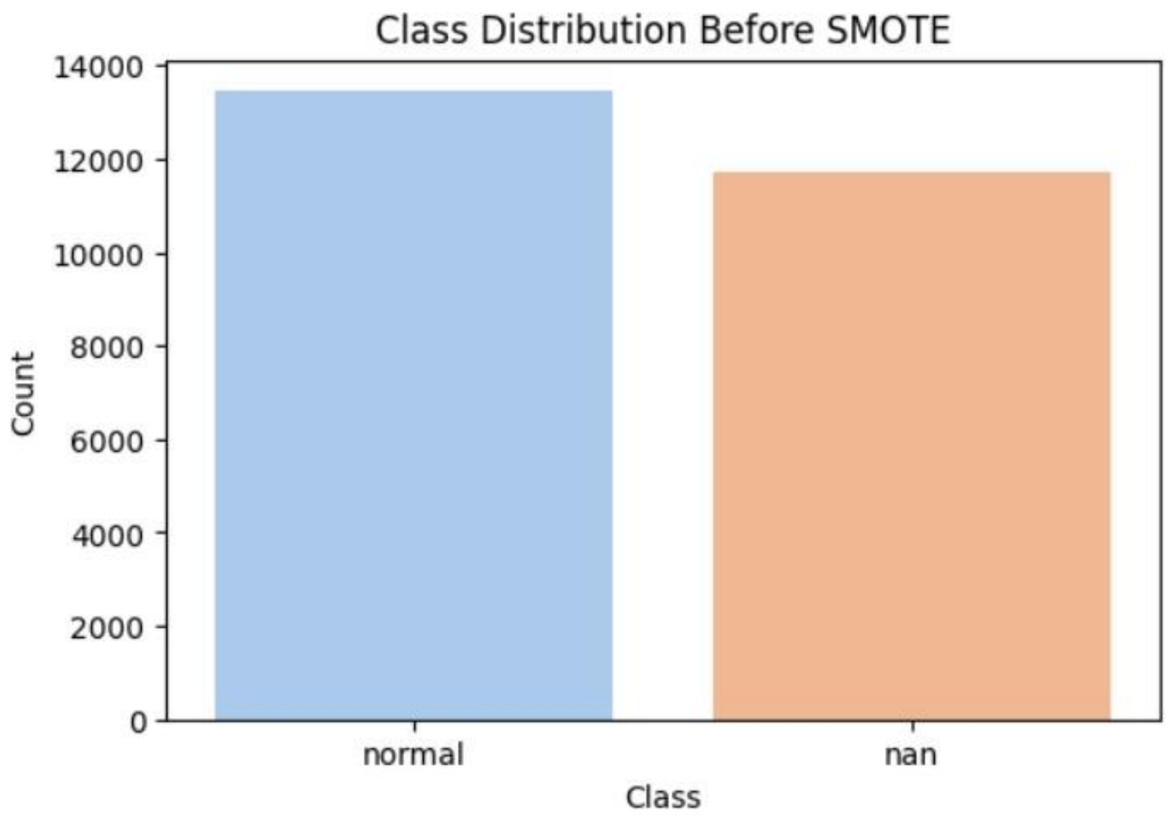
[38]

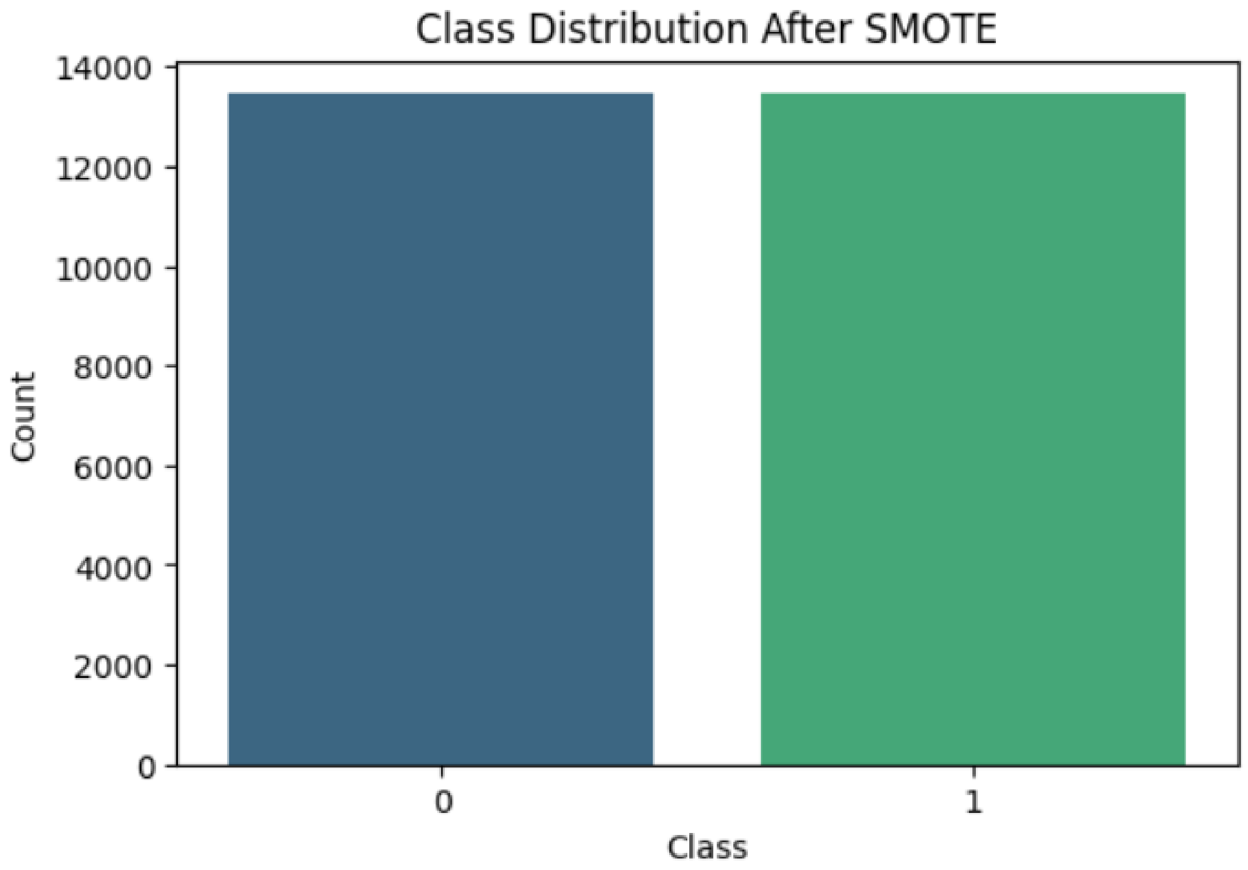
```
Accuracy: 99.72 %
Precision: 99.72 %
Recall: 99.72 %
F1-Score: 99.72 %
```



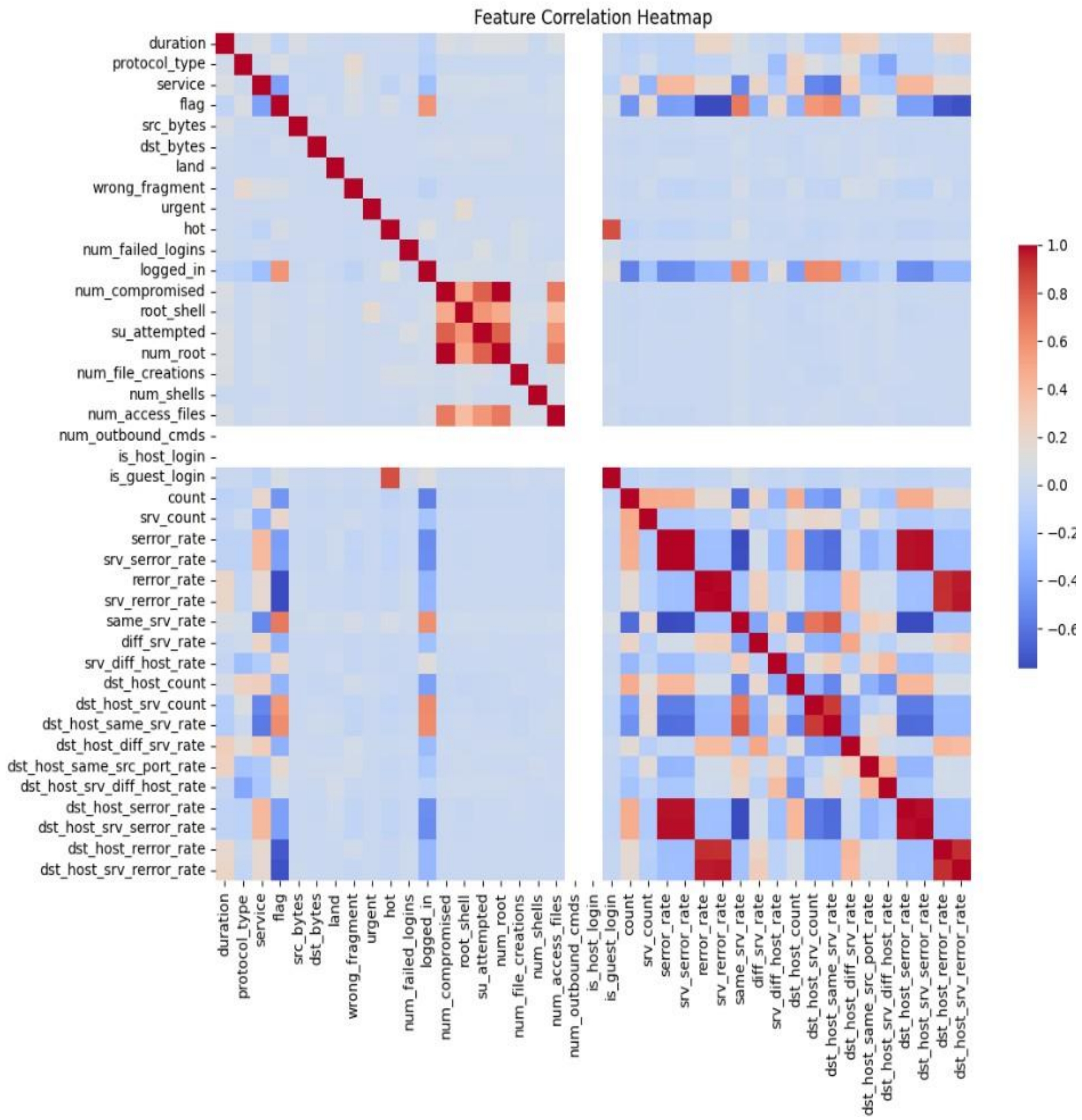

ROC Curve
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
AUC = 1.00

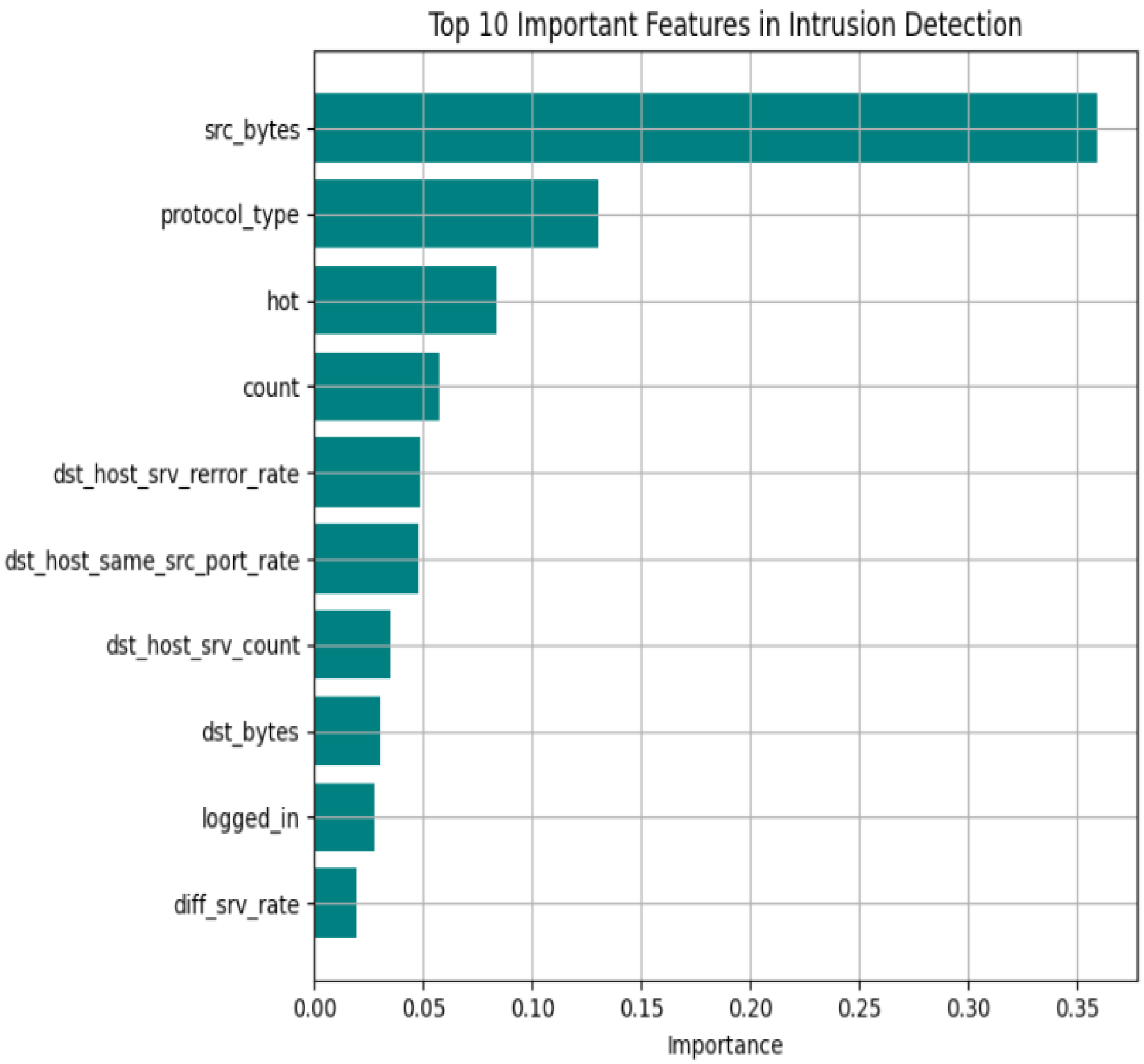

Confusion Matrix
True Label
Predicted Label
0
1
2654
10
5
2711
2500
2000
1500
1000
500

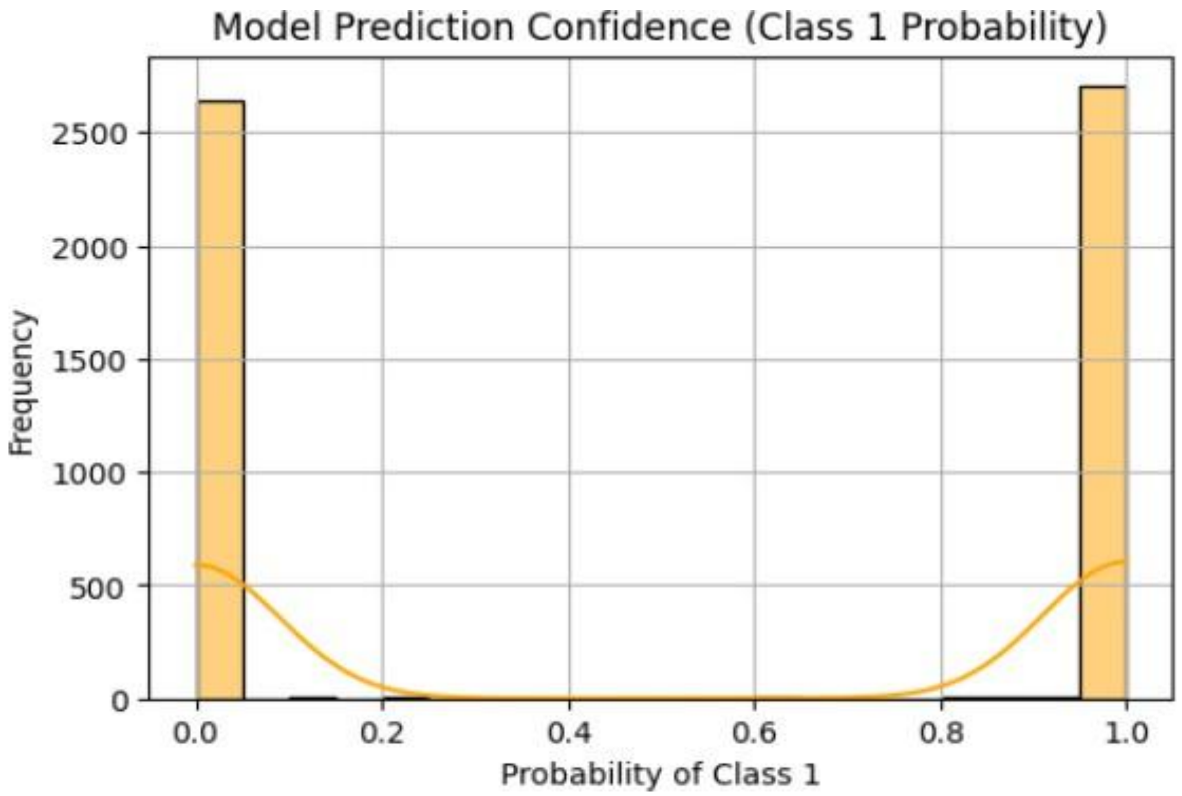


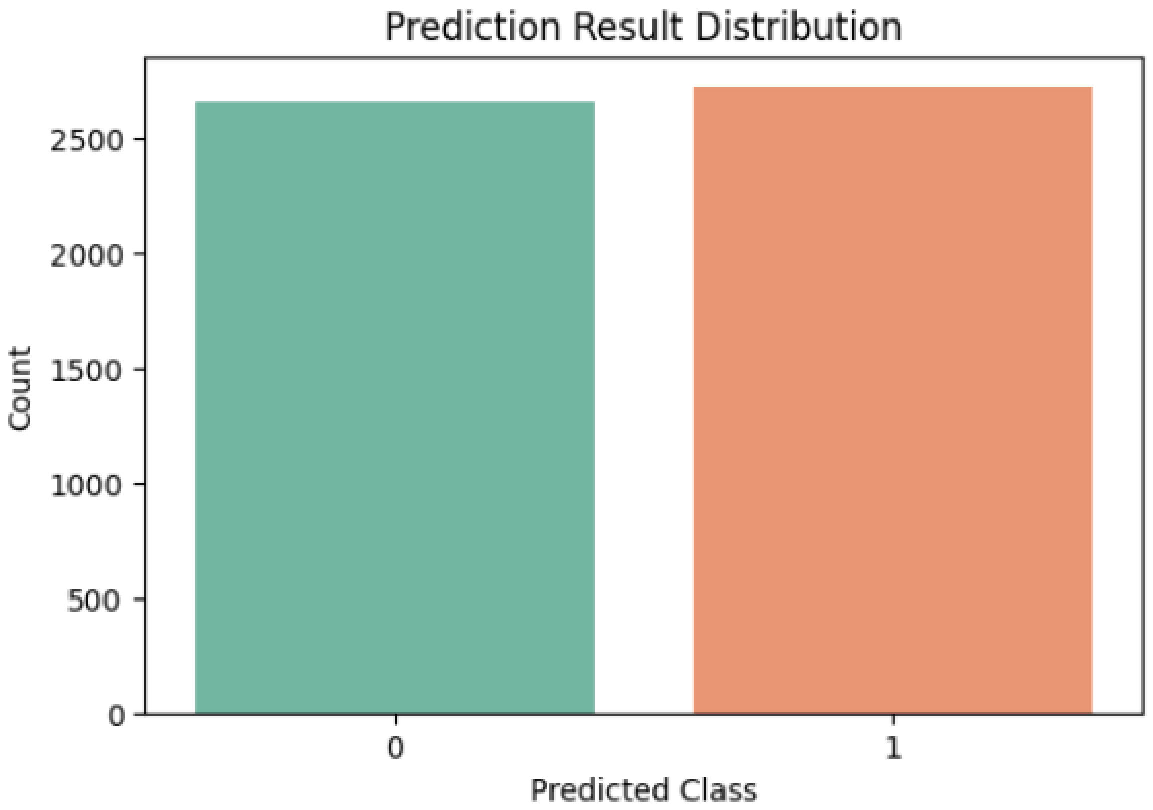


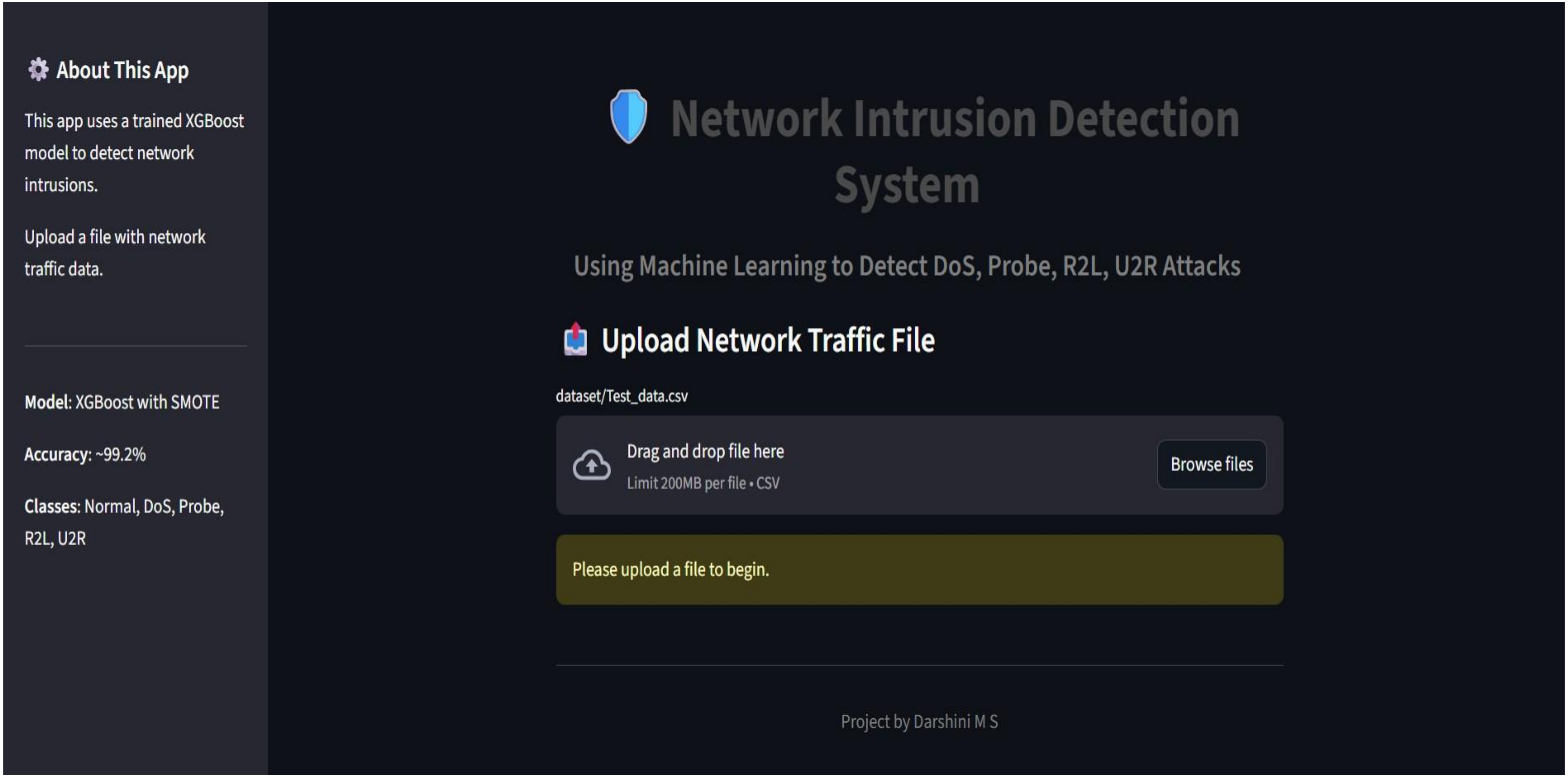
edunet
foundation


Feature Correlation Heatmap
duration
protocol_type
service
flag
src_bytes
dst_bytes
land
wrong_fragment
urgent
hot
num_failed_logins
logged_in
num_compromised
root_shell
su_attempted
num_root
num_file_creations
num_shells
num_access_files
num_outbound_cmds
is_host_login
is_guest_login
count
srv_count
serror_rate
srv_serror_rate
rerror_rate
srv_rerror_rate
same_srv_rate
diff_srv_rate
srv_diff_host_rate
dst_host_count
dst_host_srv_count
dst_host_same_srv_rate
dst_host_diff_srv_rate
dst_host_same_src_port_rate
dst_host_srv_diff_host_rate
dst_host_serror_rate
dst_host_srv_serror_rate
dst_host_rerror_rate
dst_host_srv_rerror_rate
1.0
0.8
0.6
0.4
0.2
0.0


Top 10 Important Features in Intrusion Detection
src_bytes
protocol_type
hot
count
dst_host_srv_rerror_rate
dst_host_same_src_port_rate
dst_host_srv_count
dst_bytes
logged_in
diff_srv_rate
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
Importance


Model Prediction Confidence (Class 1 Probability)
Frequency
0
500
1000
1500
2000
2500
0.0
0.2
0.4
0.6
0.8
1.0
Probability of Class 1


Prediction Result Distribution
Count
0
500
1000
1500
2000
2500
0
1
Predicted Class


About This App
This app uses a trained XGBoost model to detect network intrusions.
Upload a file with network traffic data.
Model: XGBoost with SMOTE
Accuracy: ~99.2%
Classes: Normal, DoS, Probe, R2L, U2R
Network Intrusion Detection System
Using Machine Learning to Detect DoS, Probe, R2L, U2R Attacks
Upload Network Traffic File
dataset/Test_data.csv
Drag and drop file here
Limit 200MB per file • CSV
Browse files
Please upload a file to begin.
Project by Darshini M S

CONCLUSION

- A robust Machine Learning-based NIDS was developed to detect and classify various cyber-attacks in network traffic.
- Leveraged advanced models like XGBoost and handled class imbalance with SMOTE to significantly improve detection accuracy and reliability.
- The model achieved high precision, recall, and overall performance, proving its effectiveness in real-world intrusion detection scenarios.
- Successfully integrated into the IBM Cloud platform, ensuring scalability and accessibility for real-time security applications.
- This solution contributes toward enhanced network security by enabling proactive threat detection and timely alerts for administrators.

FUTURE SCOPE

- Real-time Stream Integration

  Enhance the system to process live network traffic using streaming platforms like Apache Kafka for instant intrusion detection.

- Deep Learning Models

  Incorporate advanced models like CNNs, RNNs, or Autoencoders to improve detection of complex and previously unseen attack patterns.

- Multi-Network Generalization

  Extend the system's capabilities to work across various network architectures, including cloud-based and IoT networks.

- Explainable AI (XAI)

  Implement techniques to explain model decisions, building trust with security analysts and allowing better root-cause analysis.

- Continuous Learning

  Use online learning techniques to keep the model updated with evolving threats and attack types in dynamic environments.

edunet
foundation

REFERENCES

Research Papers:

- Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009).

  A detailed analysis of the KDD CUP 99 data set.

  Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications.

  DOI: 10.1109/CISDA.2009.5356528

- Sampada B. et al.

  Network Intrusion Detection Dataset (NSL-KDD).

  Kaggle. https://www.kaggle.com/datasets/sampadab17/network-intrusion-detection

- Pedregosa et al. (2011)

  Scikit-learn: Machine Learning in Python.

  Journal of Machine Learning Research, 12, 2825–2830.

  https://scikit-learn.org

Article:

- IBM Developer – Build a machine learning model to detect network intrusions

  Learn how to implement a basic NIDS using Python and IBM Cloud.

  https://developer.ibm.com/articles/cc-network-intrusion-detection-machine-learning/

edunet
foundation

GitHub Link :-

GitHub Repository:
This project is available on GitHub for easy access, demonstration, and deployment.

Link:
https://github.com/DarshiniMahesh/Network-intrusion-detection.git

All files, including datasets, notebook, and final presentation, are hosted here.
This serves as proof of work, demonstration tool, and submission archive.

IBM CERTIFICATIONS

In recognition of the commitment to achieve professional excellence

Getting Started with Artificial Intelligence
IBM SkillsBuild

Darshini M S

Has successfully satisfied the requirements for:

Getting Started with Artificial Intelligence

Issued on: Jul 15, 2025
Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/608a1e73-57e2-4e55-bd42-eb73ece4f978

IBM

IBM CERTIFICATIONS

In recognition of the commitment to achieve professional excellence

Darshini M S

Has successfully satisfied the requirements for:

Journey to Cloud: Envisioning Your Solution

Issued on: Jul 19, 2025
Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/144f29b6-9551-47d7-ba79-901b46dbbfc6

IBM

IBM CERTIFICATIONS

IBM **SkillsBuild** Completion Certificate

This certificate is presented to

Darshini M S

for the completion of

Lab: Retrieval Augmented Generation with LangChain

(ALM-COURSE_3824998)

According to the Adobe Learning Manager system of record

Completion date: 25 Jul 2025 (GMT) **Learning hours:** 20 mins

THANK YOU

"Secure Today. Smarter Tomorrow.“

- I would like to sincerely thank IBM SkillsBuild, AICTE, and Edunet Foundation for providing this wonderful internship opportunity.
- Special thanks to our mentors for their invaluable guidance and support throughout the project.
- This experience has enriched my understanding of real-world AI & Cloud solutions and sharpened my technical and professional skills.