Intrusion Detection System

PRESENTED BY

STUDENT NAME: ANNYATOMA DAS

COLLEGE NAME: INSTITUTE OF ENGINEERING AND MANAGEMENT KOLKATA

DEPARTMENT: COMPUTER SCIENCE AND BUSINESS SYSTEM

EMAIL ID: ANNYATOMA@GMAIL.COM

AICTE STUDENT ID:
STU680f975e9b9ed1745852254

OUTLINE

- **Problem Statement** (Should not include solution)
- **Proposed System/Solution**
- **System Development Approach** (Technology Used)
- **Algorithm & Deployment**
- **Result (Output Image)**
- **Conclusion**
- **Future Scope**
- **References**

PROBLEM STATEMENT

With the exponential increase in network traffic, identifying unauthorized or malicious activities in real time has become a major cybersecurity concern. Traditional rule-based systems are slow to adapt to new threats and lack scalability. There is a need for an intelligent system that can detect anomalies or cyberattacks automatically and accurately.

PROPOSED SOLUTION

We propose a Machine Learning–based Intrusion Detection System (IDS) that uses historical network connection data to classify behavior as normal or malicious.
Key components:

- Data Preprocessing (Label encoding, normalization)
- Model Training (Decision Tree and ANN)
- Evaluation using classification metrics

SYSTEM APPROACH

System Requirements:

•Python 3.x

•Google Colab

•Libraries: pandas, numpy, scikit-learn, keras, matplotlib

Dataset Used:

•NSL-KDD Dataset

(Improved version of KDD99 for intrusion detection)

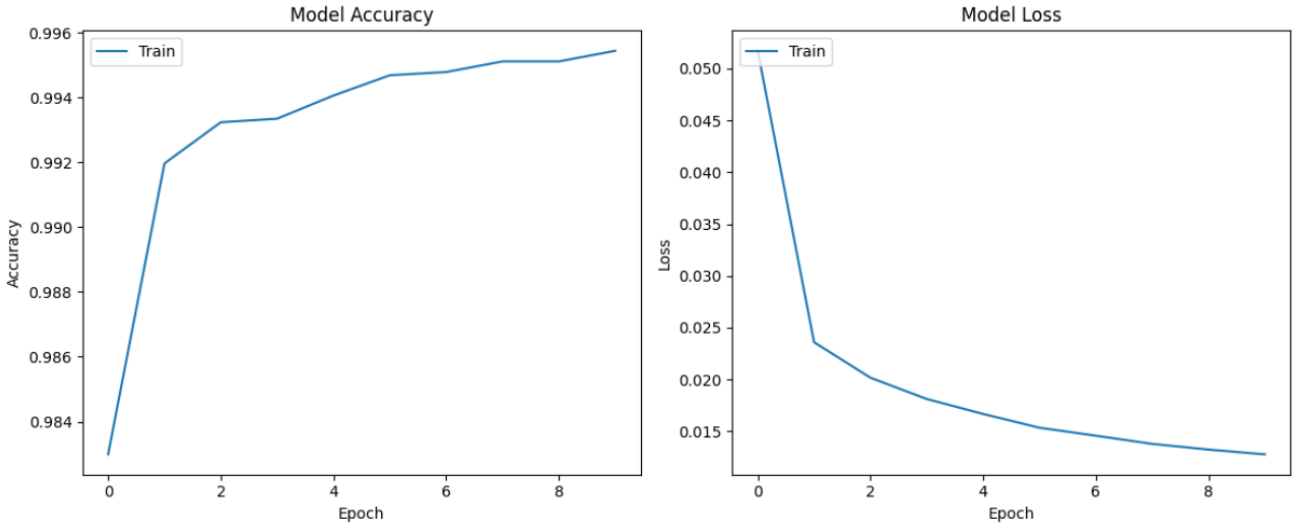
ALGORITHM & DEPLOYMENT

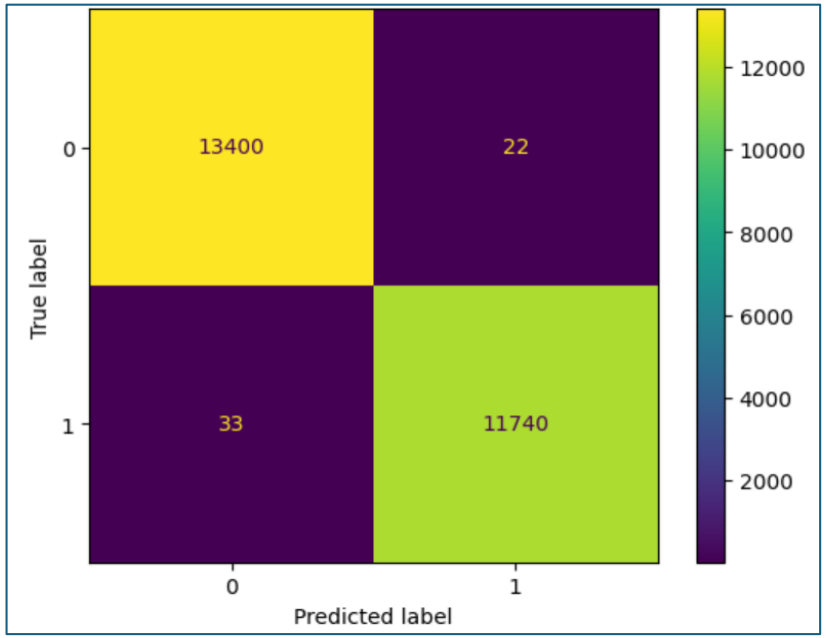
Algorithm Selection:
•Decision Tree (for interpretability and speed)
•ANN (for deeper pattern recognition)
Input Features:
•41 features per network connection, including protocol_type, service, src_bytes, etc.
Training Process:
•Data cleaned, categorical variables encoded
•StandardScaler used for normalization
•Model trained with 95/5 train-test split
Deployment:
•Model saved (.keras)

```
Epoch 1/10
3150/3150 ━━━━━━━━━━━━━━━━━━━━ 14s 3ms/step - accuracy: 0.9658 - loss: 0.0949
Epoch 2/10
3150/3150 ━━━━━━━━━━━━━━━━━━━━ 8s 2ms/step - accuracy: 0.9913 - loss: 0.0256
Epoch 3/10
3150/3150 ━━━━━━━━━━━━━━━━━━━━ 11s 3ms/step - accuracy: 0.9930 - loss: 0.0215
Epoch 4/10
3150/3150 ━━━━━━━━━━━━━━━━━━━━ 11s 3ms/step - accuracy: 0.9931 - loss: 0.0192
Epoch 5/10
3150/3150 ━━━━━━━━━━━━━━━━━━━━ 10s 3ms/step - accuracy: 0.9938 - loss: 0.0172
Epoch 6/10
3150/3150 ━━━━━━━━━━━━━━━━━━━━ 11s 3ms/step - accuracy: 0.9950 - loss: 0.0147
Epoch 7/10
3150/3150 ━━━━━━━━━━━━━━━━━━━━ 7s 2ms/step - accuracy: 0.9948 - loss: 0.0143
Epoch 8/10
3150/3150 ━━━━━━━━━━━━━━━━━━━━ 8s 3ms/step - accuracy: 0.9947 - loss: 0.0144
Epoch 9/10
3150/3150 ━━━━━━━━━━━━━━━━━━━━ 9s 3ms/step - accuracy: 0.9949 - loss: 0.0141
Epoch 10/10
3150/3150 ━━━━━━━━━━━━━━━━━━━━ 8s 3ms/step - accuracy: 0.9956 - loss: 0.0129
```

RESULT

	precision	recall	f1-score	support
0	1.00	1.00	1.00	13422
1	1.00	1.00	1.00	11773
accuracy			1.00	25195
macro avg	1.00	1.00	1.00	25195
weighted avg	1.00	1.00	1.00	25195





CONCLUSION

- The developed **Intrusion Detection System (IDS)** effectively demonstrated how machine learning models can identify and classify malicious network activity with high accuracy.
- The project used the **NSL-KDD dataset** to train two models:
  - A **Decision Tree Classifier** for quick and interpretable results
  - An **Artificial Neural Network (ANN)** for deeper pattern recognition
- All training and testing were conducted on **Google Colab**, providing an efficient and collaborative environment with access to hardware acceleration (GPU), making model training faster.
- The ANN model achieved an accuracy of over 95%, significantly improving detection rates compared to traditional rule-based systems.
- Google Colab allowed seamless experimentation, visualization, and sharing of results through interactive notebooks

FUTURE SCOPE

- Use real-time traffic with tools like Wireshark, scapy
- Try more advanced models like LSTM or transformers
- Integrate with firewall and alert systems
- Deploy on cloud (AWS/GCP)
- Extend detection to encrypted traffic using deep packet inspection

REFERENCES

Kaggle Dataset: https://www.kaggle.com/datasets/hassan06/nslkdd

GitHub Link: Link

Thank you