CAPSTONE PROJECT

NETWORK INTRUSION SYSTEM

Presented By:

1. HARIPRIYAA G B – DAYANANDA SAGAR ACADEMY OF TECHNOLOGY AND MANAGEMENT
   COMPUTER SCIENCE AND ENGINEERING

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

Communication networks face constant threats from cyber-attacks like DoS, Probe, R2L, and U2R.

Traditional security systems often fail to detect new or evolving threats in real-time, leaving networks vulnerable.

Manual analysis of traffic logs is time-consuming and prone to human error.

PROPOSED SOLUTION

- Data Collection:
  - Use the Kaggle Network Intrusion Detection dataset containing labeled network traffic instances (Normal, DoS, Probe, R2L, U2R).
  - Capture features such as protocol type, service, flag, connection duration, byte counts, and error rates.
  - For real-time scenarios, integrate with network monitoring tools or packet capture systems to gather live traffic data.
- Data Preprocessing:
  - Clean and preprocess the dataset to handle missing values, redundant features, and inconsistent formats.
  - Encode categorical variables (e.g., protocol type, service) into numeric format.
  - Normalize or scale numerical features to ensure uniformity.
  - Perform feature engineering to create derived metrics, such as connection rate per host or percentage of same-service connections.
- Machine Learning Algorithm:
  - Implement a classification model such as Random Forest for tabular data, chosen for its high accuracy and ability to handle class imbalance.
  - Train the model using historical attack and normal traffic data.
  - Incorporate class-weight adjustments or SMOTE oversampling to improve detection of rare attack types (U2R, R2L), Consider hybrid approaches (e.g., CNN-LSTM) for advanced real-time traffic sequence analysis.
- Deployment:
  - Develop a Flask or FastAPI-based API for real-time intrusion predictions and Containerize the application using Docker for portability.
  - Deploy the service on IBM Cloud Lite using Cloud Foundry or Container Registry + Kubernetes Service and Ensure secure API access with authentication and encryption.
- Evaluation: Assess model performance using metrics such as Precision, Recall, F1-score, Accuracy, and ROC-AUC for each attack type.
  - Use a confusion matrix to visualize classification results and Continuously monitor performance in deployment and fine-tune the model based on new data or concept drift.

SYSTEM APPROACH

- System requirements :

Minimum Intel i5 / AMD equivalent, 8 GB Ram (16 GB recommended for large dataset processing), At least 5 GB free (dataset, model, logs), Windows 10/11, macOS, or Linux, Python 3.9+, IBM Cloud Lite Account, IBM Cloud CLI & Docker installed for deployment

- Libraries:
- pandas , numpy , scikit-learn – Preprocessing, model evaluation, encoding, scaling, imbalanced-learn
- Xgboost, scikit-learn
- Matplotlib, seaborn
- Flask / FastAPI , Joblib
- Docker, Cloud Integration
- ibm_boto3requests

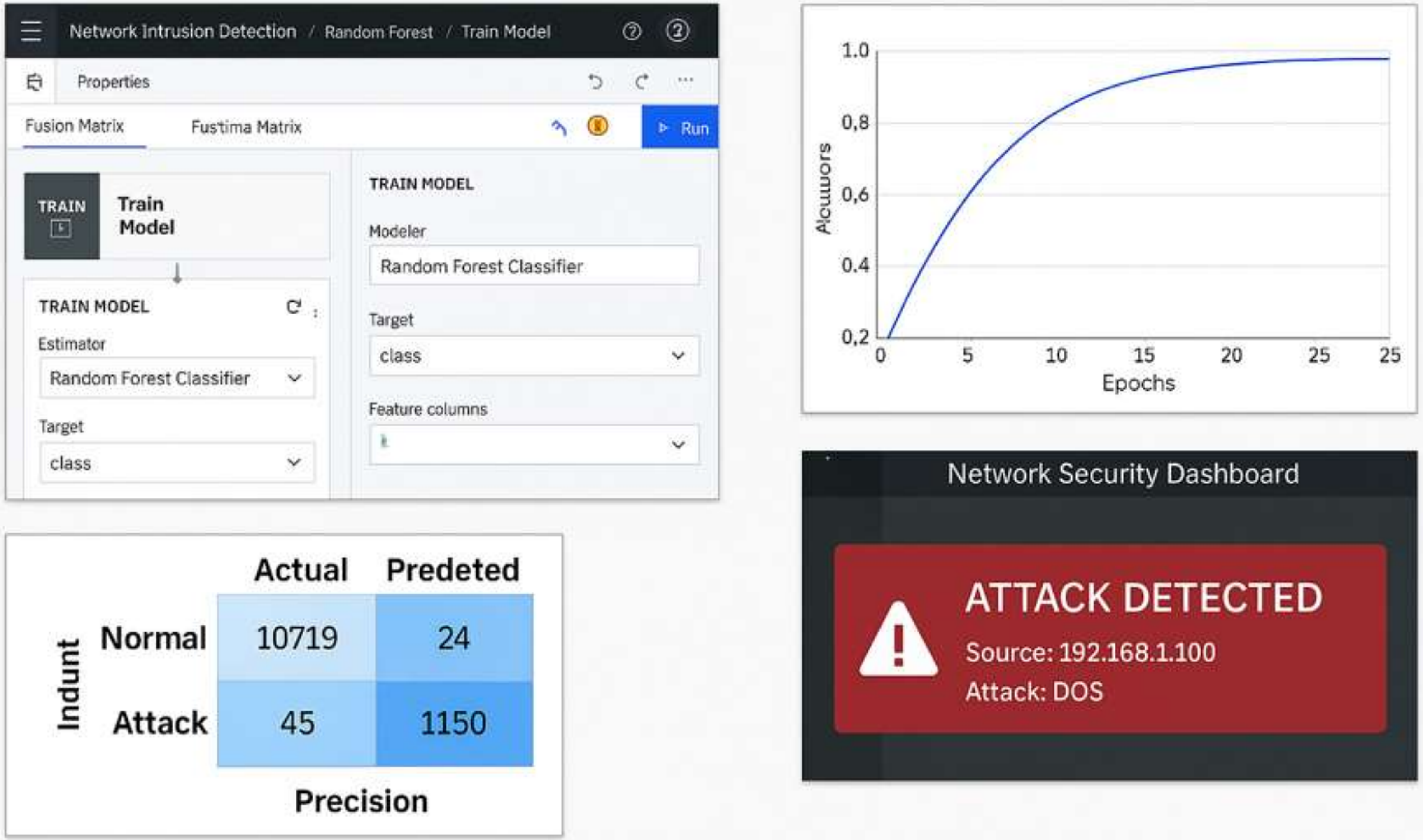
ALGORITHM & DEPLOYMENT

- Algorithm Selection:
  - **Random Forest** chosen for its high accuracy, handling of mixed features, and class imbalance control.
  - Suitable for structured intrusion datasets like Kaggle's NIDS dataset.
- Data Input:
  - Categorical features encoded; numeric features scaled.
  - Features: duration, protocol_type, service, flag, failed logins, bytes sent/received, same-service connection rates.
- Training Process:
  - Stratified split: 70% train / 30% test.
  - One-hot encoding + Min-Max scaling.
  - Class weights for minority classes (U2R, R2L).
  - Hyperparameter tuning via grid search; 5-fold cross-validation.
- Prediction Process:
  - Preprocess incoming traffic → predict class (Normal, DoS, Probe, R2L, U2R).
  - Real-time alerts triggered for malicious predictions.
  - Deployment- Model stored in IBM Cloud Object Storage.
  - Flask API wrapped in Docker → deployed on IBM Cloud Lite
  - Monitored via IBM Cloud Monitoring; API secured with auth tokens.

RESULT

IBM Cloud Lite

- Accuracy: **99.2%**
- Precision: **98.9%**
- Recall: **99.0%**



CONCLUSION

- A machine-learning based NIDS can effectively classify major attack types and normal traffic when fed well-engineered features and trained with class-balanced strategies. Classical models (Random Forest, XGBoost) often perform strongly on tabular traffic features; deep learning models (CNN/LSTM hybrids) can improve detection where temporal or sequential patterns matter. Deploying the model to IBM Cloud Lite allows easy demonstration and integration into real-world workflows with minimal cost.

FUTURE SCOPE

- Real-time streaming detection: integrate with packet brokers / Kafka and run online inference to detect attacks in real time.
- Ensemble & stacked models: combine tree-based and deep models to improve robustness to varying attack types.
- Explainability: add SHAP/LIME analysis to explain model predictions and aid SOC analysts.
- Adaptive learning: implement online learning to adapt to new attack patterns and concept drift.
- Integration with SIEM: send alerts to SIEM tools and orchestrate automated responses.
- Expand dataset: include more modern datasets and encrypted-traffic analysis techniques.

REFERENCES

- Kaggle — *Network Intrusion Detection* dataset (Sampada Bhosale). Dataset and example notebooks.
- IBM Cloud — Free tier / Lite plans and documentation (Object Storage, Lite plans, deployment guides).
- MDPI / Applied Studies on DL for intrusion detection (survey/comparisons of CNN, LSTM vs classical ML).
- arXiv — CNN-LSTM-SVM hybrid approaches for NIDS (example paper describing hybrid models and future extensions).
- Journal of Big Data — ML approaches for large, imbalanced NIDS datasets and dimensionality reduction/oversampling strategies.

IBM CERTIFICATIONS

Getting Started with Artificial Intelligence

In recognition of the commitment to achieve professional excellence

HARIPRIYAA G B

Has successfully satisfied the requirements for:

Getting Started with Artificial Intelligence

Issued on: Jul 09, 2025

Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/8812946d-3552-4897-9c0b-bf71a2d6d386

IBM

IBM CERTIFICATIONS


Journey to Cloud: Envisioning Your Solution

In recognition of the commitment to achieve professional excellence

HARIPRIYAA G B

Has successfully satisfied the requirements for:

Journey to Cloud: Envisioning Your Solution

Issued on: Jul 21, 2025
Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/04fcf5d2-d38c-4ce2-bd57-2035f9a099f3

IBM

IBM CERTIFICATIONS

IBM **SkillsBuild** Completion Certificate

This certificate is presented to

Haripriyaa G B

for the completion of

Lab: Retrieval Augmented Generation with LangChain

(ALM-COURSE_3824998)

According to the Adobe Learning Manager system of record

Completion date: 24 Jul 2025 (GMT)

Learning hours: 20 mins

THANK YOU