CAPSTONE PROJECT

NETWORK INTRUSION DETECTION SYSTEM

Presented By:
1. Hritam Brahmachari-Heritage Institute of Technology-IT

edunet foundation

OUTLINE

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

PROBLEM STATEMENT

Problem statement No.40

Create a robust network intrusion detection system (NIDS) using machine learning. The system should be capable of analyzing network traffic data to identify and classify various types of cyber-attacks (e.g., DoS, Probe, R2L, U2R) and distinguish them from normal network activity. The goal is to build a model that can effectively secure communication networks by providing an early warning of malicious activities.

PROPOSED SOLUTION

- Approach: We adopted a no-code, cloud-based approach to rapidly develop and deploy a high-quality machine learning model.
- Utilized the IBM Cloud platform and Watson Studio environment.
- Employed the AutoAI tool to automate the process of data preparation, model selection, and hyperparameter tuning.
- Trained a binary classification model to distinguish between "normal" and "anomaly" traffic.
- Deployed the best-performing model as a live web service (API) for real-time predictions.
- Key Technologies Used: IBM Cloud, IBM Watson Studio, IBM AutoAI.

SYSTEM APPROACH

System Requirements:

- Platform: An active IBM Cloud account with a provisioned Watson Studio service.
- Hardware: A standard computer capable of running a modern web browser.
- Internet: A stable internet connection to access the IBM Cloud platform.
- Environment: A "Deployment Space" configured within Watson Studio to host the final model.

Libraries Required to Build the Model:

No specific programming libraries were required to be installed or coded with for this project. Our system approach was to use IBM's AutoAI, a no-code tool that automates the entire model-building process. AutoAI handles the implementation of various machine learning algorithms and data processing techniques internally, using libraries such as Scikit-learn and Pandas in the background. However, as the user, we did not need to write any code or directly manage these libraries.

ALGORITHM & DEPLOYMENT

Algorithm Selection:

- LightGBM Classifier.is used. LightGBM (Light Gradient Boosting Machine) is a gradient boosting framework that uses tree-based learning algorithms. It is designed to be distributed and efficient.

Several key advantages:

- Faster training speed and higher efficiency.
- Lower memory usage. Better accuracy.
- Support for parallel and GPU learning.
- Capable of handling large-scale data.

Data Input:

- The Network Intrusion Detection dataset was sourced from Kaggle.
- The Train_data.csv file, containing labeled examples of network traffic, was selected as the input for model training.
- This file was uploaded directly into the IBM Watson Studio project, where it became a usable data asset.

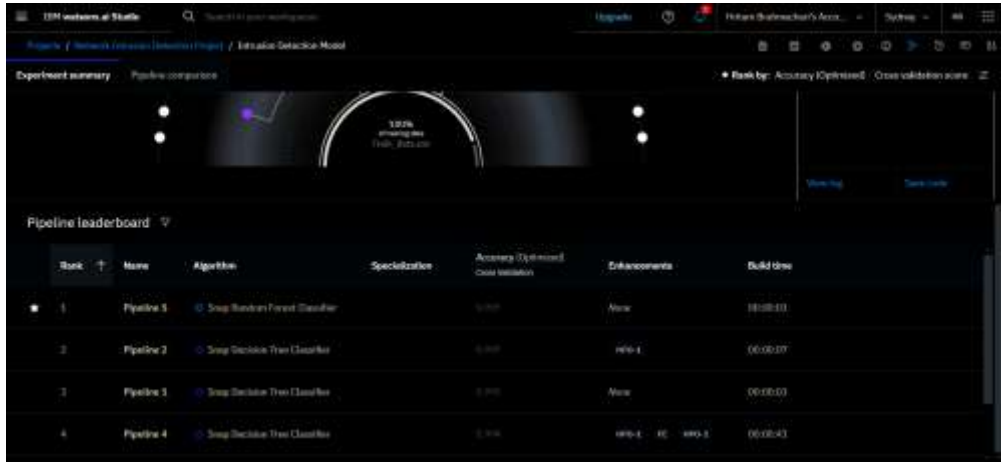
Training Process:

- AutoAI automatically performs a process called data hold-out. It splits the training data, using one part to train the models and a separate, unseen part to validate their performance. This ensures the model is evaluated on data it hasn't seen before, preventing overfitting.
- Hyperparameter Tuning: This is a core function of AutoAI. The tool automatically and systematically searches for the best combination of hyperparameters for each algorithm it tests. This automated tuning process is what allows it to optimize models like the LightGBM Classifier for the highest possible accuracy and efficiency without any manual intervention.

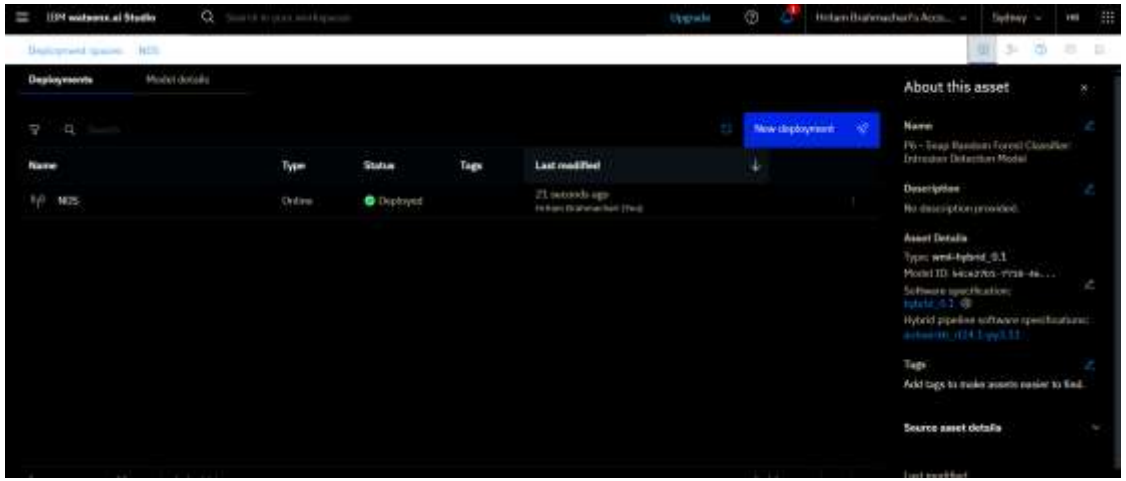
Prediction Process:

- The process begins when an application or monitoring system needs to evaluate a new network connection.
- Data Input: The data for this single connection is formatted as a JSON object. This object contains the values for all the input features the model was trained on (e.g., duration, protocol_type, src_bytes)
- API Request: The JSON object is sent as a request to the deployed model's secure REST API endpoint. Model Prediction: The live LightGBM Classifier model receives the input data, instantly processes its features, and applies the patterns it learned during training
- Output: The API returns a prediction in real-time. This output includes the final classification (anomaly or normal) along with a confidence score that indicates the model's certainty.

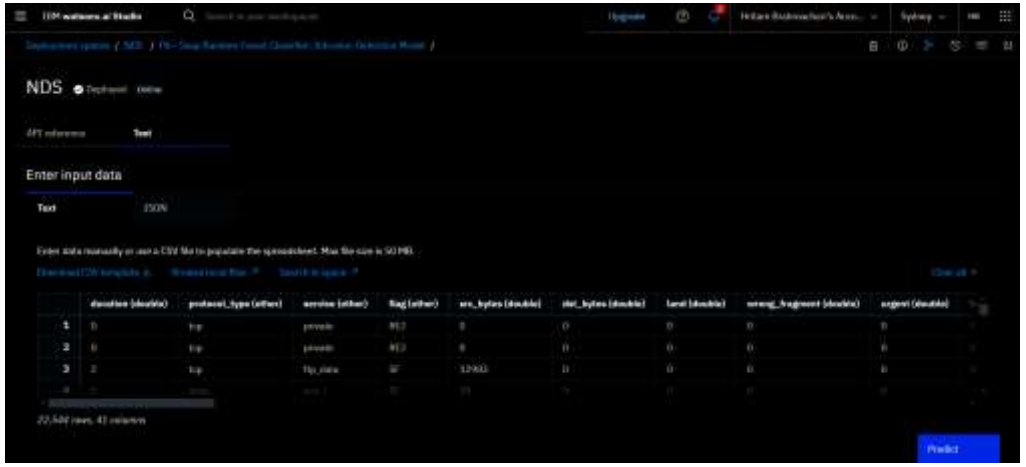
RESULT



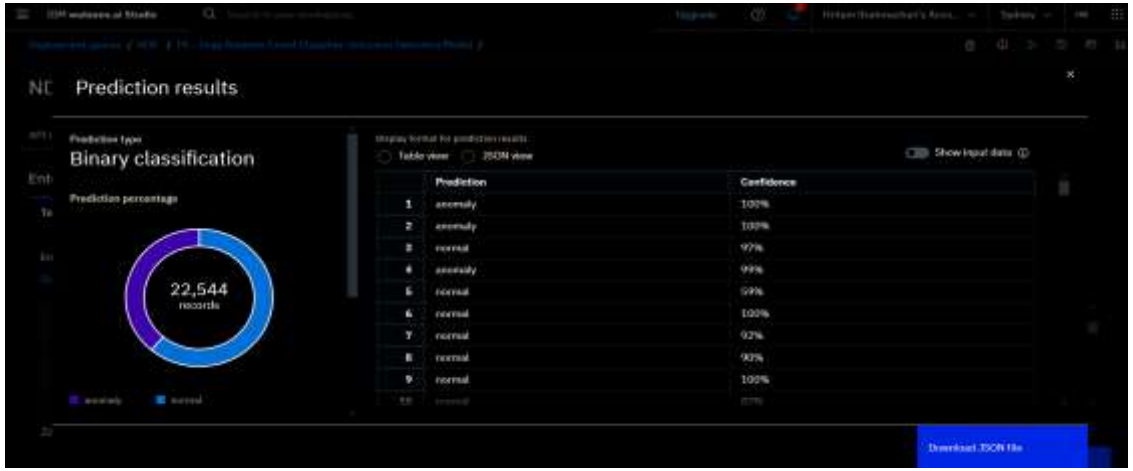
TRAINING



DEPLOYMENT



TEST DATA INPUT



PREDICTION

edunet foundation

CONCLUSION

- Findings: The project demonstrated that a no-code approach using IBM AutoAI can rapidly produce and deploy an effective LightGBM classifier for live network threat detection. The deployed API provides a practical and efficient solution.
- Challenges: The main challenges involved interpreting the dataset's scope (binary vs. multiclass) and correctly configuring the model's "Positive Class" to align with the project's goal of finding attacks.
- Potential Improvements: Future work includes enhancing the model for multiclass attack detection, integrating it with real-time monitoring tools, and implementing an automated retraining pipeline.
- Importance: This solution is critical as it serves as a vital defense for maintaining network security, protecting sensitive data, and ensuring an organization's operational stability against cyber threats.

FUTURE SCOPE

Potential Enhancements

- **Algorithm and Data:** The model can be enhanced by incorporating real-time **threat intelligence feeds** and upgrading it to perform **multiclass classification** to identify specific attack types.
- **System Expansion:** The system's coverage can be expanded by deploying models across multiple **network segments or cloud environments**, managed via a centralized monitoring dashboard.
- **Emerging Technologies:** Future versions could integrate **Edge Computing** for on-device detection and advanced techniques like **Federated Learning** to train models without sharing private data.
- **Advanced Machine Learning:** The system could adopt more advanced techniques like **federated learning** to train a shared model across isolated networks without compromising data privacy. Furthermore, using **transformer models** to analyze sequences of network packets could help detect more sophisticated, low-and-slow attacks that are invisible to connection-based analysis.

REFERENCES

The proposed solution for the **Network Intrusion Detection** project was developed using the provided project documentation and the automated capabilities of the IBM Cloud platform, rather than external academic papers.
The following sources were instrumental in its development:

GITHUB: https://github.com/HritamBrahmachari/Network-Intrusion-Detection-System

Project Definition Document

- This document provided the official problem statement, which was to create a network intrusion detection system using machine learning.
- It specified the mandatory use of IBM Cloud services.
- It provided the direct link to the dataset on Kaggle.

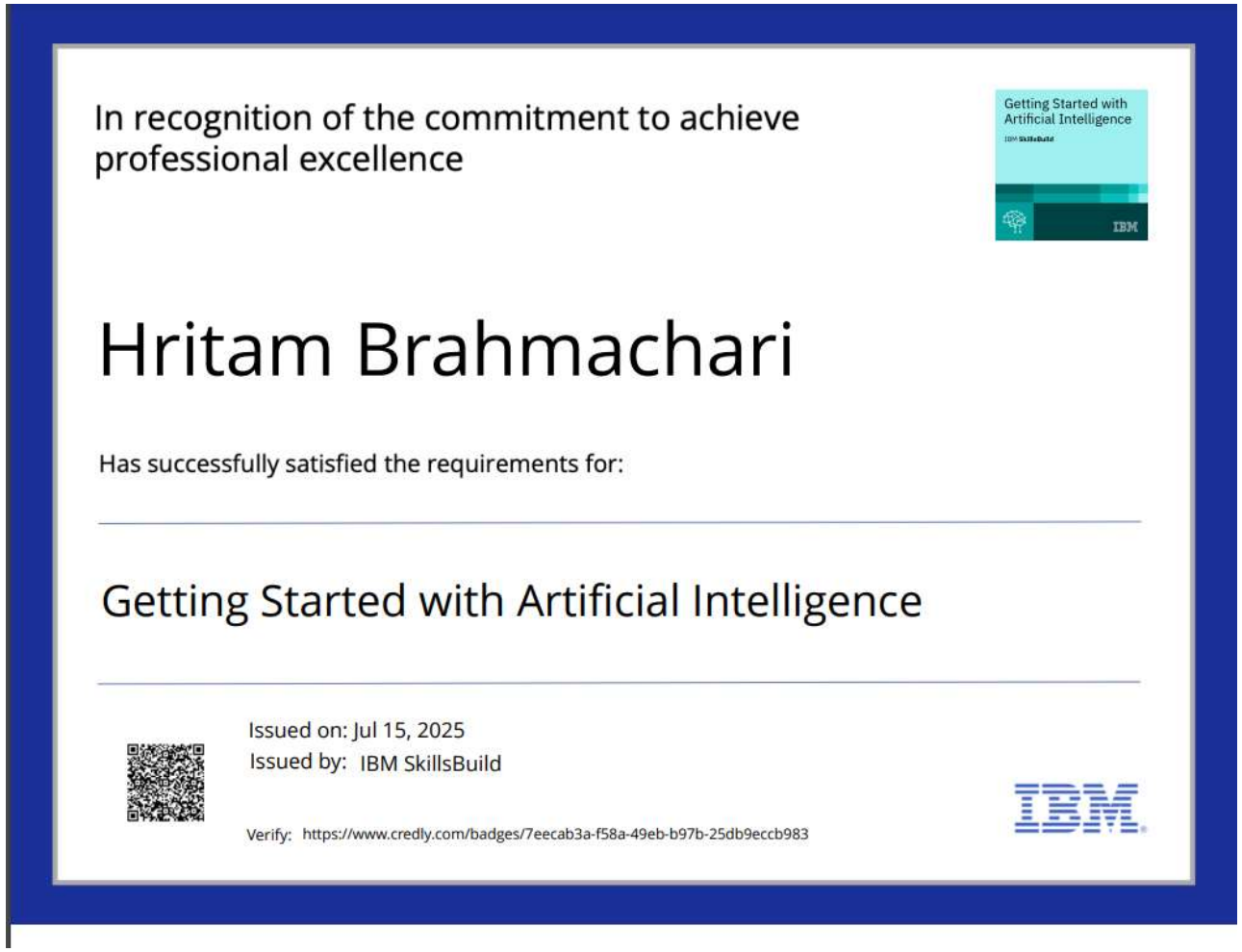
Network Intrusion Detection Dataset

- The solution was trained on this specific dataset, sourced from Kaggle as per the project requirements. The structure and features of this dataset dictated the modeling approach.

IBM AutoAI Platform

- This platform was the core tool for development. Its automated features for data preprocessing, algorithm selection, hyperparameter tuning, and model evaluation were used to build the solution without needing to reference external best practices or research papers on these topics.

IBM CERTIFICATIONS



edunet foundation

IBM CERTIFICATIONS

In recognition of the commitment to achieve professional excellence

Journey to Cloud: Envisioning Your Solution

IBM SkillsBuild

IBM

Hritam Brahmachari

Has successfully satisfied the requirements for:

Journey to Cloud: Envisioning Your Solution

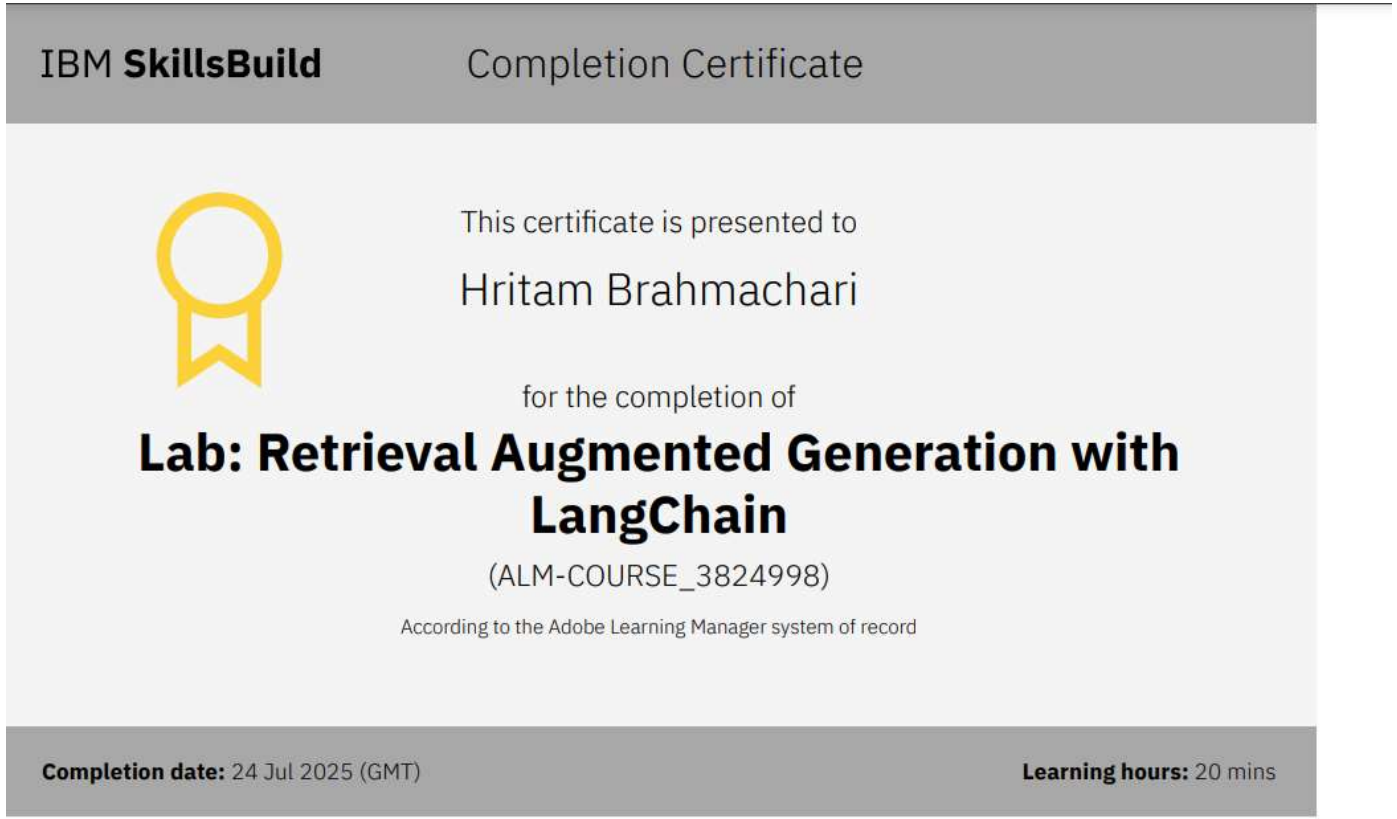
Issued on: Jul 17, 2025

Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/dffeb9e2-34aa-4235-853b-45df0ee1bfd5

IBM

IBM CERTIFICATIONS



IBM **SkillsBuild** Completion Certificate

This certificate is presented to

Hritam Brahmachari

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

edunet foundation