
CAPSTONE PROJECT

PROJECT TITLE

Presented By:-Suman Pathy
College Name-Nist University
Department:- Computer Science

OUTLINE

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

PROBLEM STATEMENT

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

Develop a machine learning model for analysing network traffic data to identify and classify various types of cyber-attacks and distinguish them from normal network activity .The model can effectively secure communication networks by providing an early warning of malicious activities.

- Key components:

- Data collection: Data set collection from Kaggle
- Preprocessing: Clean and normalise the dataset
- Model training: Train a classification model(e.g, Decision Tree, Random Forest, svm)
- Evaluation: validate the model using accuracy, precession, recall and F-1 score

SYSTEM APPROACH

The "System Approach" section outlines the overall strategy and methodology for developing and implementing the Network Intrusion Detection system. Here's a suggested structure for this section:

- **System requirements**
 - IBM Cloud (mandatory)
 - IBM Watson studio for model development and deployment
 - IBM Cloud object storage for dataset handling

ALGORITHM & DEPLOYMENT

- **Algorithm Selection:**

- Snap Decision Tree Classifier

- **Data Input:**

- Specify the input features used by the algorithm, such as protocol_type , flag , src_bytes , dst_bytes , count , dst_host_srv_error_rate

- **Training Process:**

- Supervised learning using labeled faulty types

- **Prediction Process:**

- Model deployed on IBM Watson API Studio with API endpoint for real time predictions.

RESULT

IBM watsonx.ai Studio

Search in your workspaces

Upgrade

Suman Pathy's Account

Sydney

SP

Projects / final_project / Network Intrusion Detector

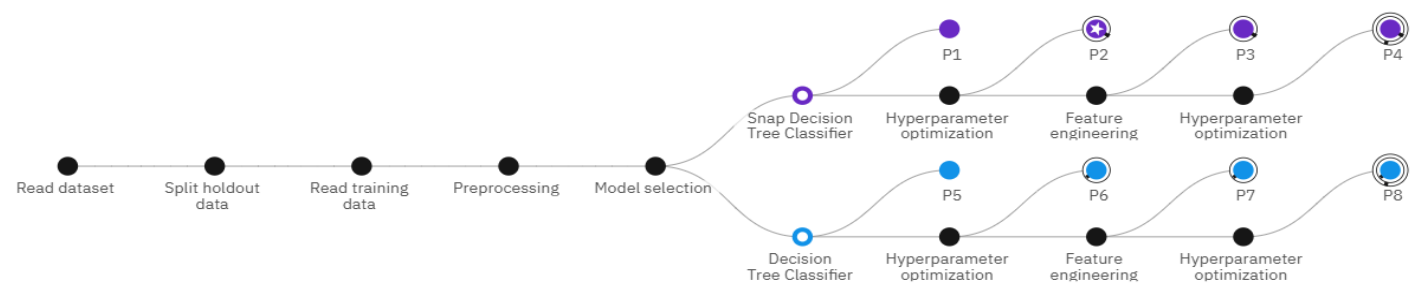
Experiment summary

Pipeline comparison

★ Rank by: Accuracy (Optimized) | Cross validation score

Progress map ⓘ

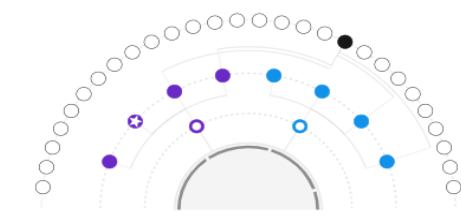
Prediction column: class



```
graph LR; A[Read dataset] --> B[Split holdout data]; B --> C[Read training data]; C --> D[Preprocessing]; D --> E[Model selection]; E --> F[Snap Decision Tree Classifier]; E --> G[Decision Tree Classifier]; F --> H[P1]; F --> I[P5]; H --> J[Hyperparameter optimization]; I --> K[Hyperparameter optimization]; J --> L[Feature engineering]; K --> M[Feature engineering]; L --> N[Hyperparameter optimization]; M --> O[Hyperparameter optimization]; N --> P[P3]; O --> Q[P7]; P --> R[P4]; Q --> S[P8];
```

Relationship map

Swap view ↔



Experiment completed ✓

8 PIPELINES GENERATED

8 pipelines generated from algorithms. See pipeline leaderboard below for more detail.

Time elapsed: 14 minutes

View log

Save code

Pipeline leaderboard ⌵

RESULT

IBM watsonx.ai Studio

Search in your workspaces

Upgrade

Suman Pathy's Account

Sydney

SP

Projects / final_project / Network Intrusion Detector

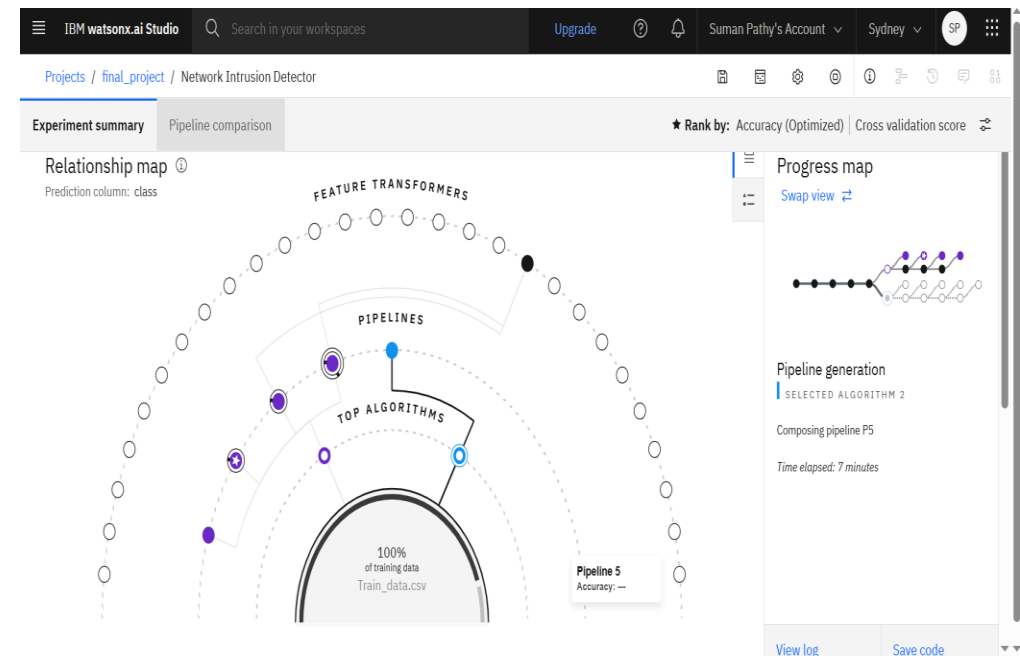
Experiment summary Pipeline comparison

★ Rank by: Accuracy (Optimized) | Cross validation score

[View log](#) [Save code](#)

Pipeline leaderboard ▾

	Rank ↑	Name	Algorithm	Accuracy (Optimized) Cross Validation	Enhancements	Build time
★	1	Pipeline 2	Snap Decision Tree Classifier	0.995	HPO-1	00:00:58
	2	Pipeline 1	Snap Decision Tree Classifier	0.995	None	00:00:54
	3	Pipeline 5	Decision Tree Classifier	0.994	None	00:00:03
	4	Pipeline 4	Snap Decision Tree Classifier	0.994	HPO-1 FE HPO-2	00:02:21



RESULT

IBM watsonx.ai Studio

Search in your workspaces

Upgrade

Summarize

Deployment spaces / Network Intrusion Detection dep_1 / P2 - Snap Decision Tree Classifier: Network Intrusion Detector /

Network Intrusion Detection dep_2 Deployed Online

API reference

Test

Enter input data

Text

JSON

Enter data manually or use a CSV file to populate the spreadsheet. Max file size is 50 MB.

[Download CSV template](#) [Browse local files](#) [Search in space](#)

	duration (double)	protocol_type (other)	service (other)	flag (other)	src_bytes (double)	dst_bytes (double)	land (double)	wrong_fragment (double)	urgent (double)	hot (double)
1	0	tcp	private	REJ	0	0	0	0	0	0
2	0	tcp	private	REJ	0	0	0	0	0	0
3	2	tcp	ftp_data	SF	12983	0	0	0	0	0
4	0	icmp	eco_i	SF	20	0	0	0	0	0
5	1	tcp	telnet	RSTO	0	15	0	0	0	0
6	0	tcp	http	SF	267	14515	0	0	0	0
7	0	tcp	smtp	SF	1022	387	0	0	0	0
8	0	tcp	telnet	SF	129	174	0	0	0	0
9	0	tcp	http	SF	327	467	0	0	0	0
10	0	tcp	ftp	SF	26	157	0	0	0	0

10 rows, 41 columns

IBM watsonx.ai Studio

Search in your workspaces

Upgrade

Suman Pathy's Account

London

SP

Deployment spaces / Network Intrusion Detection dep_1 / P2 - Snap Decision Tree Classifier: Network Intrusion Detector /

API reference

Test

Enter input data

JSON

Enter data manually or use a CSV file to populate the spreadsheet. Max file size is 50 MB.

[Download CSV template](#) [Browse local files](#) [Search in space](#)

10 rows, 41 columns

Prediction results

Prediction type

Binary classification

Prediction percentage

10 records

Confidence level distribution

Display format for prediction results

Table view

JSON view

Show input data

	Prediction	Confidence
1	anomaly	100%
2	anomaly	100%
3	normal	100%
4	anomaly	100%
5	normal	100%
6	normal	100%
7	normal	100%
8	normal	100%
9	normal	100%
10	anomaly	100%
11		

Download JSON file

CONCLUSION

- In this project, i successfully developed a Machine Learning-based **Network Intrusion Detection System (NIDS)** capable of analyzing network traffic data to detect and classify malicious activities. Using a labeled dataset containing various network features and class labels (normal vs anomaly), the system learned to differentiate between legitimate traffic and potential cyber-attacks.

FUTURE SCOPE

- **Multi-Class Attack Classification**

Upgrade the system to not only detect anomalies but also classify them into specific attack types like DoS, Probe, R2L, and U2R, improving threat identification and response.

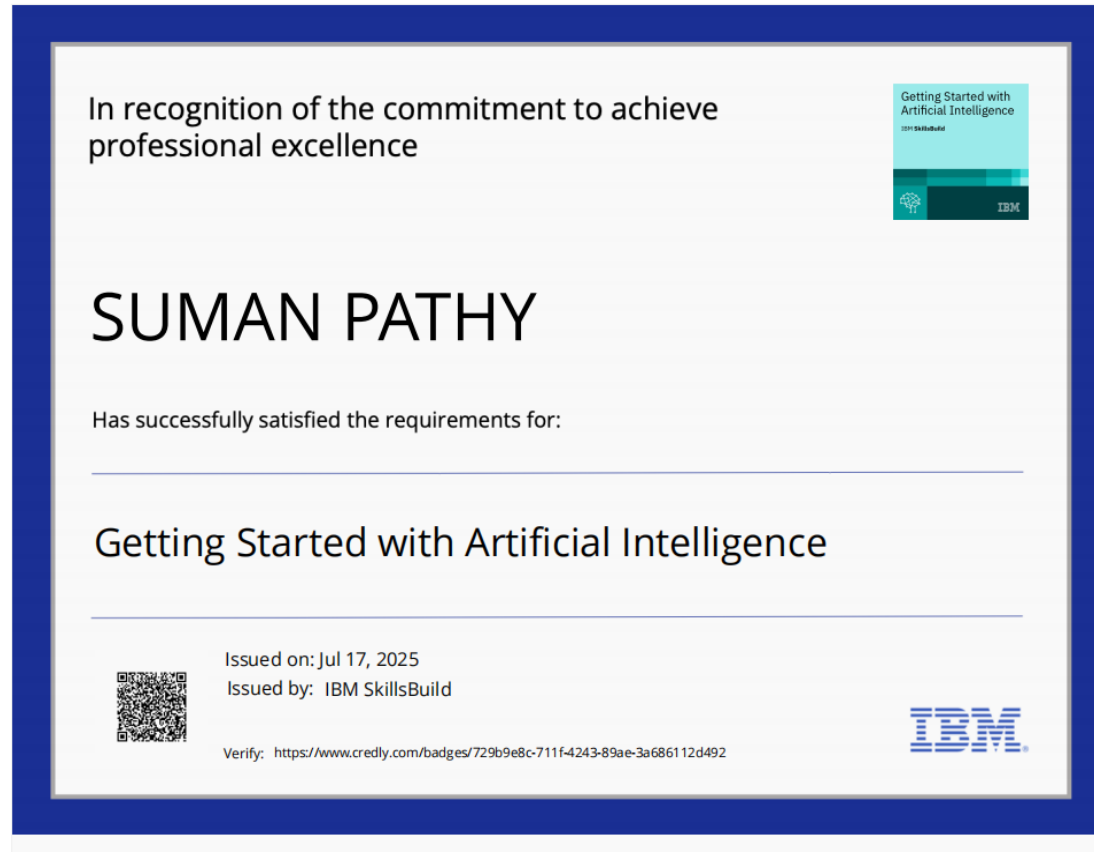
- **Real-Time Intrusion Detection**

Integrate with live network monitoring tools and IBM Event Streams to detect and respond to threats in real time, enhancing the system's practical applicability.

- **Explainable AI Integration**

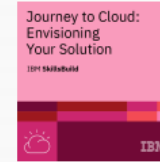
Use explainability tools such as SHAP or LIME to make the model's decisions transparent, enabling better trust and understanding by security analysts.

IBM CERTIFICATIONS



IBM CERTIFICATIONS

In recognition of the commitment to achieve
professional excellence



SUMAN PATHY

Has successfully satisfied the requirements for:

Journey to Cloud: Envisioning Your Solution

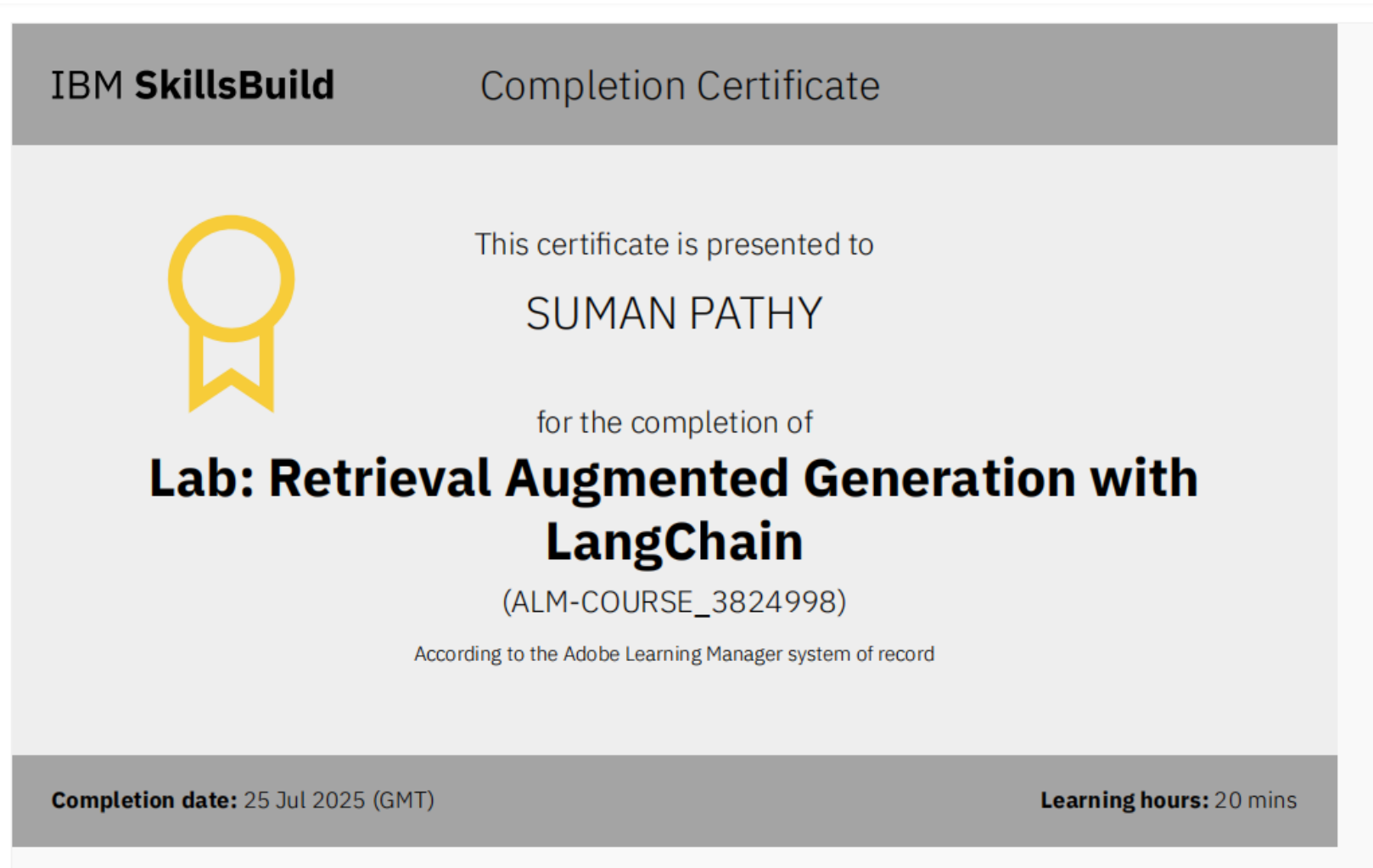


Issued on: Jul 20, 2025
Issued by: IBM SkillsBuild

Verify: <https://www.credly.com/badges/7abbab08-fc88-408d-9454-a952c7104882>



IBM CERTIFICATIONS





THANK YOU