

---

# CAPSTONE PROJECT

## NETWORK INTRUSION DETECTION

**Presented By:**

**1. Aditya Singh – Graphic Era Hill University – B Tech CSE**

# 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 increasing reliance on digital communication, networks are becoming frequent targets for a variety of cyber-attacks. Traditional intrusion detection systems often rely on static rules and signatures, making them ineffective against new or evolving threats.
- This project aims to build a Machine Learning-based Network Intrusion Detection System (NIDS) capable of analyzing network traffic to automatically detect and classify different types of attacks such as DoS, Probe, R2L, and U2R. The system will also distinguish between normal and malicious activity with high accuracy.
- By leveraging ML models, the goal is to create a scalable, adaptive, and efficient defense mechanism that provides early warning signs of network intrusions, helping secure systems against potential threats.

# PROPOSED SOLUTION

The proposed solution aims to detect anomalous network behavior using a binary classification model developed in IBM Watsonx.ai Studio. By leveraging AutoAI, the system automates the pipeline building process — including data preprocessing, model selection, feature transformation, and hyperparameter optimization — to create a scalable, high-accuracy Network Intrusion Detection System (NIDS).

## ■ Data Collection:

- Utilized a labeled network intrusion dataset (train\_data.csv) from Kaggle, containing both normal and attack traffic instances.
- The dataset includes features such as protocol type, service, flag, src\_bytes, and many others representing connection behaviors.

## ■ Data Preprocessing:

- Cleaned and preprocessed the data to handle:
  - Categorical variables (protocol\_type, service, flag) through encoding
  - Feature scaling of numerical attributes
- Labelled the target column (class) into binary categories: **normal** and **anomaly**.

## ■ Machine Learning Algorithm:

- Used IBM Watsonx.ai AutoAI, which automatically Performs feature Engineering, hyper optimization and generating 8 ML pipelines.
- Top algorithms used include:

1. Snap Decision Tree Classifier	2. Random Forest Classifier
----------------------------------	-----------------------------
- The best model (Pipeline 2) achieved **99.5% cross-validation accuracy**.

# PROPOSED SOLUTION

- **Deployment:**

- The best pipeline is ready for deployment as a REST API through IBM Cloud Lite.
- Enables integration with network monitoring tools for real-time intrusion detection and alerting.

- **Evaluation:**

- Performance was evaluated using:
  - Confusion matrix (99.9% precision for normal, 99.7% for anomaly)
  - Cross-validation accuracy of 99.5%

- **Result:**

- Pipeline 2 using Snap Decision Tree Classifier and HPO-1 achieved:
  - Accuracy (CV): 99.5%
- Model is highly accurate and suitable for real-time, scalable intrusion detection applications.

# SYSTEM APPROACH

The System Approach section outlines the complete setup for building and deploying the Machine Learning-based Network Intrusion Detection System (NIDS) using IBM's cloud ecosystem.

This includes data handling, model development, and real time deployment infrastructure.

- **System requirements**

- Platform: IBM Watsonx.ai Studio
- Deployment: IBM Cloud Lite
- IBM Watsonx Runtime
- IBM Watsonx Storage (Bucket)
- No local hardware needed (cloud-based)

- **Library required to build the model**

- AutoAI (Watsonx)
- pandas, numpy – Data handling
- scikit-learn – Model training
- Watson Machine Learning – Model deployment

# ALGORITHM & DEPLOYMENT

- In the Algorithm section, describe the machine learning algorithm chosen for detecting network intrusions. Here's an example structure for this section:
- **Algorithm Selection:**
  - The machine learning algorithm selected for this project is the **Snap Decision Tree Classifier**, an optimized version of the standard Decision Tree. It was automatically chosen by IBM Watsonx.ai AutoAI, which evaluated multiple models and pipelines to determine the best performer. This algorithm was selected due to its high interpretability and a perfect fit for detecting anomalies in structured network traffic data.
- **Data Input:**
  - The model was trained using features extracted from the network traffic dataset. Input variables included:
    - **Basic connection information:** duration, protocol type, service, flag.
    - **Traffic content features:** number of failed login attempts, source bytes, destination bytes.
    - **Statistical summaries:** connection count, same host rate, same service rate, etc.
  - The target column was the class label, consisting of:
    - "normal" — representing benign traffic
    - "anomaly" — representing any malicious or suspicious activity
  - All attack types were grouped under the "anomaly" class to simplify into a **binary classification** task.

# ALGORITHM & DEPLOYMENT

- **Training Process:**

- Training was conducted using **historical network traffic data** within IBM Watsonx.ai Studio. AutoAI handled:
  - **Data preprocessing**, including encoding and scaling
  - **Cross-validation**, to ensure model generalization
  - **Hyperparameter optimization**, to fine-tune parameters like tree depth and splitting criteria

- **Prediction Process:**

- After training, the model predicts whether new traffic records are normal or anomalies. Each prediction includes:
  - A **binary label** (normal or anomaly)
  - A **confidence score** indicating the certainty of the prediction
- The system supports both **batch and real-time predictions**, making it suitable for live network monitoring and automated threat detection in production environments.



# CONFUSION MATRIX

Pipeline details

Pipeline 2 ▼

Rank

1

Accuracy (Optimized)

0.998 (Holdout)

Algorithm

Snap Decision Tree Classifier

Enhancements

HPO-1

Save as

Model viewer

Model information

Feature summary

Evaluation

Model evaluation

Confusion matrix

Precision recall

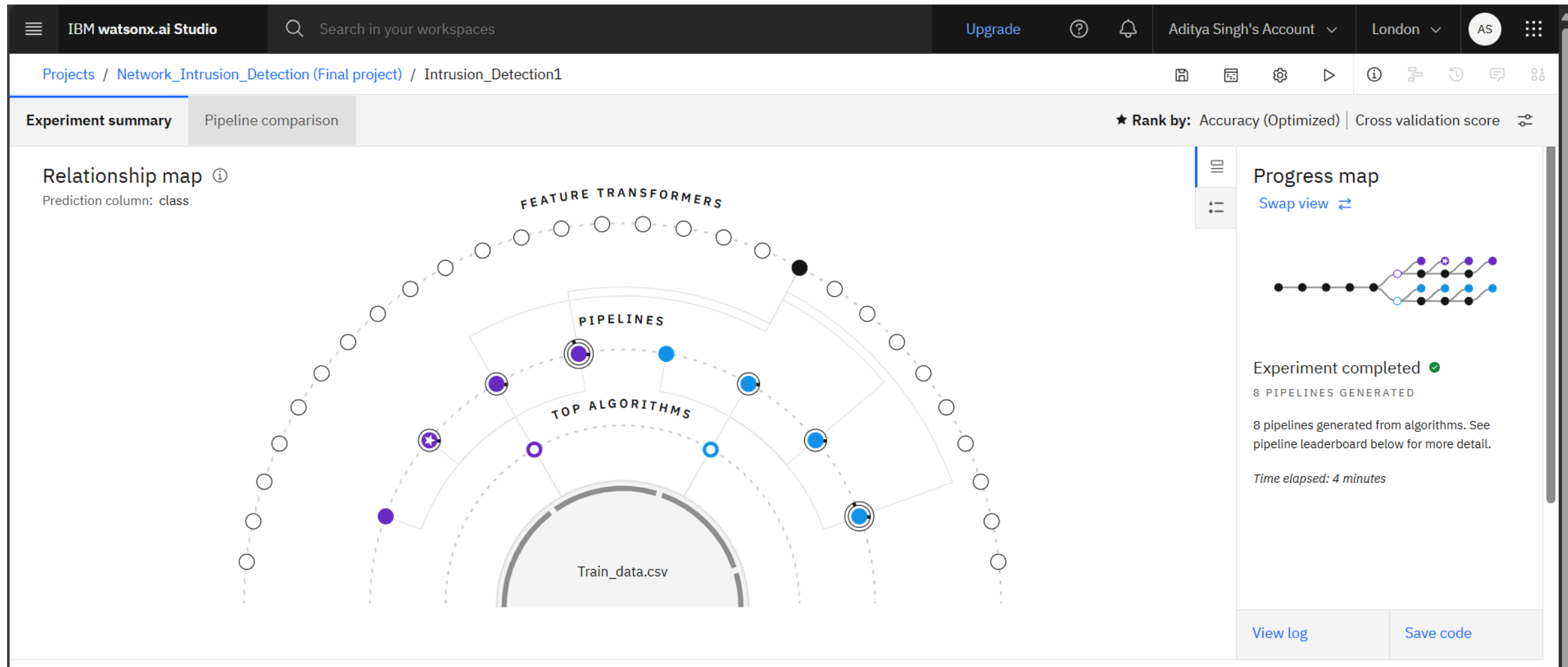
Confusion matrix ⓘ

Observed	Predicted		
	normal	anomaly	Percent correct
normal	1343	2	99.9%
anomaly	4	1171	99.7%
Percent correct	99.7%	99.8%	99.8%

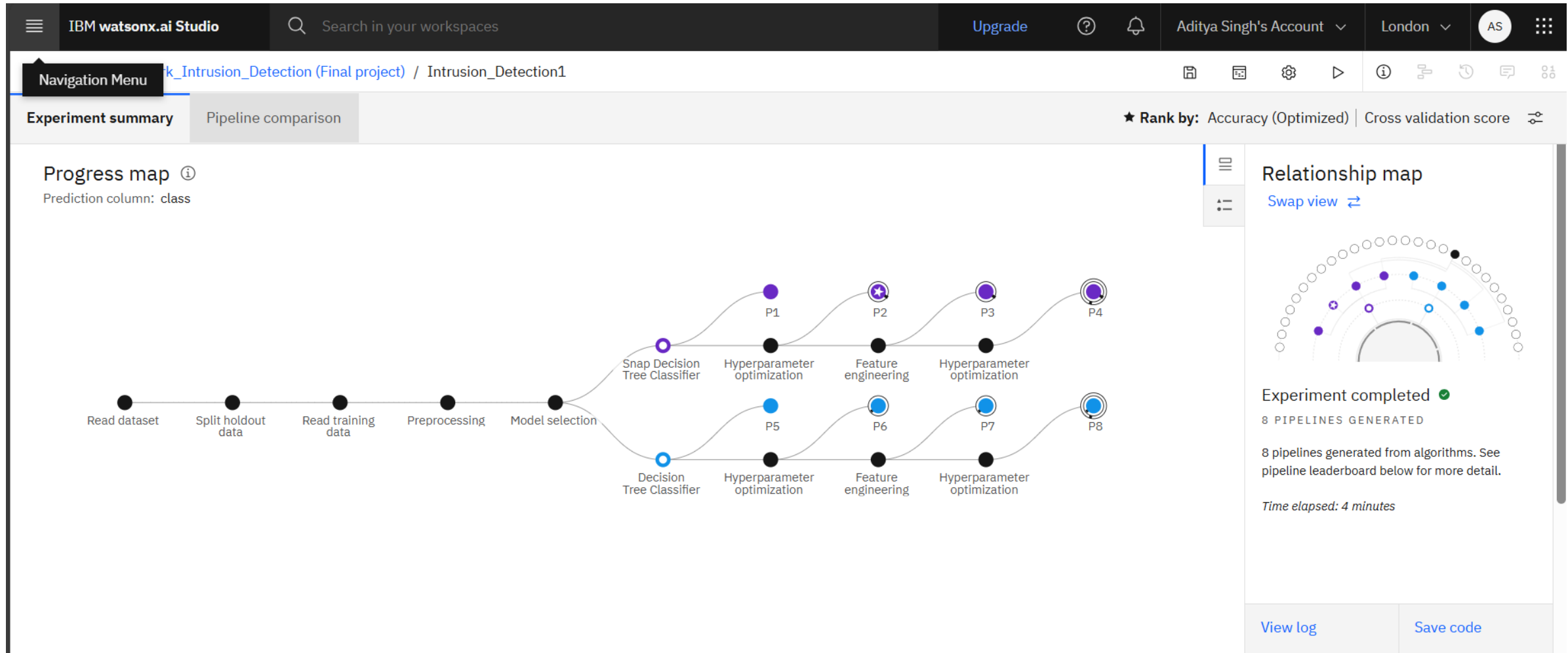
Less correct

More correct


# RELATIONSHIP MAP







# PROGRESS MAP

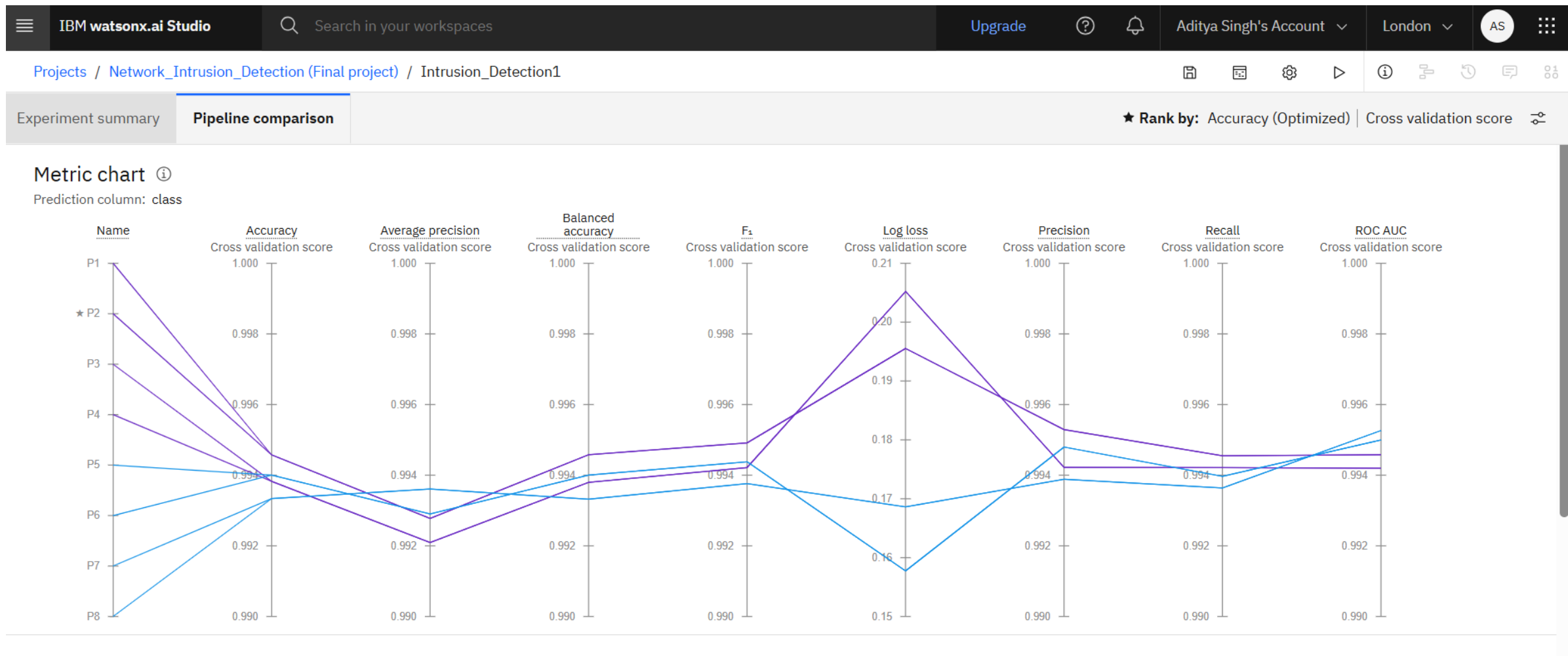


# PIPELINE LEADERBOARD

Pipeline leaderboard 

	Rank	↑	Name	Algorithm	Accuracy (Optimized) Cross Validation	Enhancements	Build time
★	1		Pipeline 2	 Snap Decision Tree Classifier	0.995	HPO-1	00:00:11
	2		Pipeline 1	 Snap Decision Tree Classifier	0.995	None	00:00:05
	3		Pipeline 6	 Decision Tree Classifier	0.994	HPO-1	00:00:09
	4		Pipeline 5	 Decision Tree Classifier	0.994	None	00:00:04

# METRIC CHART

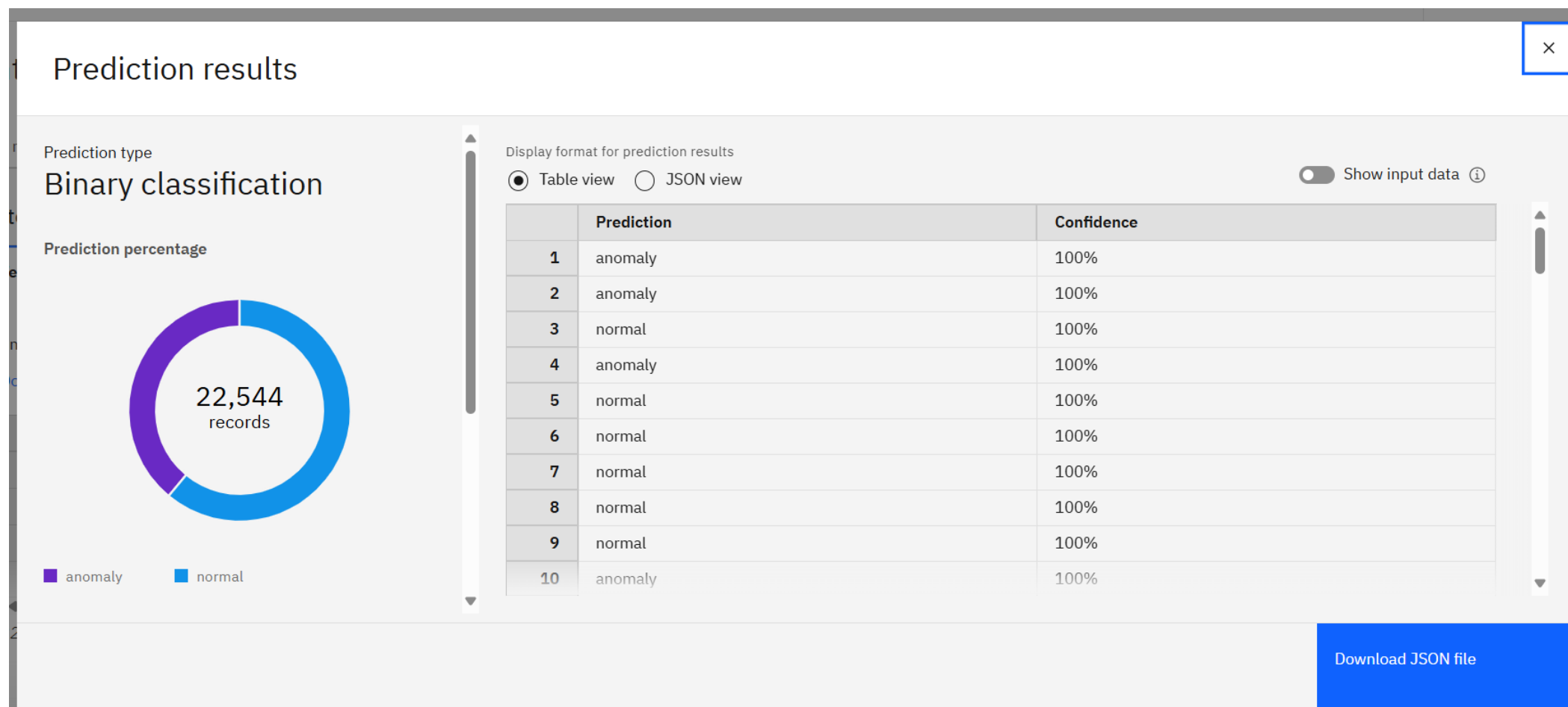


# RESULT

- **Best Model:**
  - Snap Decision Tree Classifier
  - Training Accuracy: 99.5% (via cross-validation)
- **Prediction Results:**
  - Correctly classified "normal" and "anomaly" traffic
  - Confidence scores up to 100% on holdout data
- **Visualization:**
  - Class distribution pie chart
  - Leader board comparing multiple model pipelines
- **Deployment-Ready :**
  - Fast and accurate predictions through IBM Cloud Lite.

# PREDICTION RESULTS

- Results of test.csv



# CONCLUSION

- Successfully implemented a binary classification-based Intrusion Detection System using IBM Watsonx.ai AutoAI.
- Utilized the Snap Decision Tree Classifier, which provided high accuracy and fast prediction for real-time anomaly detection.
- Trained the model on a labeled network traffic dataset with a mix of categorical and numerical features.
- AutoAI streamlined the process through automated preprocessing, model selection, and hyperparameter tuning.
- The deployed model is capable of detecting normal vs anomalous network activity, with a confidence score for each prediction.
- The solution is scalable and deployable on IBM Cloud, suitable for both batch and real-time network monitoring.
- Demonstrates the effectiveness of machine learning in enhancing cybersecurity through early detection of potential threats.




# FUTURE SCOPE

- **Multiclass Classification:**
  - Extend from binary to multiclass detection (e.g., DoS, Probe, R2L, U2R) for more detailed attack identification.
- **Real-time Data Integration:**
  - Incorporate live network traffic streams for continuous intrusion monitoring and faster response.
- **Model Enhancement:**
  - Experiment with advanced algorithms like Random Forest, XGBoost, or Deep Learning (LSTM, CNN) for improved accuracy.
- **Threat Intelligence Integration:**
  - Combine with external threat databases to detect emerging and zero-day attacks.
- **Explainable AI (XAI):**
  - Add interpretability layers to explain why certain traffic is flagged as anomalous.
- **Cloud Security Extension:**
  - Adapt and deploy the IDS in hybrid/multi-cloud environments for broader coverage.

# API PREFERENCES

Deployment spaces / Intrusion\_Detect1 / P2 - Snap Decision Tree Classifier: Intrusion\_Detection1 /


     

Intusion\_detect2  Deployed Online

API reference Test

## Endpoints for scoring

Private endpoint


<https://private.eu-gb.ml.cloud.ibm.com/ml/v4/deployments/2d58e5ca-d7bd-4b41-97c3-58f91f87bc0a/predictions?version=2021-05-01> 

Public endpoint

<https://eu-gb.ml.cloud.ibm.com/ml/v4/deployments/2d58e5ca-d7bd-4b41-97c3-58f91f87bc0a/predictions?version=2021-05-01> 

[Learn more](#) about the 2021-05-01 version query parameter

## Code snippets

Bearer <token> 

IAM

## About this deployment

**Name** 

Intusion\_detect2

**Description** 

No description provided.

### Deployment Details

Deployment ID: 2d58e5ca-d7bd-4b...

Serving name: 

No serving name.

Software specification: 


hybrid\_0.1 

Hybrid pipeline software specifications:

autoai-kb\_rt24.1-py3.11

Copies: 

1

**Tags** 

Add tags to make assets easier to find.

# REFERENCES

- IBM Watsonx.ai Studio Documentation(AutoAI and Model Deployment on IBM Cloud)
  - <https://dataplatform.cloud.ibm.com>
- IBM Cloud Lite(Free-tier platform for deploying AI models and services)
  - <https://www.ibm.com/cloud/lite>
- Network Intrusion Detection Dataset – Kaggle
  - [Click me](#)

# IBM CERTIFICATIONS

In recognition of the commitment to achieve professional excellence



Aditya Singh

Has successfully satisfied the requirements for:

Getting Started with Artificial Intelligence



Issued on: Jul 19, 2025

Issued by: IBM SkillsBuild

Verify: <https://www.credly.com/badges/05033827-9503-4ea9-8b36-652e749f2b22>



# IBM CERTIFICATIONS

In recognition of the commitment to achieve professional excellence



Aditya Singh

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/b4976a9d-dc71-4466-85c5-f03e5bae69ae>




# IBM CERTIFICATIONS

7/23/25, 6:39 PM

Completion Certificate | SkillsBuild

IBM SkillsBuild

Completion Certificate



This certificate is presented to

Aditya Singh

for the completion of

**Lab: Retrieval Augmented Generation with LangChain**

(ALM-COURSE\_3824998)

According to the Adobe Learning Manager system of record

Completion date: 19 Jul 2025 (GMT)

Learning hours: 20 mins

[https://skills.yourlearning.ibm.com/certificate/ALM-COURSE\\_3824998](https://skills.yourlearning.ibm.com/certificate/ALM-COURSE_3824998)

1/1



**THANK YOU**