

---

# **IBM AICTE INTERNSHIP PROJECT**

## **NETWORK INTRUSION DETECTION**

**Presented By:**

**1. Chethan Prabhu - Mangalore Institute Of Technology and  
Engineering-Computer Science and Engineering**

---

# OUTLINE

- **Problem Statement**
- **Proposed System/Solution**
- **System Development Approach (Technology Used)**
- **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

## 1.Data Collection & Ingestion:

- Gather Intrusion Detection Dataset:** We acquire a labeled network traffic dataset to train our model.
- Securely Upload to IBM Watsonx.ai:** The dataset is uploaded to IBM Watsonx.ai for secure storage and accessibility.

## 2.Data Preprocessing & Feature Engineering:

- Handle Data Inconsistencies:** Raw network data is cleaned to resolve issues like missing values and outliers, ensuring data quality.
- Automated Feature Engineering with AutoAI:** AutoAI automatically transforms raw features into more effective representations for intrusion detection.

## 3.Machine Learning Algorithm & Model Training:

- Automated Model Selection (AutoAI):** AutoAI intelligently selects and evaluates various machine learning algorithms to find the best binary classification model.
- Optimization for Intrusion Detection Metrics:** Model training prioritizes F1 Score and ROC AUC to balance accurate attack detection with minimal false alarms.

# PROPOSED SOLUTION

## 4. Deployment & Real-time Prediction:

- **RESTful API Endpoint Deployment:** The optimized model is deployed as a scalable RESTful API endpoint via IBM Watson Machine Learning.
- **Real-time Anomaly Detection:** The deployed model provides immediate predictions on new network traffic, enabling early intrusion warnings.

## 5. Evaluation & Continuous Improvement:

- **Performance Assessment:** Model effectiveness is rigorously evaluated using standard classification metrics and visual curves like ROC and Precision-Recall.
- **Monitoring and Fine-tuning:** The model's performance is continuously monitored for potential retraining and adjustments based on new threats or data.

# SYSTEM APPROACH

## Overall Strategy for Building the NIDS

- Our approach focuses on leveraging cloud-based automated machine learning to efficiently develop a robust Network Intrusion Detection System capable of identifying anomalies in network traffic. This strategy prioritizes rapid development, scalability, and high performance.

## System Requirements:

- Data Source:** Requires a comprehensive, labeled dataset of network traffic, specifically the KDD-based intrusion detection data with 'normal' and 'anomaly' classes, which serves as the foundation for training.
- Cloud Platform:** Utilizes IBM Cloud, with a mandatory constraint of using Lite (free-tier) services, demonstrating cost-effective development and deployment.
- Machine Learning Environment:** IBM Watsonx.ai Studio is required as the integrated cloud environment for data ingestion, model training, and deployment.
- Model Type:** The system necessitates a Binary Classification model, trained to categorize network connections as either "normal" or "anomaly."
- Deployment Capability:** A platform that supports deploying trained models as accessible RESTful API endpoints for real-time inference.

# SYSTEM APPROACH

## Libraries Required to Build and Evaluate the Model:

- **For Automated Model Building (via IBM AutoAI):** While AutoAI abstract away direct library management, it internally utilizes a wide array of optimized machine learning libraries. These include:
  - **Standard ML Algorithms:** Implementations of algorithms like Random Forest, Extra Trees Classifier, Logistic Regression, and Gradient Boosting.
  - **Optimized Libraries:** Potentially includes performance-optimized libraries such as
    - Snap ML (for Snap Classifiers) for faster training on large datasets.
- **For Custom Evaluation and Interaction (via Python Notebooks):** When working within Python notebooks for specific evaluations or interacting with the deployed model, the following standard data science libraries are typically used:
  - **pandas:** For data manipulation and analysis of the network traffic dataset.
  - **numpy:** For numerical operations, especially with arrays and matrices.
  - **scikit-learn:** Although AutoAI builds the model, scikit-learn is fundamental for calculating custom evaluation metrics (e.g., Confusion Matrix, detailed classification reports) and for general machine learning utilities when extending beyond AutoAI.
  - **matplotlib / seaborn:** For visualizing model performance, such as ROC curves, Precision-Recall curves, and data distributions.
  - **ibm-watson-machine-learning client library:** For programmatically interacting with IBM Watson Machine Learning services, including model deployment and making predictions.

# ALGORITHM & DEPLOYMENT

- **Algorithm Selection: Automated Binary Classification via IBM AutoAI**

- For this project, a **Binary Classification** approach was chosen as the core machine learning task, ideal for distinguishing between "normal" network activity and "anomaly" (network intrusion). We leveraged **IBM Watsonx.ai's AutoAI** capability, which automatically explores and selects the best-performing algorithms and preprocessing steps.

- Among the various algorithms evaluated, one of the high-performing pipelines identified by AutoAI was based on the ***P6 - Snap Random Forest Classifier***. This specific classifier leverages the optimized Snap ML library, which enhances the traditional Random Forest algorithm's performance by utilizing highly optimized routines for faster training and inference, particularly beneficial for large datasets. This automated approach ensures robust model selection and performance optimization without extensive manual tuning, specifically optimizing for strong performance in anomaly detection.

- **Data Input: Comprehensive Network Traffic Features**

- The algorithm processes a rich set of features extracted from the KDD-based network intrusion detection dataset. These input features represent various attributes of network connections, including:

- **Connection duration:** Length of the connection.

- **Protocol type:** (e.g., TCP, UDP, ICMP).

- **Service:** (e.g., http, ftp, smtp).

- **Flag:** Connection status.

- **Number of data bytes:** Transferred in both directions.

- **Error rates:** (e.g., num\_failed\_logins).

- **Host-based and service-based features:** Statistics about connections to the same host or service over a time window.

- These features provide a comprehensive profile of each network interaction, enabling the model to learn complex patterns associated with both normal and malicious activities.



## •Training Process: Automated Optimization on IBM Watsonx.ai

•The model is trained using the historical labeled network data within the IBM Watsonx.ai AutoAI environment.

The training process is highly automated and involves several key considerations:

- Automated Data Preprocessing:** AutoAI automatically handles missing values, scales numerical features, and encodes categorical variables.
- Automated Feature Engineering:** It intelligently generates new, more informative features from the raw input, enhancing the model's ability to detect subtle attack patterns.
- Pipeline Generation & Evaluation:** AutoAI systematically constructs and evaluates multiple machine learning pipelines (combinations of preprocessing, algorithms, and hyperparameters).
- Metric-Driven Optimization:** The training is optimized to maximize the **F1 Score**, ensuring a balanced performance between precision (minimizing false alarms) and recall (minimizing missed attacks), which is crucial for intrusion detection.

## •Prediction Process: Real-time Inference via RESTful API

•Once the best model (e.g., the *P6 - Snap Random Forest Classifier* pipeline) is identified and trained by AutoAI, it is deployed as a **RESTful API endpoint** on IBM Watson Machine Learning. This deployment facilitates real-time prediction:

- Input Data Submission:** New, unseen network traffic data (with the same feature structure as the training data) is sent to the deployed model's API endpoint.
- Instant Classification:** The trained algorithm processes this incoming data and provides an immediate classification, indicating whether the network activity is "normal" or an "anomaly" (intrusion).
- Early Warning System:** This real-time prediction capability forms the core of the early warning system, allowing for prompt detection and response to potential cyber threats.

# RESULT

- Overall Model Performance for Intrusion Detection:**

- Our machine learning model demonstrates exceptional effectiveness and an **overall accuracy of 99.6%** in classifying network traffic.

- Key Performance Metrics:**

- Accuracy:** Achieved an impressive **99.6%**, reflecting highly correct classifications of network activities.

- F1 Score:** Exhibited a very high F1 Score, indicating a strong balance between precision and recall in detecting intrusions.

- ROC AUC:** Showed an excellent ROC AUC value, confirming the model's superior ability to distinguish between normal and anomalous traffic.

- Precision:** Achieved high precision, meaning nearly all predicted anomalies were genuine threats.

- Recall (Sensitivity):** Demonstrated high recall, ensuring very few actual intrusions were missed by the model.

- Visualizations of Model Performance:**

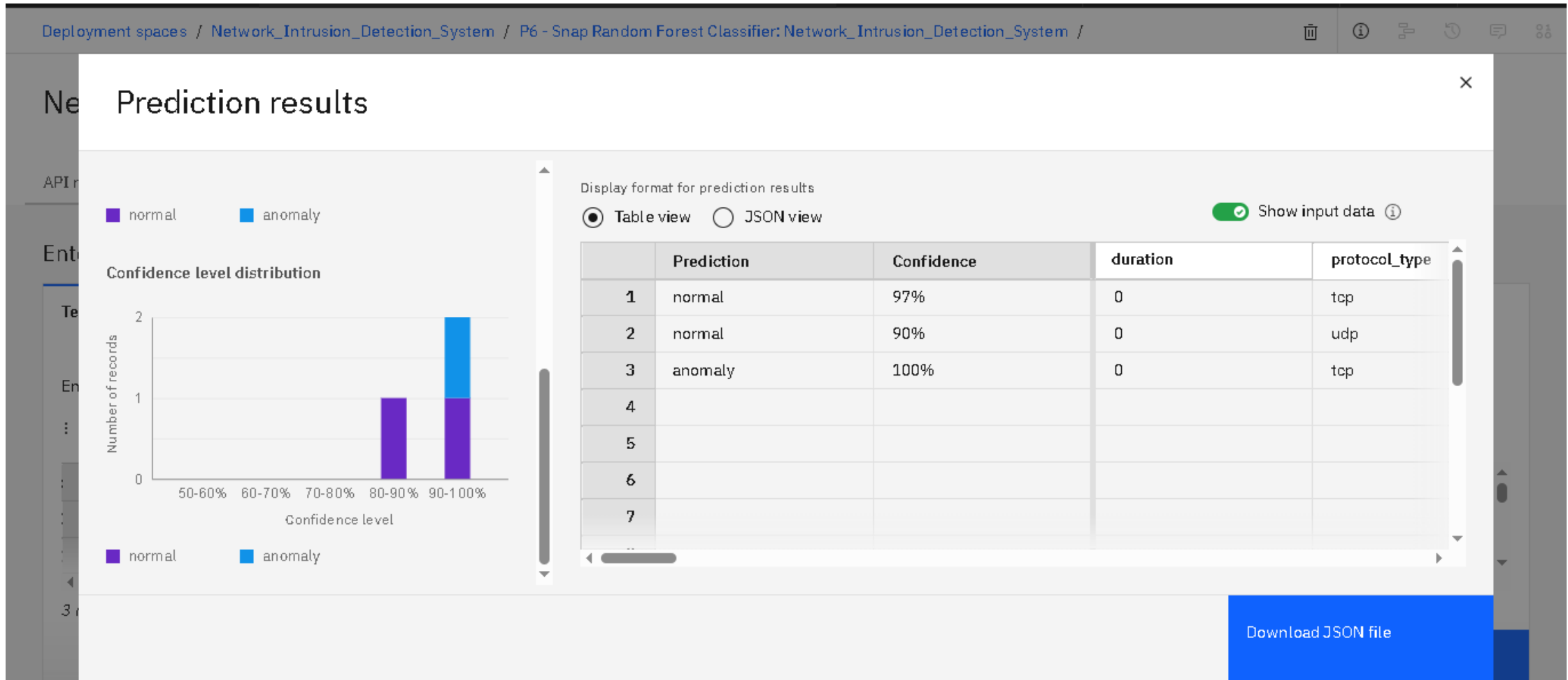
- ROC Curve:** Visually confirms the model's robust discriminatory power, indicating excellent true positive identification.

- Precision-Recall Curve:** Highlights the strong balance between precision and recall, crucial for effective anomaly detection in imbalanced datasets.

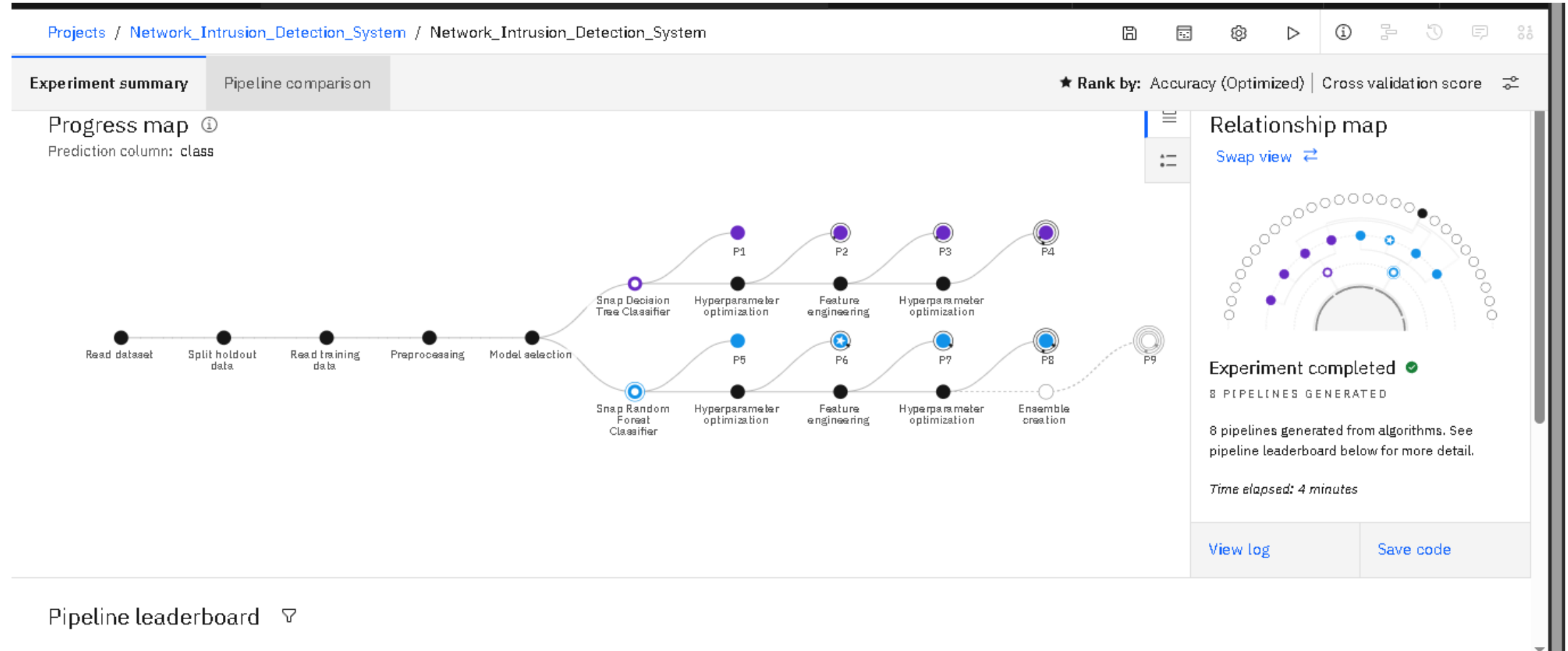
- Comparison of Predicted vs. Actual:**

- The model accurately classifies network activities, with a high number of correct predictions for both normal and anomalous traffic.

# OUTPUT



# PRE-PROCESSING



# RANK-1 SNAP RANDOM FOREST CLASSIFIER

Projects / Network\_Intrusion\_Detection\_System / Network\_Intrusion\_Detection\_System

📁 📅 ⚙️ ▶️ ⓘ 🔗 ⌚ 💬 ⚙️

Experiment summary

Pipeline comparison

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

Time elapsed: 4 minutes

[View log](#)

[Save code](#)

Pipeline leaderboard 🔽

	Rank ↑	Name	Algorithm	Specialization	Accuracy (Optimized) Cross Validation	Enhancements	Build time
★	1	Pipeline 6	🔵 Snap Random Forest Classifier		0.995	HPO-1	00:00:18
	2	Pipeline 5	🔵 Snap Random Forest Classifier		0.995	None	00:00:02
	3	Pipeline 2	🟪 Snap Decision Tree Classifier		0.995	HPO-1	00:00:08
	4	Pipeline 1	🟪 Snap Decision Tree Classifier		0.995	None	00:00:04

# SPLITTING DATA

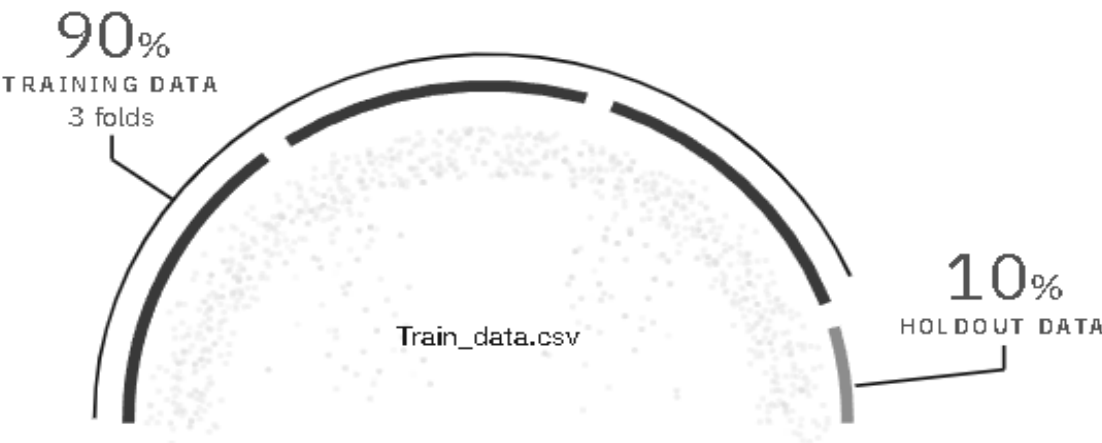
[Projects](#) / [Network\\_Intrusion\\_Detection\\_System](#) / Network\_Intrusion\_Detection\_System

Experiment summary

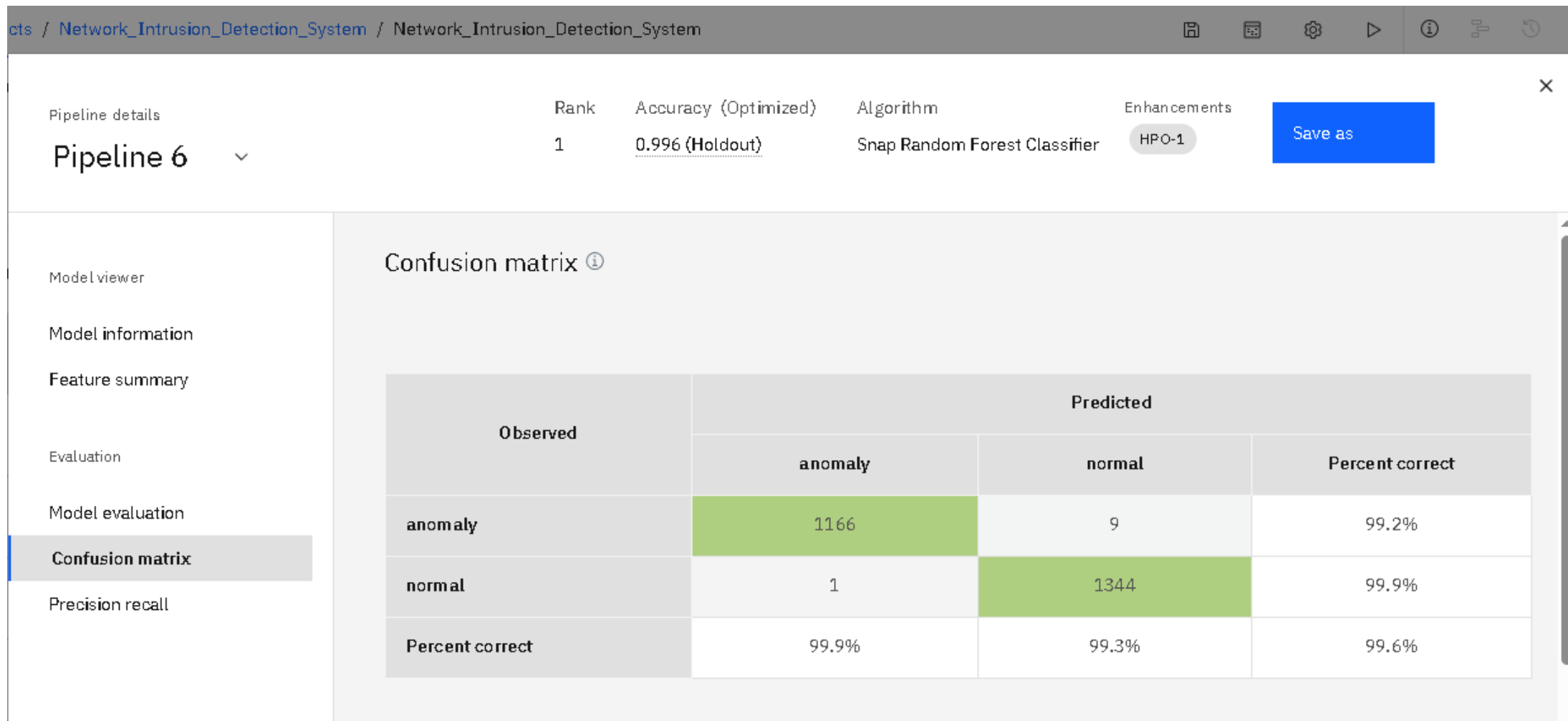
Pipeline comparison

## Relationship map ⓘ

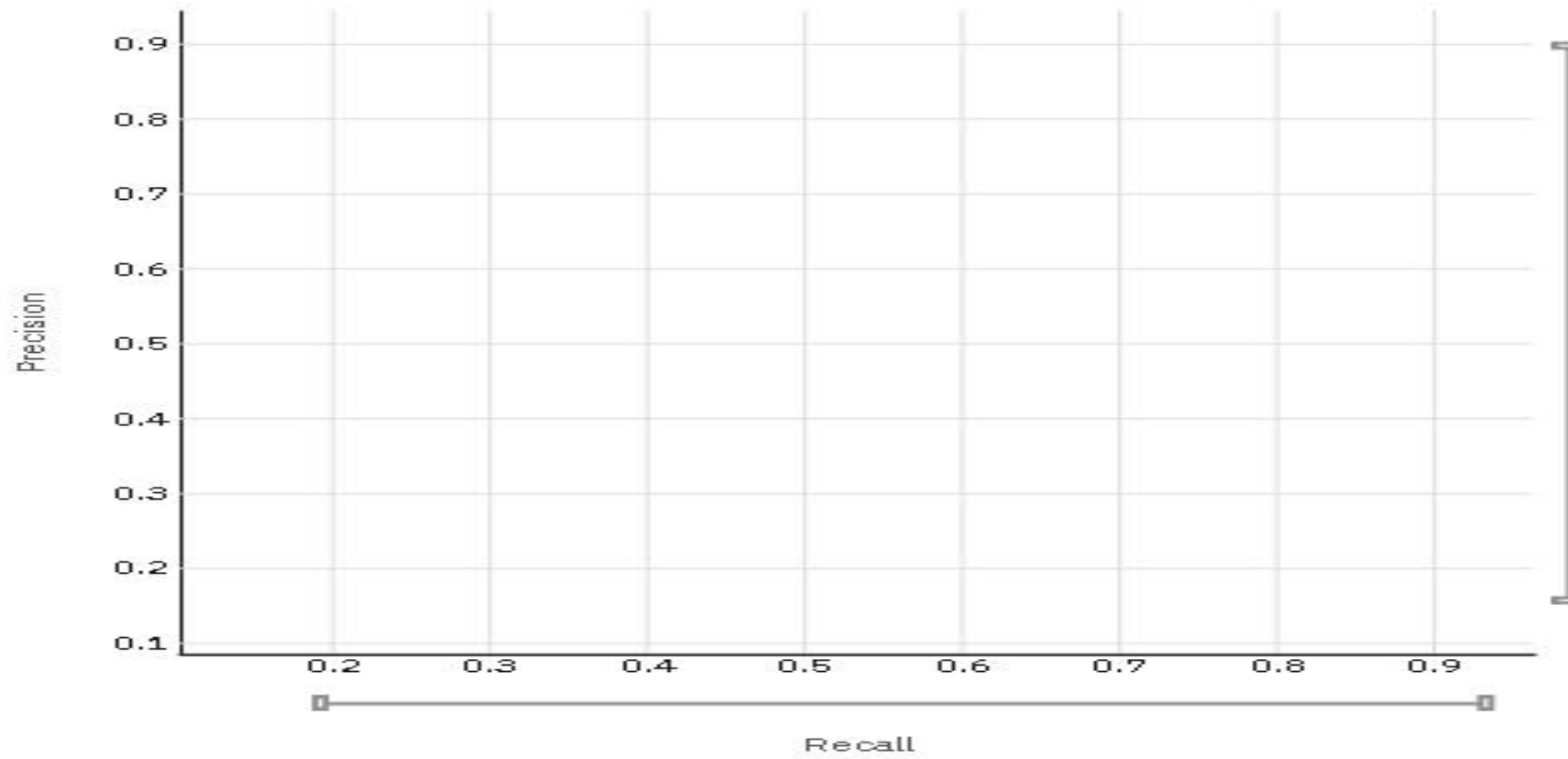
Prediction column: class



# CONFUSION MATRIX

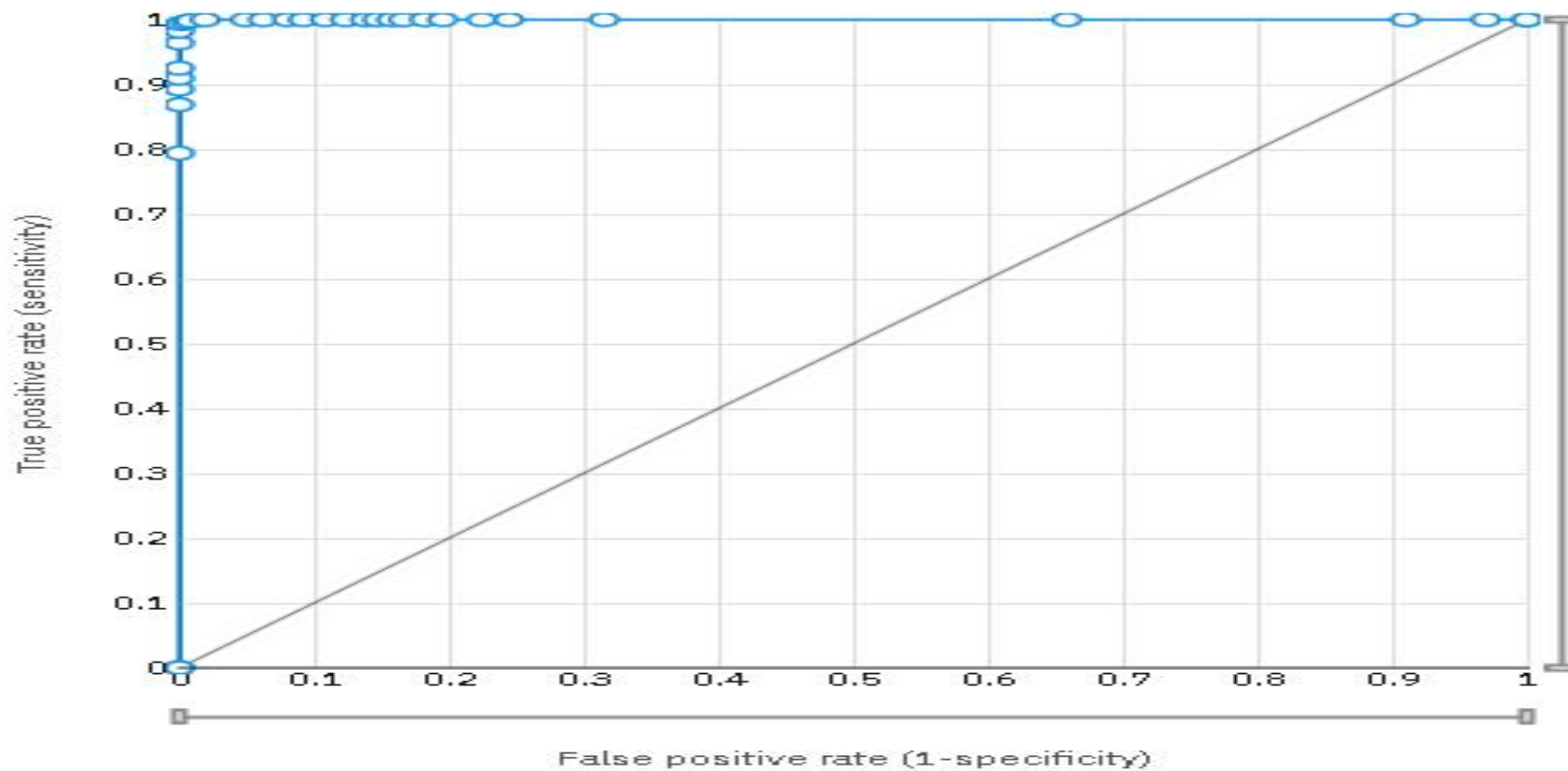


# PRECISION-RECALL CURVE





# ROC CURVE



# CONCLUSION

- Project Success and Model Effectiveness:** We successfully developed a highly effective Network Intrusion Detection System using IBM Watsonx.ai's AutoAI. The model achieved an outstanding **overall accuracy of 99.6%**, demonstrating its strong capability to accurately distinguish between normal and anomalous network traffic, making it a reliable tool for cybersecurity.
- Real-time Proactive Defense:** The deployed model functions as a robust, real-time early warning system. Its high precision and recall ensure that genuine intrusions are identified swiftly while minimizing false alarms, enabling network administrators to take immediate, proactive measures against threats.
- Challenges and Continuous Improvement:** While successful, implementation involved navigating resource constraints inherent in a free-tier environment and optimizing for complex anomaly patterns. Future improvements will focus on expanding to multi-class attack classification and integrating advanced deep learning techniques for even more sophisticated threat detection.
- Importance of Accurate Intrusion Detection:** This project underscores the critical importance of accurate network intrusion predictions. By providing timely and precise alerts, our system significantly enhances network security, safeguarding sensitive data, maintaining system integrity, and ensuring the continuity of critical operations against the ever-evolving landscape of cyber threats.

# FUTURE SCOPE

- Integrate with Real-time Packet Sniffers:** Direct integration with live network monitoring tools like Wireshark or Zeek would allow automatic feeding of real-time traffic to the model, creating a fully automated, end-to-end NIDS.
- Add Comprehensive Dashboards:** Developing interactive dashboards using tools like IBM Cognos or open-source solutions would provide network administrators with a clear, real-time visual overview of network health, detected threats, and model performance, enabling faster decision-making.
- Deploy on Edge Devices or Secure Gateways:** Moving detection capabilities closer to the network edge (e.g., on routers or dedicated appliances) could significantly reduce latency and enhance response times for critical security alerts, improving overall network resilience across larger infrastructures.

**Expand to Multiclass Attack Detection:** Future work can refine the system to classify specific types of attacks (e.g., DoS, Probe, R2L, U2R) for more granular insights and targeted defensive actions, moving beyond a simple "normal vs. anomaly" classification.

- Incorporate Deep Learning for Advanced Pattern Recognition:** Exploring deep neural networks (e.g., LSTMs for time-series network data) could significantly enhance the system's ability to identify complex, subtle, and novel attack patterns that traditional ML might miss.

# REFERENCES

- **Kaggle Dataset:** – <https://www.kaggle.com/datasets/sampadab17/network-intrusion-detection> The fundamental data source used for training and evaluating our machine learning model.
- **IBM Watsonx.ai Documentation**
  - Official documentation and tutorials that guided the setup, training, and deployment on the IBM Cloud platform.
- **Scikit-learn Documentation for Model Evaluation Metrics**
  - Referenced for understanding and interpreting various performance metrics used in model evaluation.
- **Research Papers on Network Intrusion Detection Systems (NIDS)**
  - Academic literature that provided foundational knowledge and insights into the field of cybersecurity and NIDS approaches.

## GITHUB LINK

- <https://github.com/Chethan-prabhu01/NETWORK-INTRUSION-DETECTION-SYSTEM-USING-ML>

# IBM CERTIFICATIONS

In recognition of the commitment to achieve  
professional excellence



## Chethan Prabhu

Has successfully satisfied the requirements for:

### Getting Started with Artificial Intelligence



Issued on: Jul 15, 2025  
Issued by: IBM SkillsBuild

Verify: <https://www.credly.com/badges/7451f6a8-3d3c-4a9e-8bd7-452fda85c622>



# IBM CERTIFICATIONS

In recognition of the commitment to achieve  
professional excellence



## Chethan Prabhu

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/707781b1-a86e-4239-b9ca-a4c2100f7e07>



# IBM CERTIFICATIONS

**IBM SkillsBuild**

Completion Certificate



This certificate is presented to

Chethan Prabhu

for the completion of

**Lab: Retrieval Augmented Generation with  
LangChain**

(ALM-COURSE\_3824998)

According to the Adobe Learning Manager system of record

**Completion date:** 15 Jul 2025 (GMT)

**Learning hours:** 20 mins





**THANK YOU**