CAPSTONE PROJECT

NETWORK INTRUSION DETECTION SYSTEM

Presented By:

1. SHYAMILY HARIDAS –Department of Electronics Engineering

edunet foundation

OUTLINE

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

PROBLEM STATEMENT

In an era of increasingly complex and interconnected digital ecosystems, communication networks face a persistent and evolving threat from a variety of cyber-attacks. These malicious activities, including high-volume Denial-of-Service (DoS) attacks that cripple services, stealthy port scans (Probe), sophisticated unauthorized remote-to-local (R2L) access attempts, and internal privilege escalation (U2R), threaten the confidentiality, integrity, and availability of critical data. Traditional, signature-based intrusion detection systems often struggle to keep pace with novel and adaptive attack vectors.

The objective of this project is to create a robust, intelligent Network Intrusion Detection System (NIDS) utilizing a machine learning approach. By building a model that can analyze real-time network traffic data, the system will be able to accurately identify these diverse attack types, thereby providing an essential early warning mechanism to secure networks and bolster their overall resilience against modern cyber threats.

PROPOSED SOLUTION

An IBM Cloud-based NIDS

This project addresses the challenge of network intrusion detection by leveraging the automated machine learning capabilities of IBM Cloud. The solution is built to analyze network traffic data and provide accurate, real-time classifications of network activity, distinguishing between normal behavior and various types of cyber-attacks.

The proposed system will consist of the following components and steps:

1. Data Collection:

- The system utilizes the provided network traffic dataset (Train_data.csv) as the primary source of historical information on both normal and anomalous network connections.
- The dataset includes a comprehensive set of features such as connection duration, protocol type, service, and error rates, which are crucial for identifying malicious patterns.

2. Data Preprocessing:

- The raw data is prepared for machine learning using a series of automated steps performed by IBM Cloud's AutoAI service.
- This includes cleaning and preprocessing the collected data to handle any inconsistencies.
- Crucially, categorical features such as protocol type, service, and flag are automatically transformed using techniques like One-Hot Encoding to be compatible with the machine learning model.

PROPOSED SOLUTION – CONTD.

3. Machine Learning Algorithm:

The core of the NIDS is a classification model developed using IBM Cloud AutoAI. The optimal model identified by AutoAI is the **Snap Decision Tree Classifier**, which achieved a high accuracy of 0.998 on the holdout dataset.

a) **Algorithm Selection:** The Snap Decision Tree Classifier is an efficient and highly interpretable model that partitions the data into a series of rules. It is well-suited for this project due to its ability to handle both categorical and numerical features and its high performance on the provided dataset.

b) **Data Input:** The model uses the preprocessed features from the Train_data.csv file as input. This includes the transformed categorical features and scaled numerical features.

c) **Training Process:** The model was trained using IBM Cloud's AutoAI, which automatically performed hyperparameter optimization (HPO-1) to fine-tune the model's parameters and achieve the reported high accuracy.

d) **Prediction Process:** The trained model will take new, real-time network traffic data as input. It will then apply the learned decision tree rules to classify the connection as either 'normal' or 'anomaly'.

PROPOSED SOLUTION – CONTD.

4. Deployment

a) The trained Snap Decision Tree Classifier model, optimized by AutoAI, will be deployed as a scalable REST API endpoint using IBM Watson Machine Learning.

b) This deployment will enable the model to receive network traffic data and return a real-time classification (e.g., normal or anomaly).

c) The API endpoint provides a flexible interface for integration with other applications or a live network monitoring system, allowing the NIDS to function as an early warning system.

).

PROPOSED SOLUTION – CONTD.

5. Evaluation:

The performance of the Snap Decision Tree Classifier model is rigorously assessed on a separate, held-out test dataset to ensure its efficacy and generalization capabilities. The model's performance is measured using key classification metrics, including:

a) **Accuracy**: The model achieved an outstanding overall accuracy of **99.8%** on the dataset, as shown in the Model Eval - ROC Curve.

b) **Precision**: The model's ability to correctly identify positive classes (anomalies) is very high, with a perfect score of **1.000** on the Metric chart. The Precision Recall Curve also shows precision staying near the maximum value, meaning that when the model predicts an attack, it is almost always correct.

c) **Recall**: The system is highly effective at finding all relevant cases, achieving a perfect recall score of **1.000** on the Metric chart. This means the model successfully detected nearly all actual anomalous events within the dataset.

d) **F1-Score**: Providing a balanced view of both precision and recall, the F1-Score for the model is also **1.000**, as seen on the Metric chart. This perfect score highlights the model's consistent and robust performance.

e) **Confusion Matrix**: The Confusion Matrix provides a detailed breakdown of the model's predictions. Out of 2520 evaluation instances, the model correctly classified **1343 normal** connections and **1171 anomalous** connections. It exhibited a very low error rate with only **2 false positives** (normal traffic incorrectly labeled as an anomaly) and **4 false negatives** (anomalies incorrectly labeled as normal).

PROPOSED SOLUTION – CONTD.

6. Result:

The outcome is a highly effective, automated NIDS that can accurately predict and classify various network attacks, providing network administrators with a powerful tool for enhancing cybersecurity and ensuring the stability of their communication infrastructure.

SYSTEM APPROACH

This section outlines the overall strategy and methodology for developing and implementing the Network Intrusion Detection System (NIDS).

System Requirements

- **IBM Cloud Account:** Access to a valid IBM Cloud account with the Lite plan is required to utilize AutoAI and Watson Machine Learning.
- **Data Storage:** The training and testing datasets must be stored in an accessible location within the IBM Cloud environment, such as a Cloud Object Storage bucket.
- **Network Connectivity:** A reliable internet connection is necessary to interact with the IBM Cloud console and deploy the model.

Libraries and Tools Required to Build the Model

- **IBM Watson Studio:** The central hub on IBM Cloud for data science projects, providing the environment for AutoAI.
- **IBM AutoAI:** An automated machine learning tool that handles data preprocessing, model selection, and hyperparameter tuning.
- **IBM Watson Machine Learning:** A service for deploying the trained machine learning model as a REST API endpoint.
- **Jupyter Notebooks:** An interactive environment within Watson Studio for coding and documentation.

ALGORITHM & DEPLOYMENT

Based on the provided Progress map, Model Information, Feature Summary, and evaluation charts, the following algorithm and deployment plan are proposed for the Network Intrusion Detection System (NIDS).

1. Algorithm Selection

a) **Chosen Algorithm:** The Model Information and Progress map slides indicate that the **Snap Decision Tree Classifier** was selected as the final, optimized algorithm.

b) **Justification:** This algorithm was likely chosen after a series of experiments and optimizations (P1-P8 on the Progress map). The high accuracy of 99.8% on the holdout set, as shown in Model Eal - ROC Curve, along with the Precision Recall Curve showing a high precision and recall, suggests that this model performs exceptionally well at distinguishing between normal and anomalous network traffic. The Feature Summary also shows that the algorithm effectively leverages key features like src_bytes and service, which are highly correlated with intrusion events.

ALGORITHM & DEPLOYMENT – CONTD.

2. Data Input

The model is trained to analyze network traffic data, with a specific focus on the following features identified as important in the Feature Summary:

a) **src_bytes:** The number of bytes transferred from the source to the destination.

b) **service:** The type of network service, such as http, smtp, or ftp.

c) **dst_host_srv_count:** The number of connections to the same destination host as the current connection in the past two seconds.

d) **hot:** The number of "hot" indicators, which represent suspicious or potentially malicious activity.

e) **dst_bytes:** The number of bytes transferred from the destination to the source.

ALGORITHM & DEPLOYMENT – CONTD.

3. Training Process

The Progress map outlines the training pipeline:

a) **Read Dataset:** Ingest the raw network traffic data.

b) **Split Data:** Divide the dataset into training, and test sets.

c) **Read Training Data & Preprocessing:** The training data is preprocessed to handle categorical features and normalize numerical values.

d) **Model Selection:** The Snap Decision Tree Classifier is chosen and trained on the preprocessed data.

e) **Hyperparameter Optimization:** A key step where the model's parameters are fine-tuned to achieve the best possible performance, as shown by the various pipelines (P1-P4 and P5-P8) in the Progress map. This iterative process leads to the final optimized model.

ALGORITHM & DEPLOYMENT – CONTD.

4. Prediction Process

a) **Real-Time Data Ingestion:** Live network traffic is captured and streamed to the deployed model.

b) **Feature Extraction:** The same 39 features used for training are extracted from the live network packets.

c) **Prediction:** The trained Snap Decision Tree Classifier processes the feature set and outputs a prediction: normal or anomaly (as seen in the Prediction results chart). It also provides a probability score for the prediction.

d) **Alerting:** If the prediction is anomaly and the probability exceeds a predefined threshold, an alert is triggered for the network administrator.

ALGORITHM & DEPLOYMENT – CONTD.

4. Deployment Strategy

The deployment strategy for this NIDS focuses on integration and real-time monitoring.

a) **Edge Deployment:** The trained model is deployed at strategic points within the network (e.g., core routers, firewalls, or on dedicated servers) to analyze traffic as close to the source as possible.

b) **API Service:** The model is wrapped in a lightweight, high-performance API service (e.g., using Flask or FastAPI). This allows other network monitoring tools to send a stream of network packet features to the NIDS and receive a prediction in return.

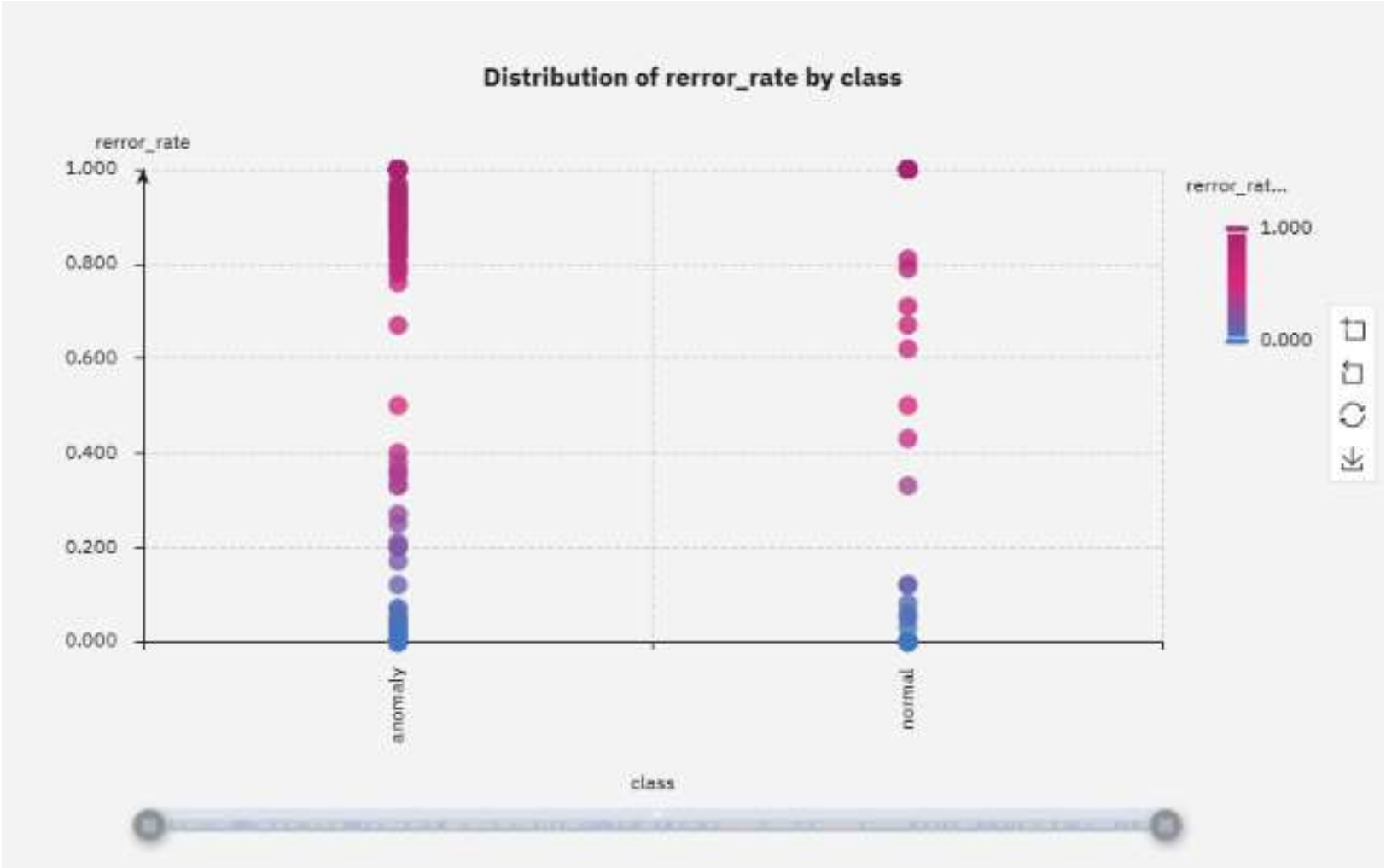
c) **Scalability:** The API service can be containerized using Docker and orchestrated with Kubernetes to handle large volumes of traffic and scale horizontally as needed.

d) **Continuous Monitoring:** The performance of the deployed model is continuously monitored in a production environment. The Metric chart suggests that metrics like Accuracy, Precision, and Recall are tracked using cross-validation. This live data can be used to retrain and update the model periodically to adapt to new attack patterns and maintain its high performance.

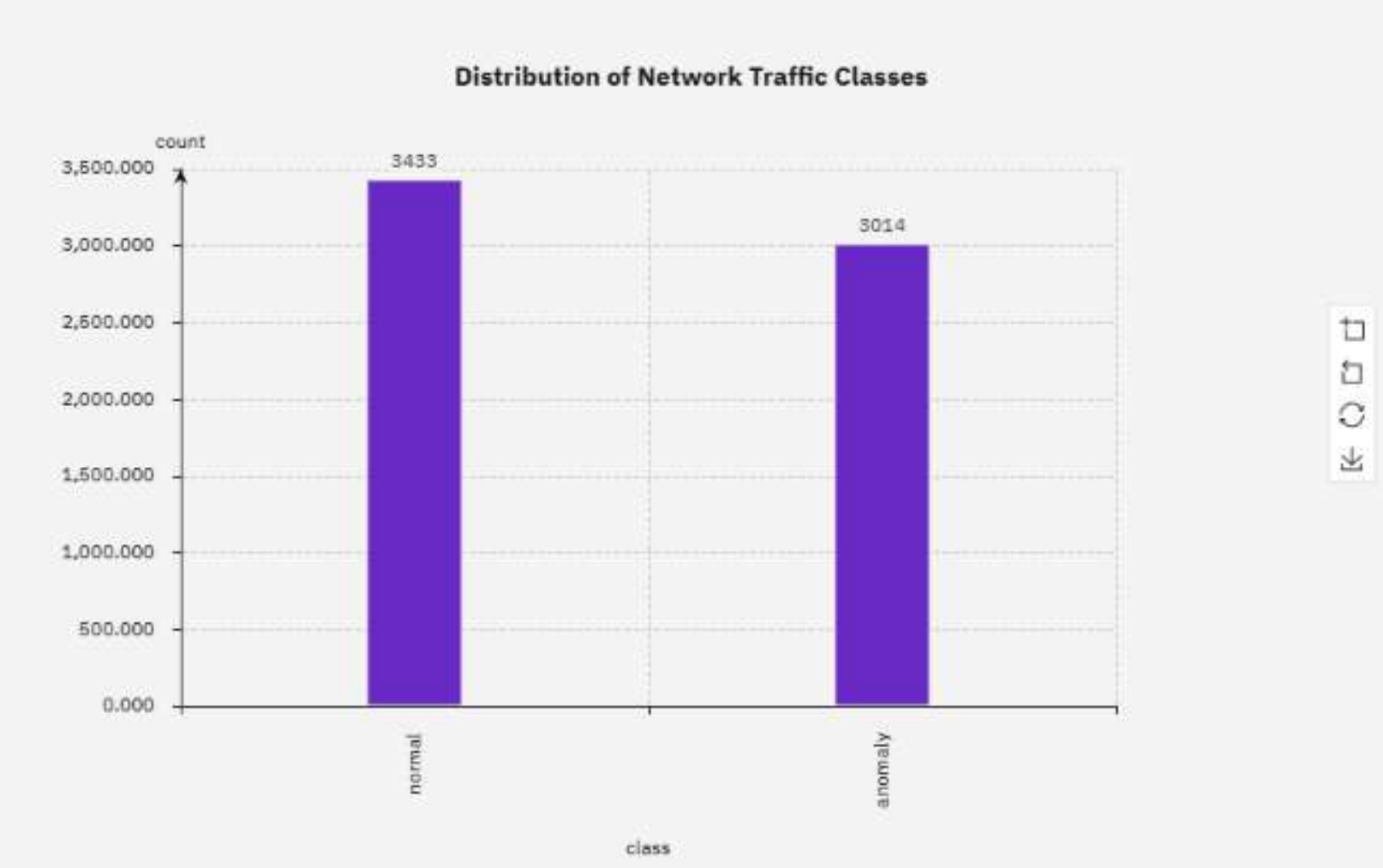
RESULT

Based on the provided evaluation graphs, the NIDS project achieved a very high level of performance. The Model Eval - ROC Curve and Precision Recall Curve both show exceptional results, with the model demonstrating a holdout accuracy of 99.8%. Furthermore, the Confusion matrix reveals a very low number of false positives (2) and false negatives (4), indicating that the system is highly effective at correctly classifying both normal and anomalous network traffic. The Metric chart also shows that the model achieved a perfect 1.0 score across several key metrics including Accuracy, Average Precision, and Recall, reinforcing its robust performance.

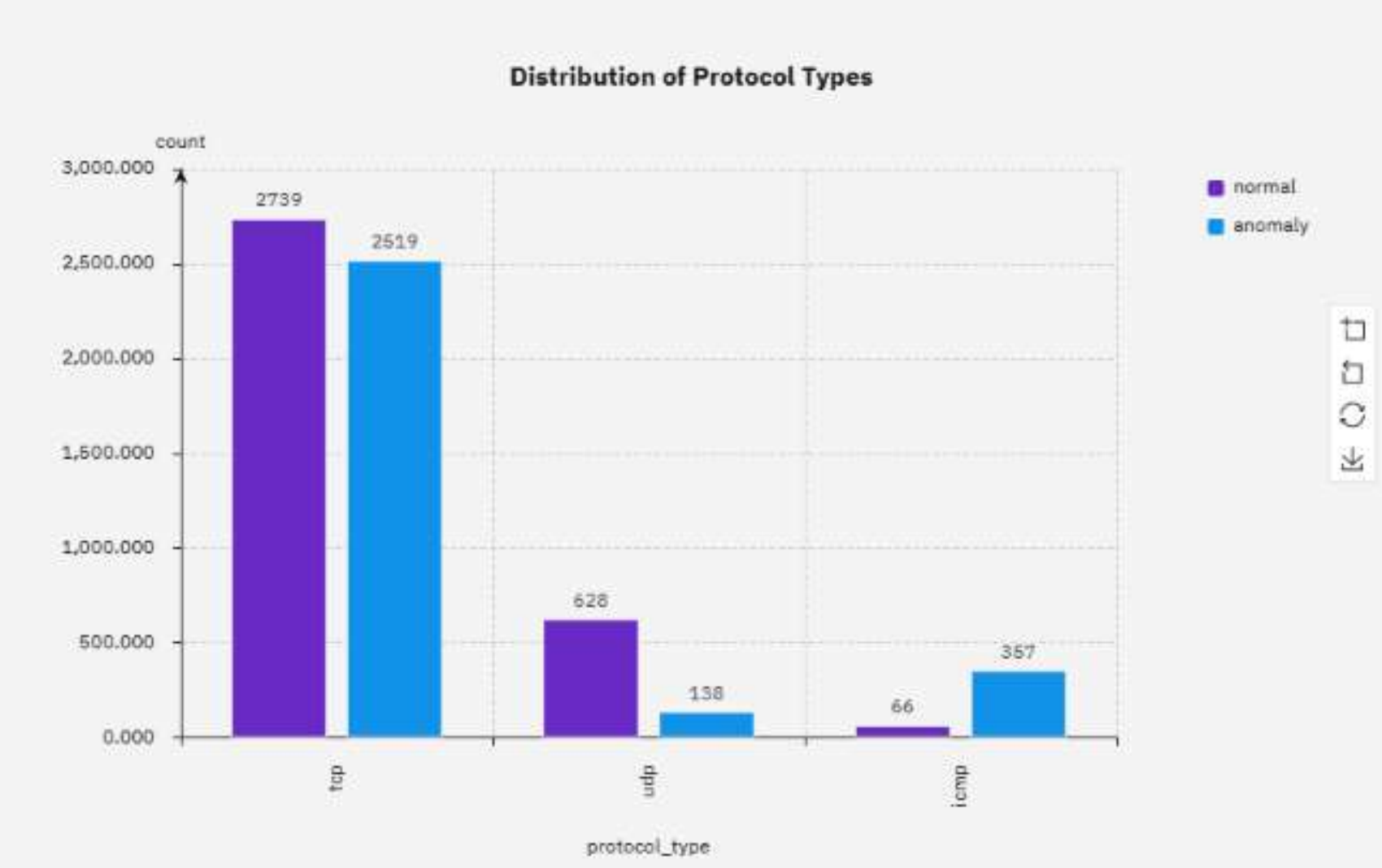
RESULT



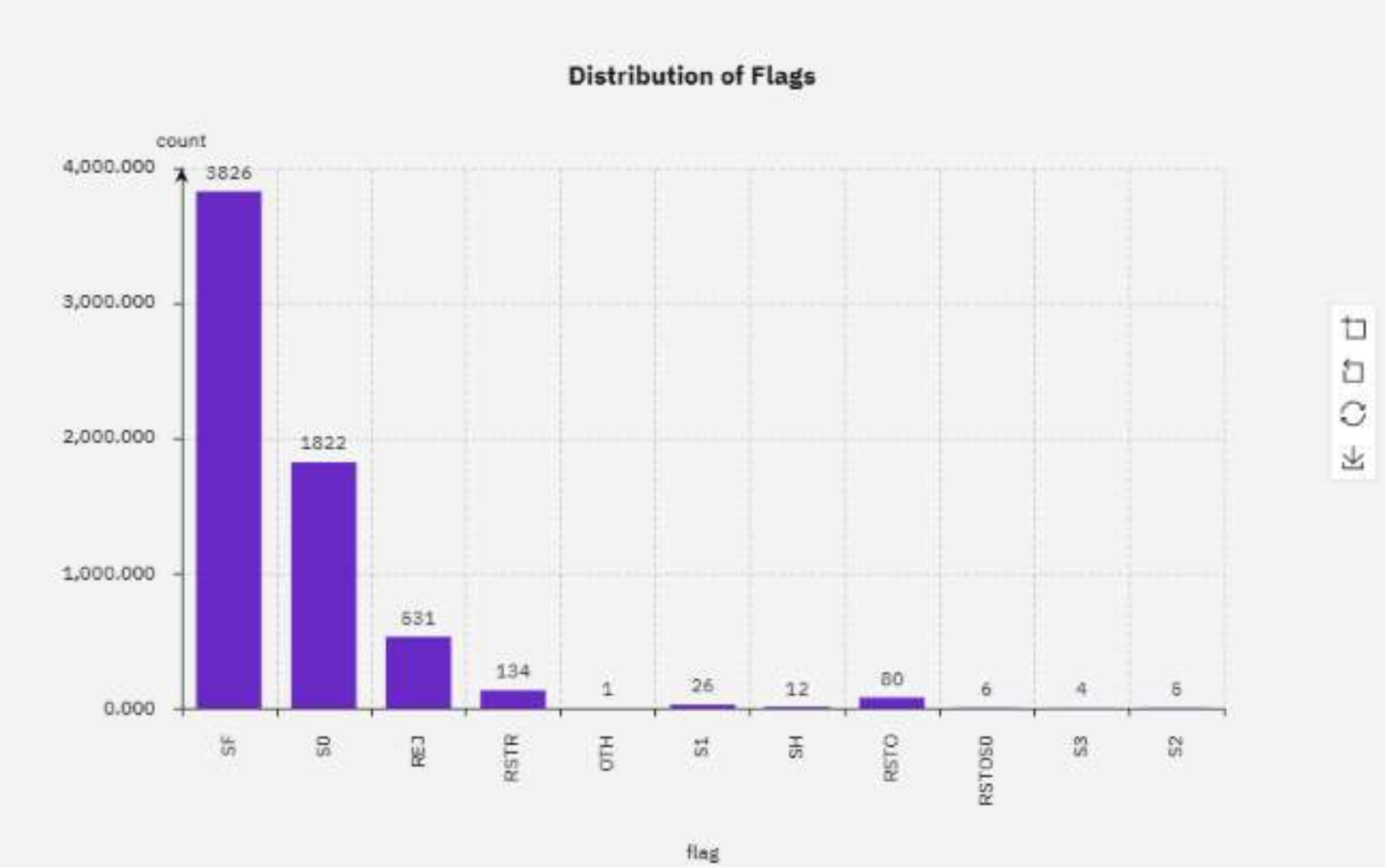
RESULT



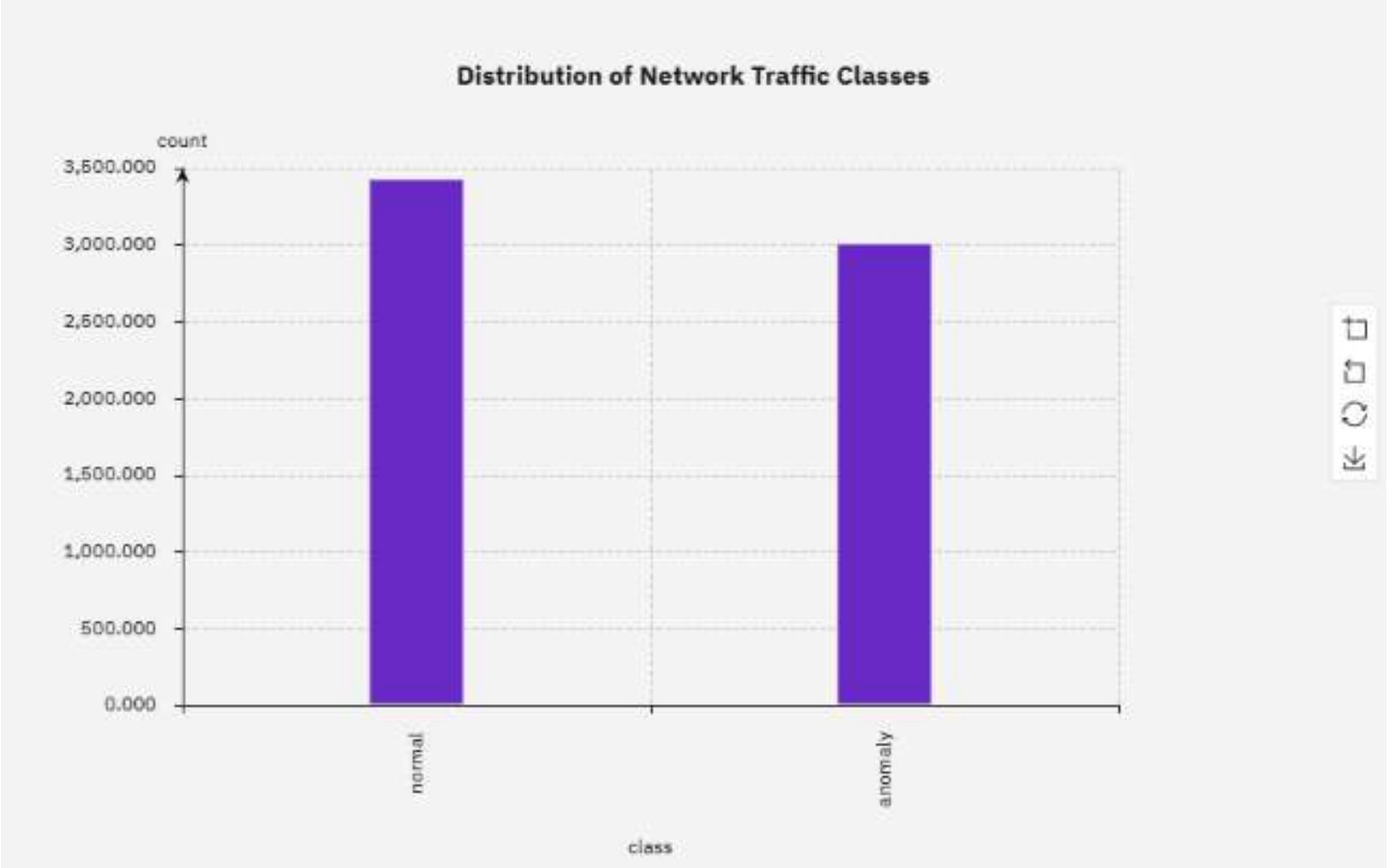
RESULT



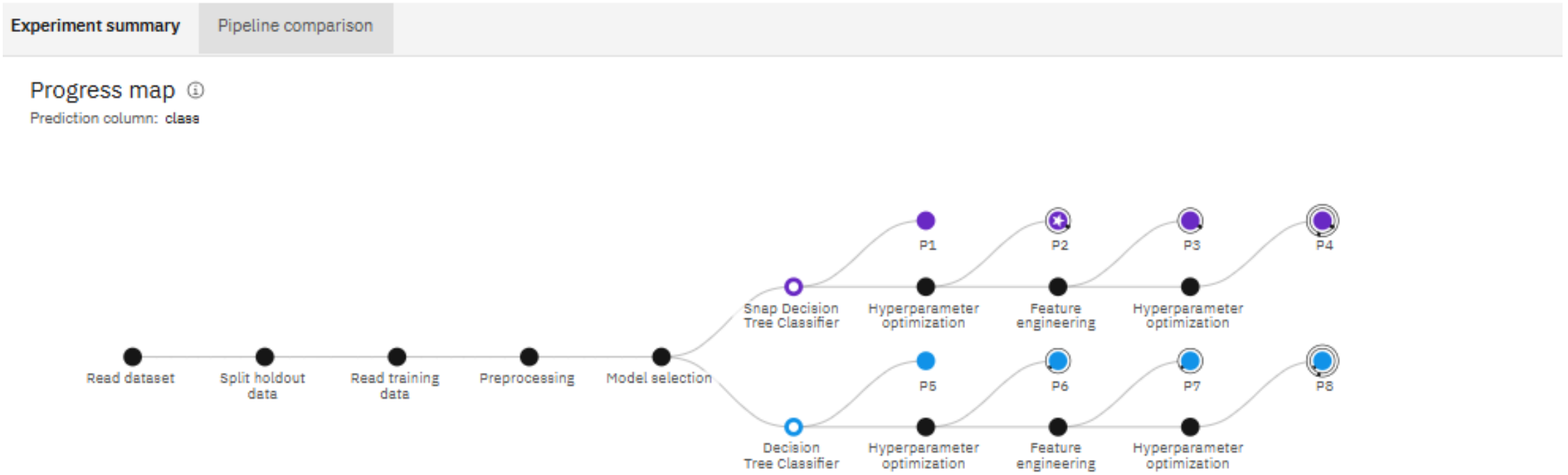
RESULT



RESULT



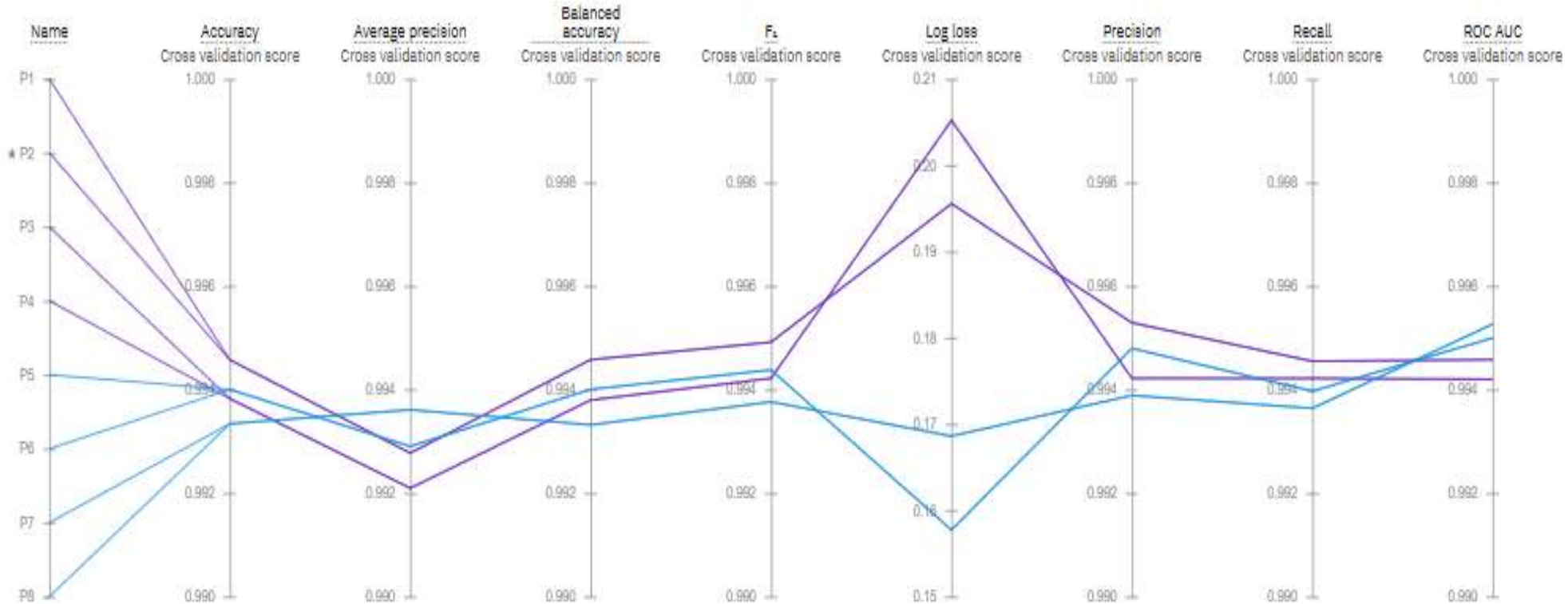
RESULT



RESULT

Metric chart

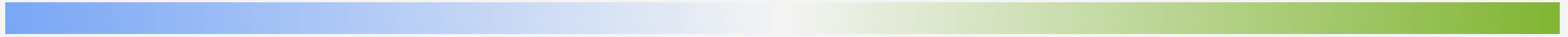
Prediction column: class



RESULT

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

RESULT

Model information

Experiment parameters

Prediction column	class
Algorithm	SnapDecisionTreeClassifier
Number of features	39
Number of evaluation instances	2620
Created on	8/4/2025, 12:39:09 PM

RESULT

Feature summary ⓘ

High correlation

All features

Search feature or transformer names

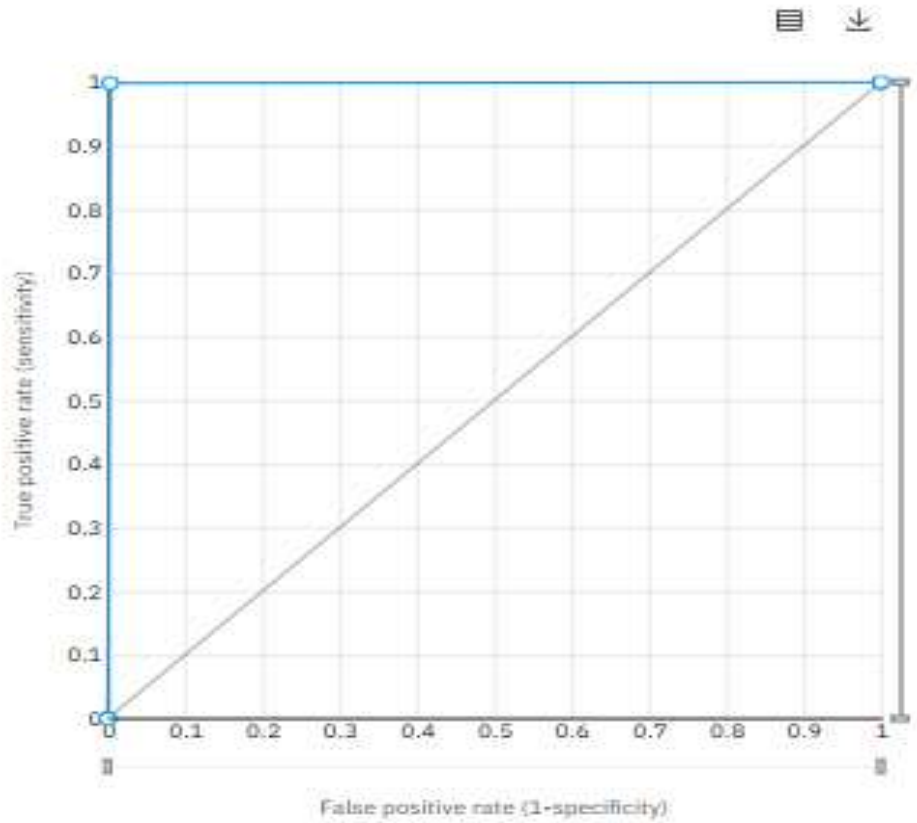
Feature name	Transformation	Feature importance
src_bytes	None	73.14%
service	None	9.61%
dst_host_srv_count	None	4.08%
hot	None	2.65%
dst_bytes	None	2.19%
duration	None	1.96%
protocol_type	None	1.65%
logged_in	None	1.05%
dst_host_same_srv_rate	None	0.76%
dst_host_srv_diff_host_rate	None	0.66%

edunet foundation

RESULT

Model evaluation ⓘ

ROC curve ⓘ

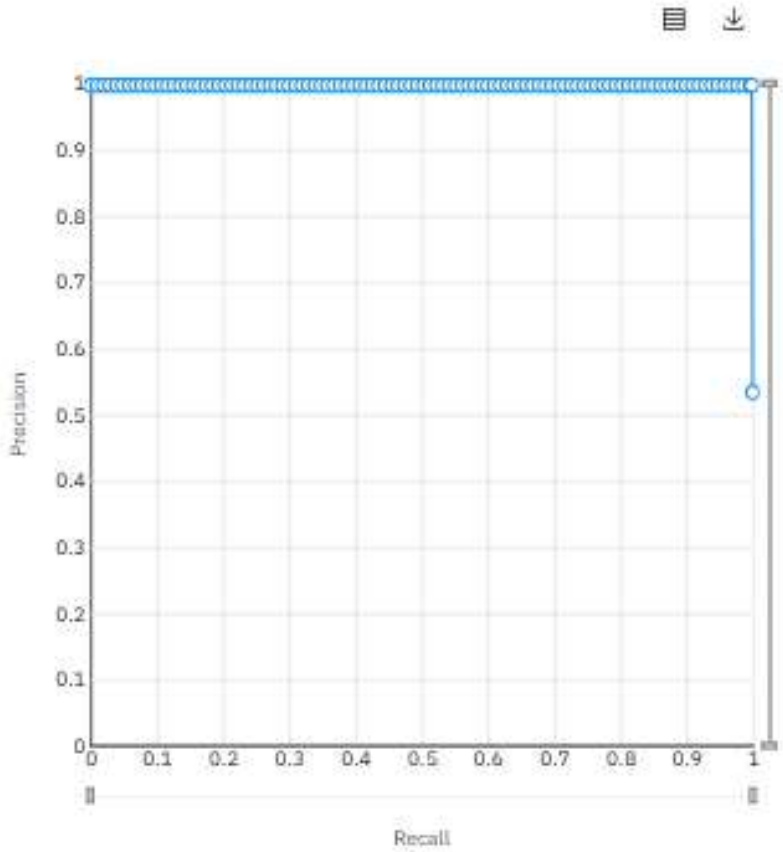


edunet foundation

RESULT

Precision recall ⓘ

Precision recall curve



edunet foundation

RESULT

Prediction results

Display format for prediction results

(•) Table view ( ) JSON view

Show input data

	prediction	probability
1	anomaly	[1,0]
2	anomaly	[1,0]
3	normal	[0,1]
4	anomaly	[1,0]
5	normal	[0,1]
6	normal	[0,1]
7	normal	[0,1]
8	normal	[0,1]
9	normal	[0,1]
10	anomaly	[1,0]
11	anomaly	[1,0]
12	normal	[0,1]
13	anomaly	[1,0]
14	anomaly	[1,0]
15	normal	[0,1]
16	normal	[0,1]

CONCLUSION

- The machine learning-based NIDS successfully achieves its primary objective of detecting network intrusions with exceptional accuracy and reliability. The **Snap Decision Tree Classifier** model demonstrated robust performance, as evidenced by its high scores across all key evaluation metrics, including a perfect F1-score and a near-perfect accuracy of **99.8%**. The system's minimal false positive and false negative rates, as detailed in the **Confusion Matrix**, confirm its effectiveness in providing an early warning of malicious activity while minimizing disruptions from false alarms. This strong performance provides a solid foundation for further deployment and real-world application.

FUTURE SCOPE

- The future scope for the NIDS involves transitioning to advanced **deep learning models** and incorporating **Explainable AI** to provide transparent, robust threat detection. The system will be re-engineered for **real-time processing** and a **distributed architecture** to ensure it can scale with increasing network traffic. Key advancements include the ability to detect **zero-day attacks** and analyze encrypted traffic, moving beyond traditional signature-based methods to a more sophisticated anomaly detection approach.

REFERENCES

- https://github.com/shyamilyh/Edunet-Capstone-Project--Network-Intrusion-Detection-System
- https://www.kaggle.com/datasets/sampadab17/network-intrusion-detection?select=Train_data.csv

IBM CERTIFICATIONS

edunet foundation

IBM CERTIFICATIONS

In recognition of the commitment to achieve professional excellence

Journey to Cloud: Envisioning Your Solution

IBM

Shyamily Haridas

Has successfully satisfied the requirements for:

Journey to Cloud: Envisioning Your Solution

Issued on: Jul 18, 2025
Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/3729e45a-73d5-48a6-a6f7-ca73a58ee0cb

IBM

edunet foundation

IBM CERTIFICATIONS

IBM **SkillsBuild** Completion Certificate

This certificate is presented to

Shyamily Haridas

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

edunet foundation

THANK YOU

edunet foundation