CAPSTONE PROJECT

NETWORK INTRUSION DETECTION

Presented By:
DIVYAVARSHINI R
BANNARI AMMAN INSTITUTE OF TECHNOLOGY
INFORMATION SCIENCE AND ENGINEERING

OUTLINE

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

edunet foundation

PROBLEM STATEMENT

Example: In today's digital landscape, communication networks are increasingly targeted by various cyber-attacks such as DoS, Probe, R2L, and U2R. Traditional security measures often fail to detect these threats effectively. The key challenge is to accurately identify and classify malicious network activity in real time without disrupting normal operations.

PROPOSED SOLUTION

- The proposed system aims to detect network intrusions using machine learning techniques. It processes and analyzes network traffic data to classify it into known attack types—such as DoS, Probe, R2L, and U2R—or as normal activity. The system is designed to provide early warning signals to network administrators, enabling faster response to threats and strengthening the organization's cybersecurity posture.

The solution will consist of the following components:

- **Data Ingestion:** Network traffic data is collected from a labeled dataset (e.g., KDD or NSL-KDD) with relevant features such as protocol type, duration, and byte size.
- **Preprocessing:** Raw data is cleaned, encoded, and normalized for optimal model performance.
- **Model Training:** Machine learning algorithms such as Random Forest or SVM are trained to detect and classify traffic behavior.
- **Deployment:** The model is deployed using IBM Cloud Lite services for scalable, real-time threat detection.
- **Alert System:** Detected intrusions are flagged and categorized to assist in rapid security response

SYSTEM APPROACH

System Requirements:

- IBM Cloud Lite
- Python
- Libraries: Pandas, NumPy, Scikit-learn, Seaborn, Matplotlib

Tools Used:

- Dataset: Kaggle NIDS dataset
- Platform: IBM Cloud Watson Studio (Lite Tier)
- Environment: Python 3.x runtime

ALGORITHM & DEPLOYMENT

Algorithm Selection:

- Decision Tree, Random Forest, and Support Vector Machine (SVM) were tested.
- Final model: [Your chosen algorithm] (e.g., Random Forest) due to high accuracy and speed.

Data Input:

- Features include duration, protocol_type, service, flag, src_bytes, dst_bytes, and more.
- Label: Attack type (DoS, Probe, R2L, U2R, Normal)

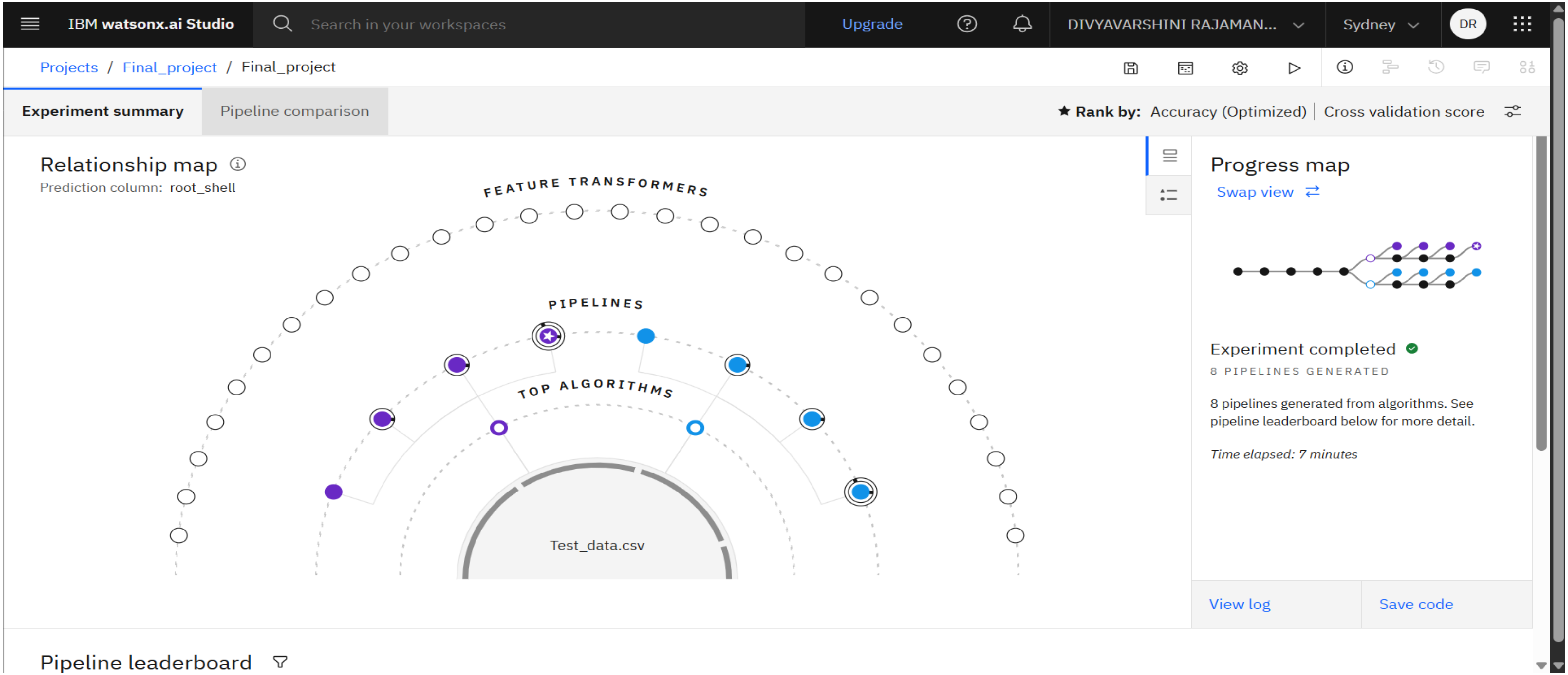
Training Process:

- Dataset split into training and testing (e.g., 80/20)
- Preprocessing includes label encoding and normalization

Deployment:

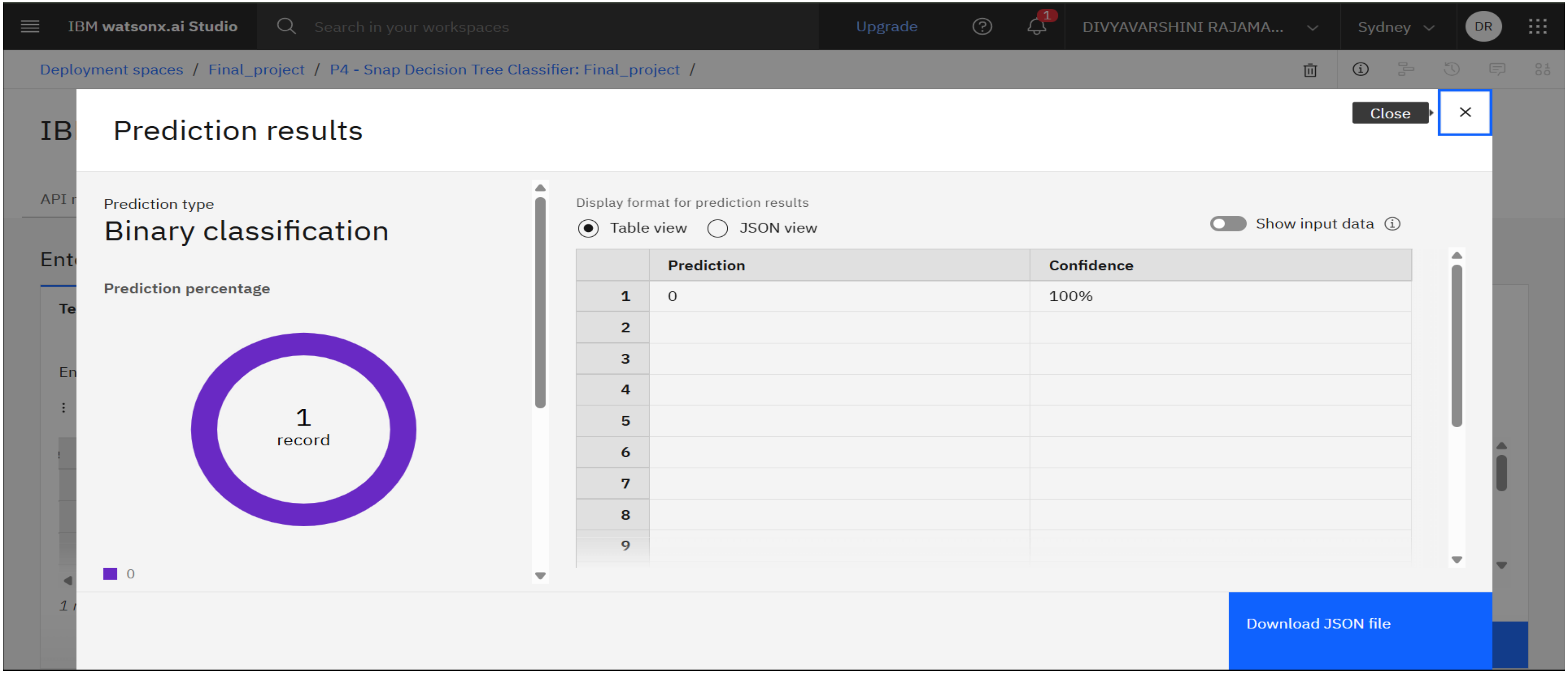
- Model deployed on IBM Watson Machine Learning
- Integrated with a dashboard for visualizing alerts

RESULT



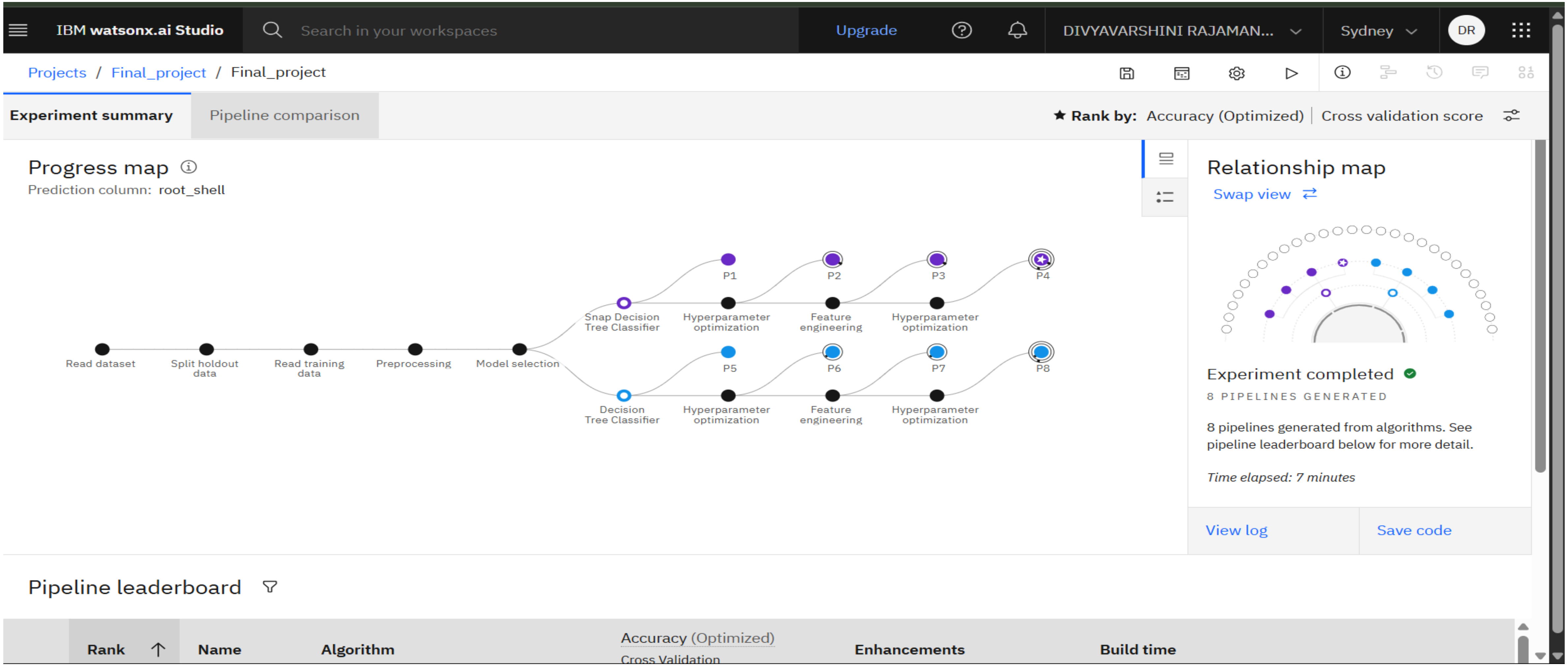
IBM watsonx.ai Studio
Search in your workspaces
Upgrade
DIVYAVARSHINI RAJAMAN...
Sydney
DR
Projects / Final_project / Final_project
Experiment summary
Pipeline comparison
Rank by: Accuracy (Optimized) | Cross validation score
Relationship map
Prediction column: root_shell
FEATURE TRANSFORMERS
PIPELINES
TOP ALGORITHMS
Test_data.csv
Progress map
Swap view
Experiment completed
8 PIPELINES GENERATED
8 pipelines generated from algorithms. See pipeline leaderboard below for more detail.
Time elapsed: 7 minutes
View log
Save code
Pipeline leaderboard

RESULT



IBM watsonx.ai Studio
Search in your workspaces
Upgrade
DIVYAVARSHINI RAJAMA...
Sydney
DR
Deployment spaces / Final_project / P4 - Snap Decision Tree Classifier: Final_project /
Close
Prediction results
Prediction type
Binary classification
Prediction percentage
1
record
0
Display format for prediction results
Table view
JSON view
Show input data
Prediction
Confidence
1
0
100%
2
3
4
5
6
7
8
9
Download JSON file

RESULT



IBM watsonx.ai Studio
Search in your workspaces
Upgrade
DIVYAVARSHINI RAJAMAN...
Sydney
DR
Projects / Final_project / Final_project
Experiment summary
Pipeline comparison
Rank by: Accuracy (Optimized) | Cross validation score
Progress map
Prediction column: root_shell
Read dataset
Split holdout data
Read training data
Preprocessing
Model selection
Snap Decision Tree Classifier
Decision Tree Classifier
Hyperparameter optimization
Feature engineering
Hyperparameter optimization
P1
P2
P3
P4
P5
P6
P7
P8
Relationship map
Swap view
Experiment completed
8 PIPELINES GENERATED
8 pipelines generated from algorithms. See pipeline leaderboard below for more detail.
Time elapsed: 7 minutes
View log
Save code
Pipeline leaderboard
Rank
Name
Algorithm
Accuracy (Optimized)
Cross Validation
Enhancements
Build time

CONCLUSION

- The developed NIDS effectively detects various intrusion types and distinguishes them from normal traffic. The use of machine learning improves adaptability to evolving attack patterns. The system contributes to proactive cybersecurity defense in real-time environments.

FUTURE SCOPE

- Include deep learning models (e.g., LSTM or Autoencoders)
- Real-time traffic integration with live packet capture
- Scalable deployment for enterprise-level networks
- Enhanced visualization dashboard with alerting mechanism

REFERENCES

- Kaggle Dataset: https://www.kaggle.com/datasets/sampadab17/network-intrusion-detection
- IBM Cloud Documentation
- Research papers on ML-based NIDS models

edunet foundation

IBM CERTIFICATIONS

In recognition of the commitment to achieve professional excellence

Getting Started with Artificial Intelligence

IBM SkillsBuild

IBM

DIVYAVARSHINI R

Has successfully satisfied the requirements for:

Getting Started with Artificial Intelligence

Issued on: Jul 20, 2025

Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/16241ec4-e9e7-4a4b-9bd5-b56f39b123f4

IBM

edunet foundation

IBM CERTIFICATIONS

In recognition of the commitment to achieve professional excellence

Journey to Cloud: Envisioning Your Solution

IBM SkillsBuild

IBM

DIVYAVARSHINI R

Has successfully satisfied the requirements for:

Journey to Cloud: Envisioning Your Solution

Issued on: Jul 21, 2025

Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/f2e94a3d-95fd-4247-bc28-5f56d59436ac

IBM

IBM CERTIFICATIONS

Rotate (Ctrl+])

IBM **SkillsBuild** Completion Certificate

This certificate is presented to

DIVYAVARSHINI R

for the completion of

Lab: Retrieval Augmented Generation with LangChain

(ALM-COURSE_3824998)

According to the Adobe Learning Manager system of record

Completion date: 25 Jul 2025 (GMT)

Learning hours: 20 mins

THANK YOU