CAPSTONE PROJECT

NETWORK INTRUSION DETECTION

Presented By:
1. Divyanshe-Graphic Era Hill University-Computer Science

edunet foundation

OUTLINE

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

PROBLEM STATEMENT

With the rapid growth of internet-based services, ensuring the security of network systems has become a critical concern. Modern networks are increasingly vulnerable to cyber-attacks, which can result in data breaches and service disruptions. To address this, intelligent systems are required to automatically monitor and identify suspicious activity in real time. This project focuses on developing a machine learning based Network Intrusion Detection System that distinguishes anomalous network behavior from normal traffic, assisting cybersecurity teams in identifying potential threats and improving overall network defense.

PROPOSED SOLUTION

- The proposed system aims to develop a machine learning-based Network Intrusion Detection System that analyzes network traffic data to detect and classify cyber-attacks. It involves data preprocessing, model training, and real-time deployment using IBM Cloud Lite to provide early threat detection and enhance network security.
- **Data Collection:**
  - The dataset used for this project was obtained from Kaggle. The dataset contains simulated network traffic data, where each record is labeled either as normal or anomalous, representing various cyber-attacks including Denial-of-Service (DoS), Distributed DoS (DDoS), and other intrusion types.
- **Data Preprocessing:**
  - The dataset was cleaned and prepared using label encoding for categorical variables and normalization for numerical features to ensure consistency and model readiness.
- **Machine Learning Algorithm:**
  - The Snap Decision Tree Classifier was selected due to its high accuracy (0.995) under HPO-1 hyperparameter configuration. It demonstrated better performance compared to other tested models in terms of accuracy, efficiency, and suitability for intrusion detection tasks.

PROPOSED SOLUTION

- **Deployment:**
  - Model training and testing were conducted in Python 3.11 environment within Watson.ai Studio. IBM Cloud Object Storage was used for managing the dataset and storing the model. The deployment was carried out in the IBM Cloud Lite runtime environment.
- **Evaluation:**
  - The model was evaluated using the holdout method (90% training, 10% testing), along with cross-validation which was used during training to reduce overfitting and assess model consistency across multiple subsets of data.
- **Result:**
  - The proposed system delivers an intelligent intrusion detection solution capable of identifying and classifying potential cyber threats in real-time. The classifier showed consistent performance across various classes, including DoS and probing attacks, with minimal false positives. With high prediction accuracy and seamless deployment, it enhances the security of communication networks by detecting anomalies efficiently and accurately.

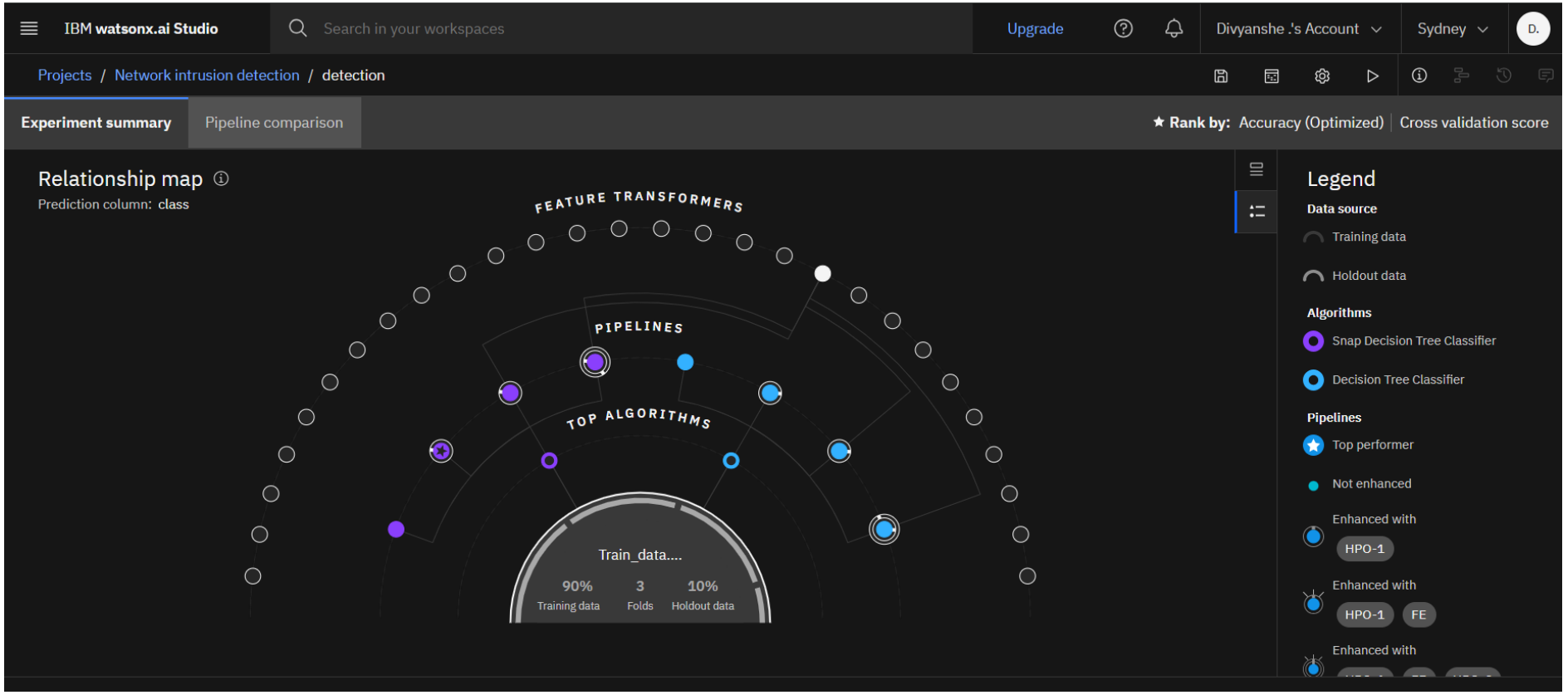
SYSTEM APPROACH

- **System Requirements:** Required software, libraries, and hardware are listed below.
- **Software Requirements:**
  - IBM Cloud Account
  - IBM Watson.AI Studio
  - IBM Cloud Object Storage
- **Libraries/Packages:** ibm-watsonx-ai, autoai-libs, lale, scikit-learn, xgboost, lightgbm, snapml
- **Hardware Requirements :**
  - Windows 10 or higher (64-bit)
  - Minimum 4 GB RAM
  - Stable internet connection for accessing IBM Cloud services
- **Runtime Environment**
  - Python 3.11 notebook environment in IBM Watson Studio was used for training, testing, and evaluation.

ALGORITHM & DEPLOYMENT

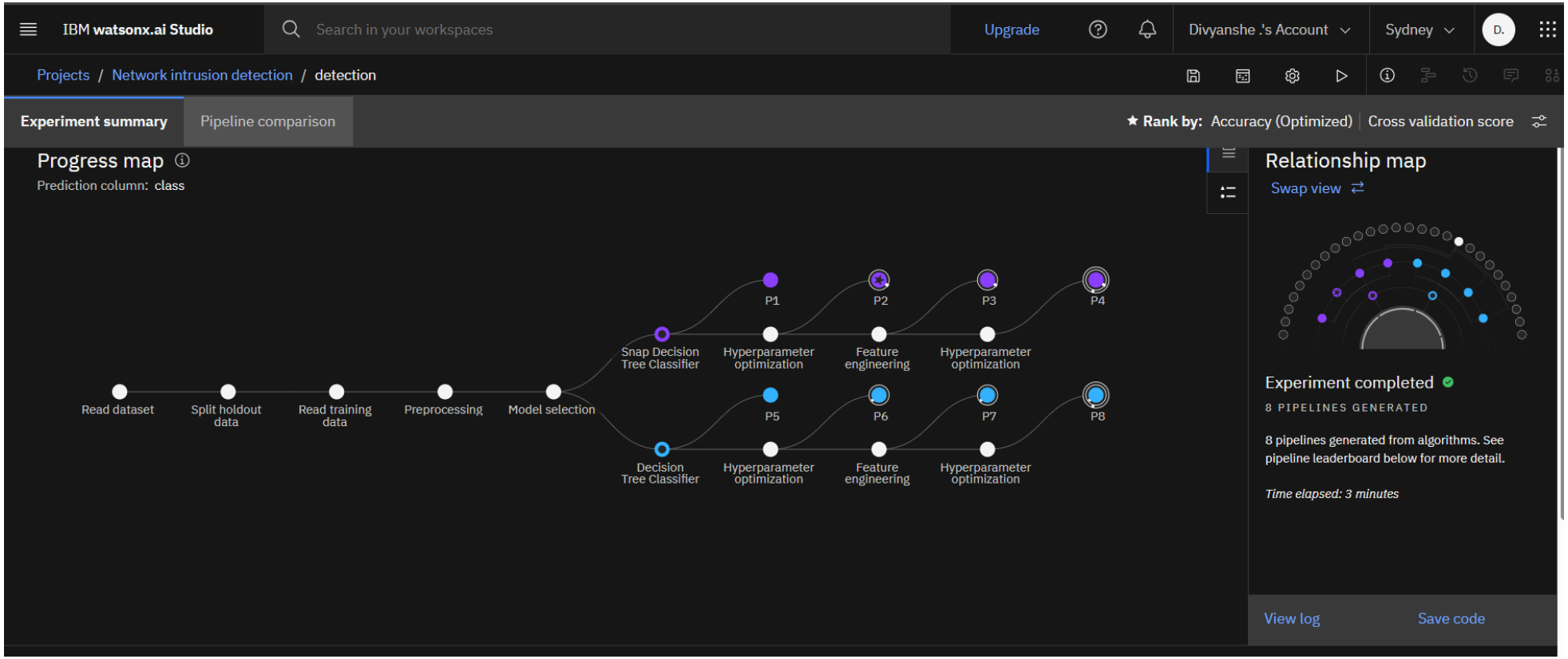
- **Algorithm Selection:**
  - The Snap Decision Tree Classifier with HPO-1 (Hyperparameter Optimization level 1) was selected because it showed the highest accuracy (99.5%) during training and validation phases. It is known for its efficiency, high performance, and suitability for handling large-scale network intrusion detection problems involving mixed feature types.
- **Data Input:**
  - Input features such as protocol type, service, flag, source bytes, destination bytes, and connection status were used. These features were extracted from the train_data.csv file, which was obtained from Kaggle for the purpose of model development.
- **Training Process:**
  - The dataset was preprocessed using label encoding and normalization techniques. The holdout method was used for evaluation—90% of the data was used for training, and 10% was reserved for testing. This approach helped validate the model's performance and avoid overfitting. The model was trained using supervised learning and further validated using cross-validation techniques to ensure robustness.
- **Prediction Process:**
  - The trained model was tested on the test_data.csv file to evaluate its accuracy. It predicts whether each new record represents normal or anomaly (malicious) activity based on its features.

RESULT



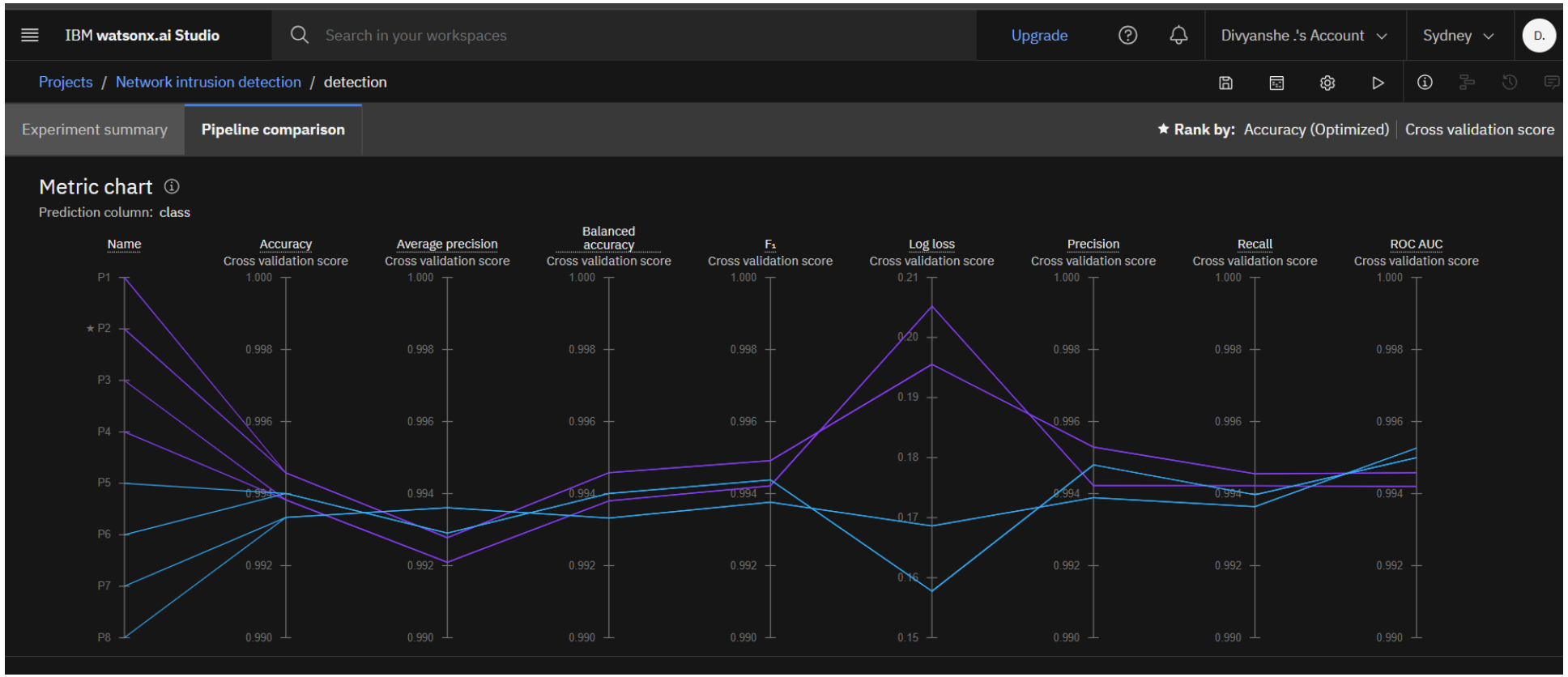
edunet foundation

RESULT



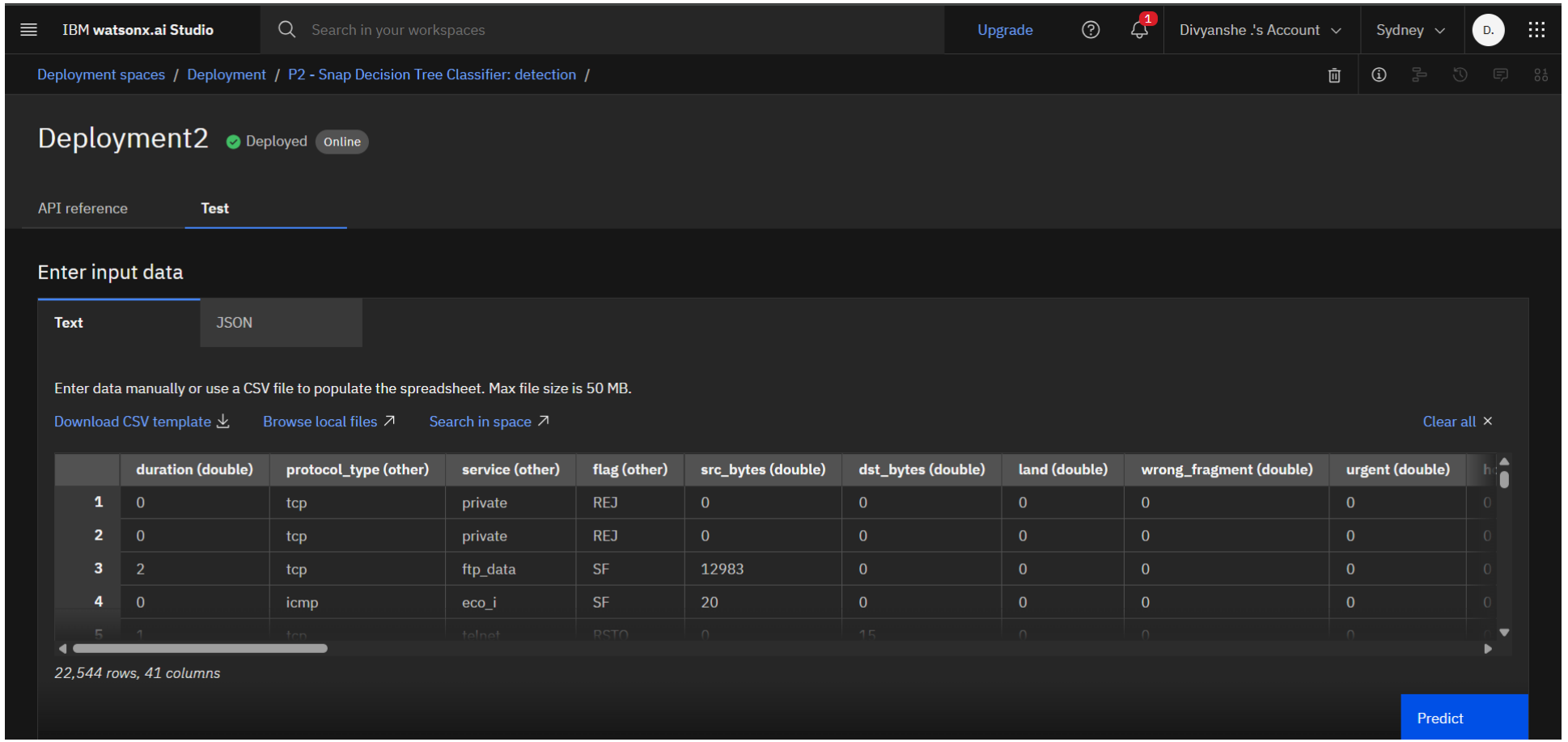
edunet foundation

RESULT



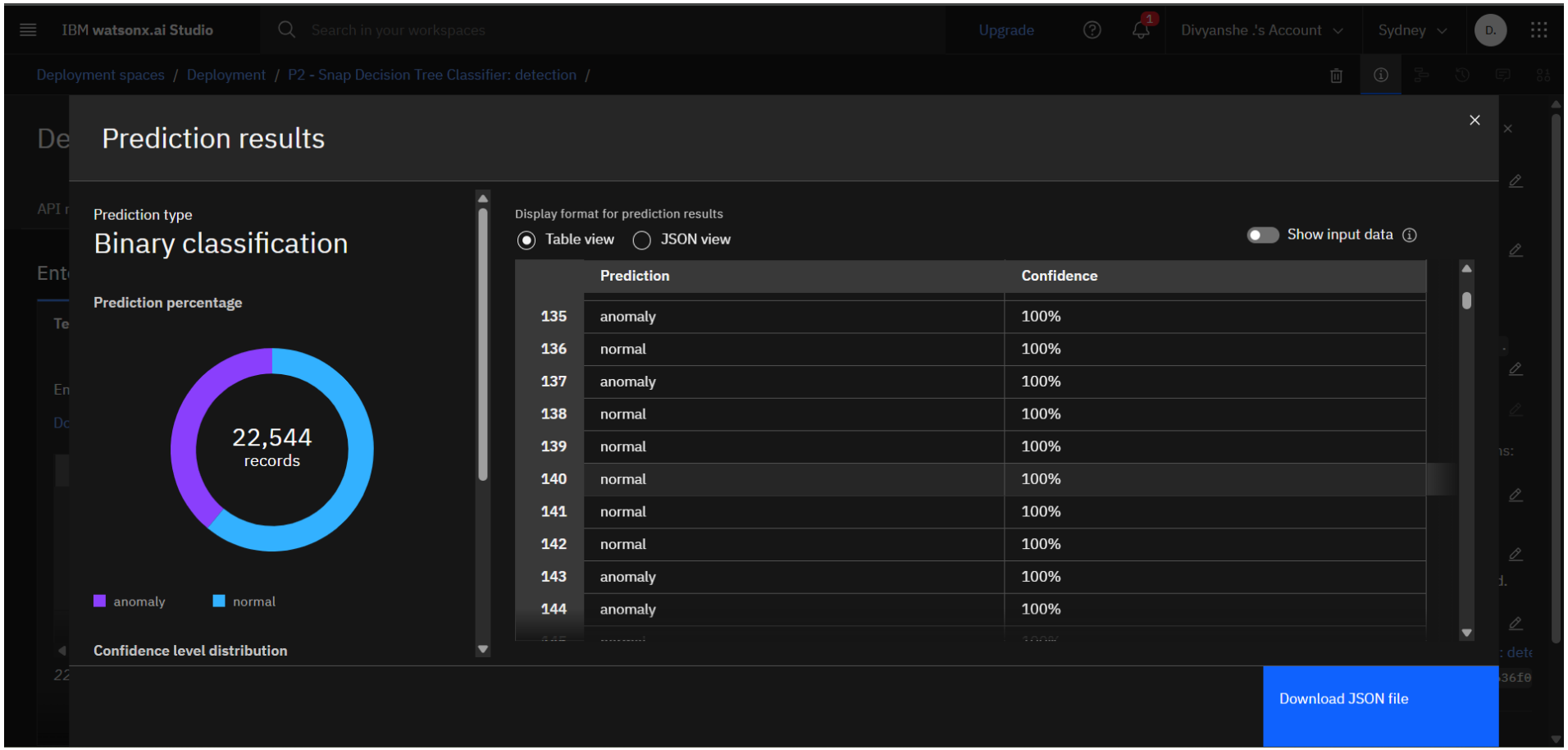
edunet foundation

RESULT



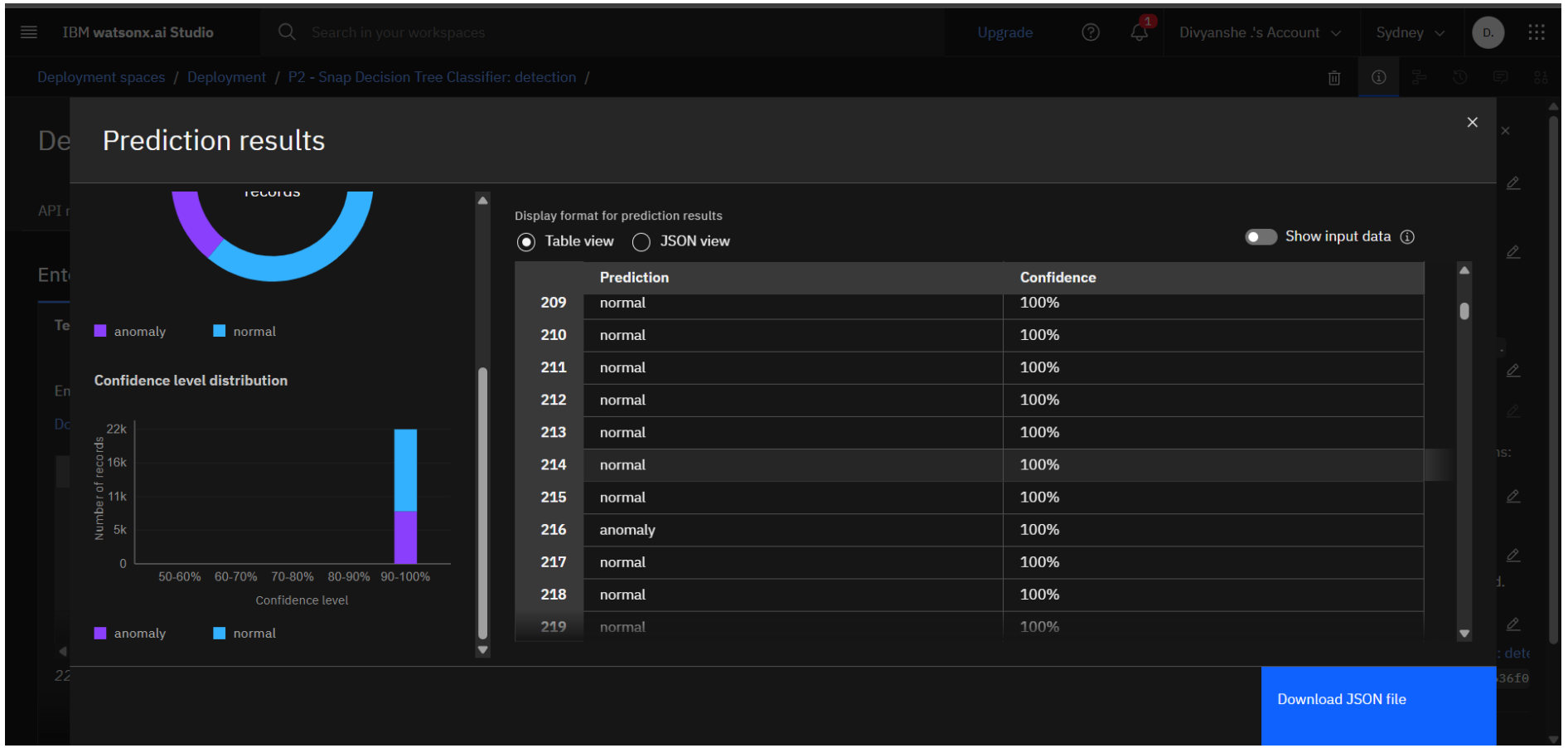
edunet foundation

RESULT



edunet foundation

RESULT



edunet foundation

CONCLUSION

- The proposed machine learning-based Network Intrusion Detection System effectively distinguishes between normal and malicious network activity using the Snap Decision Tree Classifier. With high accuracy achieved through proper preprocessing, feature selection, and validation techniques, the system demonstrates strong potential for real-world deployment. The use of IBM Cloud services provided a scalable and flexible environment for model training and testing.
- Overall, the project emphasizes the growing need for intelligent intrusion detection in today's digital world and demonstrates the feasibility of using machine learning to enhance network security.

FUTURE SCOPE

- **Real-Time Alerting:**
  - Integrating a real-time alerting mechanism to notify administrators about detected intrusions can greatly improve the system's practical usability.
- **Advanced Models:**
  - Future implementations can explore deep learning techniques for handling more complex intrusion patterns and improving detection capabilities.
- **Website and Mobile App Integration:**
  - A user-friendly web interface and mobile application can be developed to monitor intrusion alerts, view logs, and manage system settings remotely.
- **Scalability:**
  - Adapting the model to handle larger and more diverse datasets from different network environments can enhance robustness and generalization.
- **Multi-Tenant Support:**
  - Build support for multiple users in a cloud-based interface with role-based access controls.

REFERENCES

- [1] M. A. Ambusaidi, X. He, P. N. S. S. R. B. Priyadarshini, and Z. Tian, "Building an intrusion detection system using a filter-based feature selection algorithm," *IEEE Transactions on Computers*, vol. 65, no. 10, pp. 2986–2998, Oct. 2016.
- [2] S. Revathi and A. Malathi, "A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection," *Int. J. Eng. Res. Technol.*, vol. 2, no. 12, pp. 1848–1853, Dec. 2013.
- [3] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, D. Breitenbacher, and Y. Elovici, "N-BaIoT—Network-based detection of IoT botnet attacks using deep autoencoders," *IEEE Pervasive Comput.*, vol. 17, no. 3, pp. 12–22, Jul.–Sep. 2018.
- [4] T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE Commun. Surv. Tutor.*, vol. 10, no. 4, pp. 56–76, Fourth Quarter 2008.
- [5] IBM, "Getting started with IBM Watson Studio," *IBM Cloud Docs*, 2024. [Online]. Available: https://www.ibm.com/cloud/watson-studio
- [6] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE Symp. Comput. Intell. Security Defense Appl. (CISDA)*, Ottawa, ON, Canada, Jul. 2009, pp. 1–6.
- [7] Kaggle, "Network intrusion detection dataset," [Online]. Available: https://www.kaggle.com/datasets

IBM CERTIFICATIONS

In recognition of the commitment to achieve professional excellence

Getting Started with Artificial Intelligence
IBM SkillsBuild

Divyanshe .

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/c5727dff-027d-4446-9842-2be570e32e77

IBM

edunet foundation

IBM CERTIFICATIONS

edunet foundation

IBM CERTIFICATIONS

IBM **SkillsBuild** Completion Certificate

This certificate is presented to

Divyanshe .

for the completion of

Lab: Retrieval Augmented Generation with LangChain

(ALM-COURSE_3824998)

According to the Adobe Learning Manager system of record

Completion date: 23 Jul 2025 (GMT) **Learning hours:** 20 mins

edunet foundation

THANK YOU

edunet foundation