
CAPSTONE PROJECT

NETWORK INTRUSION DETECTION

Presented By:

**1. Shaikh Ayesha- M.H Saboo Siddik College of Engineering-
CSE(AI&ML)**

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 rapid expansion of digital communication, cyberattacks have become more frequent and sophisticated. Traditional rule-based intrusion detection systems often fail to detect newer, unseen patterns of attacks. There is a need for intelligent systems that can detect abnormal behavior in network traffic in real time. This project addresses the problem of building a machine learning-based Network Intrusion Detection System (NIDS) that can classify network traffic as normal or attack.

PROPOSED SOLUTION

- The proposed system is a Network Intrusion Detection System (NIDS) that uses machine learning to classify network traffic as either normal or anomaly. The goal is to build an intelligent, automated system that can be deployed to monitor communication networks and raise early warnings of possible intrusions.
- **Data Collection:**
 - The dataset used (Train_data.csv) from [Kaggle network intrusion system](#) contains 41 features representing characteristics of network connections, such as protocol type, bytes sent, login attempts, etc.
 - This data mimics real-world network traffic, both normal and malicious.
- **Data Upload & Preprocessing:**
 - The dataset was uploaded to IBM Watsonx.ai Studio
 - AutoAI handled Encoding of categorical features, Scaling of numerical values, Splitting into training and validation sets
- **Model Training (AutoAI)**
 - AutoAI automatically tested various algorithms and pipelines
 - The goal was to optimize model performance for binary classification (class: normal or anomaly)
- **Model Selection & Deployment:**
 - The best performance model (Pipeline 2 using algorithm Snap Decision Tree Classifier) was selected and promoted to a deployment space
 - It allowed the model to accept new inputs and return live predictions
- **Prediction & Testing**
 - **Model was tested using both:**
 - Manually created test samples (realistic network behaviors)
 - test_data.csv from [Kaggle network intrusion system](#) (official test set with unseen records)

SYSTEM APPROACH

Technology Used:

- **Language:** Python
- **IBM Services:** Watsonx.ai Studio, Watson Machine Learning, Cloud Object Storage
- **Tools:** IBM AutoAI, Jupyter Notebook
- **Dataset:** Train_data.csv from [Kaggle network intrusion system](#)

Steps Followed on IBM Cloud Lite:

- 1.Created Watsonx.ai Studio instance (Lite plan)
- 2.Created a new project and attached Cloud Object Storage
- 3.Uploaded the dataset
- 4.Started an AutoAI experiment and selected class as the prediction column
- 5.Allowed AutoAI to train and rank models automatically
- 6.Saved and promoted the best-performing model
- 7.Deployed the model as an online API for prediction
- 8.Tested the model using realistic JSON inputs

ALGORITHM & DEPLOYMENT

- **Algorithm Selection:**
 - AutoAI selected a top-performing classification algorithm
 - Binary classification (target: class) to distinguish between normal and anomaly
- **Model Inputs:**
 - 41 network traffic features (e.g., protocol_type, service, src_bytes, count, etc.)
- **Training Process:**
 - AutoAI automatically handled:
 - Data preprocessing (encoding, scaling)
 - Model selection and training
 - Performance evaluation (accuracy, F1-score, etc.)
- **Prediction Process:**
 - The trained model was deployed using Watson Machine Learning
 - A REST API was created to accept input values and return predictions
 - Users can input values manually or from CSV and get the prediction (normal or anomaly)

RESULT

After training and deploying the best-performing model from the AutoAI experiment, we tested its accuracy and reliability using both:

- Manually created test inputs (3 real-world inspired samples)
- The official test_data.csv provided as part of the project

The deployed model correctly classified inputs as either normal or anomaly using all 41 input features. Testing was done through the IBM Watson Machine Learning “Test” interface.

RESULT

■ Key Results:

The model achieved **high accuracy (99.5 %)** in binary classification

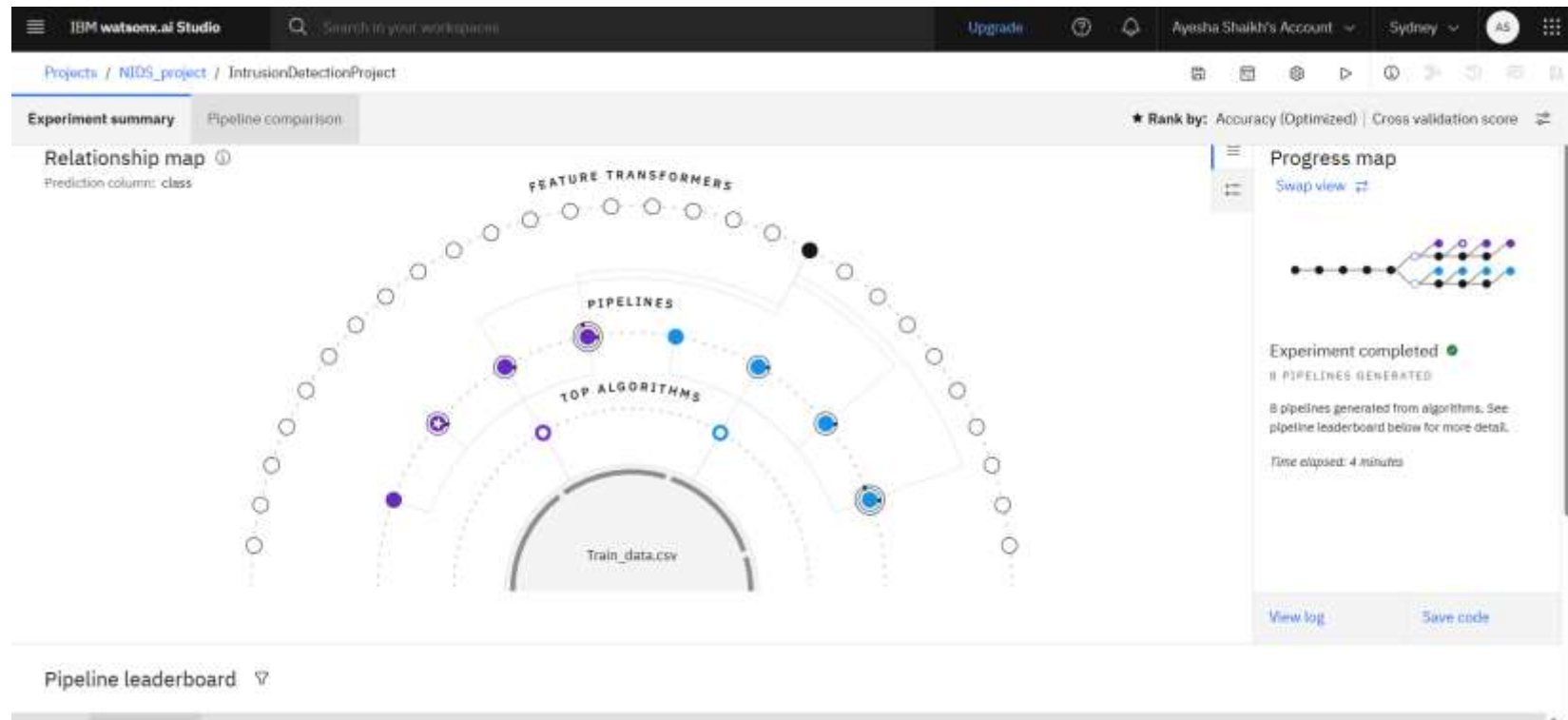


Fig 1. Relationship Map

RESULT

■ Key Results:

The model achieved **high accuracy (99.5 %)** in binary classification

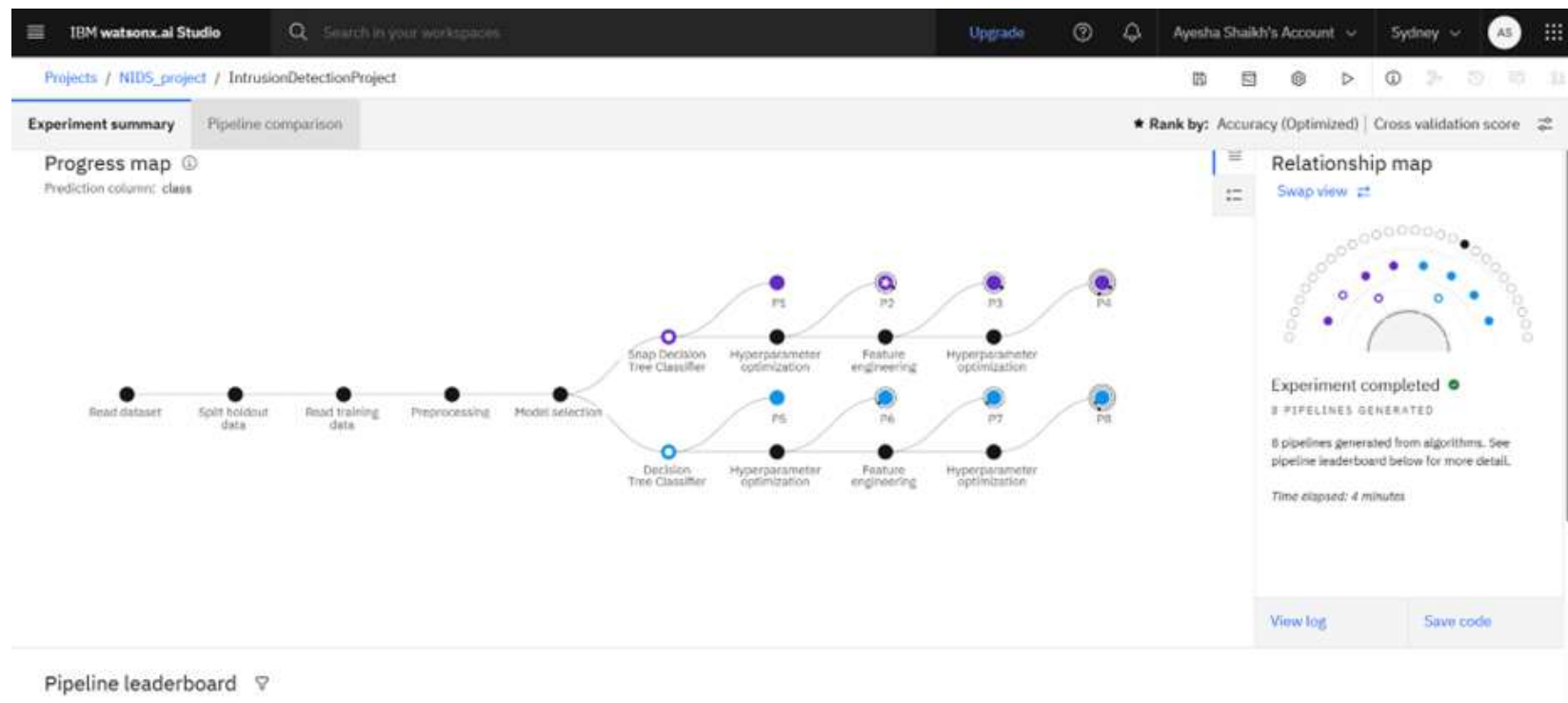
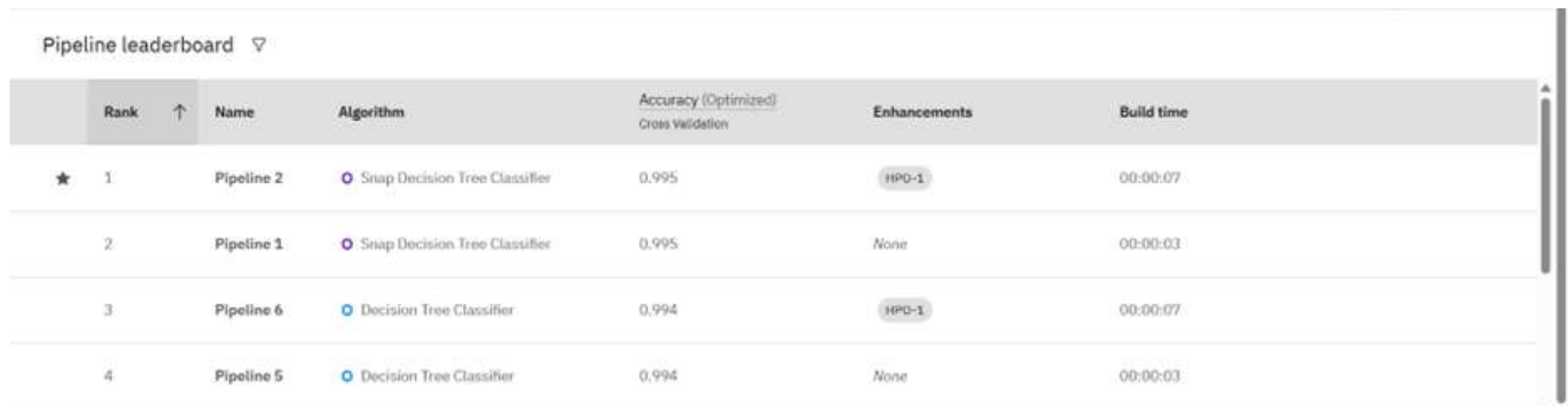


Fig 2. Progress Map

RESULT

■ Key Results:

The model achieved **high accuracy (99.5 %)** in binary classification



Pipeline leaderboard ▾

	Rank	↑	Name	Algorithm	Accuracy (Optimized) Cross Validation	Enhancements	Build time
★	1		Pipeline 2	ⓘ Snap Decision Tree Classifier	0.995	HPD-1	00:00:07
	2		Pipeline 1	ⓘ Snap Decision Tree Classifier	0.995	None	00:00:03
	3		Pipeline 6	ⓘ Decision Tree Classifier	0.994	HPD-1	00:00:07
	4		Pipeline 5	ⓘ Decision Tree Classifier	0.994	None	00:00:03

Fig 3. Pipeline Leaderboard

RESULT

■ Key Results:

The model correctly classified the networks as anomaly or normal using the manually entered data

The screenshot displays the IBM Watsonx.ai Studio interface. At the top, the navigation bar includes the IBM Watsonx.ai Studio logo, a search bar, an 'Upgrade' button, and user account information for 'Ayesha Shaikh's Account' in the 'Sydney' region. The breadcrumb trail indicates the current location: 'Deployment spaces / network_deploy / P2 - Snap Decision Tree Classifier: IntrusionDetectionProject /'. The main content area shows the 'NetworkIntrusionDetection' model, which is 'Deployed' and 'Online'. Below this, the 'Test' tab is selected, and the 'Enter input data' section is active. The 'Text' input type is chosen, and a table for manual data entry is displayed. The table has 10 columns: 'duration (double)', 'protocol_type (other)', 'service (other)', 'flag (other)', 'src_bytes (double)', 'dst_bytes (double)', 'land (double)', 'wrong_fragment (double)', 'urgent (double)', and an unlabeled column. Three rows of data are entered. A 'Predict' button is visible at the bottom right of the interface.

	duration (double)	protocol_type (other)	service (other)	flag (other)	src_bytes (double)	dst_bytes (double)	land (double)	wrong_fragment (double)	urgent (double)
1	0	tcp	ftp_data	SF	491	0	0	0	0
2	0	udp	other	SF	146	0	0	0	0
3	2	tcp	private	S0	0	0	0	0	0

Fig 4. Manually entered records

RESULT

■ Key Results:

The model correctly classified the networks as anomaly or normal using the manually entered data

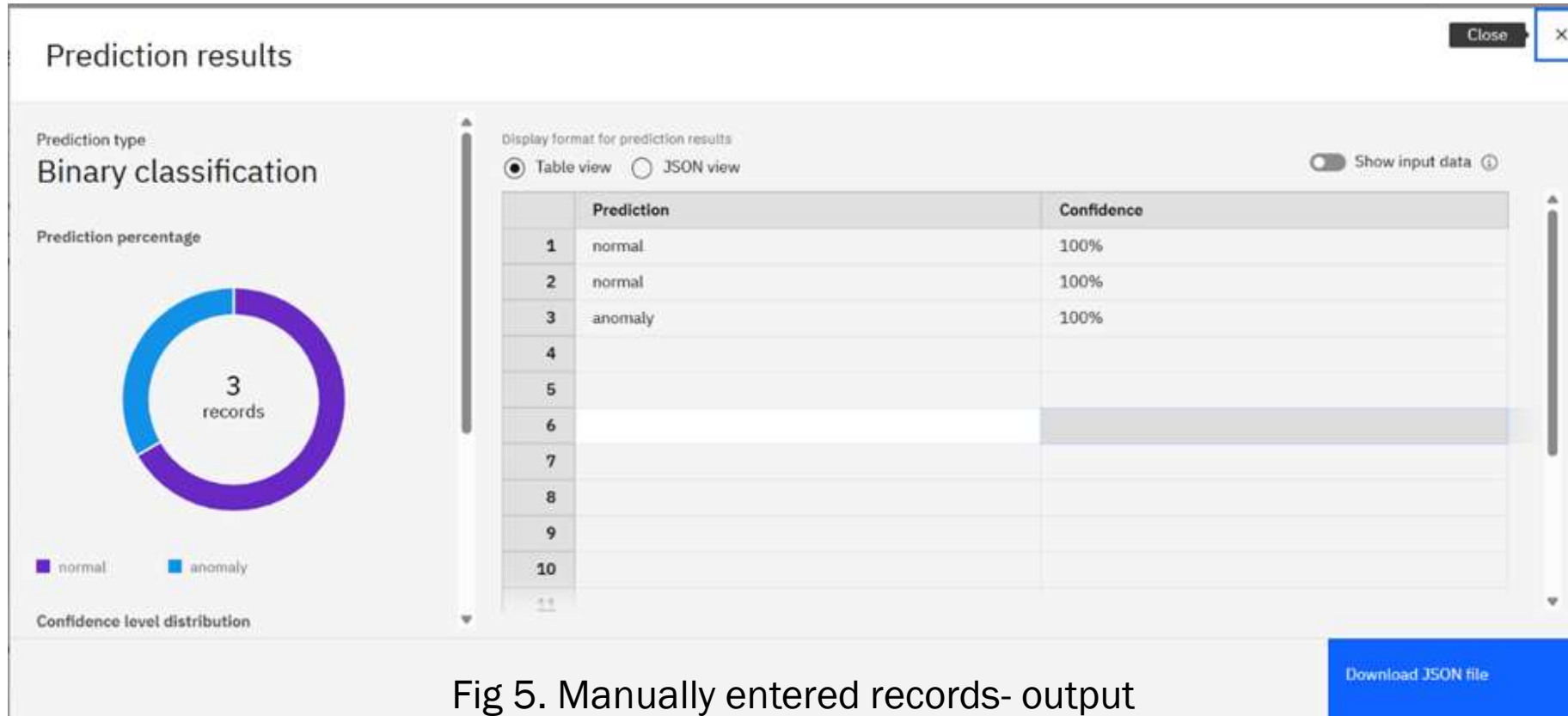


Fig 5. Manually entered records- output

RESULT

■ Key Results:

The model also correctly classified the networks as anomaly or normal using the original data (test.csv) from the Kaggle Network Intrusion Detection Dataset

The screenshot displays the IBM Watsonx.ai Studio interface. At the top, the header includes the IBM Watsonx.ai Studio logo, a search bar, and user account information for Ayesha Shaikh. The main content area shows a deployment space for 'network_deploy' with a project named 'P2 - Snap Decision Tree Classifier: IntrusionDetectionProject'. Below this, the model 'NetworkIntrusionDetection' is shown as 'Deployed' and 'Online'. The 'Test' tab is active, showing an 'Enter input data' section with options for 'Text' and 'JSON'. A table of input data is displayed, with columns for various network metrics. The table contains 4 rows of data. At the bottom right, there is a 'Predict' button.

	duration (double)	protocol_type (other)	service (other)	flag (other)	src_bytes (double)	dst_bytes (double)	land (double)	wrong_fragment (double)	urgent (double)
1	0	tcp	private	REJ	0	0	0	0	0
2	0	tcp	private	REJ	0	0	0	0	0
3	2	tcp	ftp_data	SF	12983	0	0	0	0
4	0	icmp	eco_i	SF	20	0	0	0	0

22,544 rows, 41 columns

Predict

Fig 6. test.csv

RESULT

■ Key Results:

The model also correctly classified the networks as anomaly or normal using the original data (test.csv) from the Kaggle Network Intrusion Detection Dataset

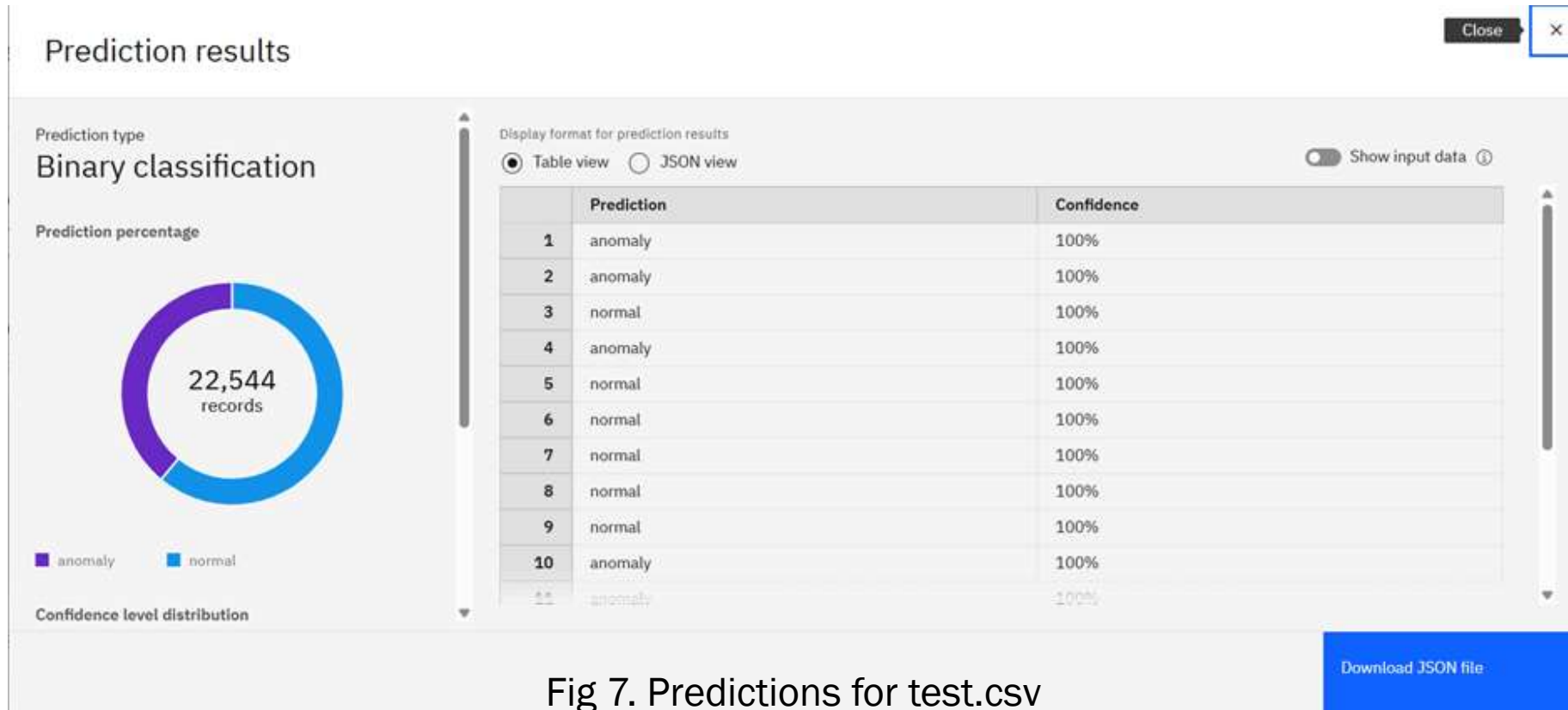


Fig 7. Predictions for test.csv

CONCLUSION

- This project successfully demonstrates how machine learning, combined with IBM Watson AutoAI, can be used to create an effective **Network Intrusion Detection System (NIDS)**.
- By training on the Train_data.csv dataset and validating with test_data.csv and manual inputs, the system accurately classified network traffic as either normal or anomalous.
- The IBM Cloud environment made it simple to:
 - Build and test models without writing much code
 - Deploy a working API
 - Test predictions in real-time
- This solution shows strong potential for real-world applications in cybersecurity, providing an automated, scalable approach to detect threats early.

FUTURE SCOPE

- In the future, this Network Intrusion Detection System can be enhanced by integrating real-time traffic monitoring tools like Wireshark to enable live detection of threats. Support for encrypted traffic analysis can also be explored using advanced techniques without compromising data privacy. To improve detection of complex patterns, deep learning models such as LSTM or GRU could be incorporated, especially for sequence-based attacks. The system can be further expanded to integrate with enterprise-grade security platforms like SIEM tools for automated alerting and reporting. Additionally, transitioning from binary classification to multi-class classification would allow identification of specific attack types, making the system more informative and actionable in real-world cybersecurity environments.

REFERENCES

- Dataset Source: https://www.kaggle.com/datasets/sampadab17/network-intrusion-detection?select=Train_data.csv
- IBM Watson Studio: <https://dataplatform.cloud.ibm.com>
- Getting Started with AutoAI on IBM Cloud
<https://cloud.ibm.com/docs/autoai>

IBM CERTIFICATIONS



IBM CERTIFICATIONS

In recognition of the commitment to achieve
professional excellence



Ayesha Shaikh

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/b0d7144e-78db-4dda-b58a-ab844090c140>



IBM CERTIFICATIONS

IBM SkillsBuild

Completion Certificate



This certificate is presented to

Ayesha Shaikh

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



THANK YOU