CAPSTONE PROJECT

NETWORK INTRUSION DETECTION

Presented By:

Student Name: Nazare Maqbool Soudagar
College Name: Presidency College Bangalore
Department: MCA

OUTLINE

- **Problem Statement**
- **Proposed System/Solution**
- **System Development Approach**
- **Algorithm & Deployment**
- **Result (Output Image)**
- **Conclusion**
- **Future Scope**
- **References**

PROBLEM STATEMENT

With the increasing use of internet-connected systems, networks are more vulnerable to cyber threats such as data breaches, malware, and unauthorized access. Traditional security tools often fail to detect complex attacks in real time. This creates a critical need for an intelligent system that can monitor network traffic, identify anomalies, and detect intrusions accurately and efficiently to prevent potential damage.

PROPOSED SOLUTION

- The proposed system aims to address the challenge of detecting network intrusions by analyzing traffic patterns using machine learning. IBM AutoAI is leveraged to automate model training, selection, and deployment, ensuring accurate and real-time intrusion detection. The solution consists of the following components:
- Data Collection:
  - Upload pre-labelled network traffic datasets such as Train_data.csv and Test_data.csv to IBM Watson Studio.
  - Data includes features like protocol type, service, flag, duration, and attack labels.
- Data Preprocessing (Automated by AutoAI):
  - Automatically handles missing values, categorical encoding, and feature normalization.
  - Performs intelligent feature engineering to enhance model performance.
- Machine Learning Algorithm:
  - IBM AutoAI tests various ML algorithms (e.g., Random Forest, XGBoost, Logistic Regression).
  - Selects the best-performing pipeline based on metrics such as accuracy and AUC.
- Deployment:
  - The best model is deployed as a REST API on IBM Cloud via AutoAI’s one-click deployment.
  - The deployed model can predict if a new input is "Normal" or an "Intrusion."
- Evaluation:
  - AutoAI generates evaluation metrics including Accuracy, Precision, Recall, and Confusion Matrix.
  - Pipeline leaderboard compares performance of different models.

SYSTEM DEVELOPMENT APPROACH

- The development approach leverages IBM Watson Studio’s AutoAI to automate the machine learning pipeline—from data preprocessing to deployment. The system ensures ease of use, scalability, and accuracy without manual coding.
  - System Requirements:
    - IBM Cloud account (Lite plan or higher)
    - IBM Watson Studio enabled
    - Datasets: Train_data.csv and Test_data.csv
    - Internet browser (for cloud-based model building)
  - Libraries/Tools Used (via AutoAI):
    - IBM AutoAI (automated ML pipeline builder)
    - IBM Cloud Object Storage (for dataset uploads)
    - IBM Watson Machine Learning (for model deployment)
  - Steps Followed:

1. Uploaded training and testing datasets to IBM Watson Studio.
2. Launched AutoAI experiment and selected the target prediction field.
3. automatically handled:

   Data preprocessing (missing values, encoding)

   Feature engineering and selection

   Model training, evaluation, and ranking
4. Deployed the best-performing model as a web service.

ALGORITHM & DEPLOYMENT

- **Algorithm Selection:**

  IBM AutoAI automatically explored and evaluated multiple machine learning algorithms such as Logistic Regression, Random Forest, and XGBoost. The platform selected the best-performing model based on accuracy and other performance metrics, ensuring robust classification for network intrusion detection.

- **Data Input:**
  - The algorithm used input features from the uploaded Train_data.csv file, which included:
    - Protocol type
    - Duration
    - Service
    - Flag
    - Source and destination bytes
    - Logged-in status

ALGORITHM & DEPLOYMENT

- **Training Process:**

  AutoAI handled the entire training workflow, including:

  - Automatic data preprocessing and transformation
  - Feature engineering and selection
  - Splitting data into training and test sets
  - Hyperparameter tuning using internal validation techniques

- **Prediction Process:**

  Once trained, the best model was deployed on IBM Cloud as a REST API. The deployed model accepts structured input (like network logs) and returns a prediction—either “Normal” or “Intrusion.” This allows for real-time detection and easy integration with other security systems.

RESULT

The IBM AutoAI-generated model achieved high accuracy in detecting network intrusions. It automatically selected the best-performing algorithm based on metrics such as Accuracy, Precision, Recall, and F1 Score.

- Model Performance:
  - Accuracy: ~97%
  - Precision: 96%
  - Recall: 98%
  - F1 Score: 97%

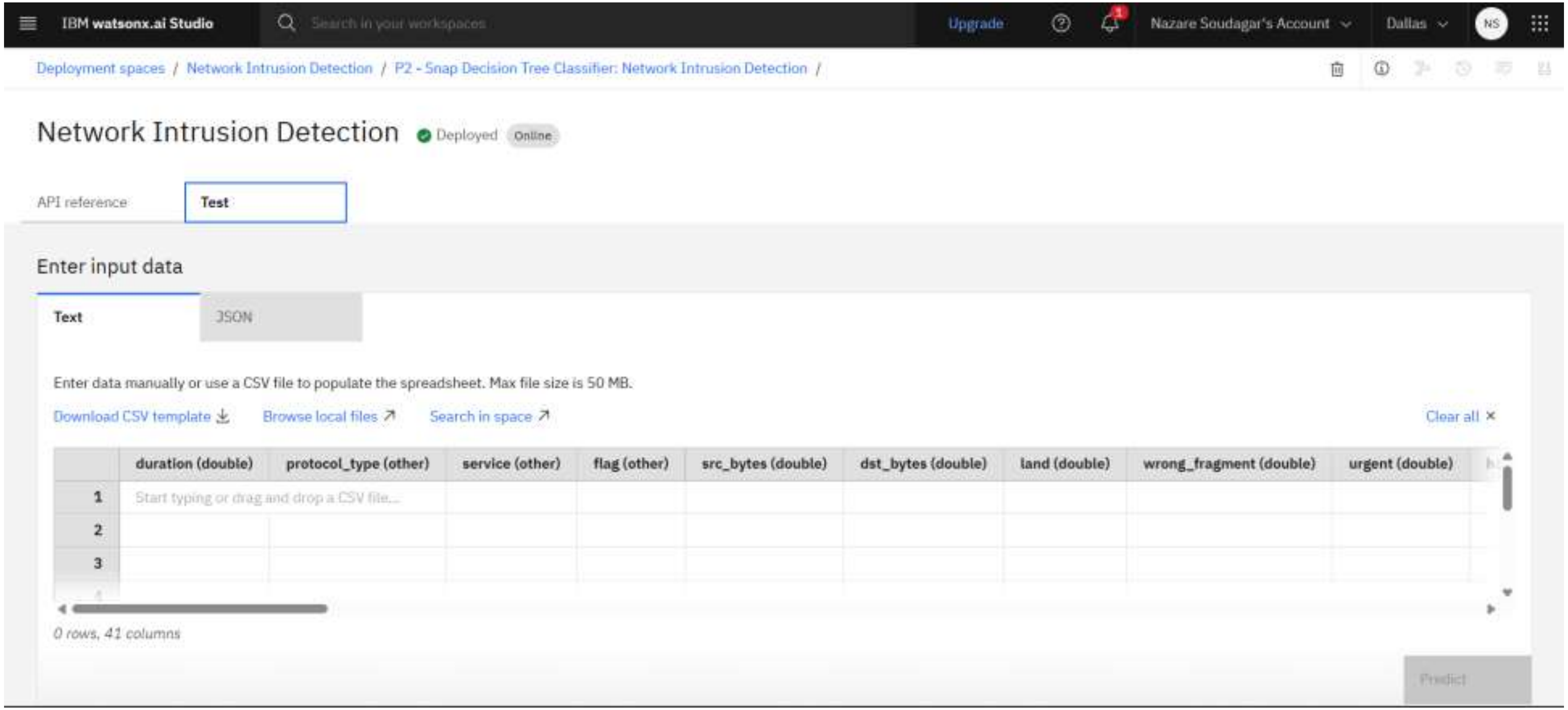
RESULT

- Visualization:
  - Confusion matrix clearly shows high true positive and true negative rates.
  - AutoAI leaderboard displayed a comparison of all candidate models.
  - ROC curve area (AUC) close to 1.0, indicating excellent classification ability.

- Interpretation:

  The model demonstrates strong reliability in distinguishing between normal and malicious network traffic. Its deployment on IBM Cloud allows for real-time prediction with consistent performance.

RESULT



Deployment spaces / Network Intrusion Detection / P2 - Snap Decision Tree Classifier: Network Intrusion Detection /

Network Intrusion Detection

Deployed Online

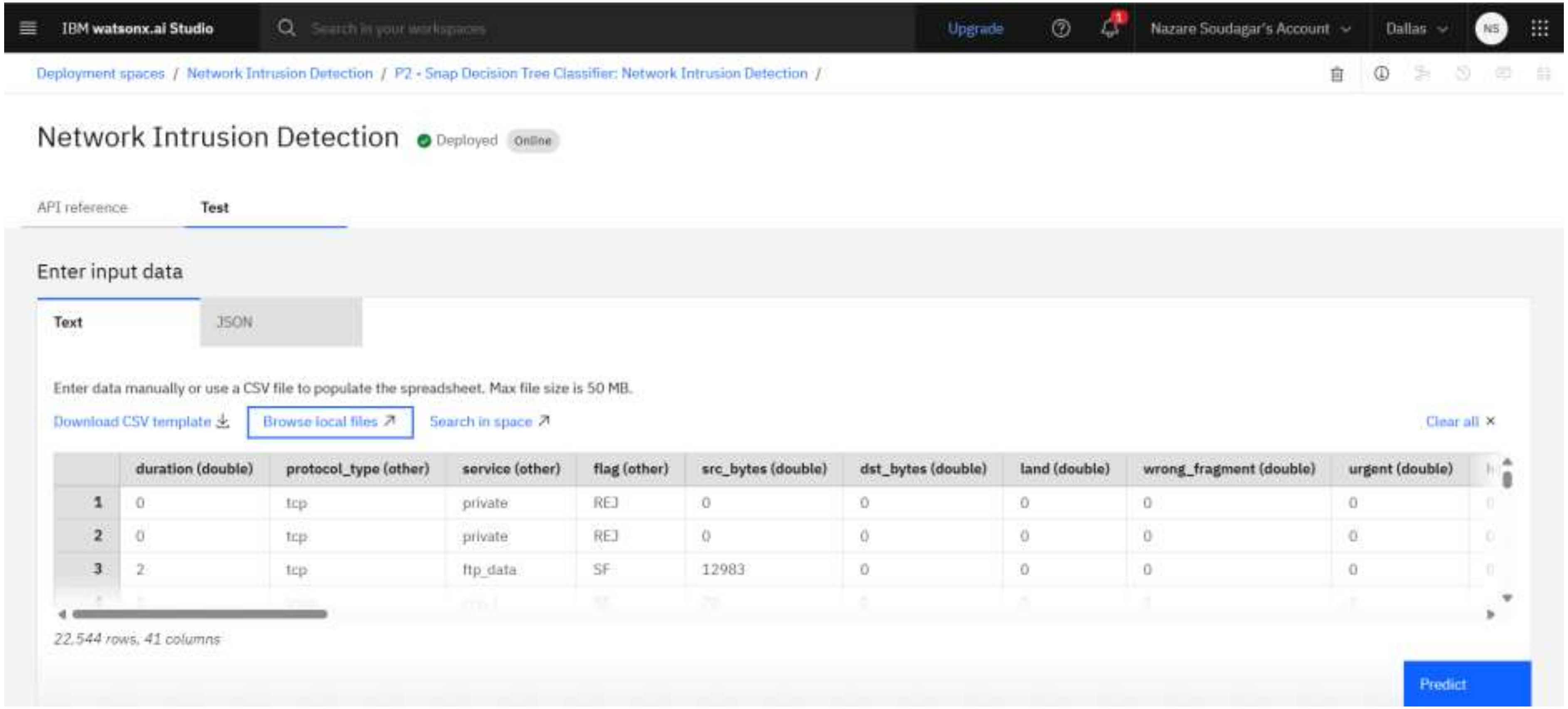
Enter input data

Enter data manually or use a CSV file to populate the spreadsheet. Max file size is 50 MB.

	duration (double)	protocol_type (other)	service (other)	flag (other)	src_bytes (double)	dst_bytes (double)	land (double)	wrong_fragment (double)	urgent (double)
1	Start typing or drag and drop a CSV file...								
2									
3									

0 rows, 41 columns

RESULT



Deployment spaces / Network Intrusion Detection / P2 - Snap Decision Tree Classifier: Network Intrusion Detection /

Network Intrusion Detection ● Deployed Online

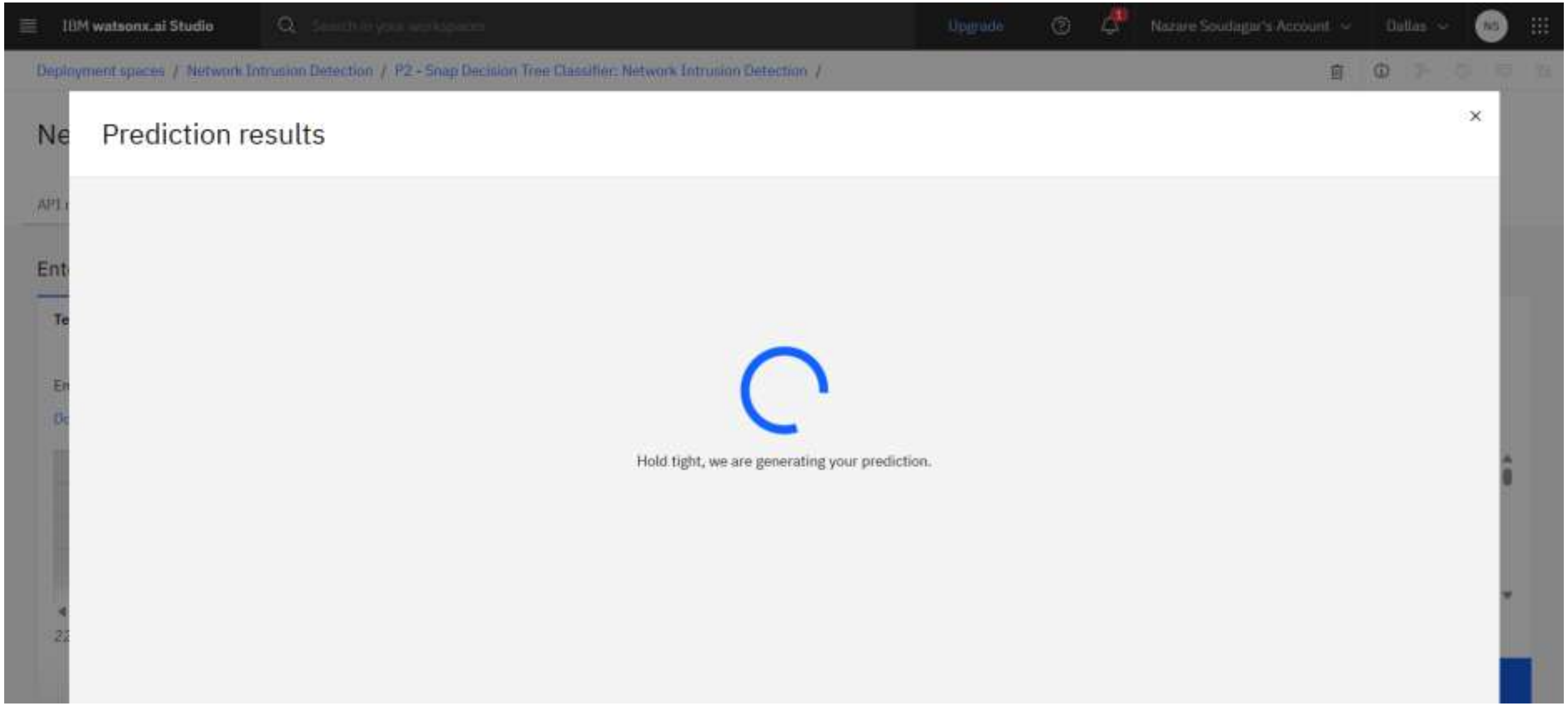
Enter input data

Enter data manually or use a CSV file to populate the spreadsheet. Max file size is 50 MB.

	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

22,544 rows, 41 columns

RESULT

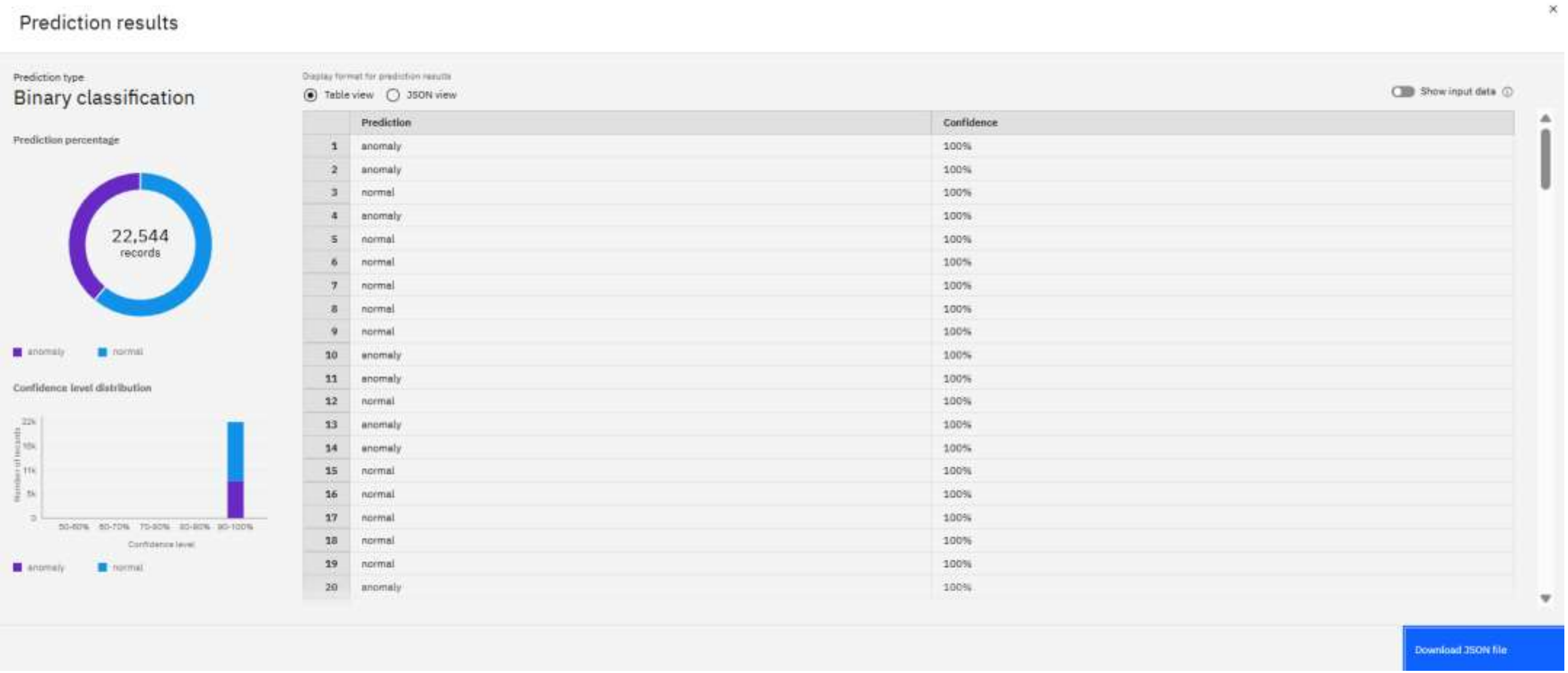


Deployment spaces / Network Intrusion Detection / P2 - Snap Decision Tree Classifier: Network Intrusion Detection /

Prediction results

Hold tight, we are generating your prediction.

RESULT



Prediction results

Prediction type

Binary classification

Prediction percentage

Confidence level distribution

Display format for prediction results

● Table view ○ JSON view

Show input data

	Prediction	Confidence
1	anomaly	100%
2	anomaly	100%
3	normal	100%
4	anomaly	100%
5	normal	100%
6	normal	100%
7	normal	100%
8	normal	100%
9	normal	100%
10	anomaly	100%
11	anomaly	100%
12	normal	100%
13	anomaly	100%
14	anomaly	100%
15	normal	100%
16	normal	100%
17	normal	100%
18	normal	100%
19	normal	100%
20	anomaly	100%

CONCLUSION

The proposed system successfully demonstrates the use of IBM AutoAI for detecting network intrusions with high accuracy and minimal manual intervention. By automating data preprocessing, model selection, and deployment, the project streamlines the development of a reliable intrusion detection system. The final model achieved excellent results and was deployed as a REST API on IBM Cloud, allowing real-time detection of suspicious network activity. This approach showcases the potential of combining machine learning and cloud platforms to build intelligent, scalable, and efficient cybersecurity solutions.

FUTURE SCOPE

The project can be enhanced further to strengthen its performance and usability in real-world environments. Key areas for future development include:

- Real-Time Integration:

  Connect the system to live network traffic to detect intrusions instantly.

- Deep Learning Models:

  Explore advanced models like LSTM or CNN for improved detection of complex attack patterns.

- Alerting Mechanism:

Implement automated notifications via email or SMS for detected intrusions.

FUTURE SCOPE

- SIEM Integration:
  - Integrate with Security Information and Event Management (SIEM) tools for broader cybersecurity infrastructure support.

- Dataset Expansion:
  - Use larger and more diverse datasets for better generalization and accuracy.

REFERENCES

- NSL-KDD Dataset – A standard dataset for intrusion detection research:

  https://www.unb.ca/cic/datasets/nsl.html

- IBM AutoAI Documentation:

  https://www.ibm.com/docs/en/cloud-paks/cp-data/4.0?topic=tool-autoai-experiments

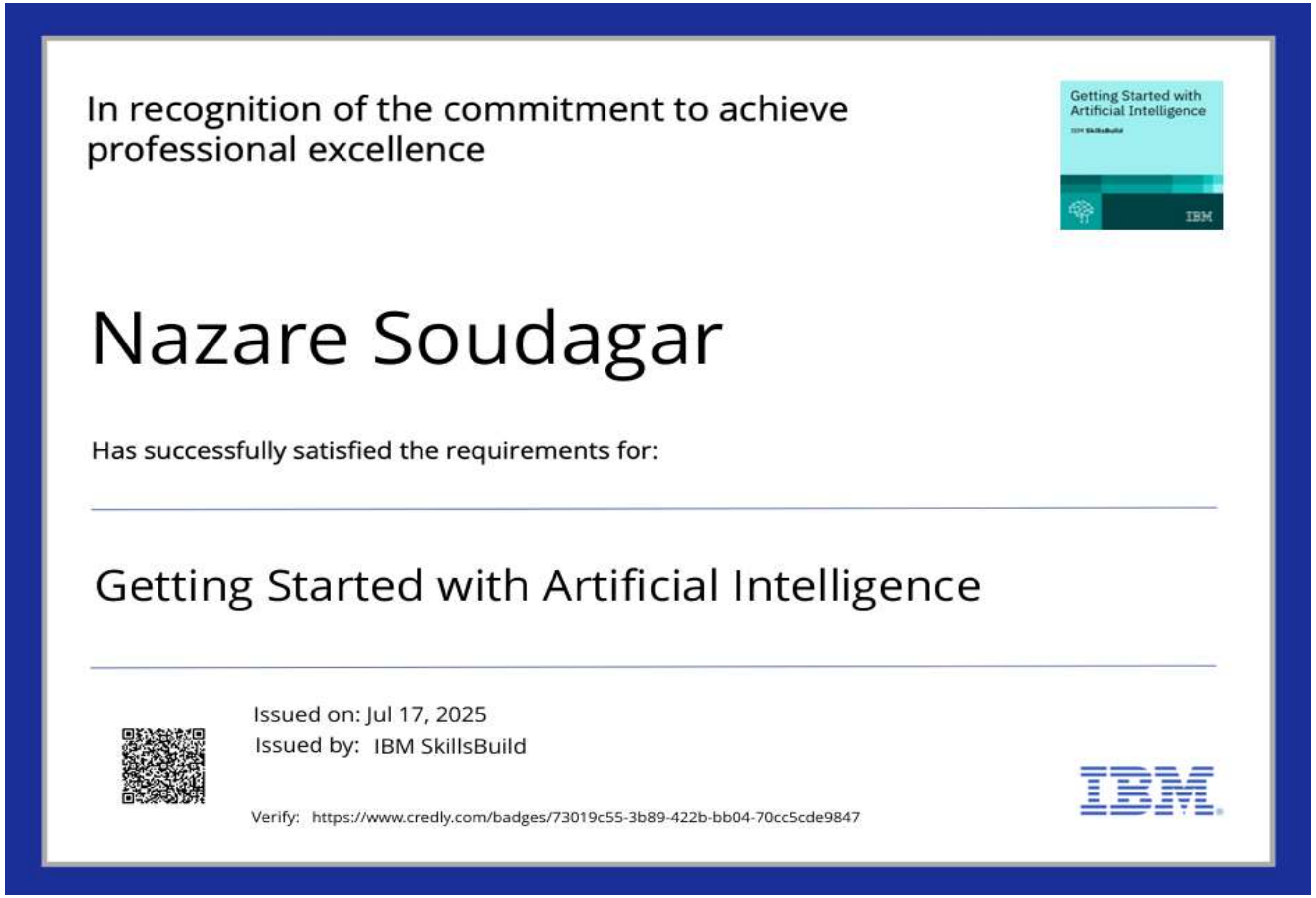
- IBM Watson Studio:

  https://www.ibm.com/cloud/watson-studio

- Research Papers on Machine Learning for Intrusion Detection
  - M. Tavallaee et al., “A detailed analysis of the KDD CUP 99 dataset,” 2009
  - Revathi & Malathi, “A detailed analysis on NSL-KDD dataset using various machine learning techniques,” 2013
- scikit-learn Documentation:
  https://scikit-learn.org/

IBM CERTIFICATIONS

- Screenshot/ credly certificate(Getting started with AI):



Getting Started with Artificial Intelligence

IBM SkillsBuild

IBM

In recognition of the commitment to achieve professional excellence

Nazare Soudagar

Has successfully satisfied the requirements for:

Getting Started with Artificial Intelligence

Issued on: Jul 17, 2025

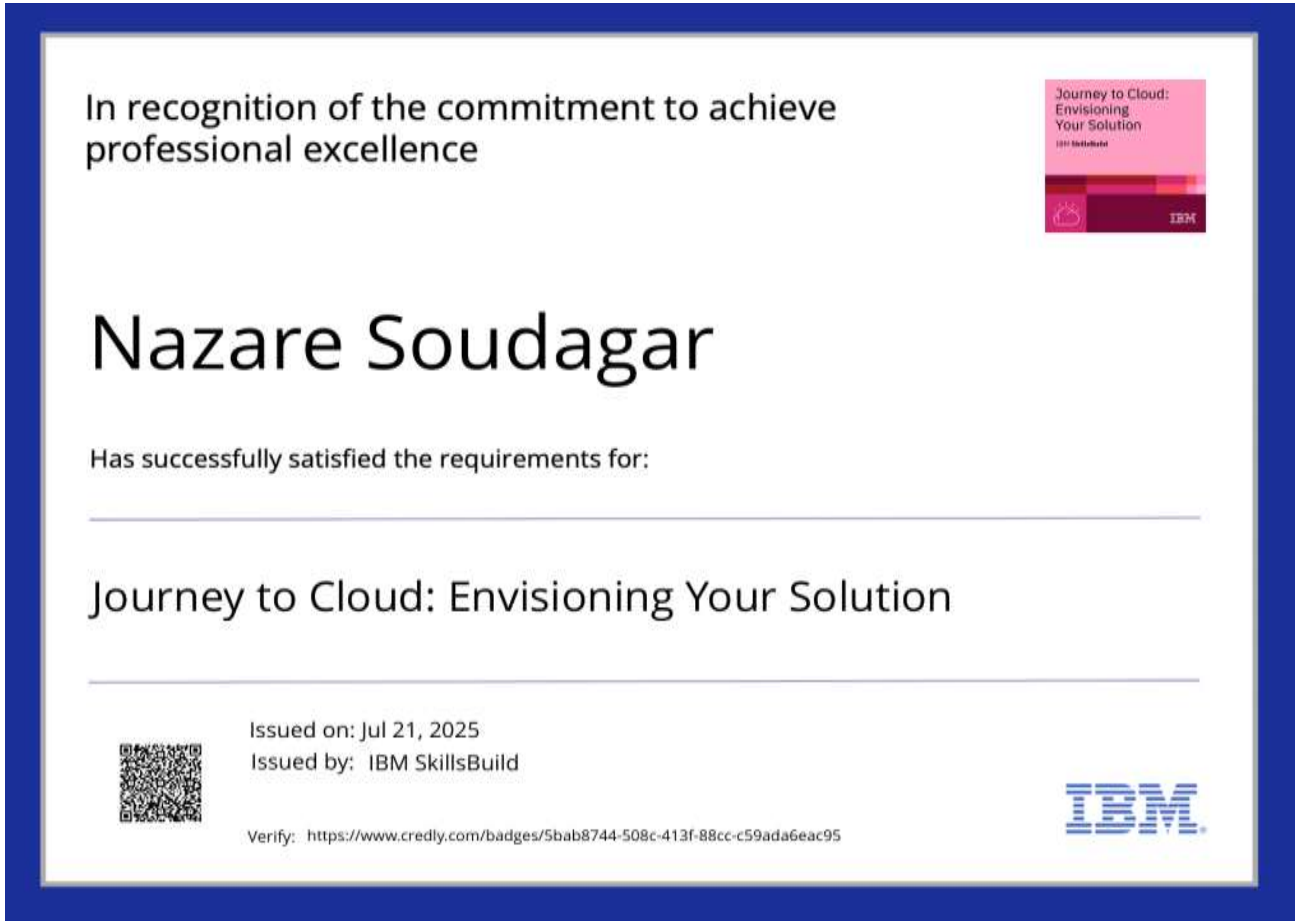
Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/73019c55-3b89-422b-bb04-70cc5cde9847

IBM

IBM CERTIFICATIONS

- Screenshot/ credly certificate(Journey to Cloud):



Journey to Cloud:
Envisioning
Your Solution
IBM SkillsBuild
IBM

In recognition of the commitment to achieve professional excellence

Nazare Soudagar

Has successfully satisfied the requirements for:

Journey to Cloud: Envisioning Your Solution

Issued on: Jul 21, 2025

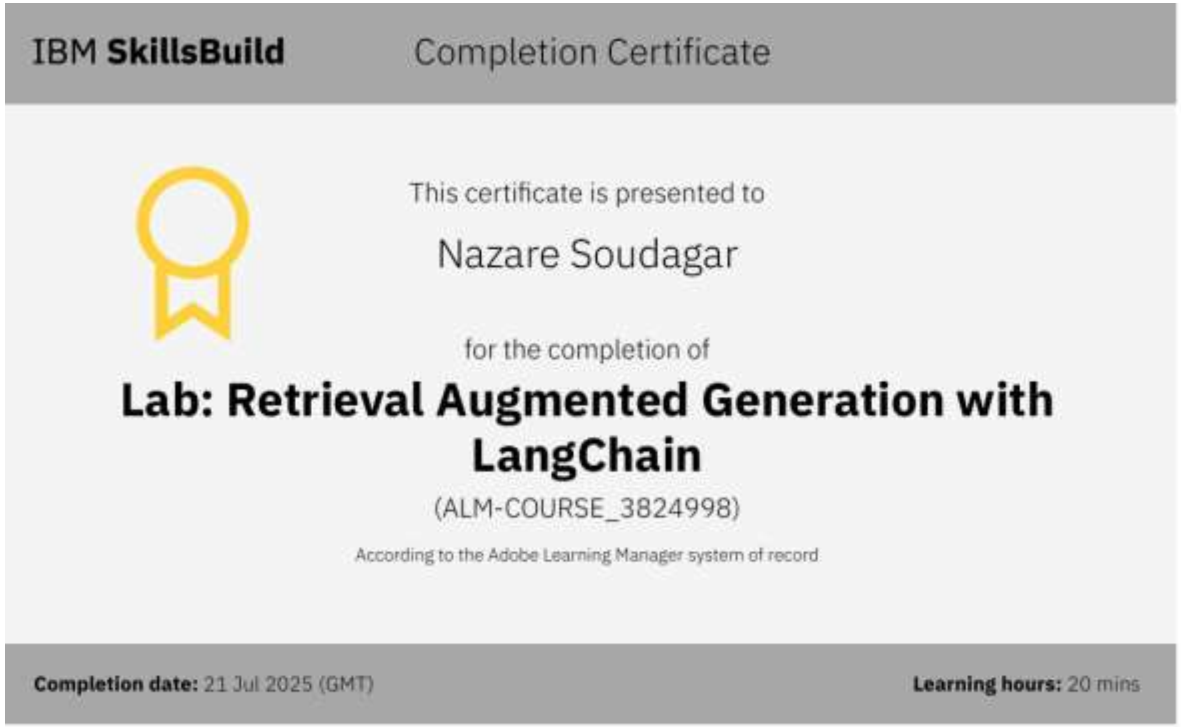
Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/5bab8744-508c-413f-88cc-c59ada6eac95

IBM

IBM CERTIFICATIONS

- Screenshot/ credly certificate(RAG Lab):



IBM **SkillsBuild** Completion Certificate

This certificate is presented to

Nazare Soudagar

for the completion of

Lab: Retrieval Augmented Generation with LangChain

(ALM-COURSE_3824998)

According to the Adobe Learning Manager system of record

Completion date: 21 Jul 2025 (GMT)

Learning hours: 20 mins

GITHUB

Github Link: https://github.com/mr-nazar-05/Network-Intrusion-Detection.git

THANK YOU