
CAPSTONE PROJECT

NETWORK INTRUSION DETECTION SYSTEM USING AUTOAI

Presented By:
Mohd Faizan
Nawab Shah Alam Khan College of Engineering and Technology
CSE(AI/ML)

OUTLINE

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

Problem Statement

With the exponential growth of internet-connected systems, cybersecurity threats have become more sophisticated, targeting networks with diverse attack vectors. Traditional intrusion detection systems often rely on predefined signatures and cannot adapt to new or evolving attack patterns.

This results in undetected breaches, data theft, and potential damage to critical infrastructure.

There is a pressing need for an intelligent, automated system capable of detecting multiple types of attacks in real-time and differentiating them from normal network activity.

Proposed Solution

The proposed solution is an AI-powered Network Intrusion Detection System (NIDS) built using IBM Watson Studio's AutoAI service. The system is designed to automatically train and optimize machine learning models to classify network traffic as either Normal or Anomalous.

Key features of the solution include:

- Automated data preprocessing and feature engineering.
- Evaluation of multiple algorithms to select the best-performing model.
 - Deployment as a cloud-based API for real-time prediction.
- Scalability to handle large-scale network traffic data and evolving attack patterns.

System Approach

System Requirements:

- IBM Cloud Lite account with Watson Studio and Watson Machine Learning services.
 - AutoAI experiment setup in Watson Studio.
- Dataset: KDD-based intrusion detection dataset with Normal and Anomalous labels.

Libraries/Technologies Used:

- IBM Watson Studio AutoAI
 - Python for API testing
- Pandas, Requests for data handling and integration
 - Cloud Object Storage for data and model storage

System Approach

Approach

1. Data Preparation

- Upload Train_data.csv to Watson Studio project assets
- Ensure target column class is correctly labeled

2. Model Building with AutoAI

- Create AutoAI experiment and select dataset
- Set problem type to Classification
- Enable automated preprocessing, feature engineering, and pipeline generation

3. Model Selection & Deployment

- Review leaderboard and select top-performing pipeline
- Save as model and deploy as an Online Deployment
- Obtain Scoring URL and API Key

4. Testing & Validation

- Send sample input data via Python script using requests
- Verify predictions and probabilities returned by API

Algorithm & Deployment

Algorithm Selection:

- AutoAI automatically evaluates multiple models (Random Forest, Gradient Boosting, XGBoost, Logistic Regression, etc.) and selects the one with the highest F1-score for optimal classification.

Data Input:

- 41 features per network connection, including 3 categorical and 38 numerical attributes.

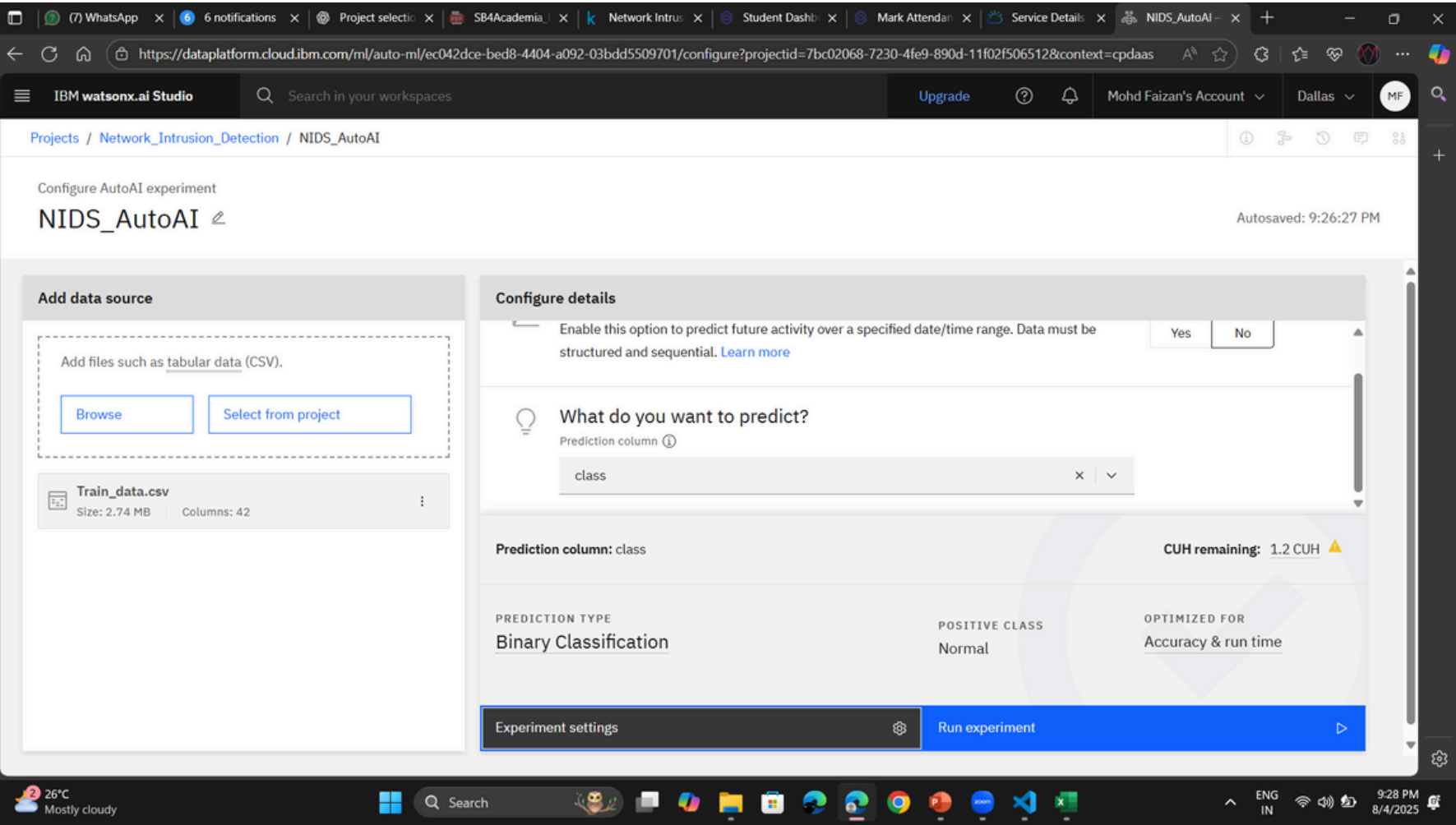
Training Process:

- AutoAI performs automated preprocessing, feature engineering, model training, and hyperparameter tuning.
 - Pipelines are ranked based on evaluation metrics.

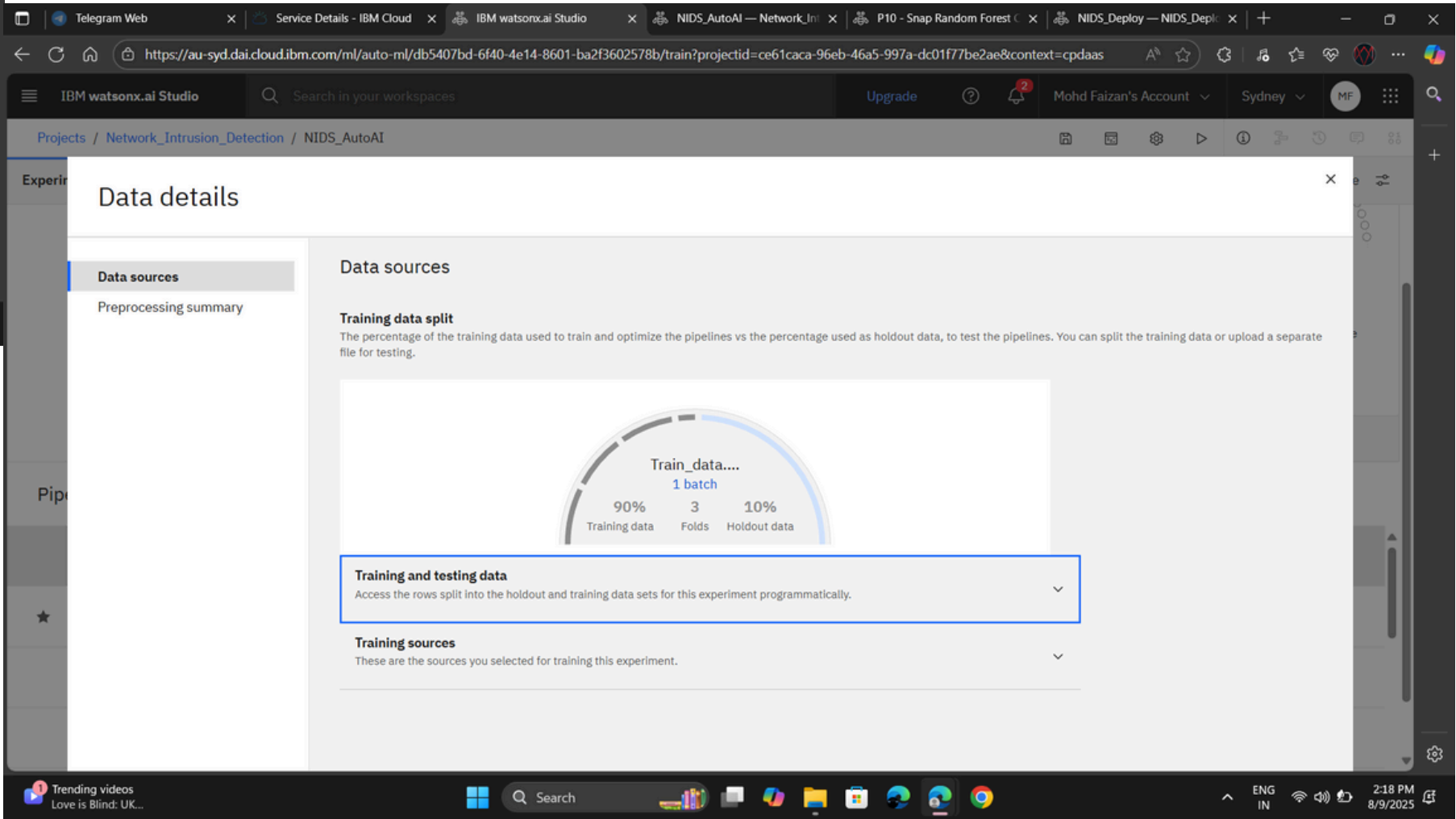
Deployment:

- The best pipeline is saved as a model in Watson Machine Learning.
- Deployed as an Online API endpoint for real-time intrusion detection.

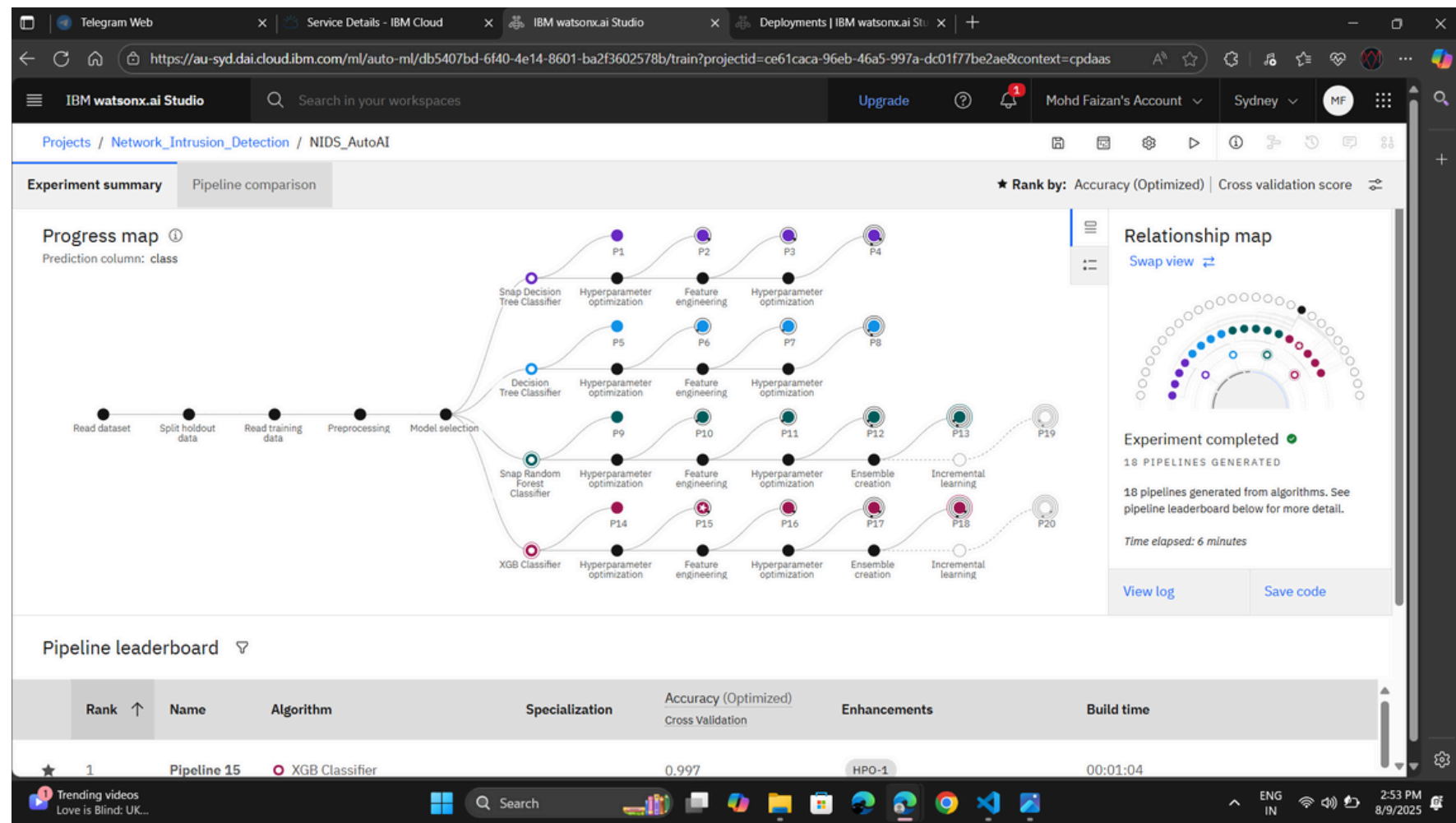
Result



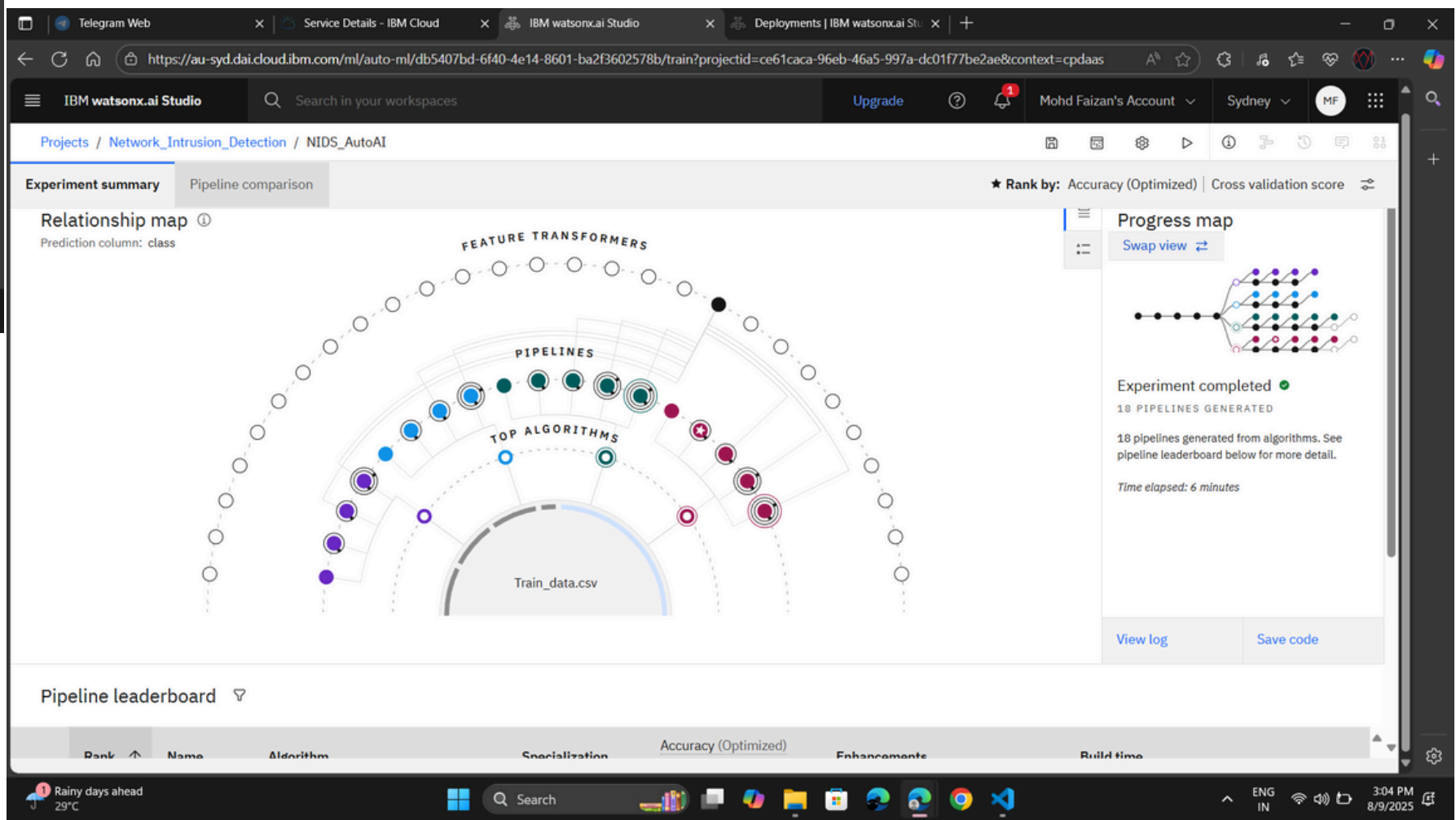
Data



Result











































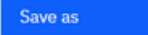














Progress Map & Relationship Map



Result

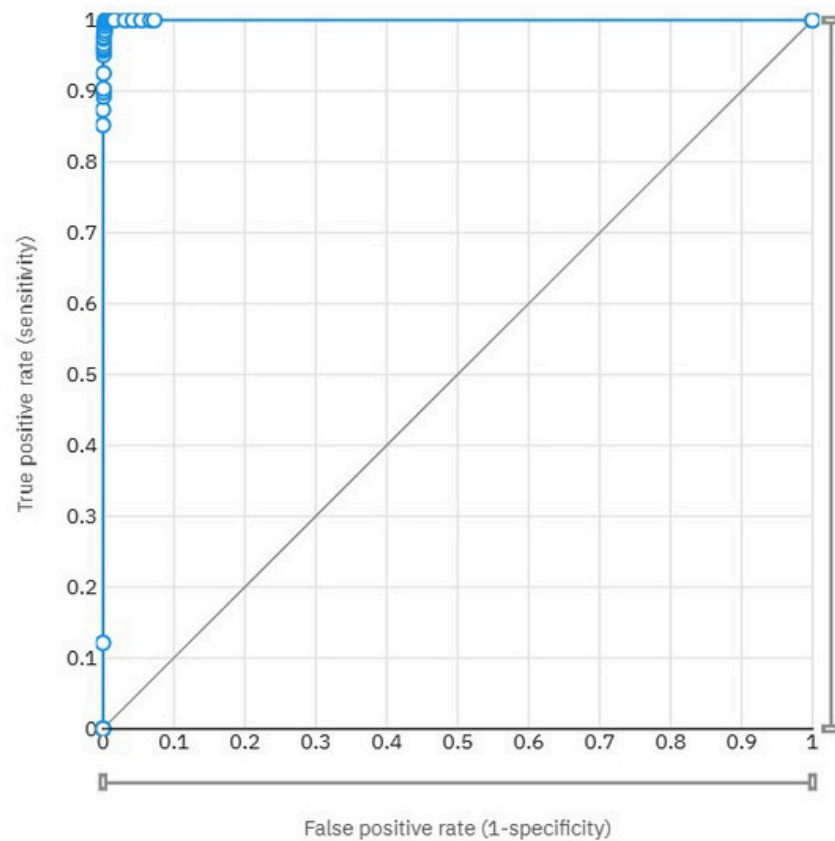
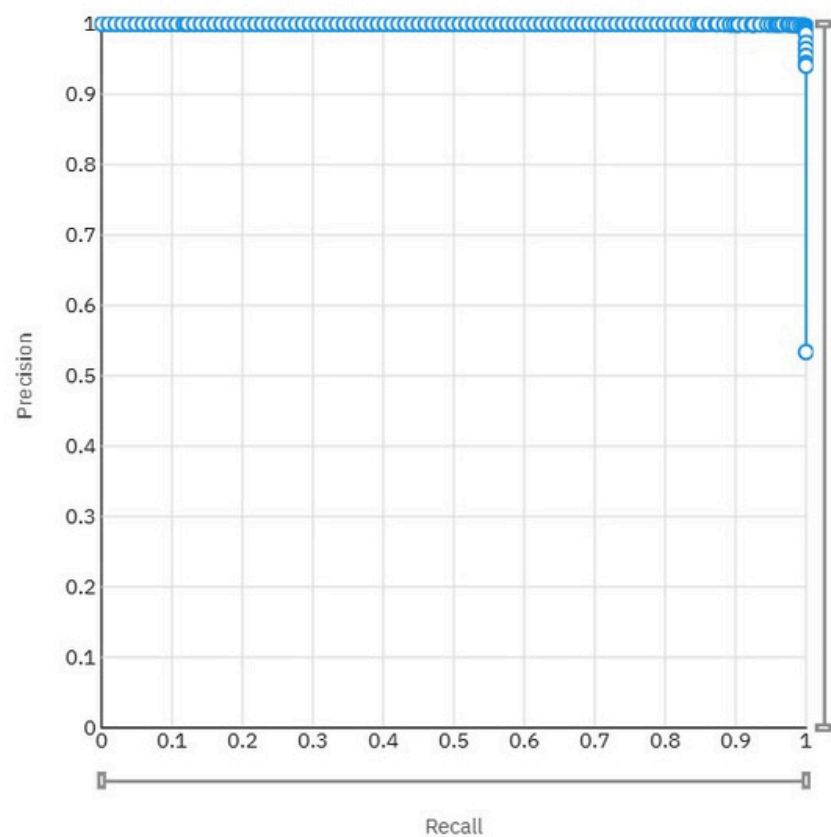
Pipeline Leaderboard

Pipeline leaderboard 

	Rank 	Name	Algorithm	Specialization	Accuracy (Optimized) Cross Validation	Enhancements	Build time
★	1	Pipeline 15	 XGB Classifier		0.997		00:01:04
	2	Pipeline 14	 XGB Classifier		0.997	None	00:00:50
	3	Pipeline 18	 Batched Tree Ensemble Classifier (XGB Classifier)		0.997	   	00:01:13
	4	Pipeline 17	 XGB Classifier		0.997	  	00:01:06
	5	Pipeline 16	 XGB Classifier		0.996	 	00:00:39
	6	Pipeline 10	 Snap Random Forest Classifier		0.995		00:01:07
	7	Pipeline 9	 Snap Random Forest Classifier		0.995	None	00:00:51
	8	Pipeline 2	 Snap Decision Tree Classifier		0.995		00:00:56
	9	Pipeline 1	 Snap Decision Tree Classifier		0.995	None	00:00:52
	10	Pipeline 13	 Batched Tree Ensemble Classifier (Snap Random Forest Classifier)		0.994	   	00:00:46
	11	Pipeline 12	 Snap Random Forest Classifier		0.994	  	00:00:42
	12	Pipeline 11	 Snap Random Forest Classifier		0.994	 	00:00:29
	13	Pipeline 6	 Decision Tree Classifier		0.994		00:00:07
	14	Pipeline 5	 Decision Tree Classifier		0.994	None	00:00:03 
	15	Pipeline 4	 Snap Decision Tree Classifier		0.994	  	00:01:26
	16	Pipeline 3	 Snap Decision Tree Classifier		0.994	 	00:01:21
	17	Pipeline 8	 Decision Tree Classifier		0.993	  	00:00:48
	18	Pipeline 7	 Decision Tree Classifier		0.993	 	00:00:42

Result

XGB Classifier Evaluation Metrics

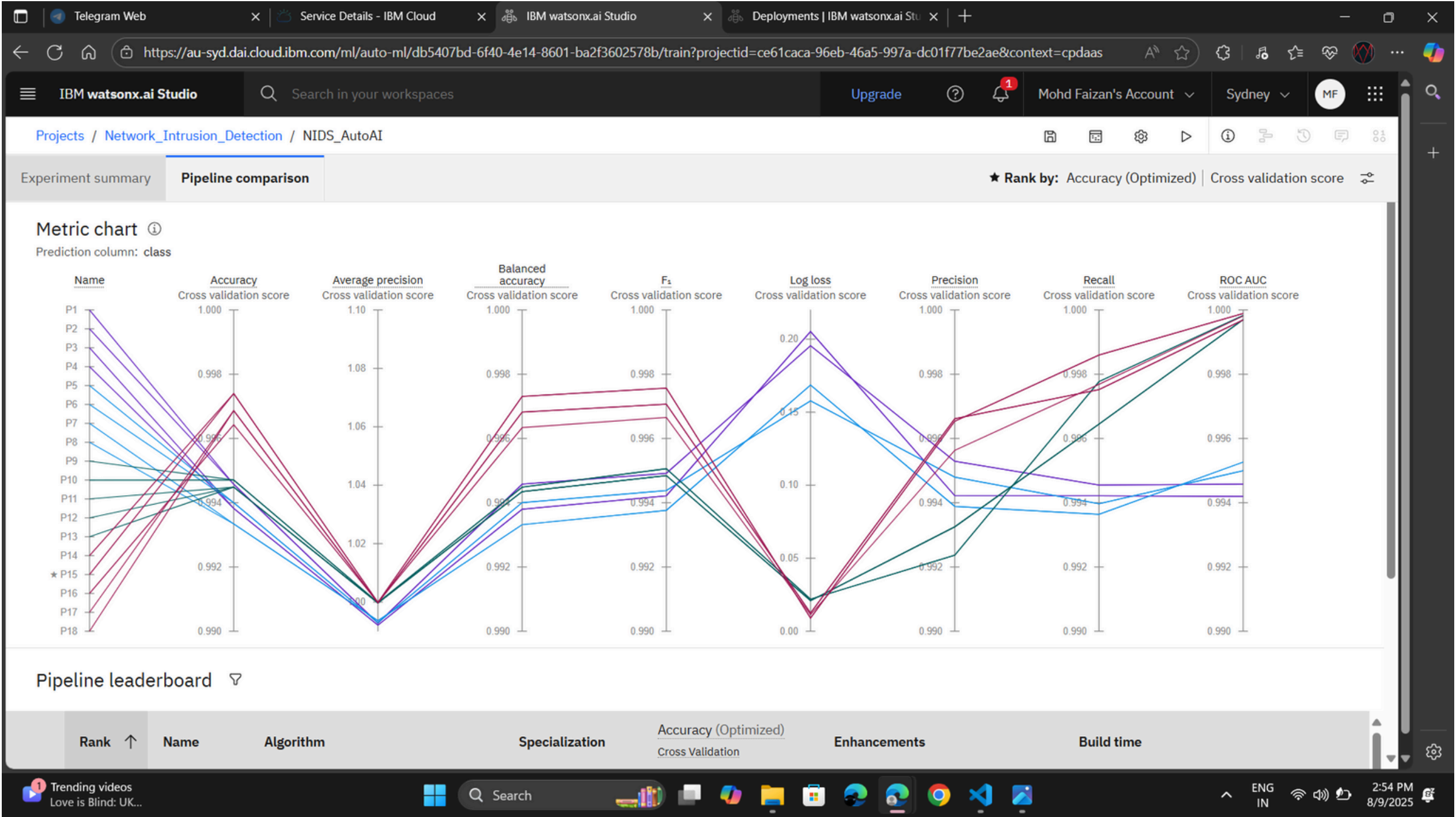


Model evaluation measure		
Measures	Holdout score	Cross validation score
Accuracy	0.998	0.996
Area under ROC	1.000	0.999
Precision	0.996	0.995
Recall	0.999	0.997
F1	0.998	0.996
Average precision	1.000	0.999
Log loss	0.009	0.022

Confusion matrix ⓘ			
Observed	Predicted		
	normal	anomaly	Percent correct
normal	1344	1	99.9%
anomaly	5	1170	99.6%
Percent correct	99.6%	99.9%	99.8%

Less correctMore correct

Result



PIPELINE Comparison

Result

IBM watsonx.ai Studio

Search in your workspaces

Upgrade

Mohd Faizan's Account

Sydney

MF

Deployment spaces / NIDS_Deploy / P15 - Random Forest Classifier: NIDS_AutoAI

Deployments

Model details

Search

New deployment

Name	Type	Status	Tags	Last modified
NIDS_Deploy	Online	Deployed		20 seconds ago Mohd Faizan (You)

Items per page: 201-1 of 1 items1 of 1 pages

About this asset

NameP15 - Random Forest Classifier: NIDS_AutoAI

DescriptionNo description provided.

Asset DetailsType: wml-hybrid_0.1
Model ID: e3439638-11fd-4b...
Software specification: hybrid_0.1
Hybrid pipeline software specifications: autoai-kb_rt24.1-py3.11

TagsAdd tags to make assets easier to find.

Source asset details

Last modified1 minute ago by Service

Created onAug 9, 2025 by Mohd Faizan

DEPLOYMENT

Telegram Web

Service Details - IBM Cloud

IBM watsonx.ai Studio

P15 - Random For

NIDS_Deploy — N

NIDS_AutoAI — N

NIDS_AutoAI.pdf

(7) WhatsApp

https://au-syd.dai.cloud.ibm.com/ml-runtime/deployments/4261a149-2dd5-4dd4-a23c-752d2058546e/implementation?space_id=c4d54d94-8e03-4df2-bc83-ead2622f63a6...

Upgrade

Mohd Faizan's Account

Sydney

MF

Deployment spaces / NIDS_Deploy / P15 - Random Forest Classifier: NIDS_AutoAI /

NIDS_Deploy

Deployed

Online

API reference

Test

Endpoints for scoring

Private endpointhttps://private.au-syd.ml.cloud.ibm.com/ml/v4/deployments/4261a149-2dd5-4dd4-a23c-752d2058546e/predictions?version=2021-05-01Bearer <token> IAM

Public endpointhttps://au-syd.ml.cloud.ibm.com/ml/v4/deployments/4261a149-2dd5-4dd4-a23c-752d2058546e/predictions?version=2021-05-01

Learn more about the 2021-05-01 version query parameter

Code snippets

cURLJavaJavaScriptPythonScala

NOTE: you must set \$API_KEY below using information retrieved from your IBM Cloud account (https://au-syd.dai.cloud.ibm.com/docs/content/wsj/analyze-data/ml-authentication.ht...
export API_KEY=<your API key>
export IAM_TOKEN=\$(curl --insecure -X POST --location "https://iam.cloud.ibm.com/identity/token" \ --header "Content-Type: application/x-www-form-urlencoded" \

Result

IBM watsonx.ai Studio

Search in your workspaces

Upgrade

Mohd Faizan's Account

Sydney

MF

Deployment spaces / NIDS_Deploy / P15 - Random Forest Classifier: NIDS_AutoAI /

NIDS_Deploy Deployed Online

API referenceTest

Enter input data

TextJSON

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

Clear all

	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
5	1	tcp	telnet	RSTO	0	15	0	0	0

22,544 rows, 41 columns

Predict

TESTING

IBM watsonx.ai Studio

Search in your workspaces

Upgrade

Mohd Faizan's Account

Sydney

MF

Deployment spaces / NIDS_Deploy / P15 - Random Forest Classifier: NIDS_AutoAI /

Prediction results

Prediction type

Binary classification

Prediction percentage

22,544 records

anomaly

normal

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	anomaly	60%
6	normal	100%
7	normal	99%
8	normal	100%
9	normal	100%
10	anomaly	50%
11	anomaly	60%

Download JSON file

Conclusion

- The AI-powered Network Intrusion Detection System successfully demonstrates the application of AutoAI for cybersecurity.
- By automating the process of model selection, training, and deployment, the system achieved high accuracy and adaptability against diverse network attacks.
- The deployment as a cloud API enables real-time threat detection without manual intervention.
- This approach reduces the dependency on static rule-based systems and improves resilience against evolving cyber threats.

GITHUB LINK

[https://github.com/FaizanSyntaX/Network Intrusion
Detection IBMCloud.git](https://github.com/FaizanSyntaX/Network_Intrusion_Detection_IBMCloud.git)

Future scope

- Integrate real-time streaming data analysis for live network monitoring. Expand the system to multi-class classification for identifying specific attack types (DoS, Probe, R2L, U2R).
- Incorporate anomaly detection techniques for zero-day threats. Enhance explainability to provide detailed reasons for flagged anomalies.
- Deploy on edge devices for local, low-latency detection in IoT and critical systems.

References

- Dataset: <https://www.kaggle.com/datasets/sampadab17/network-intrusion-detection>
- IBM Watson Studio Documentation: <https://dataplatform.cloud.ibm.com/docs>
- “A Detailed Analysis of the KDD Cup 99 Dataset” – Research Paper
- IBM AutoAI Overview: <https://www.ibm.com/cloud/watson-studio/autoai>

IBM Certifications

In recognition of the commitment to achieve
professional excellence



Mohd Faizan

Has successfully satisfied the requirements for:

Getting Started with Artificial Intelligence



Issued on: Jul 23, 2025
Issued by: IBM SkillsBuild

Verify: <https://www.credly.com/badges/4d8f9ce3-98d1-4a05-a684-65bc45d49519>



IBM Certifications

In recognition of the commitment to achieve
professional excellence



Mohd Faizan

Has successfully satisfied the requirements for:

Journey to Cloud: Envisioning Your Solution

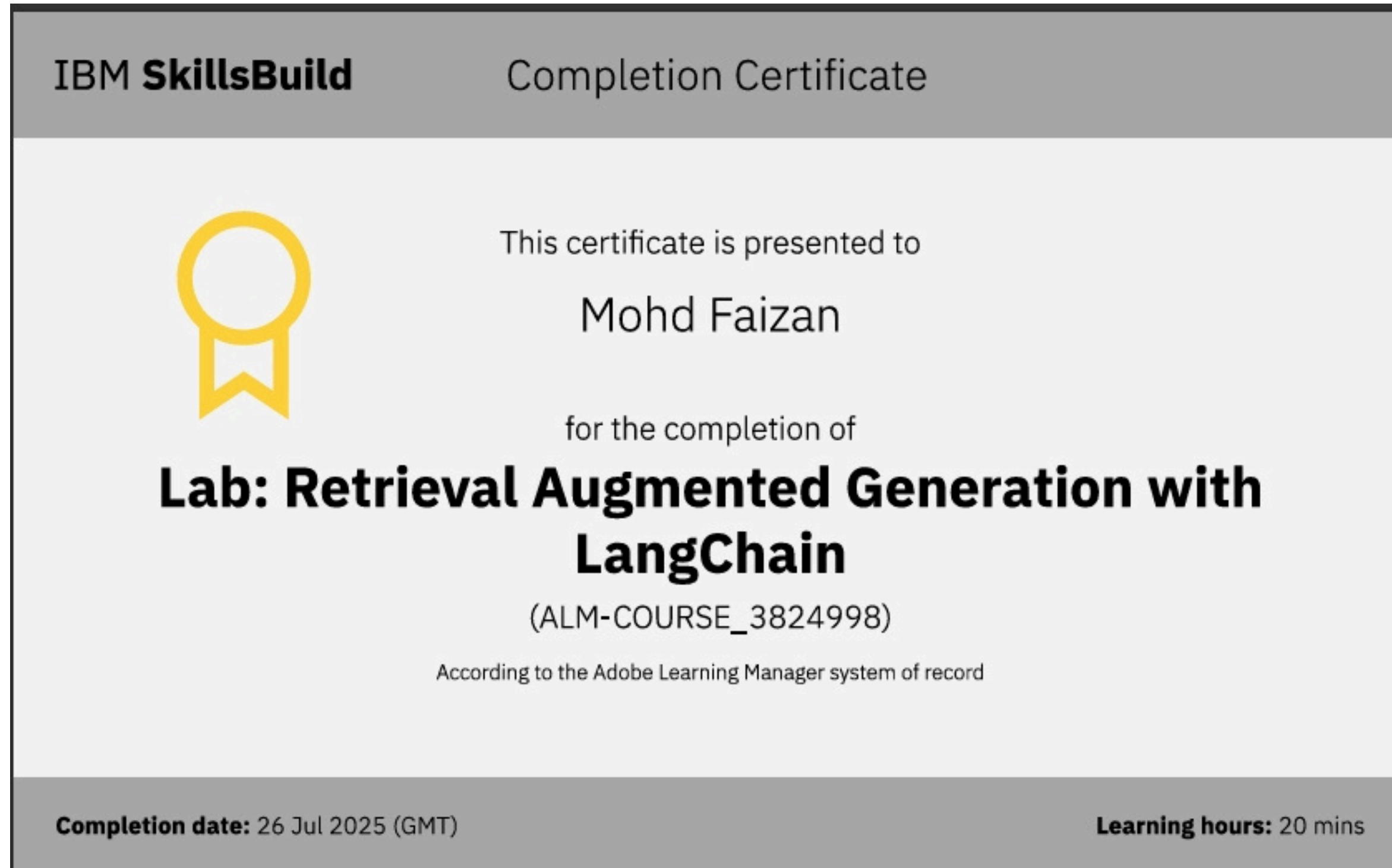


Issued on: Jul 23, 2025
Issued by: IBM SkillsBuild

Verify: <https://www.credly.com/badges/a8774713-06c1-47bb-be12-436b2ac57a9e>



IBM Certifications



THANK YOU