CAPSTONE PROJECT

NETWORK INTRUSION DETECTION USING MACHINE LEARNING

Presented By:

1. Student Name- N.Satya Srinija
2. College Name- Bhoj Reddy Engineering College for Women
3. Department- Computer Science and Engineering

edunet foundation

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

Create a robust network intrusion detection system (NIDS) using machine learning. The system should be capable of analyzing network traffic data to identify and classify various types of cyber-attacks (e.g., DoS, Probe, R2L, U2R) and distinguish them from normal network activity. The goal is to build a model that can effectively secure communication networks by providing an early warning of malicious activities.

PROPOSED SOLUTION

- Develop a machine learning model that classifies network traffic as normal or malicious using the provided dataset. The model will analyze network connection features to rapidly and accurately detect various types of intrusions, enhancing cybersecurity and enabling real-time threat response.
- Data Collection:
  - Use the Kaggle Network Intrusion Detection Dataset.
  - The dataset contains 42 features including protocol_type, service, src_bytes, dst_bytes, etc., and a target column class indicating whether the record is normal or anomaly.
- Preprocessing:
  - Normalize numerical features to ensure uniform scale.
  - Split the dataset into training and testing sets.
- Model Training:
  - Train and compare different classification models, such as (e.g., Decision Tree, Random Forest, or SVM)
- Evaluation:
  - Evaluate model performance using Accuracy, Precision, Recall, and F1-Score.

SYSTEM APPROACH

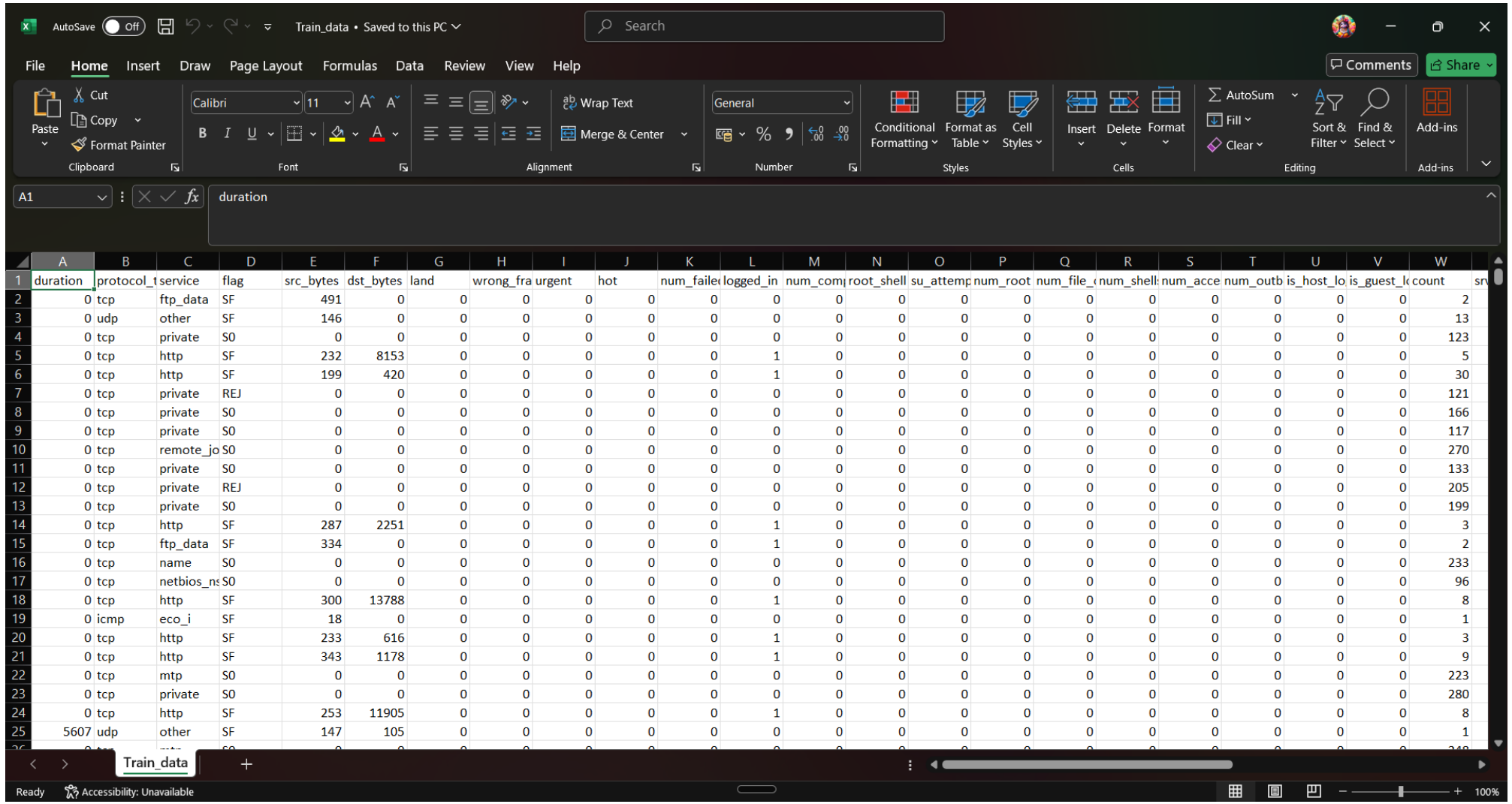
The "System Approach" section outlines the overall strategy and methodology for developing and implementing the Network Intrusion Detection System (NIDS) using machine learning techniques.

- System requirements:
  - IBM Cloud(mandatory)
  - IBM Watson studio for model development and deployment
  - IBM cloud object storage for dataset handling

ALGORITHM & DEPLOYMENT

- Algorithm Selection:
  - Chosen Algorithm: Random Forest Classifier
  - Reason for Selection:
    - Handles high-dimensional data efficiently
    - Provides high accuracy in classification problems
- Data Input:
  - Kaggle dataset link: https://www.kaggle.com/datasets/sampadab17/networkintrusion-detection
  - Features used are duration, protocol_type, service, src_bytes, num_failed_logins, logged_in, count, srv_count, dst_host_srv_count, etc.
- Training Process:
  - Supervised Learning approach using labeled class types (normal vs. attack types)
- Prediction Process:
  - Model is deployed on IBM Watson Studio and an API endpoint is generated for real-time predictions.
  - Model predicts:
    - normal – Safe connection
    - anomaly – Possible attack

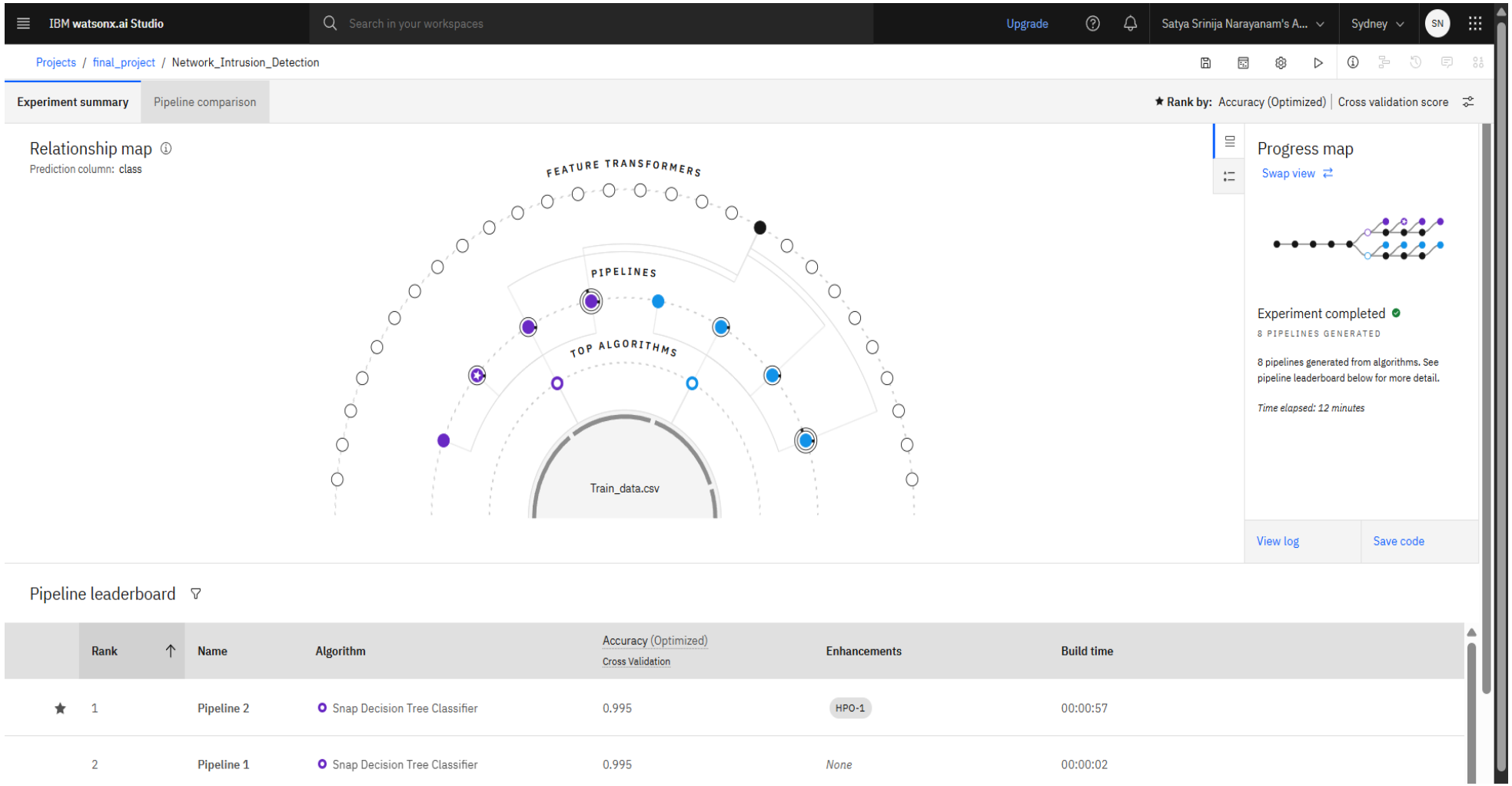
RESULT



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	duration	protocol_t	service	flag	src_bytes	dst_bytes	land	wrong_fra	urgent	hot	num_faile	logged_in	num_com	root_shell	su_attemp	num_root	num_file_	num_shell	num_acce	num_outb	is_host_lo	is_guest_l	count
2	0	tcp	ftp_data	SF	491	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
3	0	udp	other	SF	146	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13
4	0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	123
5	0	tcp	http	SF	232	8153	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	5
6	0	tcp	http	SF	199	420	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	30
7	0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	121
8	0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	166
9	0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	117
10	0	tcp	remote_jo	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	270
11	0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	133
12	0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	205
13	0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	199
14	0	tcp	http	SF	287	2251	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	3
15	0	tcp	ftp_data	SF	334	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	2
16	0	tcp	name	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	233
17	0	tcp	netbios_ns	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96
18	0	tcp	http	SF	300	13788	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	8
19	0	icmp	eco_i	SF	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
20	0	tcp	http	SF	233	616	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	3
21	0	tcp	http	SF	343	1178	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	9
22	0	tcp	mtp	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	223
23	0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	280
24	0	tcp	http	SF	253	11905	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	8
25	5607	udp	other	SF	147	105	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

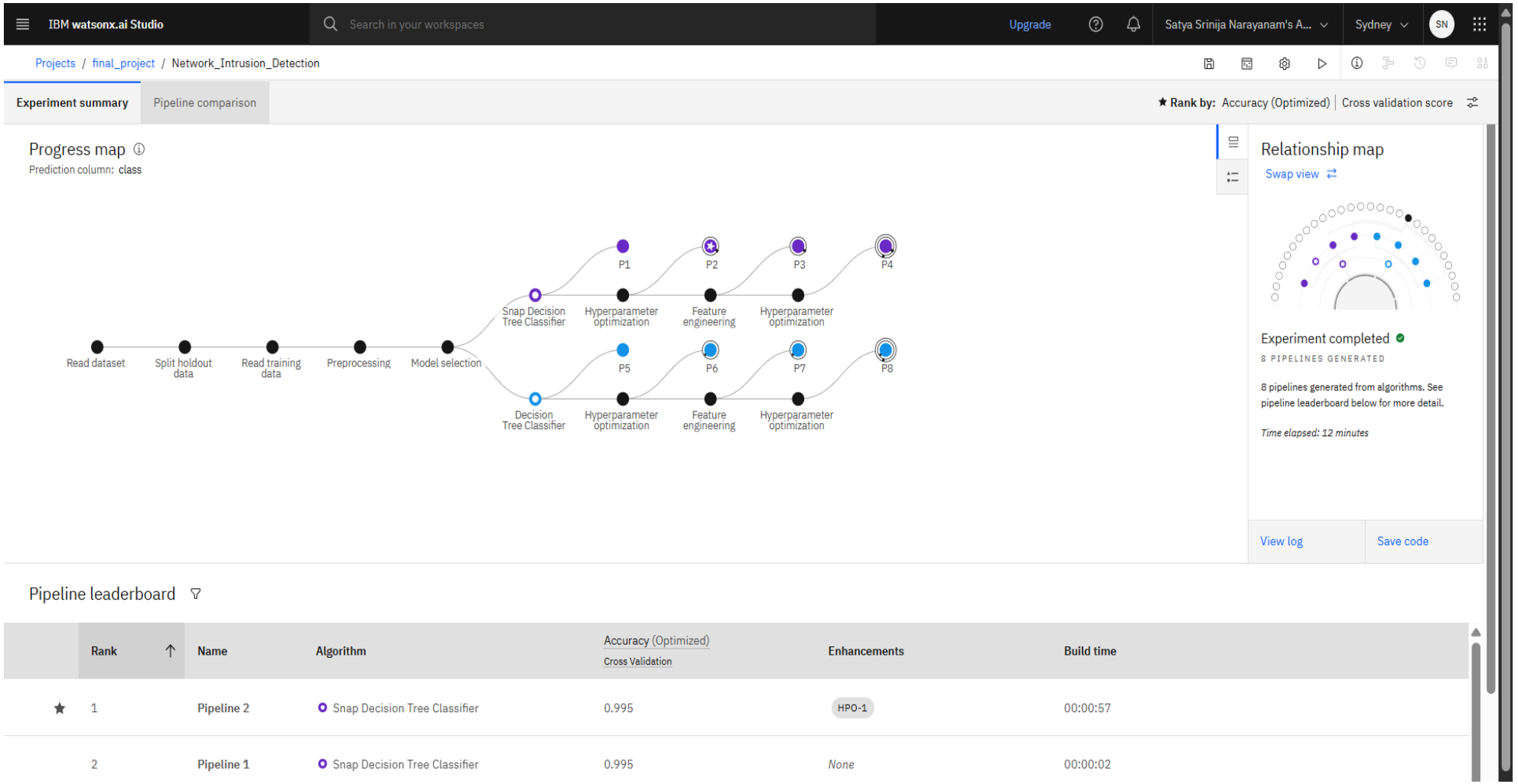
edunet foundation

RESULT



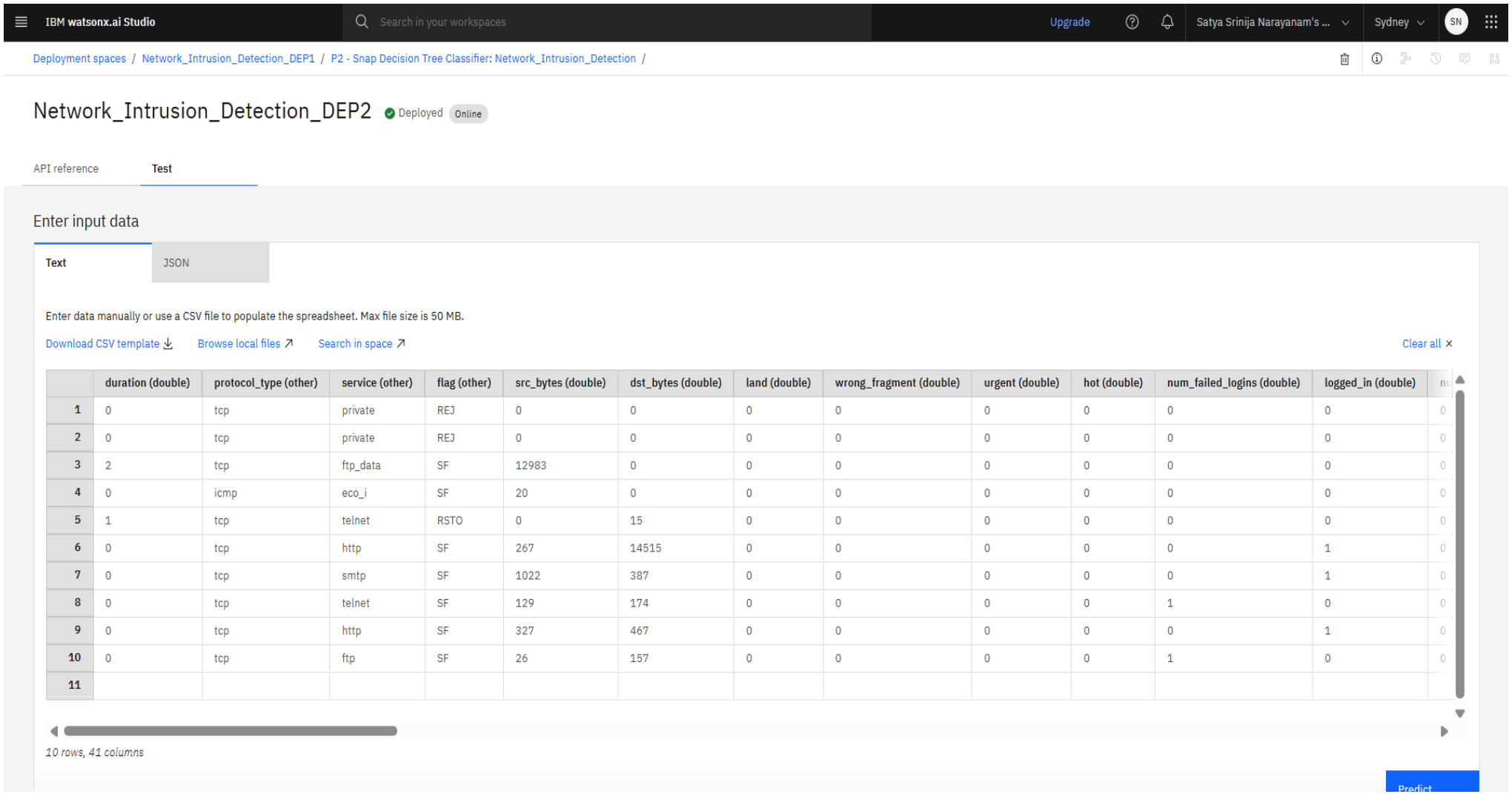
edunet foundation

RESULT



edunet foundation

RESULT



	duration (double)	protocol_type (other)	service (other)	flag (other)	src_bytes (double)	dst_bytes (double)	land (double)	wrong_fragment (double)	urgent (double)	hot (double)	num_failed_logins (double)	logged_in (double)	n
1	0	tcp	private	REJ	0	0	0	0	0	0	0	0	0
2	0	tcp	private	REJ	0	0	0	0	0	0	0	0	0
3	2	tcp	ftp_data	SF	12983	0	0	0	0	0	0	0	0
4	0	icmp	eco_i	SF	20	0	0	0	0	0	0	0	0
5	1	tcp	telnet	RSTO	0	15	0	0	0	0	0	0	0
6	0	tcp	http	SF	267	14515	0	0	0	0	0	1	0
7	0	tcp	smtp	SF	1022	387	0	0	0	0	0	1	0
8	0	tcp	telnet	SF	129	174	0	0	0	0	1	0	0
9	0	tcp	http	SF	327	467	0	0	0	0	0	1	0
10	0	tcp	ftp	SF	26	157	0	0	0	0	1	0	0
11													

edunet foundation

RESULT

Prediction results

Close

Display format for prediction results

Table view JSON view

Show input data

	prediction	probability
1	anomaly	[1,0]
2	anomaly	[1,0]
3	normal	[0,1]
4	anomaly	[1,0]
5	normal	[0,1]
6	normal	[0,1]
7	normal	[0,1]
8	normal	[0,1]
9	normal	[0,1]
10	anomaly	[1,0]
11		
12		
13		
14		
15		
16		

Download JSON file

CONCLUSION

The proposed machine learning model effectively detects and classifies network intrusions with high accuracy. By leveraging supervised learning techniques, the system identifies known attack patterns and differentiates them from normal traffic. Deployment on IBM Watson Studio allows for real-time detection via an API endpoint, enabling fast and automated responses to threats. Accurate intrusion detection is critical for protecting modern digital infrastructure, making such systems essential for cybersecurity in enterprise networks.

- Challenges faced:
  - Imbalanced dataset (more normal traffic than attacks)
  - Feature selection and preprocessing from raw network data
  - Ensuring fast predictions for live environments

FUTURE SCOPE

- Use deep learning models to detect complex or zero-day attacks.
- Add unsupervised methods for anomaly detection.
- Integrate with real-time dashboards for alerts and monitoring

REFERENCES

- Kaggle dataset link:

  https://www.kaggle.com/datasets/sampadab17/networkintrusion-detection

- IBM Watson Studio:

  Watson Machine Learning for model deployment and real-time scoring.
  https://www.ibm.com/cloud/watson-studio

IBM CERTIFICATIONS

edunet foundation

IBM CERTIFICATIONS

IBM CERTIFICATIONS

IBM **SkillsBuild** Completion Certificate

This certificate is presented to

Satya Srinija Narayanam

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

edunet foundation

THANK YOU

edunet foundation