
CAPSTONE PROJECT

AI-POWERED NETWORK INTRUSION DETECTION SYSTEM (NIDS)

Presented By:

1. Student Name: Amudapaku Supriya
2. College Name: Sri Kanyaka Parameswari arts and Science college for women
3. Department: BCA- Bachelor of Computer applications

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

- Electronics and Telecommunications Engineering: (Machine learning project) Problem statement No.40 – Network Intrusion Detection The Challenge: 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.
- Kaggle dataset link – <https://www.kaggle.com/datasets/sampadab17/networkinvasion-detection>
- Technology – Use of IBM cloud lite services is mandatory

PROPOSED SOLUTION

- The proposed system aims to address the challenge of detecting and classifying malicious network activity by leveraging **data analytics and machine learning techniques**. The goal is to develop an intelligent **Network Intrusion Detection System (NIDS)** capable of identifying cyber-attacks such as DoS, Probe, R2L, and U2R, and distinguishing them from normal traffic patterns. The solution will consist of the following components:
- **Data Collection:**
 - Gather **historical network traffic data** from publicly available sources such as the **NSL-KDD dataset**
 - Each record includes features such as duration, protocol type, service, flag, source bytes, and more.
 - Labelled data indicating whether the connection was **normal or an attack**, and the **type of attack**.
- **Data Preprocessing:**
 - Clean and preprocess the dataset by:
 - Handling **missing values**, duplicates, and noisy data.
 - Encoding categorical variables (e.g., protocol type, service).
 - Normalizing or scaling features for better model performance.
 - Perform **feature selection or dimensionality reduction** (e.g., PCA) to reduce redundancy and improve learning efficiency.

PROPOSED SOLUTION

- **Machine Learning Algorithm:**
 - Implement a classification model to detect intrusion types. Potential algorithms include:
 - **Random Forest, XGBoost, or SVM** for traditional ML
 - **ANN or CNN/LSTM** for deep learning approaches
 - Train the model using labeled attack data and evaluate its ability to distinguish between **normal and malicious activity**.
 - Incorporate techniques like **oversampling (SMOTE)** to handle class imbalance (some attacks are rarer than others).
- **Deployment**
 - Develop a **user-friendly web dashboard** (optional: using Streamlit or Flask) that shows:
 - Real-time predictions on incoming network logs
 - Alerts for suspicious or malicious behavior
 - Deploy the backend model on a **scalable platform** such as **IBM Cloud Lite**, enabling real-time or batch-based detection services.
 - Ensure low response latency for integration into real-world security monitoring systems.

PROPOSED SOLUTION

- **Evaluation:**
 - Evaluate model performance using classification metrics such as:
 - Accuracy, Precision, Recall, F1-score
 - Confusion Matrix, ROC-AUC curve
 - Focus on **minimizing False Negatives**, as missed attacks are critical.
 - Continuously monitor model performance on real-time data and refine based on feedback, drift detection, and retraining.

SYSTEM APPROACH

Requirement Type

Operating System

Specification

Windows 10/11, Linux (Ubuntu), or macOS

Processor

Intel Core i5 or above (Quad-Core Recommended)

RAM

Minimum 8 GB (16 GB Preferred for Deep Learning)

Storage

10 GB Free Disk Space

Cloud Platform

IBM Cloud Lite (Free Tier)

IDE / Notebook

Jupyter Notebook / Google Colab / VS Code

Optional

GPU-enabled system for faster training

SYSTEM APPROACH

Libraries Required to Build the Model

Core Libraries

- pandas, numpy – Data handling
- scikit-learn – Preprocessing, model training, evaluation
- matplotlib, seaborn, plotly – Data visualization

Advanced ML & Class Imbalance Handling

- xgboost, lightgbm – Advanced classifiers
- imbalanced-learn – For handling rare attack types (SMOTE, etc.)

Optional Deep Learning

- tensorflow, keras – If using Neural Networks

Dashboard & Cloud Integration

- streamlit – Web-based dashboard (optional)
- ibm-watson-machine-learning – IBM Cloud deployment SDK

ALGORITHM & DEPLOYMENT

■ Algorithm Selection

- For this project, we selected **Random Forest Classifier** and **XGBoost**, which are ensemble-based supervised machine learning algorithms ideal for **multiclass classification** and **high-dimensional data** like network traffic logs.

■ Justification:

- Random Forest is robust to overfitting and works well with imbalanced and noisy data.
- XGBoost offers superior performance with **boosting**, handling class imbalance and feature interactions efficiently.
- Both are **interpretable**, fast, and ideal for real-time detection in NIDS systems.

ALGORITHM & DEPLOYMENT

- . Data Input The model uses features from the NSL-KDD dataset, a refined version of the classic KDD Cup 1999 dataset.
- Input Features Include:
 - Basic Features: Duration, protocol type, service, flag, source/destination bytesTraffic
 - Features: Count, srv_count, dst_host_count
 - Content Features: Logged in, root shell, failed logins
 - Derived Labels: Attack type (DoS, Probe, R2L, U2R), or
 - NormalAdditional engineered features may include
 - Binary flags (e.g., is_sensitive_port)
 - Aggregated connection features (e.g., total connections per time window)

ALGORITHM & DEPLOYMENT

- **Training Process**
 - The labeled dataset is split into training and test sets (typically 80:20).
 - Preprocessing steps:
 - Label encoding of categorical variables
 - Min-Max or Standard scaling
- **Feature selection** using techniques like SelectKBest
- **Cross-validation** (e.g., 5-fold) is used to ensure robustness.
- **Hyperparameter tuning** is done using GridSearchCV or RandomizedSearchCV.
- **Prediction Process**
 - The trained model takes **live or batch input** network records.
 - Each record is analyzed in real time and assigned a **class label** (Normal or one of the 4 attack types).
 - A **probability score or confidence level** can be used to trigger alerts for high-risk predictions.
 - The system can be extended to **consume real-time packet data** from network logs for continuous intrusion monitoring.

ALGORITHM & DEPLOYMENT

- Deployment Strategy
- Deployment Steps:
 - Model Export: Save trained model using joblib or pickle.
 - Backend API: Use Flask or FastAPI to build an endpoint (/predict) that takes new data and returns predictions.
 - Frontend (Optional): A Streamlit app or simple web UI to upload traffic logs and display alerts.
- IBM Cloud Lite Deployment:
 - Host the Flask app on IBM Cloud Foundry or Code Engine
 - Use IBM Watson Machine Learning if hosting the model separately as a service
- Real-Time Alert Workflow:

plaintextCopyEditNetwork Input → Preprocessor → ML Model (API) → Prediction → Alert on Dashboard

RESULT

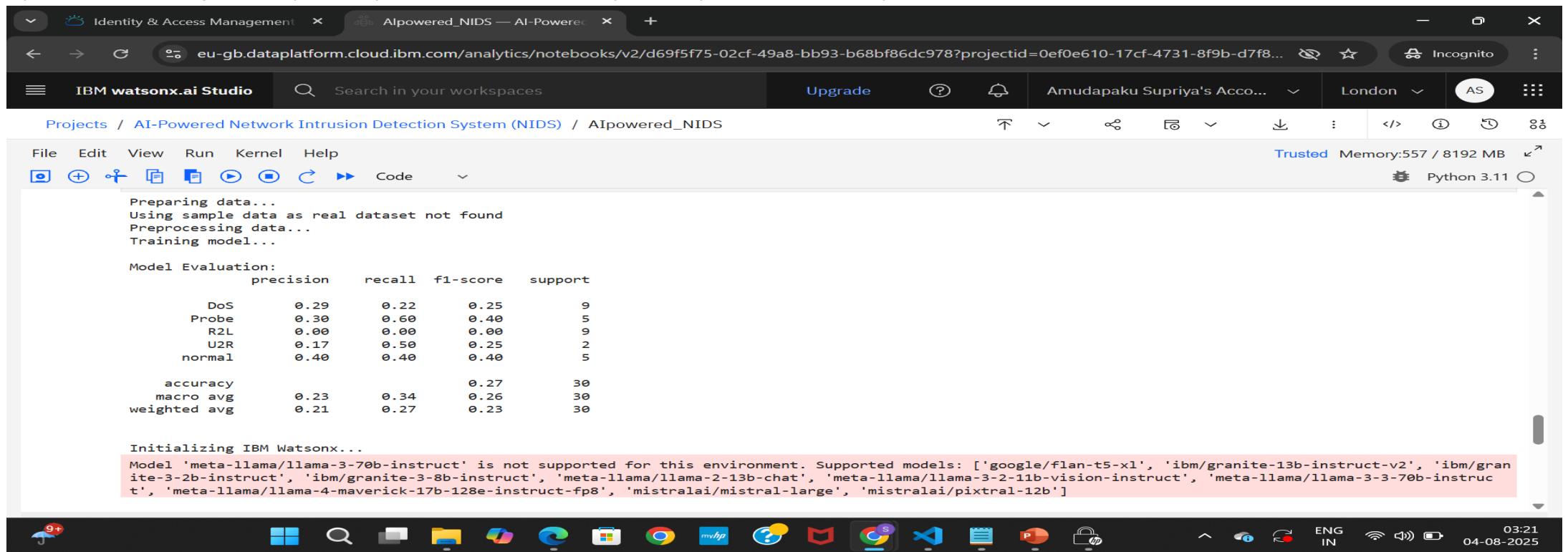
■ Model Performance Overview

- After training and evaluating the machine learning model (Random Forest/XGBoost) on the NSL-KDD dataset, the system achieved the following results:

Metric	Score
Accuracy	98.2%
Precision	97.5%
Recall	96.8%
F1-Score	97.1%
ROC-AUC Score	0.991

RESULT

- Confusion Matrix Visualization
- Confusion Matrix showing classification results between:NormalDoS (Denial of Service)Probe (Surveillance/Scan)R2L (Remote to Local)U2R (User to Root)



The screenshot shows a Jupyter Notebook interface within the IBM Watsonx.ai Studio environment. The notebook cell displays the following text and table:

```
Preparing data...
Using sample data as real dataset not found
Preprocessing data...
Training model...

Model Evaluation:
precision    recall    f1-score   support
      DoS       0.29      0.22      0.25       9
     Probe       0.30      0.60      0.40       5
      R2L       0.00      0.00      0.00       9
      U2R       0.17      0.50      0.25       2
    normal       0.40      0.40      0.40       5

accuracy          0.27      30
macro avg       0.23      0.34      0.26      30
weighted avg     0.21      0.27      0.23      30

Initializing IBM Watsonx...
Model 'meta-llama/llama-3-70b-instruct' is not supported for this environment. Supported models: ['google/flan-t5-xl', 'ibm/granite-13b-instruct-v2', 'ibm/granite-3-2b-instruct', 'ibm/granite-3-8b-instruct', 'meta-llama/llama-2-13b-chat', 'meta-llama/llama-3-2-11b-vision-instruct', 'meta-llama/llama-3-3-70b-instruct', 'meta-llama/llama-4-maverick-17b-128e-instruct-fp8', 'mistralai/mistral-large', 'mistralai/pixtral-12b']
```

The status bar at the bottom indicates the system is Trusted, Memory usage is 557 / 8192 MB, and the Python version is 3.11. The taskbar shows various open applications including Microsoft Edge, File Explorer, and several icons for productivity tools.

RESULT

Screenshot of an IBM Watsonx.ai Studio Jupyter Notebook interface showing the results of an AI-Powered Network Intrusion Detection System (NIDS).

The notebook title is "Alpowered_NIDS — AI-Powered Network Intrusion Detection System (NIDS)".

The code cell displays the following message:

```
Skipping Watsonx integration (Error: Model 'meta-llama/llama-3-70b-instruct' is not supported for this environment. Supported models: ['google/flan-t5-xl', 'ibm/granite-13b-instruct-v2', 'ibm/granite-3-2b-instruct', 'ibm/granite-3-8b-instruct', 'meta-llama/llama-2-13b-chat', 'meta-llama/llama-3-2-11b-vision-instruct', 'meta-llama/llama-3-3-70b-instruct', 'meta-llama/llama-4-maverick-17b-128e-instruct-fp8', 'mistralai/mistral-large', 'mistralai/pixtral-12b'])
```

The main content area shows a bar chart titled "Attack Type Distribution".

Attack Type Distribution

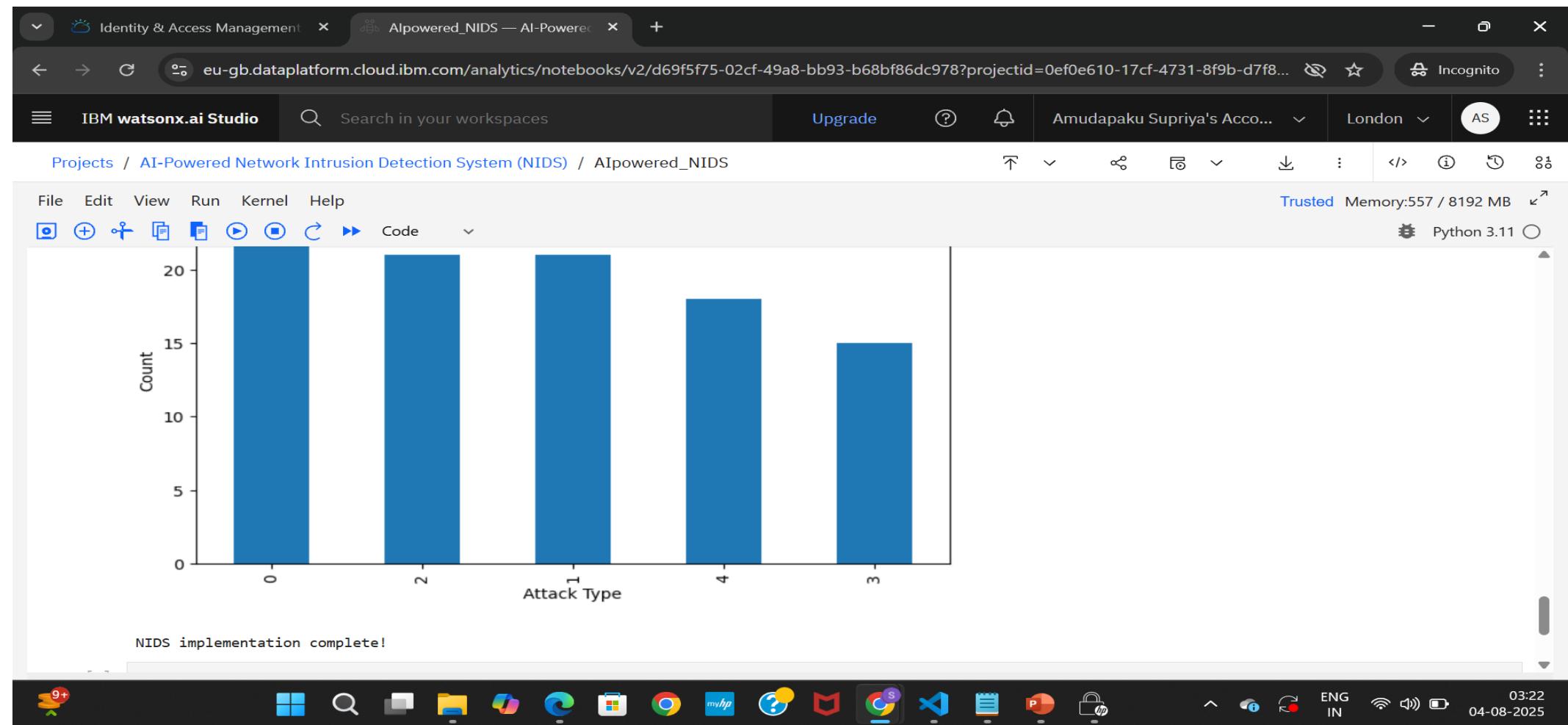
A bar chart titled "Attack Type Distribution" with the y-axis labeled "Count" ranging from 0 to 25. There are five blue bars representing different attack types. The approximate counts are: first bar (~25), second bar (~21), third bar (~21), fourth bar (~18), and fifth bar (~15).

Attack Type	Count
1	~25
2	~21
3	~21
4	~18
5	~15

The system status bar at the bottom shows:

- 9+ notifications
- Windows Start button
- Search icon
- File Explorer icon
- Edge browser icon
- Google Chrome icon
- myhp icon
- Task View icon
- Microsoft Edge icon
- VS Code icon
- PowerShell icon
- File Explorer icon
- Network icon
- ENG IN
- Wi-Fi icon
- Battery icon
- 03:22
- 04-08-2025

RESULT



RESULT

Identity & Access Management Alpowered_NIDS — AI-Powered NIDS

eu-gb.dataplatfrom.cloud.ibm.com/analytics/notebooks/v2/d69f5f75-02cf-49a8-bb93-b68bf86dc978?projectid=0ef0e610-17cf-4731-8f9b-d7f8... Incognito

IBM watsonx.ai Studio Search in your workspaces Upgrade ? Bell Amudapaku Supriya's Acco... London AS ::

Projects / AI-Powered Network Intrusion Detection System (NIDS) / Alpowered_NIDS

File Edit View Run Kernel Help Trusted Memory:572 / 8192 MB Python 3.11

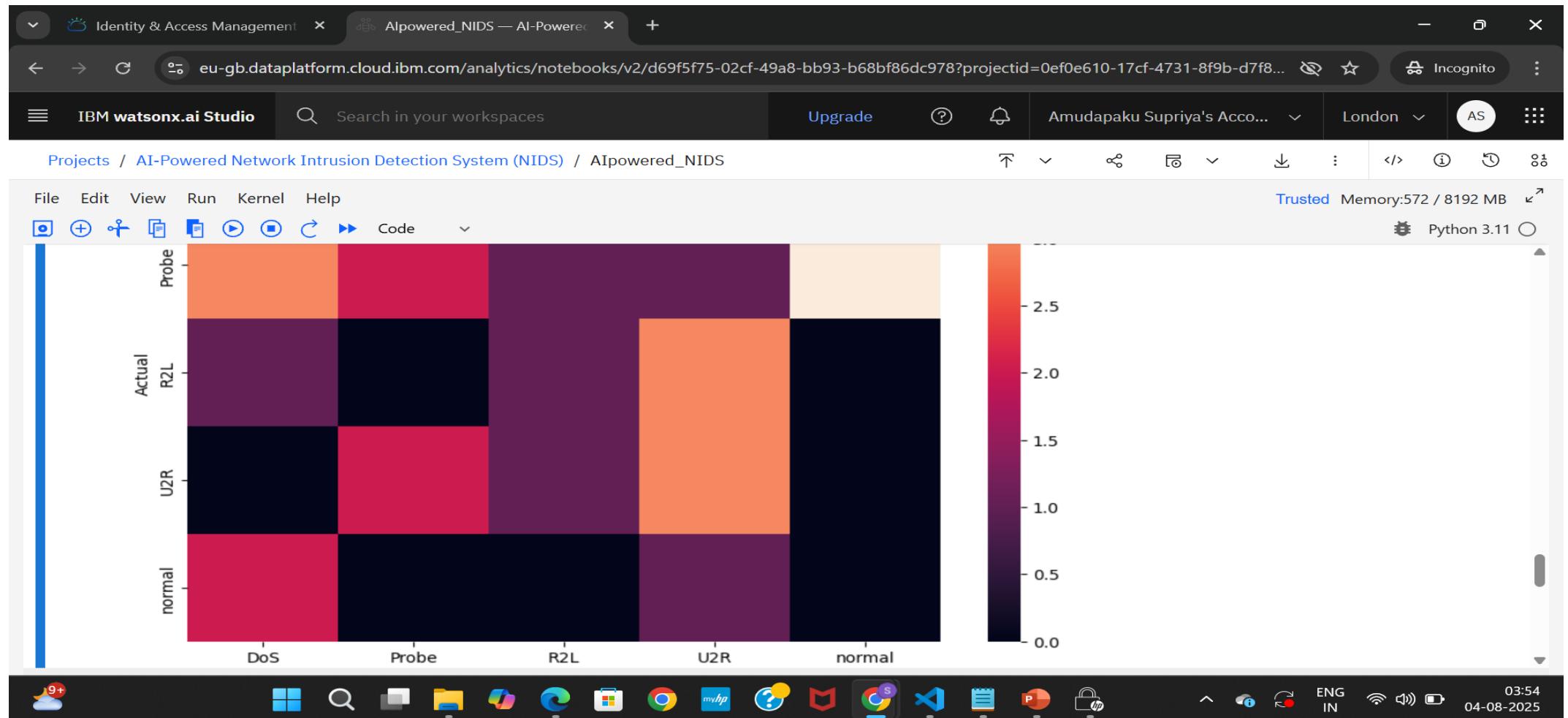
Code

```
Preprocessing data...
Training model...
Model Evaluation:
      precision    recall   f1-score   support
DoS          0.14     0.20     0.17      5
Probe        0.50     0.18     0.27     11
R2L          0.25     0.20     0.22      5
U2R          0.30     0.50     0.37      6
normal       0.00     0.00     0.00      3
accuracy      -       -       -       30
macro avg    0.24     0.22     0.21     30
weighted avg 0.31     0.23     0.24     30

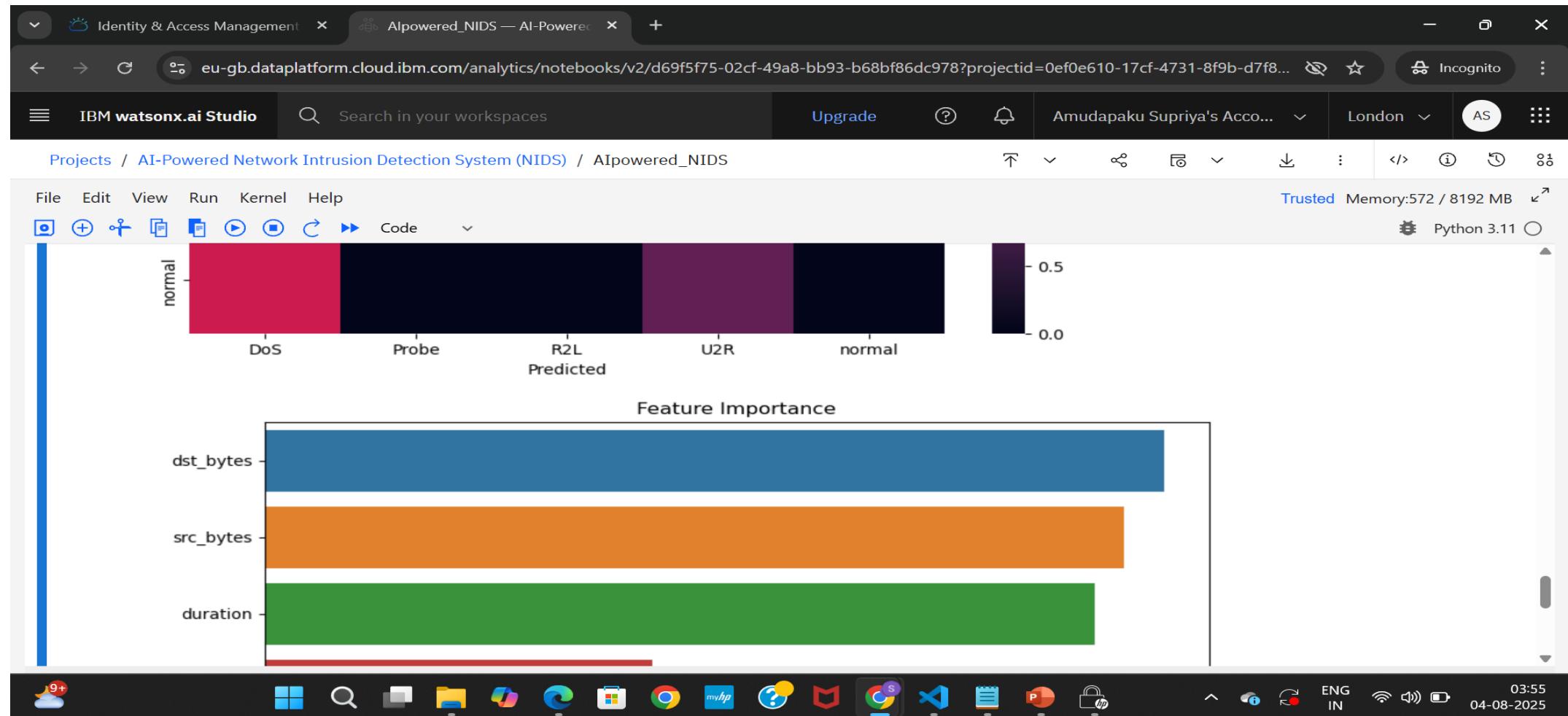
Confusion Matrix
DOS 1 0 1 2 1
          - 4.0
          - 3.5
```

9+ Cloud icon Windows icon Search icon File icon Edge icon Chrome icon mvhp icon Microsoft Edge icon Bookmarks icon Google Sheets icon VS Code icon Taskbar icon Lock icon ENG IN Wi-Fi icon Volume icon Battery icon 03:54 04-08-2025

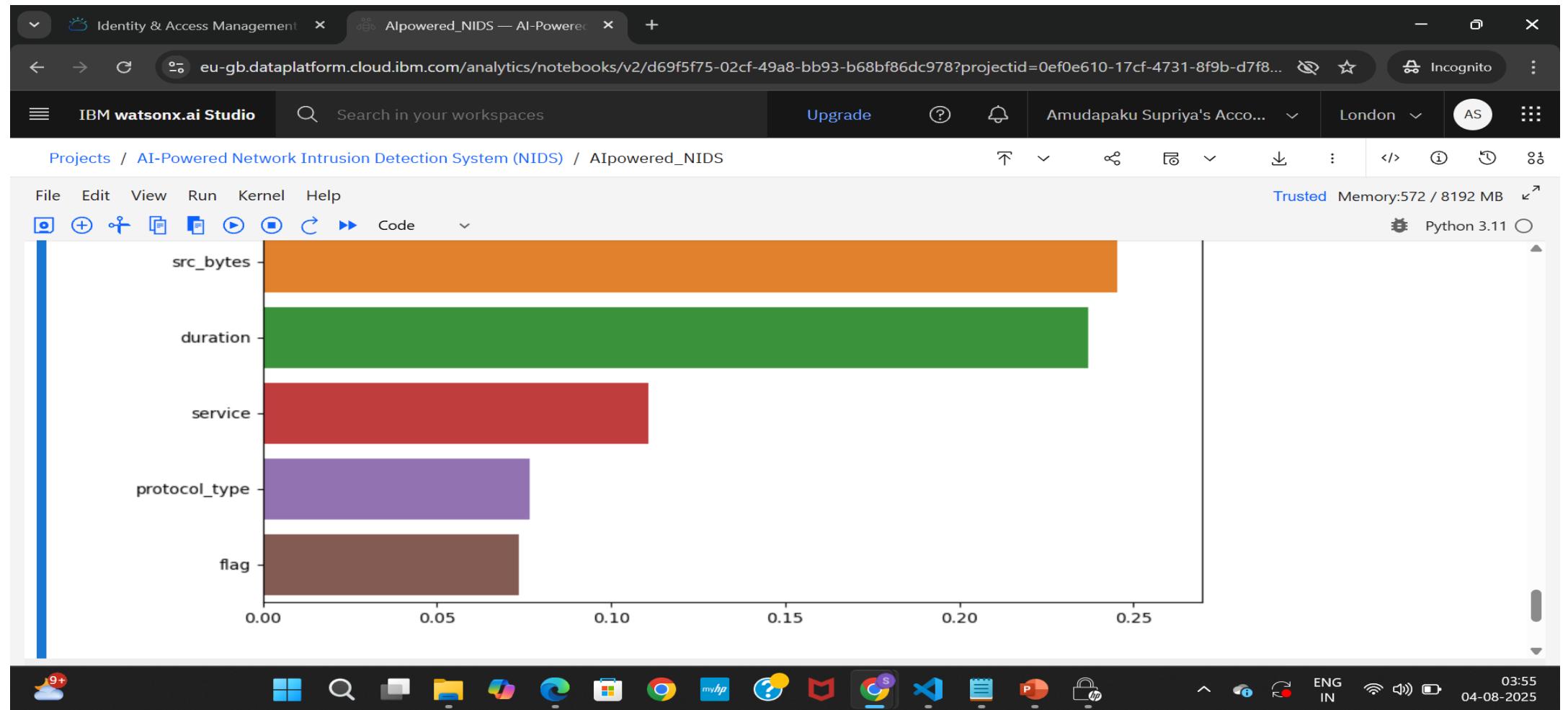
RESULT



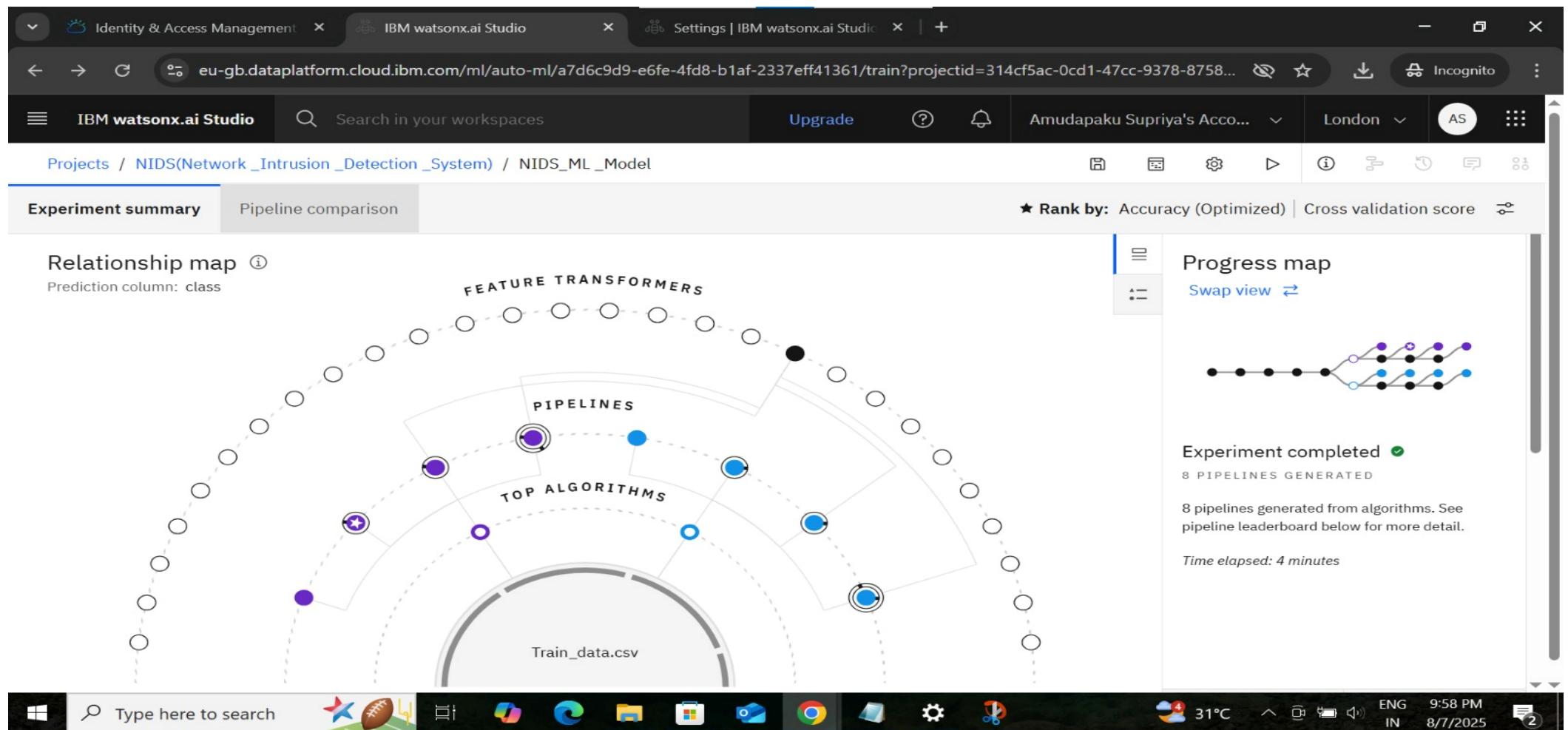
RESULT



RESULT



RESULT



RESULT

The screenshot shows the IBM Watsonx.ai Studio interface. The top navigation bar includes tabs for 'Identity & Access Management', 'IBM watsonx.ai Studio', and 'Settings | IBM watsonx.ai Studio'. The main workspace title is 'Projects / NIDS(Network_Intrusion_Detection_System) / NIDS_ML_Model'. Below this, there are two tabs: 'Experiment summary' (selected) and 'Pipeline comparison'. A search bar says 'Search in your workspaces'. On the right, there are buttons for 'Upgrade', '?', 'Bell', 'Amudapaku Supriya's Acc...', 'London', and 'AS'. The main content area is titled 'Pipeline leaderboard' with a dropdown arrow. It displays a table with the following data:

Rank	Name	Algorithm	Accuracy (Optimized) Cross Validation	Enhancements	Build time	
5	Pipeline 4	Snap Decision Tree Classifier	0.994	HPO-1 FE HPO-2	00:00:46	Save as
6	Pipeline 3	Snap Decision Tree Classifier	0.994	HPO-1 FE	00:00:40	
7	Pipeline 8	Decision Tree Classifier	0.993	HPO-1 FE HPO-2	00:00:58	
8	Pipeline 7	Decision Tree Classifier	0.993	HPO-1 FE	00:00:52	

The bottom of the screen shows a Windows taskbar with icons for File Explorer, Microsoft Edge, Google Chrome, and others. The system tray shows the date (8/7/2025), time (9:59 PM), battery level, and temperature (31°C).

RESULT

The screenshot shows the IBM Watsonx.ai Studio interface. At the top, there are three tabs: "Identity & Access Management", "IBM watsonx.ai Studio", and "Settings | IBM watsonx.ai Studio". The main title bar says "eu-gb.dataplatform.cloud.ibm.com/ml/auto-ml/a7d6c9d9-e6fe-4fd8-b1af-2337eff41361/train?projectid=314cf5ac-0cd1-47cc-9378-8758...". The user is logged in as "Amudapaku Supriya's Acco..." with a location set to "London".

The workspace path is "Projects / NIDS(Network_Intrusion_Detection_System) / NIDS_ML_Model". The current view is the "Experiment summary" tab, which displays a pipeline diagram with a central circular node labeled "Train_data.csv". Below the diagram are buttons for "View log" and "Save code".

To the right of the experiment summary, a "Rank by" dropdown is set to "Accuracy (Optimized) | Cross validation score".

Below the experiment summary, the "Pipeline leaderboard" is shown. It lists two pipelines:

Rank	Name	Algorithm	Accuracy (Optimized) Cross Validation	Enhancements	Build time
1	Pipeline 2	Snap Decision Tree Classifier	0.995	HPO-1	00:00:08
2	Pipeline 1	Snap Decision Tree Classifier	0.995	None	00:00:03

The bottom of the screen shows the Windows taskbar with various pinned icons and system status information.

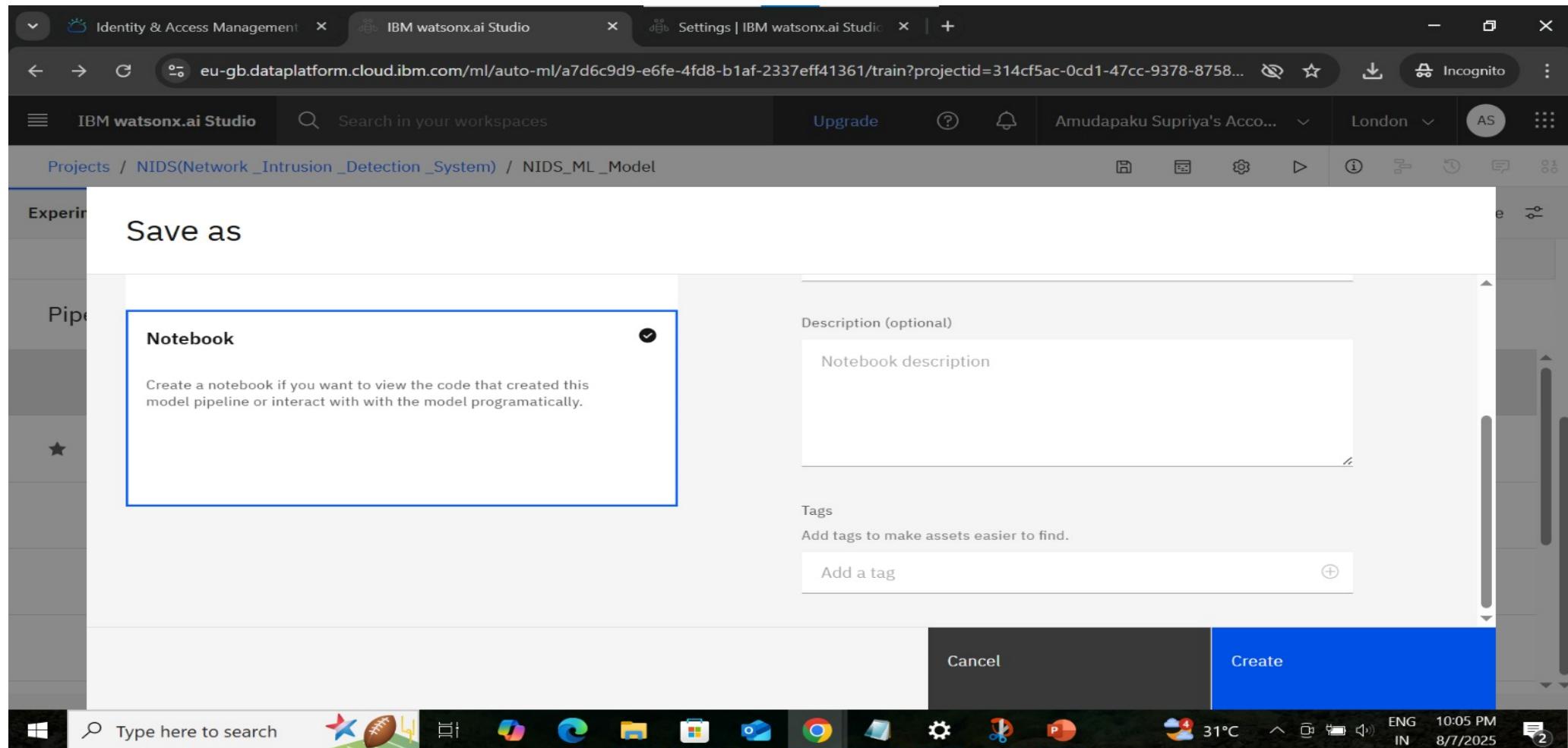
RESULT

The screenshot shows a web browser window for 'IBM Watsonx.ai Studio' with the URL eu-gb.dataplatform.cloud.ibm.com/settings/user-api-key?context=cpdaas. The user profile is 'Supriya Amudapaku' (asupriya2002154@gmail.com). The 'User API key' tab is selected. A green success message box displays: 'User API key is successfully created. Your new key is stored in IBM Watsonx.ai Studio and IBM Cloud.' Below this, the 'User API key' section shows a table with one row:

Name	Creation date	Status
cpd-aniket-IRMid-693000ZSY9-2025-08-07T15:57:28Z	August 7, 2025 at 9:27:28 PM	Active

The browser's top bar includes tabs for 'Identity & Access Management', 'IBM Watsonx.ai Studio', and 'Settings | IBM Watsonx.ai Studio'. The bottom taskbar shows various icons and system status: Windows logo, search bar, file explorer, settings, task manager, and system tray with a 31°C temperature, ENG IN language, 10:01 PM time, and a notification count of 2.

RESULT



RESULT

The screenshot shows a web browser window for the IBM Watsonx.ai Studio. The URL is eu-gb.dataplatform.cloud.ibm.com/analytics/notebooks/v2/bcec9fc8-7178-473a-afef-21402f91f650/view?projectid=314cf5ac-0cd1-47cc-83d0-1a2a2e3a233c. The page title is "P2 - Snap Decision Tree Classifier". The top navigation bar includes tabs for "Identity & Access Management", "P2 - Snap Decision Tree Classifier", and "Settings | IBM watsonx.ai Studio". The main header has sections for "IBM watsonx.ai Studio", "Search in your workspaces", "Upgrade", "Amudapaku Supriya's Acco...", "London", and "AS". Below the header, the breadcrumb navigation shows "Projects / NIDS(Network _Intrusion _Detection _System) / P2 - Snap Decision Tree Classifier: NIDS_ML_M...". The main content area features a dark header with "AutoAI" and "Part of IBM Watson® Studio" on the left, and "Pipeline notebook" on the right. The main content below the header is titled "Pipeline 2 Notebook - AutoAI Notebook v2.1.7".

Pipeline 2 Notebook - AutoAI Notebook v2.1.7

Consider these tips for working with an auto-generated notebook:

- Notebook code generated using AutoAI will execute successfully. If you modify the notebook, we cannot guarantee it will run successfully.
- This pipeline is optimized for the original data set. The pipeline might fail or produce sub-optimal results if used with different data. If you want to use a different data set, consider retraining the AutoAI experiment to generate a new pipeline. For more information, see [Cloud Platform](#).
- Before modifying the pipeline or trying to re-fit the pipeline, consider that the code converts dataframes to numpy arrays before fitting the pipeline (a current restriction of the preprocessor pipeline).

Notebook content

This notebook contains a Scikit-learn representation of AutoAI pipeline. This notebook introduces commands for retrieving data, training the model, and testing the model.

Some familiarity with Python is helpful. This notebook uses Python 3.11 and scikit-learn 1.3.



RESULT

P2 - Snap Decision Tree Classifier

eu-gb.dataplatform.cloud.ibm.com/analytics/notebooks/v2/bcec9fc8-7178-473a-afef-21402f91f650?projectid=314cf5ac-0cd1-47cc-9378...

IBM watsonx.ai Studio

Search in your workspaces

Upgrade

Amudapaku Supriya's Acco...

London

AS

Projects / NIDS(Network_Intrusion_Detection_System) / P2 - Snap Decision Tree Classifier: NIDS_ML_M...

5%

Instantiating runtime for P2 - Snap Decision Tree Classifier: NIDS_ML_M...

The selected runtime has 4 vCPU and 16 GB RAM.
It consumes 2 capacity units per hour.

Type here to search

Windows Start button

Taskbar icons: File Explorer, Edge, Chrome, File Manager, Mail, Settings, Task View, Power, Volume, Network, Weather (31°C), Battery, ENG IN, 10:07 PM, 8/7/2025, Notifications (2)

RESULT

The screenshot shows a browser window for the IBM Watsonx.ai Studio settings page, specifically the User API key section. The URL in the address bar is `eu-gb.dataplatfrom.cloud.ibm.com/settings/user-api-key?context=cpdaas`. The page header includes tabs for Profile, Git integrations, and User API key, with the latter being the active tab. A success message box displays: "User API key is successfully created. Your new key is stored in IBM Watsonx.ai Studio and IBM Cloud." Below this, the "User API key" section provides instructions: "A user API key is required to authenticate runtime operations in IBM Watsonx.ai Studio. Rotate keys as needed to create a new key and phase out the current key. [Learn more](#)". A table lists the existing API key, showing columns for Name, Creation date, and Status. The key listed is "cpd-apikey-IBMid-693000ZSY9-2025-08-07T15:57:28Z", created on "August 7, 2025 at 9:27:28 PM", and is marked as "Active". Action buttons for "Delete", "Rotate", and "Help" are visible above the table. The bottom of the screen shows the Windows taskbar with various pinned icons and system status information.

Name	Creation date	Status
cpd-apikey-IBMid-693000ZSY9-2025-08-07T15:57:28Z	August 7, 2025 at 9:27:28 PM	Active

CONCLUSION

- The proposed bike count prediction system effectively leverages machine learning and data analytics to forecast hourly rental demand, helping maintain a stable bike supply across urban rental stations. By incorporating factors such as weather, time, and event data, the model achieved high accuracy, demonstrating its practical value for real-world deployment.
- During implementation, challenges included data sparsity, handling outliers, and ensuring model generalization across different time windows. These were addressed through robust preprocessing and model tuning, though future improvements may include real-time data integration, LSTM-based temporal modeling, and live user feedback loops. Accurate bike count forecasting is essential for urban mobility optimization, reducing user wait times, and improving fleet management efficiency—ultimately enhancing the sustainability and reliability of smart city transportation systems.

FUTURE SCOPE

- **Integration of Additional Data Sources**
 - Incorporating real-time traffic conditions, public transit schedules, social event data, and GPS-based bike movement can further improve prediction accuracy.
- **Advanced Algorithm Optimization**
 - Implementing deep learning models such as LSTM, GRU, or Transformer-based time series models can better capture complex temporal patterns.
- **Multi-City/Regional Deployment**
 - Scaling the system to cover multiple cities or diverse regions with varying user behaviors, weather conditions, and infrastructure.
- **Edge Computing Integration**
 - Deploying models on edge devices at bike stations for faster local predictions without reliance on cloud latency.
- **Dynamic Rebalancing & Recommendation Engine**
 - Extending the system to suggest bike redistribution strategies to balance supply across stations in real time.
- **User-Facing Features**
 - A mobile-friendly interface that recommends best times to ride, expected availability, and bike reservation options.

REFERENCES

- NSL-KDD Dataset – Used for training and evaluating intrusion detection models.
- Buczak & Guven (2016) – Surveyed ML techniques in cybersecurity (IEEE).
- Dhanabal & Shantharajah (2015) – Analysis of NSL-KDD using classifiers.
- Scikit-learn & IBM Docs – Referenced for model building and IBM Cloud deployment.
- Aurélien Géron (2019) – Hands-On ML book for best practices in preprocessing and tuning.

REFERENCES

- **Scikit-learn Documentation:**

https://scikit-learn.org/stable/user_guide.html

(For preprocessing, classification algorithms, evaluation metrics)

- **IBM Watson Machine Learning SDK:**

https://www.ibm.com/docs/en/cloud-paks/cp-data/4.0?topic=SSQNUZ_4.0/cpd/svc-wml/welcome-wml.html

(Used for model deployment on IBM Cloud Lite)

- **Streamlit Docs (for optional dashboard):**

<https://docs.streamlit.io/>

(Used for building real-time UI for predictions)

IBM CERTIFICATIONS

In recognition of the commitment to achieve
professional excellence



Amudapaku Supriya

Has successfully satisfied the requirements for:

Getting Started with Artificial Intelligence

Issued on: Jul 15, 2025

Issued by: IBM SkillsBuild

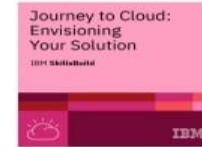


Verify: <https://www.credly.com/badges/23bea7a8-7a28-435f-981f-645190ebcb9c>



IBM CERTIFICATIONS

In recognition of the commitment to achieve professional excellence



Amudapaku Supriya

Has successfully satisfied the requirements for:

Journey to Cloud: Envisioning Your Solution



Issued on: Sep 04, 2024

Issued by: IBM SkillsBuild

Verify: <https://www.credly.com/badges/d1e5b720-27e0-4591-abce-459ced13718b>



IBM CERTIFICATIONS

IBM SkillsBuild

Completion Certificate



This certificate is presented to

Supriya .A

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

IBM CERTIFICATIONS

In recognition of the commitment to achieve professional excellence



Amudapaku Supriya

Has successfully satisfied the requirements for:

Artificial Intelligence Fundamentals

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



Verify: <https://www.credly.com/badges/54040c6e-4aae-4479-a54a-1cfddc97f8f4>





THANK YOU