CAPSTONE PROJECT

NETWORK INTRUSION DETECTION USING MACHINE LEARNING ON IBM CLOUD

Presented By:
Mann Panghal
Guru Jambheshwar University of Science and Technology
Dept : CSE(IT)

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

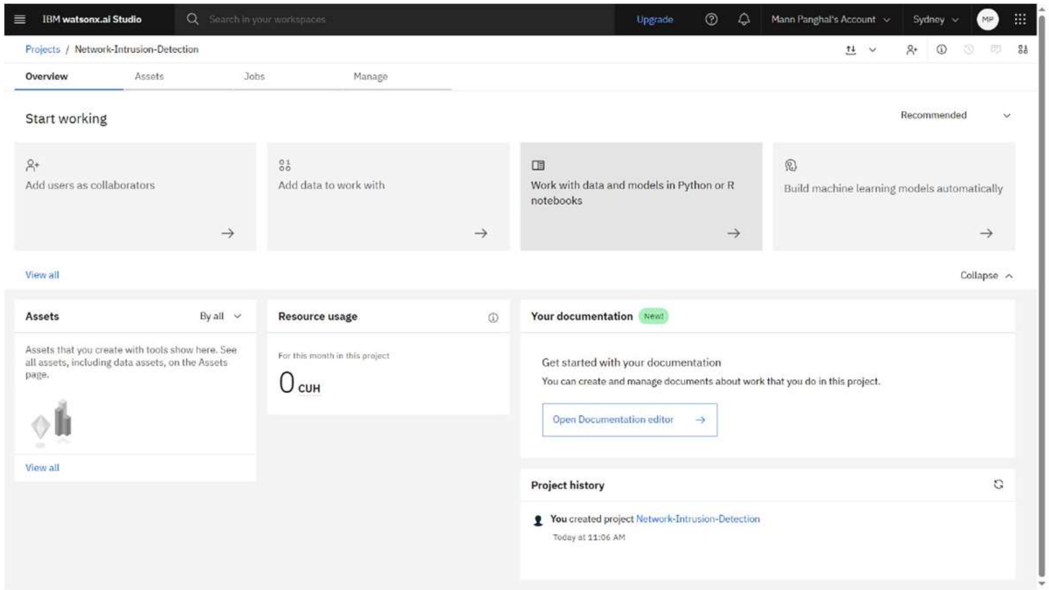
- Cybersecurity threats are constantly evolving, and it is crucial to have systems that can detect malicious network activity to protect sensitive data.
- The challenge is to create a robust Network Intrusion Detection System (NIDS) capable of analyzing network traffic data.
- The system must be able to automatically identify and classify various types of cyber-attacks (e.g., DoS, Probes) and distinguish them from normal network activity.

PROPOSED SOLUTION

- The proposed solution is to build a Machine Learning model on the IBM Cloud platform that can intelligently classify network traffic.
- Data-Driven Approach: We will use a real-world dataset containing thousands of labeled network connections.
- Model Training: The model will be trained on this data to learn the specific patterns and characteristics of both normal traffic and various cyber-attacks.
- Classification: Once trained, the model will be able to take new, unseen network data and predict whether it is "Normal" or a specific type of "Attack".
- Cloud-Based: The entire workflow, from data preparation to model evaluation, will be executed on IBM Cloud Lite services, as required.

SYSTEM APPROACH

- The project was developed entirely on the IBM Cloud platform, following a structured data science methodology.
- **Cloud Platform:** IBM Cloud (Lite Plan)
- **Primary Service:** IBM Watson Studio, which provided an integrated environment for data storage, coding, and computation.
- **Development Tool:** Jupyter Notebook, for writing and executing Python code interactively.
- **Programming Language:** Python 3.11
- **Key Libraries:**
  - pandas: For data loading, cleaning, and manipulation.
  - scikit-learn: For building and evaluating the machine learning model.



Assets that you create with tools show here. See all assets, including data assets, on the Assets page.

For this month in this project

0 CUH

Get started with your documentation

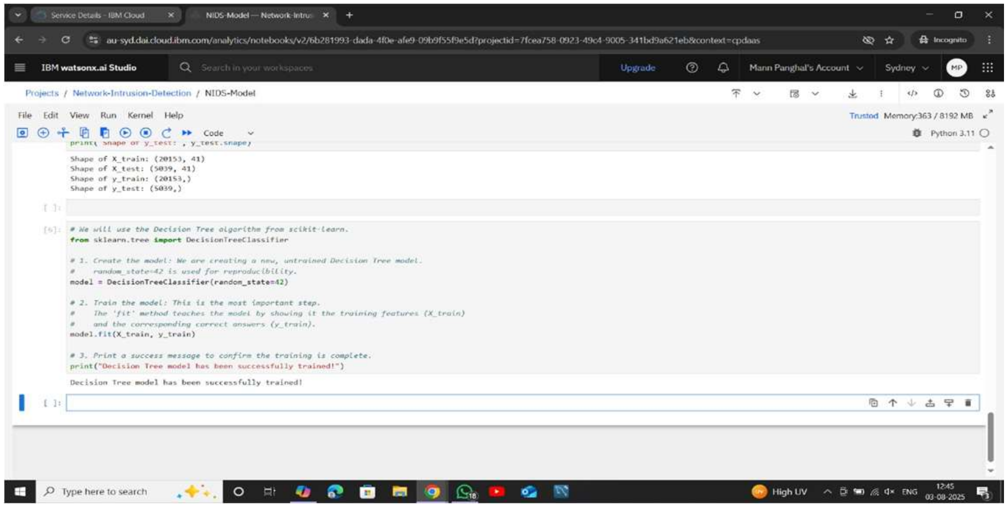
You can create and manage documents about work that you do in this project.

You created project Network-Intrusion-Detection

Today at 11:06 AM

ALGORITHM & DEPLOYMENT

- Algorithm Selection: We chose the Decision Tree Classifier algorithm. It's a powerful and interpretable model that makes predictions by learning a series of simple "if-then-else" rules from the data.
- Data Preparation:
  - The raw text data (e.g., protocol types, service names) was converted into a numerical format using Label Encoding.
  - The dataset was split into two parts: 80% for training the model and 20% for testing its performance.
- Model Training & Deployment:
  - The Decision Tree model was trained using the fit() method on the training data within our IBM Watson Studio notebook.
  - This trained model is now ready to make predictions on new data.


Service Details - IBM Cloud
NIDS-Model — Network-Intrus
au-syd.dai.cloud.ibm.com/analytics/notebooks/v2/6b281993-dada-4f0e-afe9-09b9f55f9e5d?projectid=7fcea758-0923-49c4-9005-341bd9a621eb&context=cpdaas
Incognito
IBM watsonx.ai Studio
Search in your workspaces
Upgrade
Mann Panghal's Account
Sydney
MP
Projects / Network-Intrusion-Detection / NIDS-Model
File Edit View Run Kernel Help
Code
Trusted Memory:363 / 8192 MB
Python 3.11
Type here to search
High UV
ENG
12:45
03-08-2025

```
print("Shape of y_test:", y_test.shape)
```

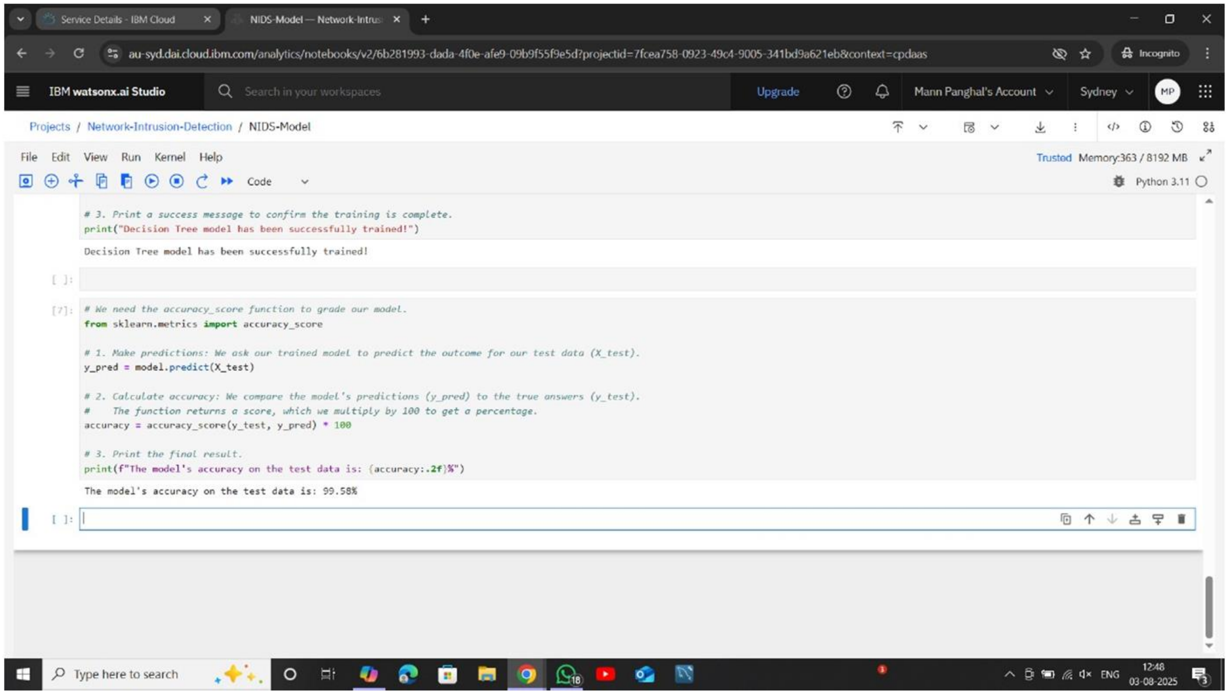
```
Shape of X_train: (20153, 41)
Shape of X_test: (5039, 41)
Shape of y_train: (20153,)
Shape of y_test: (5039,)
```

```
# We will use the Decision Tree algorithm from scikit-learn.
from sklearn.tree import DecisionTreeClassifier

# 1. Create the model: We are creating a new, untrained Decision Tree model.
#    random_state=42 is used for reproducibility.
model = DecisionTreeClassifier(random_state=42)

# 2. Train the model: This is the most important step.
#    The 'fit' method teaches the model by showing it the training features (X_train)
#    and the corresponding correct answers (y_train).
model.fit(X_train, y_train)

# 3. Print a success message to confirm the training is complete.
print("Decision Tree model has been successfully trained!")
```

```
Decision Tree model has been successfully trained!
```

RESULT

- The performance of the trained model was evaluated on the unseen test data. The model demonstrated a very high level of accuracy in correctly classifying network traffic as either normal or an attack.
- The model achieved a final accuracy of 99.58%.
- This result proves that our machine learning approach is highly effective for building a Network Intrusion Detection System.


Service Details - IBM Cloud
NIDS-Model — Network-Intrus
au-syd.dai.cloud.ibm.com/analytics/notebooks/v2/6b281993-dada-4f0e-afe9-09b9f55f9e5d?projectid=7fcea758-0923-49c4-9005-341bd9a621eb&context=cpdaas
Incognito
IBM watsonx.ai Studio
Search in your workspaces
Upgrade
Mann Panghal's Account
Sydney
MP
Projects / Network-Intrusion-Detection / NIDS-Model
File Edit View Run Kernel Help
Code
Trusted Memory:363 / 8192 MB
Python 3.11
Type here to search
ENG
12:48
03-08-2025

```
# 3. Print a success message to confirm the training is complete.
print("Decision Tree model has been successfully trained!")
```

```
Decision Tree model has been successfully trained!
```

```
[ ]:
```

```
[7]: # We need the accuracy_score function to grade our model.
from sklearn.metrics import accuracy_score

# 1. Make predictions: We ask our trained model to predict the outcome for our test data (X_test).
y_pred = model.predict(X_test)

# 2. Calculate accuracy: We compare the model's predictions (y_pred) to the true answers (y_test).
#    The function returns a score, which we multiply by 100 to get a percentage.
accuracy = accuracy_score(y_test, y_pred) * 100

# 3. Print the final result.
print(f"The model's accuracy on the test data is: {accuracy:.2f}%")
```

```
The model's accuracy on the test data is: 99.58%
```

```
[ ]:
```

CONCLUSION

- This project successfully demonstrated the creation of a robust Network Intrusion Detection System using machine learning on the IBM Cloud platform.
- The Decision Tree model proved to be highly effective, achieving an accuracy of over 99.5%, which is excellent for this type of security application.
- The project confirms that a data-driven approach is a powerful and practical way to identify and classify modern cybersecurity threats.

FUTURE SCOPE

- While the current model is very accurate, there are several ways this project could be expanded in the future:
- Try Advanced Algorithms: Implement more complex models like Random Forest or Neural Networks to potentially improve accuracy even further.
- Real-Time Deployment: Deploy the trained model as a live API service on IBM Cloud. This would allow it to analyze network traffic in real-time.
- Larger Datasets: Use a larger, more modern dataset to train the model on the very latest attack patterns.

REFERENCES

- Dataset: Network Intrusion Detection Dataset, Kaggle.

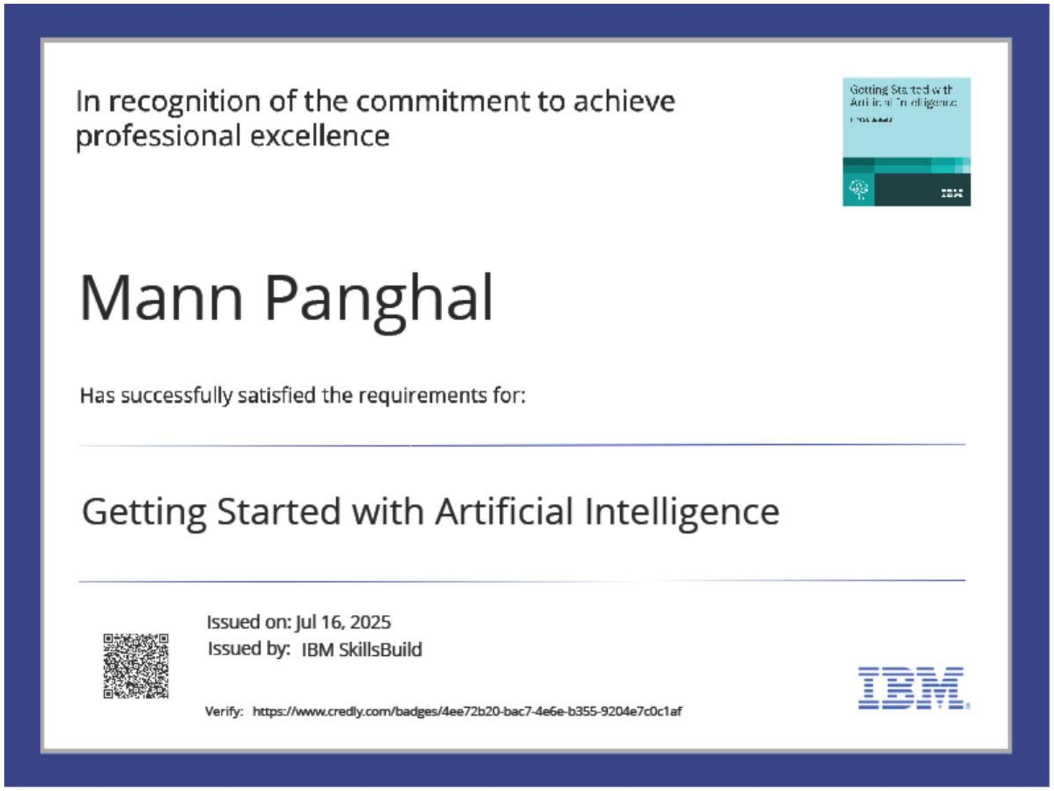
  URL: https://www.kaggle.com/datasets/sampadab17/network-intrusion-detection

- Libraries:

  Pandas Documentation: https://pandas.pydata.org/

  Scikit-learn Documentation: https://scikit-learn.org/

IBM CERTIFICATIONS



In recognition of the commitment to achieve professional excellence

Mann Panghal

Has successfully satisfied the requirements for:

Getting Started with Artificial Intelligence

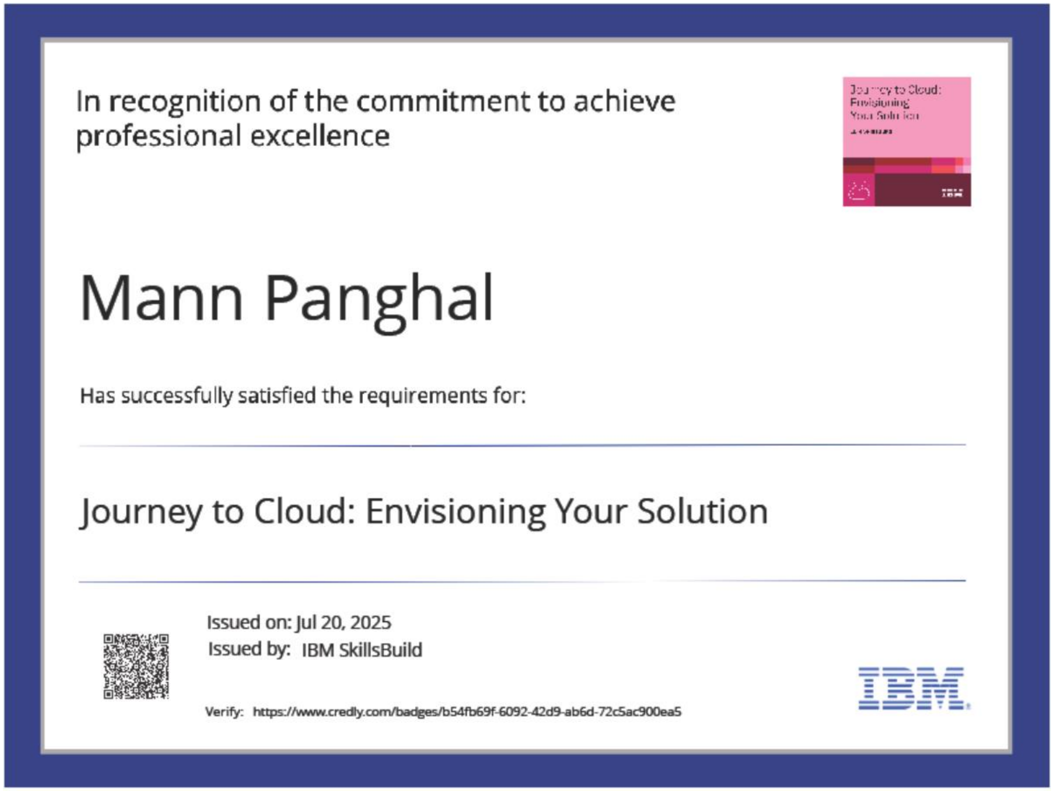
Issued on: Jul 16, 2025

Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/4ee72b20-bac7-4e6e-b355-9204e7c0c1af

Getting Started with Artificial Intelligence
IBM
IBM

IBM CERTIFICATIONS



In recognition of the commitment to achieve professional excellence

Mann Panghal

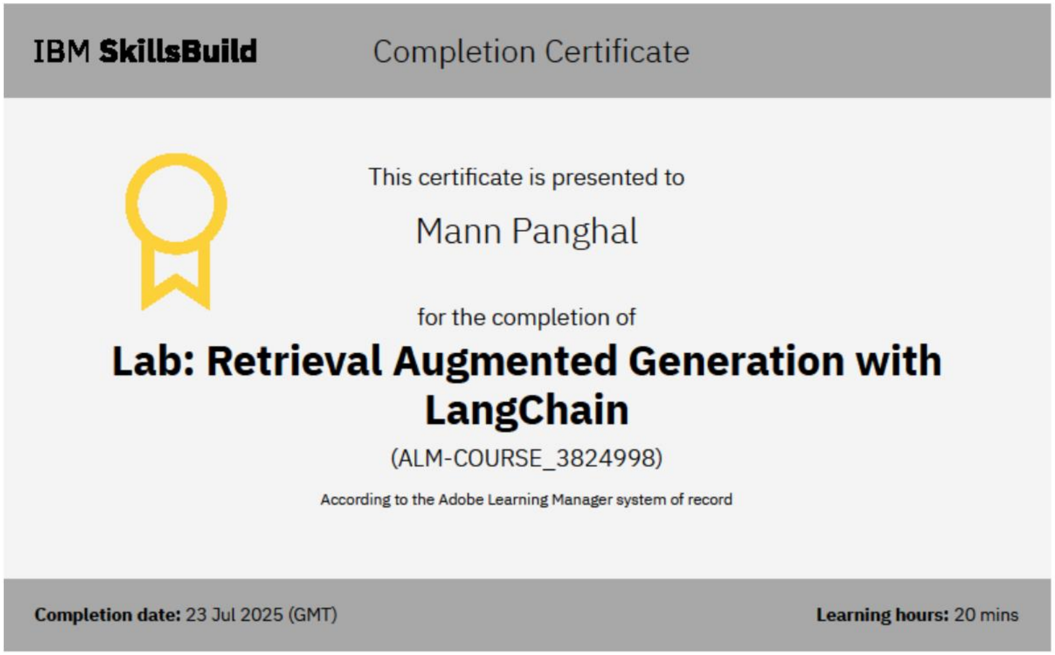
Has successfully satisfied the requirements for:

Journey to Cloud: Envisioning Your Solution

Issued on: Jul 20, 2025
Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/b54fb69f-6092-42d9-ab6d-72c5ac900ea5

IBM CERTIFICATIONS



IBM **SkillsBuild** Completion Certificate

This certificate is presented to

Mann Panghal

for the completion of

Lab: Retrieval Augmented Generation with LangChain

(ALM-COURSE_3824998)

According to the Adobe Learning Manager system of record

Completion date: 23 Jul 2025 (GMT)

Learning hours: 20 mins

THANK YOU