CAPSTONE PROJECT

NETWORK INTRUSION DETECTION
(MACHINE LEARNING)

Presented By:
Anshuman Bhandari – Graphic Era Hill University – CSE
Email – anshumanbhandari0000@gmail.com

OUTLINE

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

PROBLEM STATEMENT

- With the increasing frequency and sophistication of cyber-attacks, securing network infrastructure has become a critical challenge. This project aims to develop a Network Intrusion Detection System (NIDS) using machine learning techniques to analyze network traffic and accurately identify and classify malicious activities, such as Denial of Service (DoS), Probe, Remote to Local (R2L), and User to Root (U2R) attacks, as well as distinguish them from normal traffic. By leveraging the Kaggle network intrusion dataset and deploying the solution on IBM Cloud Lite, the goal is to create a scalable and effective system that provides real-time detection and early warning of potential security threats, enhancing the overall resilience of communication networks.

PROPOSED SOLUTION

- The proposed system aims to address the growing challenge of detecting and preventing cyber-attacks by analyzing network traffic using machine learning techniques. The solution focuses on accurately identifying and classifying different types of intrusions (DoS, Probe, R2L, U2R) and normal network activity to ensure the security of communication networks. It will consist of the following components:
- Data Collection:
  - Gather network traffic data from the Kaggle Network Intrusion Detection dataset, including features like protocol type, service, flag, source bytes, and destination bytes.
  - Split the data into training and holdout sets automatically using IBM Watson Studio.
- Data Preprocessing:
  - Clean and preprocess the dataset to handle missing values, redundant features, and class imbalances.
  - Apply label encoding and normalization techniques to convert categorical and numerical features into machine-readable formats.
  - AutoAI handles additional preprocessing and feature engineering steps automatically.

PROPOSED SOLUTION

- Machine Learning Algorithm:
  - Multiple pipelines were tested using Snap Decision Tree Classifier and Decision Tree Classifier algorithms.
  - The best-performing model was Pipeline 2, using Snap Decision Tree Classifier with Hyperparameter Optimization (HPO-1), achieving an accuracy of 99.5% via cross-validation.
  - Additional pipelines included feature engineering and HPO enhancements to improve model accuracy and generalization.
- Deployment:
  - The trained model can be exported and deployed on IBM Cloud Lite for real-time intrusion detection.
  - IBM Cloud Object Storage can be used to store the dataset and model artifacts.
  - Integration with live traffic monitoring systems or simulated network environments can enable real-time predictions and alerts.
- Evaluation:
  - Models were evaluated using cross-validation accuracy, with the top two pipelines achieving 0.995 accuracy.
  - The leaderboard guided selection of the best model for deployment
  - IBM watsonx.ai AutoAI ensured transparent comparison across 8 generated pipelines, optimizing both algorithm selection and hyperparameters automatically.
- Result:
  - Model name : Pipeline 2
  - Algorithm : Snap Decision Tree Classifier
  - Enhancement : Hyperparameter Optimization (HPO-1)
  - Accuracy : 0.995(95%)

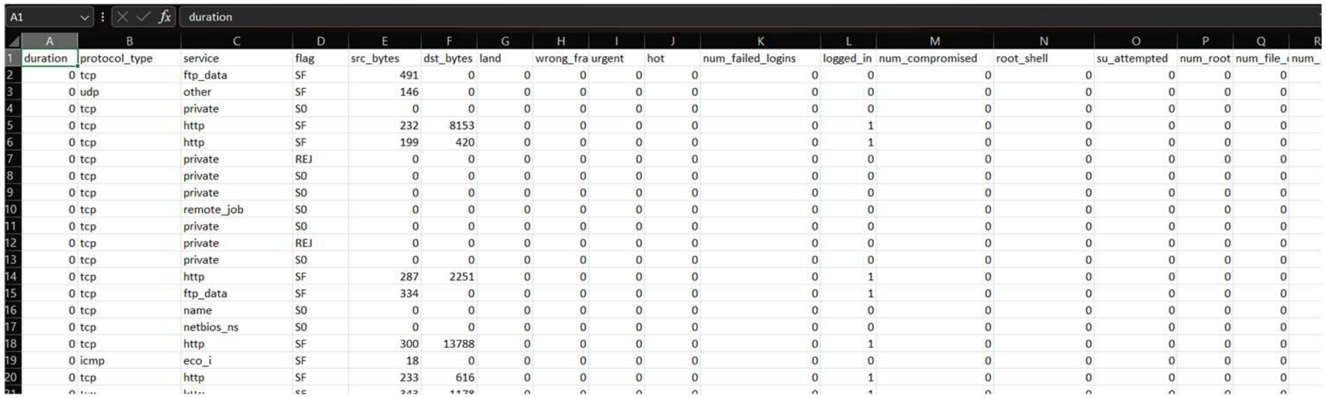
SYSTEM APPROACH

The system for Network Intrusion Detection is developed using IBM Watson Studio with the help of AutoAI. It follows a step-by-step approach to process the data, train models, and deploy the best-performing pipeline.

- System requirements : IBM cloud with Watsonx.ai and Watson Assistant access
- Library required to build the model : IBM Watson Assistant (Dialog Skill) for conversation flow and AutoAI – Automated model training and optimization (built-in with IBM Watson Studio)

ALGORITHM & DEPLOYMENT

- This section outlines the machine learning algorithm used for detecting cyber-attacks, the input data features, the training method, and the model deployment strategy.
- Algorithm Selection:
  - The best-performing model was a Snap Decision Tree Classifier, selected through IBM Watson Studio's AutoAI experiment.
  - This algorithm is well-suited for classification problems with categorical and numerical features, making it ideal for network intrusion detection tasks.
- Data Input:
  - Key input features used by the model include protocol type, service, flag, source bytes, destination bytes, count, srv_count, dst_host_count, and other network traffic-related attributes.
  - The target feature for classification was the class column, which indicates categories such as DoS, Probe, R2L, U2R, and Normal.

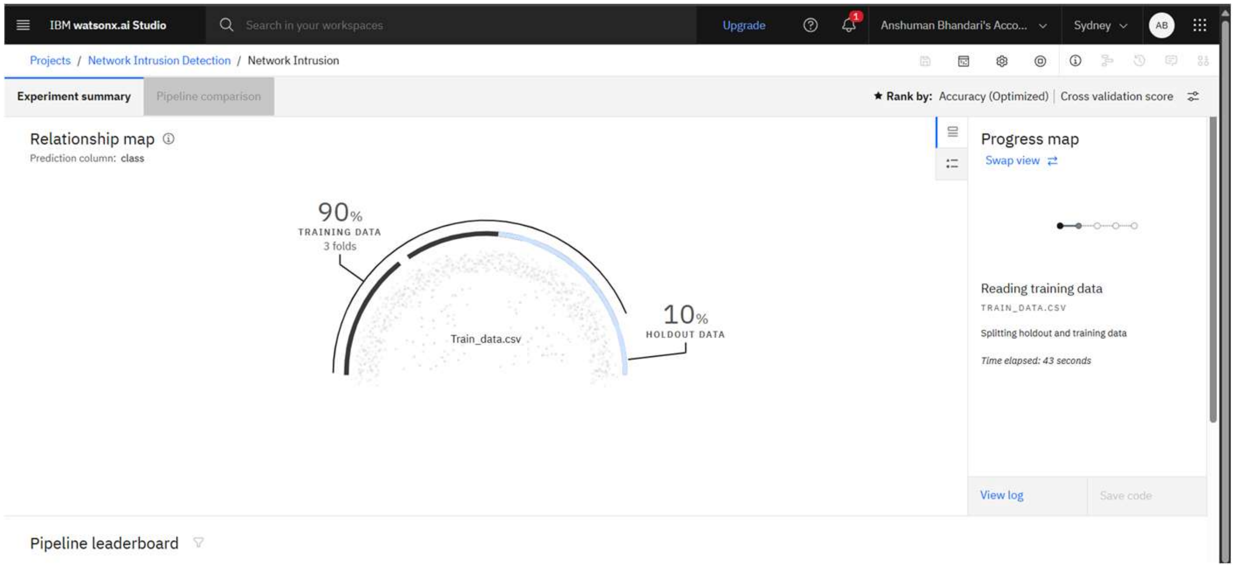


A1 duration

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fra	urgent	hot	num_failed_logins	logged_in	num_compromised	root_shell	su_attempted	num_root	num_file_(	num_
2	0	tcp	ftp_data	SF	491	0	0	0	0	0	0	0	0	0	0	0	0	
3	0	udp	other	SF	146	0	0	0	0	0	0	0	0	0	0	0	0	
4	0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	
5	0	tcp	http	SF	232	8153	0	0	0	0	0	1	0	0	0	0	0	
6	0	tcp	http	SF	199	420	0	0	0	0	0	1	0	0	0	0	0	
7	0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	
8	0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	
9	0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	
10	0	tcp	remote_job	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	
11	0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	
12	0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	
13	0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	
14	0	tcp	http	SF	287	2251	0	0	0	0	0	1	0	0	0	0	0	
15	0	tcp	ftp_data	SF	334	0	0	0	0	0	0	1	0	0	0	0	0	
16	0	tcp	name	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	
17	0	tcp	netbios_ns	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	
18	0	tcp	http	SF	300	13788	0	0	0	0	0	1	0	0	0	0	0	
19	0	icmp	eco_i	SF	18	0	0	0	0	0	0	0	0	0	0	0	0	
20	0	tcp	http	SF	233	616	0	0	0	0	0	1	0	0	0	0	0	

ALGORITHM & DEPLOYMENT

- Training Process:
  - Applied feature engineering and hyperparameter optimization (HPO).
  - Evaluated multiple pipelines using cross-validation to ensure robustness.
- Prediction Process:
  - Once trained, the model classifies incoming network traffic into one of the predefined classes (e.g., DoS, Probe).
  - The prediction can be done in real-time or on batch data, helping identify suspicious activities before they cause damage



Reading training data

TRAIN_DATA.CSV

Splitting holdout and training data

Time elapsed: 43 seconds

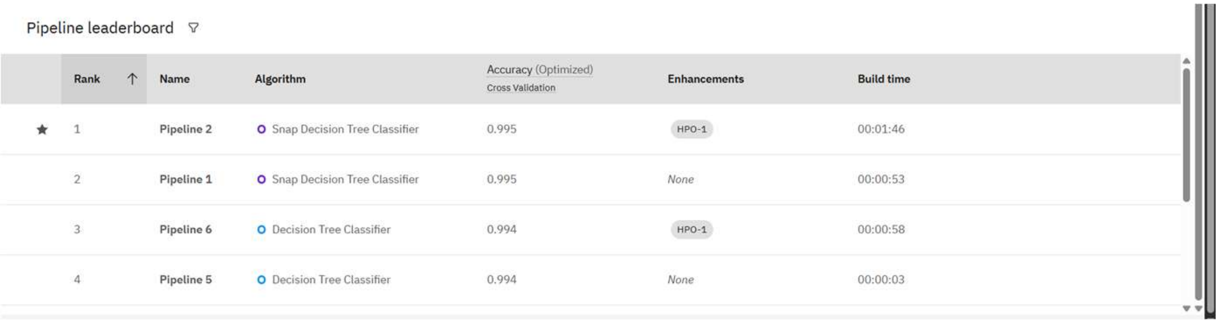
RESULT

- Total of **8 pipelines** were generated using IBM AutoAI.

- **Pipeline 2** gave the best performance. Used **Snap Decision Tree Classifier** with **Hyperparameter Optimization**.

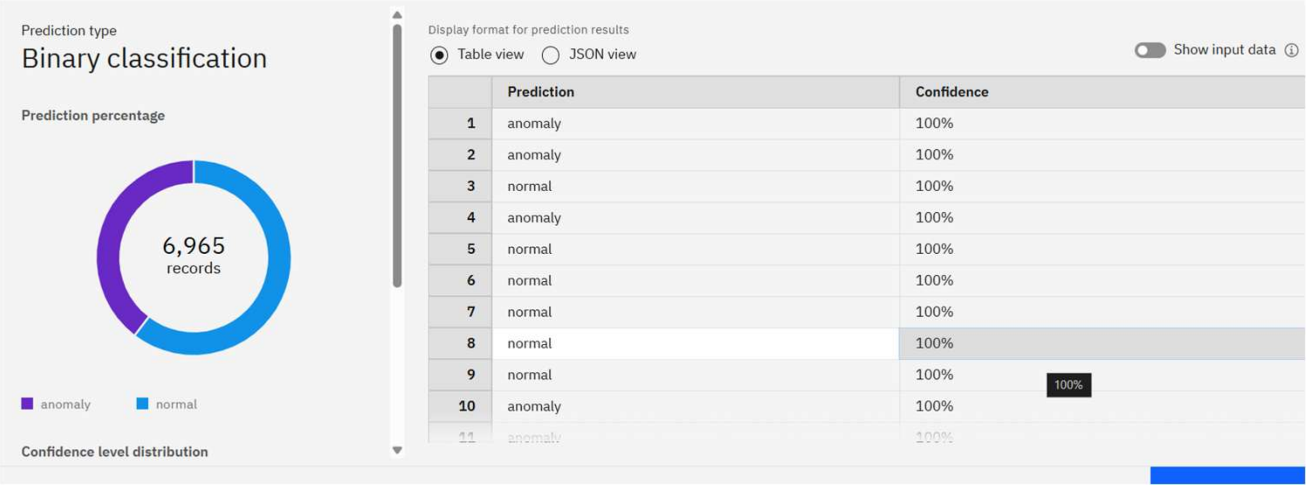
- Achieved **99.5% cross-validation accuracy**. Successfully classified attacks like **DoS, Probe, R2L, U2R**, and **Normal**.

- Model is ready for deployment on **IBM Cloud Lite**.



Pipeline leaderboard

	Rank ↑	Name	Algorithm	Accuracy (Optimized) Cross Validation	Enhancements	Build time
★	1	**Pipeline 2**	○ Snap Decision Tree Classifier	0.995	HPO-1	00:01:46
	2	**Pipeline 1**	○ Snap Decision Tree Classifier	0.995	*None*	00:00:53
	3	**Pipeline 6**	○ Decision Tree Classifier	0.994	HPO-1	00:00:58
	4	**Pipeline 5**	○ Decision Tree Classifier	0.994	*None*	00:00:03



Prediction type

Binary classification

Prediction percentage

Confidence level distribution

Display format for prediction results

● Table view ○ JSON view

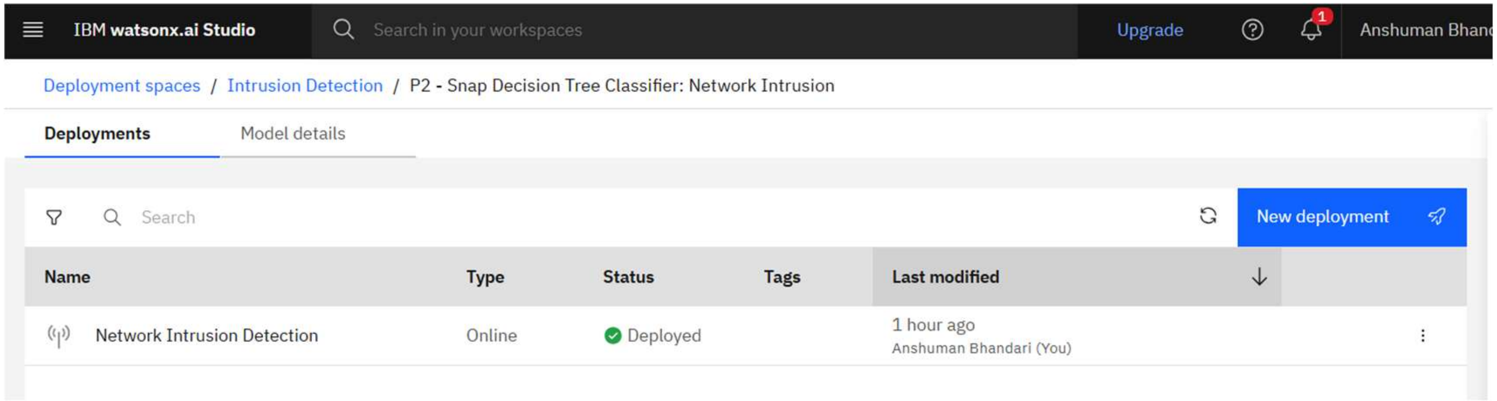
Show input data

	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%

edunet foundation

CONCLUSION

- The project successfully developed a machine learning-based Network Intrusion Detection System using IBM Watson Studio and AutoAI. The proposed solution, powered by the Snap Decision Tree Classifier, achieved a high accuracy of 99.5%, effectively classifying various types of cyber-attacks such as DoS, Probe, R2L, and U2R. This demonstrates the effectiveness of the model in securing network traffic and providing early threat detection.
- During implementation, challenges included handling class imbalance and preprocessing high-dimensional data. These were addressed through AutoAI's built-in optimization and feature engineering capabilities.



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

Name	Type	Status	Tags	Last modified
Network Intrusion Detection	Online	Deployed		1 hour ago Anshuman Bhandari (You)

FUTURE SCOPE

- Integrate real-time network traffic monitoring for live intrusion detection.
- Explore advanced models like deep learning (e.g., LSTM, CNN) for better accuracy on complex attack patterns.
- Improve detection of rare classes like R2L and U2R through data augmentation or anomaly detection techniques.
- Develop a user-friendly dashboard for security analysts to monitor threats.
- Expand the system to support multi-source data and cloud-based threat intelligence.

REFERENCES

- Intrusion Detection Dataset," *Kaggle*, https://www.kaggle.com/datasets/sampadab17/network-intrusion-detection.

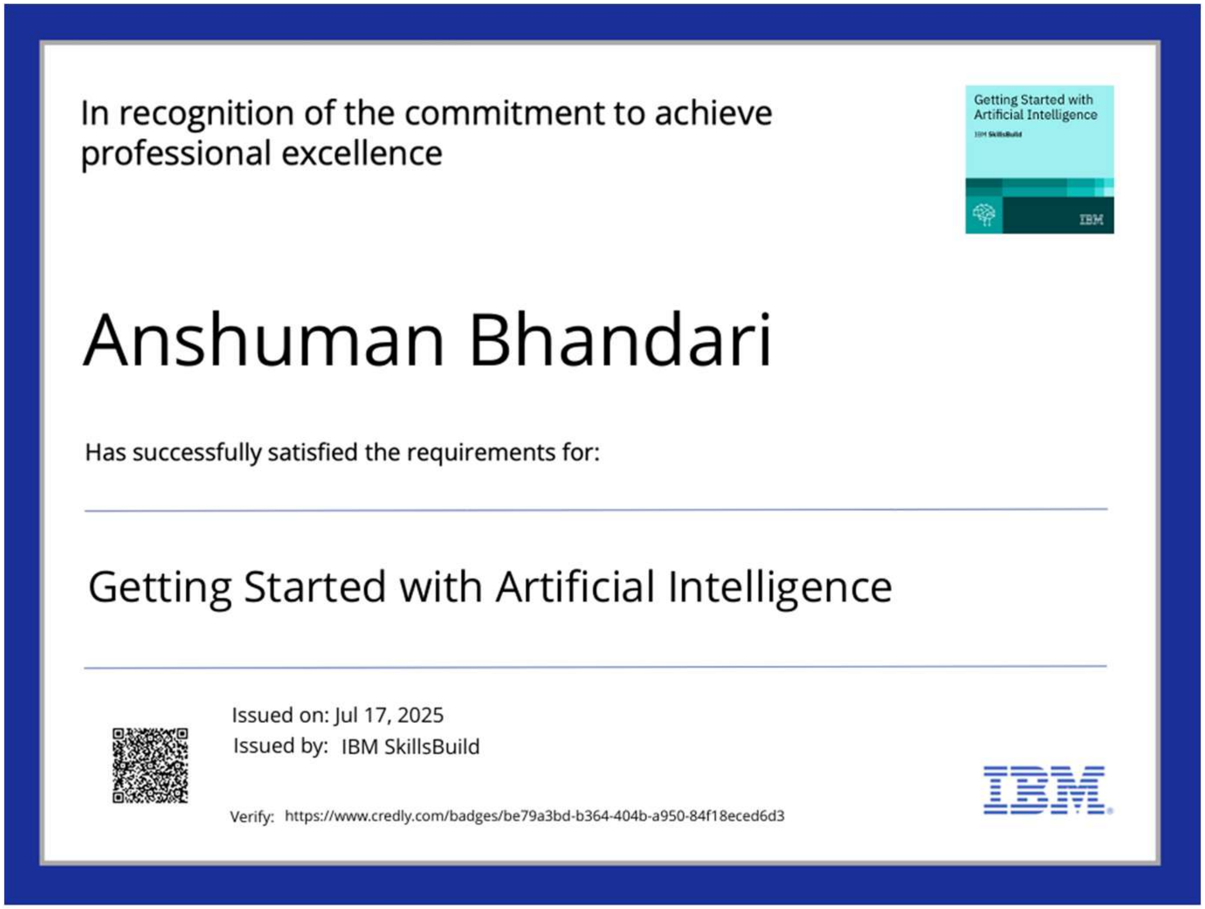
- *IBM Cloud Documentation, "Getting started with IBM Watson Studio and AutoAI," https://cloud.ibm.com/docs/watson-studio*

.

- Scikit-learn Developers, "Decision Tree Classifier — scikit-learn Documentation," https://scikit-learn.org/stable/modules/tree.html

- IBM Cloud Documentation, "Getting started with IBM Watson Studio and AutoAI," https://cloud.ibm.com/docs/watson-studio

IBM CERTIFICATIONS



Getting Started with Artificial Intelligence
IBM SkillsBuild
IBM

In recognition of the commitment to achieve professional excellence

Anshuman Bhandari

Has successfully satisfied the requirements for:

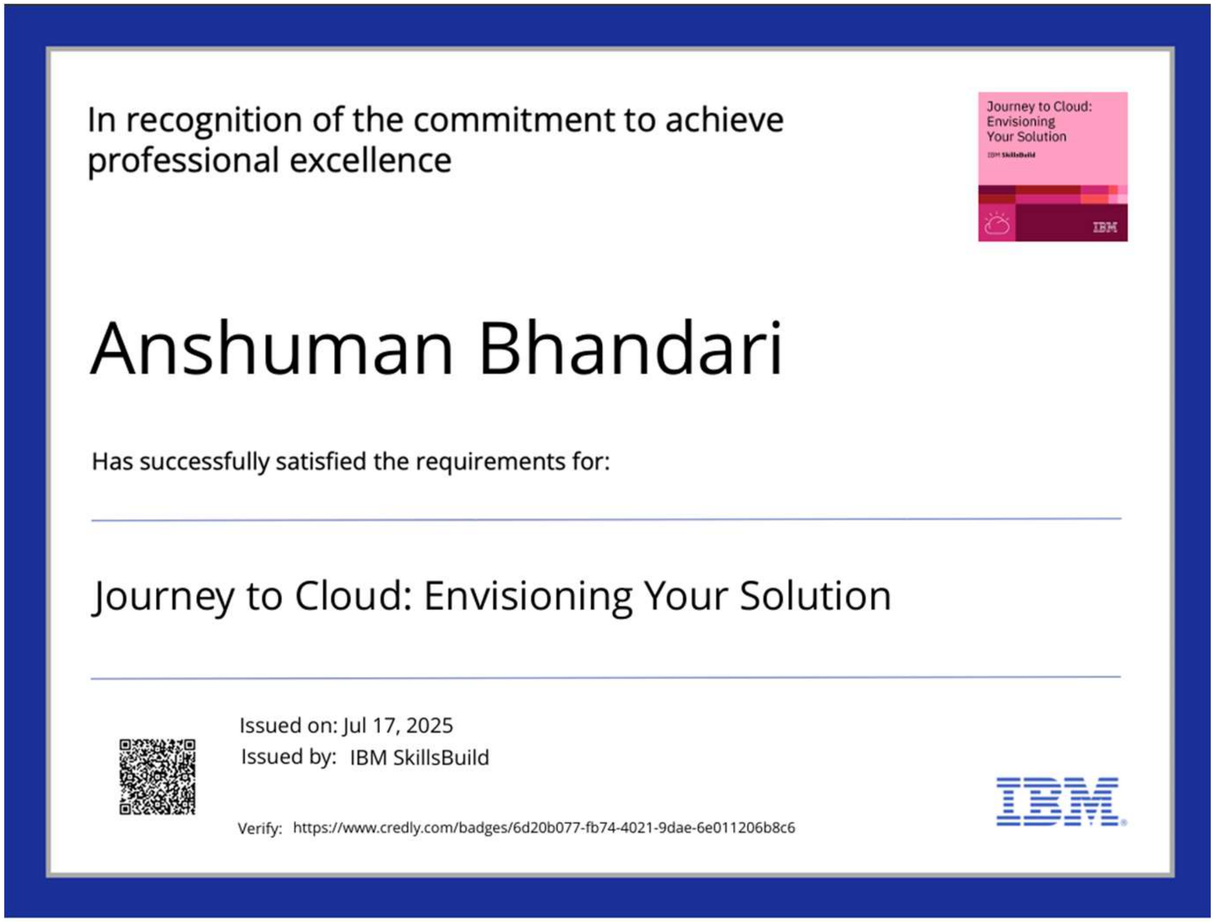
Getting Started with Artificial Intelligence

Issued on: Jul 17, 2025
Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/be79a3bd-b364-404b-a950-84f18eced6d3

IBM

IBM CERTIFICATIONS



Journey to Cloud: Envisioning Your Solution
IBM SkillsBuild
IBM

In recognition of the commitment to achieve professional excellence

Anshuman Bhandari

Has successfully satisfied the requirements for:

Journey to Cloud: Envisioning Your Solution

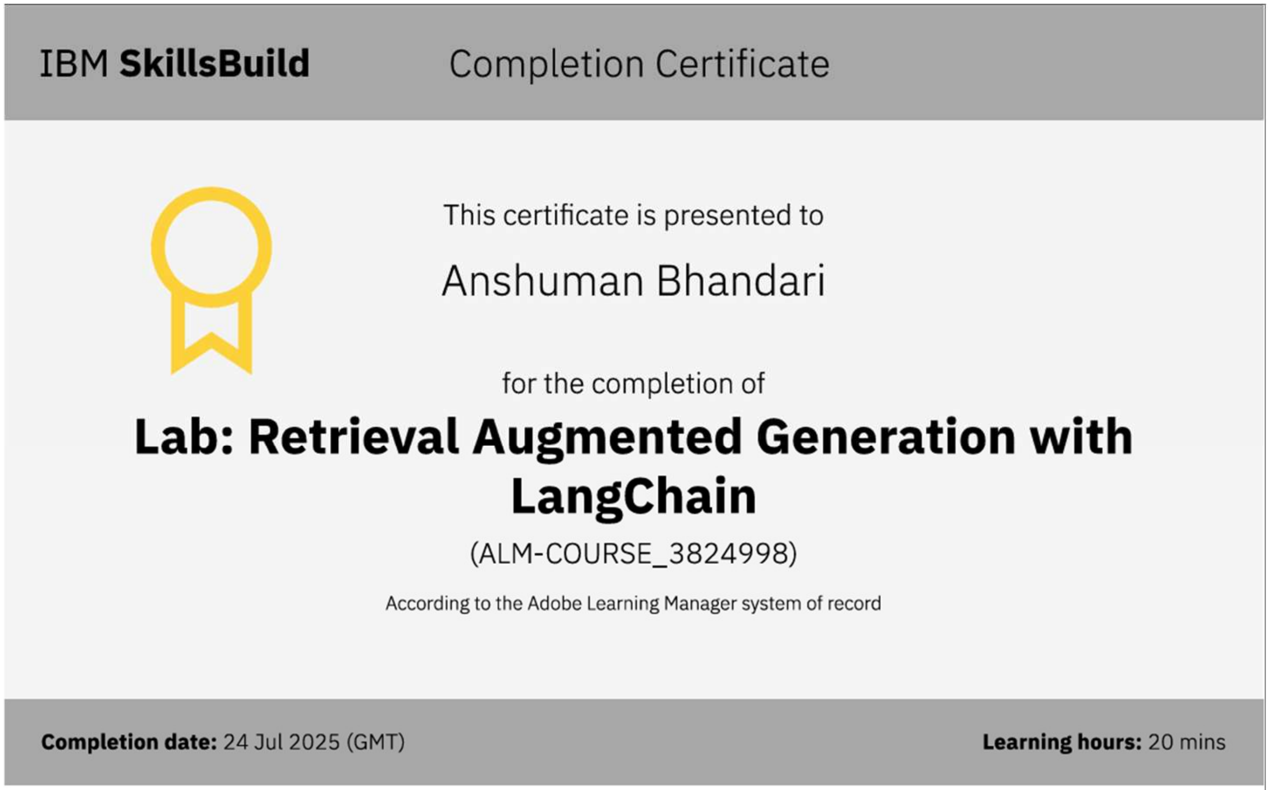
Issued on: Jul 17, 2025
Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/6d20b077-fb74-4021-9dae-6e011206b8c6

IBM

edunet
foundation

IBM CERTIFICATIONS



IBM **SkillsBuild** Completion Certificate

This certificate is presented to

Anshuman Bhandari

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