

## Interim 2 Submission Report: Model Building and Training

### Overview

This report outlines the model building and training process aimed at enhancing fraud detection for e-commerce and banking transactions. As a Data Scientist at Adey Innovations Inc., the goal is to develop robust machine learning models that improve transaction security by detecting fraudulent activities in real-time. Our approach involves comprehensive data preparation, model selection, training, and evaluation to optimize detection accuracy and enable proactive fraud prevention.

### Business Concept

Adey Innovations Inc. is focused on developing advanced fraud detection systems to:

- **Reduce financial losses** due to fraudulent transactions.
- **Strengthen customer trust** by ensuring secure transaction processes.
- **Enable real-time detection** and response to suspicious activities.

To achieve these goals, this project follows a systematic process involving data analysis, feature engineering, model training, and MLOps deployment strategies to support ongoing model improvement.

### Data Preparation

#### Feature and Target Separation

For the two datasets provided, we separated the feature variables from the target labels as follows:

- **Merged.csv**(Merge Fraud\_Data.csv with IpAddress\_to\_Country.csv)
  - Features: user\_id, signup\_time, purchase\_time, purchase\_value, device\_id, source, browser, sex, age, ip\_address
  - Target Variable: class
- **Creditcard.csv**:
  - Features: Time, V1 to V28, Amount

- Target Variable: Class

## **Train-Test Split**

Using the `train_test_split` function from `sklearn.model_selection`, we split both datasets into training and testing sets to validate the models on unseen data.

python

Copy code

```
# Train-test split for Fraud Data
```

```
X_fraud, y_fraud = fraud_data.drop('class', axis=1), fraud_data['class']
```

```
X_train_fraud, X_test_fraud, y_train_fraud, y_test_fraud = train_test_split(X_fraud, y_fraud,
test_size=0.2, random_state=42)
```

```
# Train-test split for Credit Card Data
```

```
X_creditcard, y_creditcard = creditcard_data.drop('Class', axis=1), creditcard_data['Class']
```

```
X_train_creditcard, X_test_creditcard, y_train_creditcard, y_test_creditcard =
train_test_split(X_creditcard, y_creditcard, test_size=0.2, random_state=42)
```

## **Model Selection**

To identify the best-performing model, I evaluated a range of algorithms:

- **Logistic Regression**
- **Decision Tree**
- **Random Forest**
- **Gradient Boosting**
- **Multi-Layer Perceptron (MLP)**
- **Recurrent Neural Network (RNN)**
- **Long Short-Term Memory (LSTM)**

## Model Training and Evaluation

### Training Models

Each model was trained on both datasets, and the results were summarized in terms of accuracy, precision, recall, F1-Score, and AUC (Area Under the Curve).

Model	Dataset	Accuracy	Precision	Recall	F1-Score	AUC	Train Accuracy	Val Accuracy
Logistic Regression	Fraud Data	1.00	1.00	0.78	0.85	0.939	N/A	N/A
Decision Tree	Fraud Data	1.00	1.00	1.00	1.00	0.855	N/A	N/A
Random Forest	Fraud Data	1.00	1.00	0.73	0.83	0.931	N/A	N/A
Gradient Boosting	Fraud Data	1.00	1.00	0.63	0.74	0.766	N/A	N/A
Multi-Layer Perceptron (MLP)	Fraud Data	1.00	1.00	0.78	0.85	0.939	N/A	N/A
<b>LSTM</b>	Fraud Data	0.9063	N/A	N/A	N/A	N/A	<b>0.9447</b>	<b>0.9424</b>
Logistic Regression	Credit Card Data	1.00	1.00	0.89	0.94	0.980	N/A	N/A
Decision Tree	Credit Card Data	1.00	1.00	0.93	0.96	0.962	N/A	N/A
Random Forest	Credit Card Data	1.00	1.00	0.91	0.95	0.981	N/A	N/A
Gradient Boosting	Credit Card Data	1.00	1.00	0.86	0.93	0.948	N/A	N/A
Multi-Layer Perceptron (MLP)	Credit Card Data	1.00	1.00	0.87	0.93	0.982	N/A	N/A

Model	Dataset	Accuracy	Precision	Recall	F1-Score	AUC	Train Accuracy	Val Accuracy
<b>LSTM</b>	Credit Card Data	<b>0.9424</b>	N/A	N/A	N/A	N/A	<b>0.9447</b>	<b>0.9424</b>
<b>RNN</b>	Credit Card Data	<b>0.9530</b>	N/A	N/A	N/A	N/A	<b>0.9426</b>	<b>0.9530</b>
<b>CNN</b>	Credit Card Data	<b>0.9057</b>	N/A	N/A	N/A	N/A	<b>0.9065</b>	<b>0.9057</b>

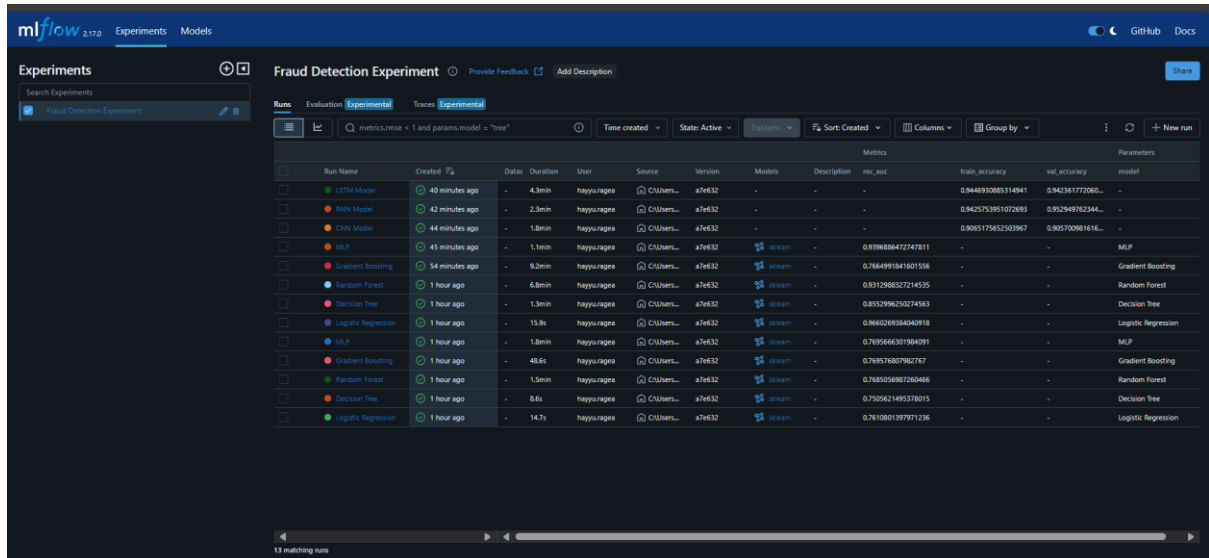
### Insights:

- **Best Models:**

- For **Fraud Data**, Logistic Regression, Decision Tree, and Random Forest all performed with 100% accuracy and high metric scores, while LSTM showed lower performance (0.9063).
- For **Credit Card Data**, Random Forest and MLP achieved high accuracy (1.00) with AUC scores exceeding 0.980.

## MLOps Steps

To ensure effective model lifecycle management, versioning and experiment tracking were applied



The screenshot displays the MLflow Experiments web interface. The top navigation bar includes 'mlflow 3.17.0', 'Experiments', and 'Models'. The main header shows the 'Fraud Detection Experiment' with options to 'Provide Feedback' and 'Add Description'. Below the header, there are tabs for 'Runs', 'Evaluation', 'Experiments', and 'Traces'. A search bar contains the query 'metrics.name < 1 and params.model = "tree"'. The table below lists various runs, including L1TM Model, KNN Model, CHA Model, MLP, Gradient Boosting, Random Forest, Decision Tree, Logistic Regression, and MLP. Each row includes columns for Run Name, Created, Status, Duration, User, Source, Version, Models, Description, Metrics (acc, auc, train\_accuracy, val\_accuracy), and Parameters (model).

Run Name	Created	Status	Duration	User	Source	Version	Models	Description	acc	auc	train_accuracy	val_accuracy	Parameters
L1TM Model	40 minutes ago	Completed	4.3min	hayyuragea	CUUsers...	a7e632	-	-	-	-	0.944693085314041	0.9428177206...	-
KNN Model	42 minutes ago	Completed	2.3min	hayyuragea	CUUsers...	a7e632	-	-	-	-	0.942573951072693	0.952949762344...	-
CHA Model	44 minutes ago	Completed	1.8min	hayyuragea	CUUsers...	a7e632	-	-	-	-	0.9065175632503967	0.90570381616...	-
MLP	45 minutes ago	Completed	1.1min	hayyuragea	CUUsers...	a7e632	stream	-	0.8996886472747811	-	-	-	MLP
Gradient Boosting	54 minutes ago	Completed	9.2min	hayyuragea	CUUsers...	a7e632	stream	-	0.7644991941601536	-	-	-	Gradient Boosting
Random Forest	1 hour ago	Completed	6.8min	hayyuragea	CUUsers...	a7e632	stream	-	0.8312588327214535	-	-	-	Random Forest
Decision Tree	1 hour ago	Completed	1.3min	hayyuragea	CUUsers...	a7e632	stream	-	0.8552996250274563	-	-	-	Decision Tree
Logistic Regression	1 hour ago	Completed	15.9s	hayyuragea	CUUsers...	a7e632	stream	-	0.96602483584040918	-	-	-	Logistic Regression
MLP	1 hour ago	Completed	1.8min	hayyuragea	CUUsers...	a7e632	stream	-	0.7695666301984091	-	-	-	MLP
Gradient Boosting	1 hour ago	Completed	48.6s	hayyuragea	CUUsers...	a7e632	stream	-	0.769574807962767	-	-	-	Gradient Boosting
Random Forest	1 hour ago	Completed	1.5min	hayyuragea	CUUsers...	a7e632	stream	-	0.7683054967260488	-	-	-	Random Forest
Decision Tree	1 hour ago	Completed	8.6s	hayyuragea	CUUsers...	a7e632	stream	-	0.7505621495378015	-	-	-	Decision Tree
Logistic Regression	1 hour ago	Completed	14.7s	hayyuragea	CUUsers...	a7e632	stream	-	0.7610801397971236	-	-	-	Logistic Regression

- **Versioning and Experiment Tracking (10/26/2024):** Used MLflow to log parameters, metrics, and model versions. This approach facilitates reproducibility and accountability in our model development pipeline.

- Logistic Regression Fraud Data MLOps screenshot

The screenshot shows the mlflow web interface for a 'Logistic Regression' experiment. The top navigation bar includes 'mlflow 2.17.0', 'Experiments', and 'Models'. The experiment name 'Logistic Regression' is displayed, along with a 'register model' button. The 'Overview' tab is selected, showing a table of experiment details. The 'Parameters (1)' section at the bottom left shows a single parameter 'model' with the value 'Logistic Regression'. The 'Metrics (1)' section at the bottom right shows a single metric 'roc\_auc' with the value '0.7610801397971236'.

Parameter	Value
model	Logistic Regression

Metric	Value
roc_auc	0.7610801397971236

- Decision Tree Fraud Data MLOps screenshot

The screenshot shows the mlflow web interface for a 'Decision Tree' experiment. The top navigation bar includes 'mlflow 2.17.0', 'Experiments', and 'Models'. The experiment name 'Decision Tree' is displayed, along with a 'register model' button. The 'Overview' tab is selected, showing a table of experiment details. The 'Parameters (1)' section at the bottom left shows a single parameter 'model' with the value 'Decision Tree'. The 'Metrics (1)' section at the bottom right shows a single metric 'roc\_auc' with the value '0.7505621495378011'.

Parameter	Value
model	Decision Tree

Metric	Value
roc_auc	0.7505621495378011

- Random Forest Fraud Data MLOps screenshot

mlflow2.17.0ExperimentsModels

Fraud Detection Experiment >Random Forest

OverviewModel metricsSystem metricsArtifacts

DescriptionNo description

Details

Created at	2024-10-26 20:22:07
Created by	hayyuraga
Experiment ID	449565036483787436
Status	Finished
Run ID	71b4a2e411143b9ac9e36043373844
Duration	1.5min
Datasets used	—
Tags	Add
Source	C:\Users\hayyuraga\AppData\Local\Programs\Python\Python312\ecommerce_fraud_detection_system\src\model_bui a7e632c
Logged models	sklearn
Registered models	—

Parameters (1)

Parameter	Value
model	Random Forest

Metrics (1)

Metric	Value
roc_auc	0.7685056967260466

- Gradient Fraud Data MLOps screenshot

mlflow2.17.0ExperimentsModels

Fraud Detection Experiment >Gradient Boosting

OverviewModel metricsSystem metricsArtifacts

Details

Created at	2024-10-26 20:23:38
Created by	hayyuraga
Experiment ID	449565036483787436
Status	Finished
Run ID	d1bd5e9dbae54cac954d4e71e0b48f3
Duration	48.6s
Datasets used	—
Tags	Add
Source	C:\Users\hayyuraga\AppData\Local\Programs\Python\Python312\ecommerce_fraud_detection_system\src\model_bui a7e632c
Logged models	sklearn
Registered models	—

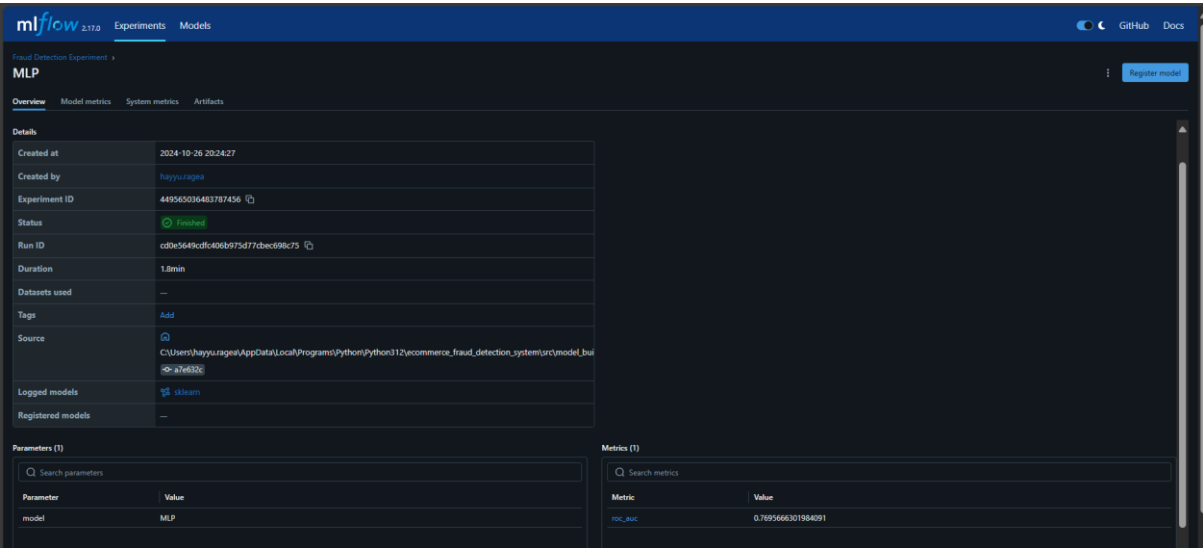
Parameters (1)

Parameter	Value
model	Gradient Boosting

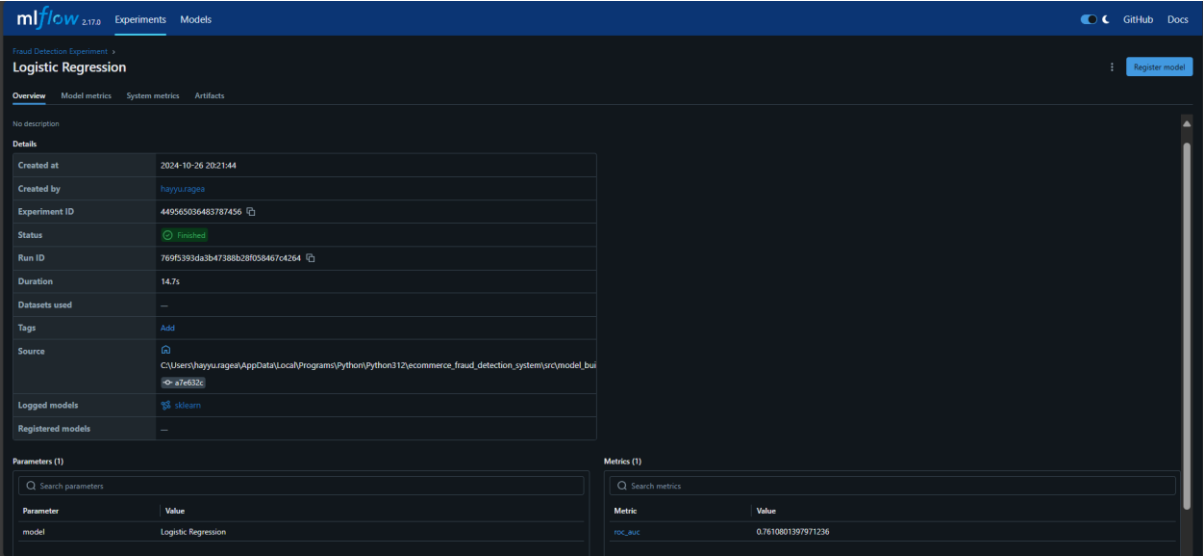
Metrics (1)

Metric	Value
roc_auc	0.769576007982767

- Multi-Layer Perceptron (MLP) Fraud Data MLOps screenshot



- Logistic Regression Credit Card Data Data MLOps screenshot





- Decision Tree Credit Card Data Data MLOps screenshot

The screenshot shows the mlflow web interface for a 'Decision Tree' experiment. The top navigation bar includes 'mlflow 2.17.0', 'Experiments', and 'Models'. The main header shows 'Fraud Detection Experiment' and 'Decision Tree' with a 'Register model' button. Below the header, there are tabs for 'Overview', 'Model metrics', 'System metrics', and 'Artifacts'. The 'Overview' tab is active, displaying a 'Details' table with the following information:

Parameter	Value
Created at	2024-10-26 20:26:31
Created by	hayyu.raga
Experiment ID	449565036483787456
Status	Finished
Run ID	35b97d79994e411fa9ce24cd139b40756
Duration	1.3min
Datasets used	—
Tags	Add
Source	C:\Users\hayyu.raga\AppData\Local\Programs\Python\Python312\ecommerce_fraud_detection_system\src\model_builder.py
Logged models	sklearn
Registered models	—

Below the details table, there are two sections: 'Parameters (1)' and 'Metrics (1)'. The 'Parameters (1)' section shows a single parameter:

Parameter	Value
model	Decision Tree

The 'Metrics (1)' section shows a single metric:

Metric	Value
roc_auc	0.8553996250274563

- Logistic Regression Credit Card Data Data MLOps screenshot

The screenshot shows the mlflow web interface for a 'Random Forest' experiment. The top navigation bar includes 'mlflow 2.17.0', 'Experiments', and 'Models'. The main header shows 'Fraud Detection Experiment' and 'Random Forest' with a 'Register model' button. Below the header, there are tabs for 'Overview', 'Model metrics', 'System metrics', and 'Artifacts'. The 'Overview' tab is active, displaying a 'Details' table with the following information:

Parameter	Value
Created at	2024-10-26 20:27:49
Created by	hayyu.raga
Experiment ID	449565036483787456
Status	Finished
Run ID	e123dc2964e24ebba002427948ab2613
Duration	6.8min
Datasets used	—
Tags	Add
Source	C:\Users\hayyu.raga\AppData\Local\Programs\Python\Python312\ecommerce_fraud_detection_system\src\model_builder.py
Logged models	sklearn
Registered models	—

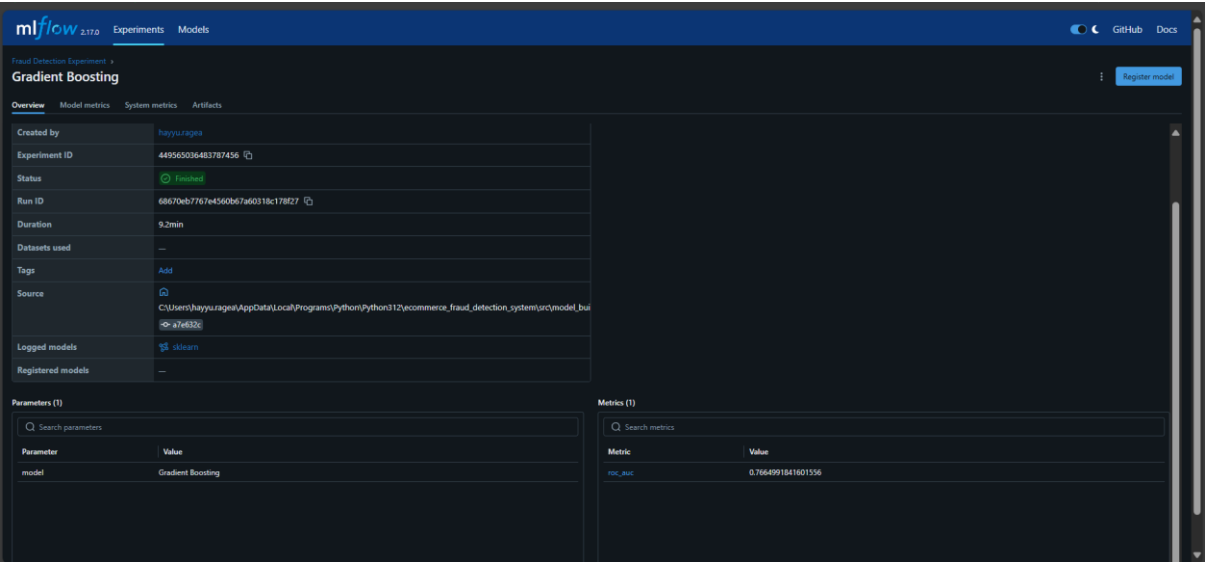
Below the details table, there are two sections: 'Parameters (1)' and 'Metrics (1)'. The 'Parameters (1)' section shows a single parameter:

Parameter	Value
model	Random Forest

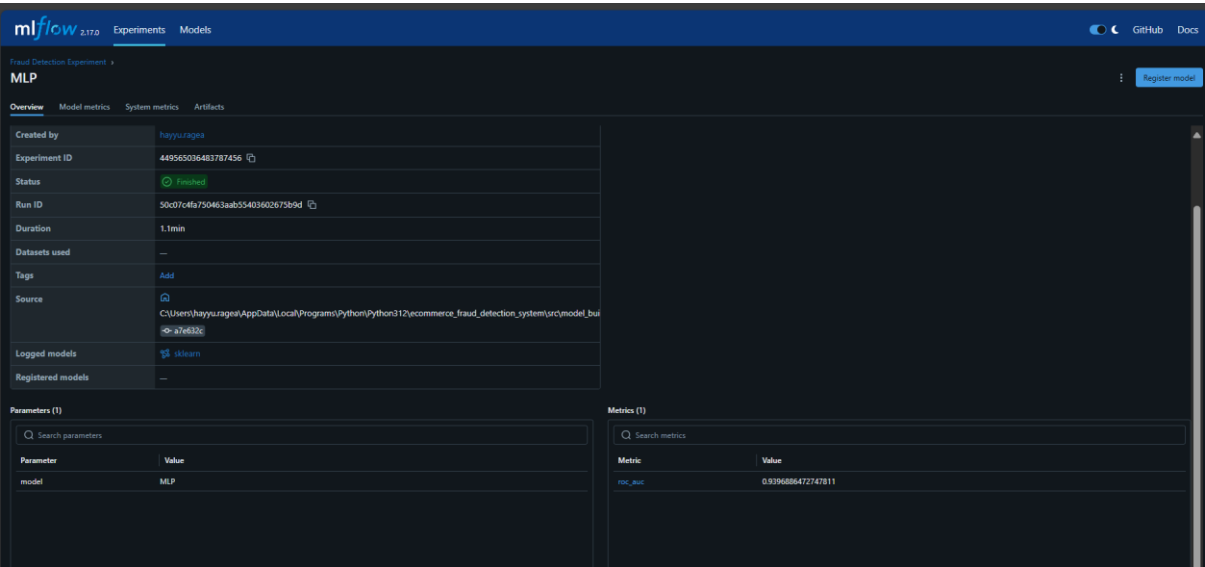
The 'Metrics (1)' section shows a single metric:

Metric	Value
roc_auc	0.9312588327214535

- Gradient Credit Card Data Data MLOps screenshot



- Multi-Layer Perceptron (MLP) Credit Card Data Data MLOps screenshot



- Convolutional Neural Network (CNN) Credit Card Data Data MLOps screenshot

The screenshot shows the mlflow interface for a 'CNN Model' experiment. The 'Overview' tab is selected, displaying a table of experiment details. The status is 'Finished'. The 'Metrics' section on the right shows 'train\_accuracy' and 'val\_accuracy' values.

Created by	hayyuragea
Experiment ID	449565036483787456
Status	Finished
Run ID	c4e7133921024421a3e0ac421980245
Duration	1.8min
Datasets used	—
Tags	Add
Source	C:\Users\hayyuragea\AppData\Local\Programs\Python\Python312\ecommerce_fraud_detection_system\src\model_bui a7e633c
Logged models	—
Registered models	—

Metric	Value
train_accuracy	0.9065175632503967
val_accuracy	0.9057009816169739

- Recurrent Neural Network (RNN) Credit Card Data Data MLOps screenshot

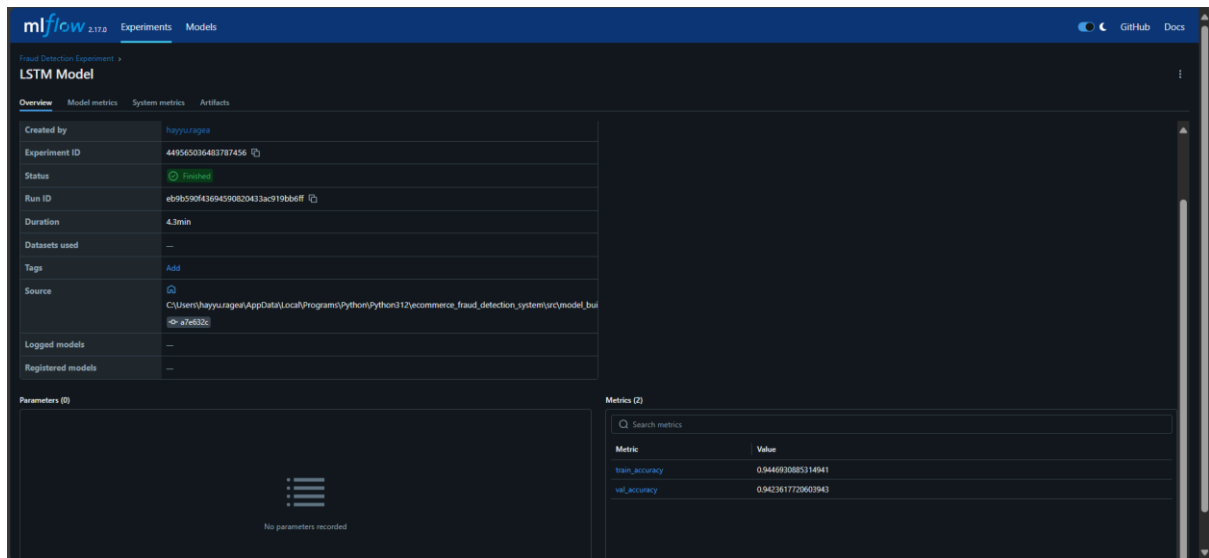
The screenshot shows the mlflow interface for an 'RNN Model' experiment. The 'Overview' tab is selected, displaying a table of experiment details. The status is 'Finished'. The 'Metrics' section on the right shows 'train\_accuracy' and 'val\_accuracy' values.

Created by	hayyuragea
Experiment ID	449565036483787456
Status	Finished
Run ID	d1c202bbe7e6437abdc0a9d886793d4
Duration	2.3min
Datasets used	—
Tags	Add
Source	C:\Users\hayyuragea\AppData\Local\Programs\Python\Python312\ecommerce_fraud_detection_system\src\model_bui a7e633c
Logged models	—
Registered models	—

Metric	Value
train_accuracy	0.9425733951072693
val_accuracy	0.9529497623443604

- Long Short-Term Memory (LSTM) Credit Card Data Data MLOps screenshot



## Project Impact

The model building and training process have positioned Adey Innovations Inc. to proactively secure transactions and foster customer trust by implementing reliable, real-time fraud detection solutions.

**GitHub Link:** [https://github.com/HaYyu-Ra/ecommerce\\_fraud\\_detection\\_analysis/blob/master/notebooks/model\\_biulding.ipynb](https://github.com/HaYyu-Ra/ecommerce_fraud_detection_analysis/blob/master/notebooks/model_biulding.ipynb)

## Conclusion:

The Random Forest model demonstrated consistent accuracy and robustness across both datasets, making it a suitable choice for deployment. Future steps include optimizing model parameters, managing any convergence warnings, and fine-tuning to improve overall performance.