National College of Ireland

Autonomous Cloud Resource Allocation: A Hybrid Machine Learning Ensemble Approach - Configuration Manual

MSc Research Project
MSc Cloud Computing

Abhishek Malik
Student ID: x22165843

School of Computing
National College of Ireland

Supervisor: Ahmed Makki

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Abhishek Malik
Student ID:	x22165843
Programme:	MSc Cloud Computing
Year:	2023
Module:	MSc Research Project
Supervisor:	Ahmed Makki
Submission Due Date:	14/12/2023
Project Title:	Autonomous Cloud Resource Allocation: A Hybrid Machine Learning Ensemble Approach - Configuration Manual
Word Count:	518
Page Count:	10

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Abhishek Malik
Date:	14th December 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Autonomous Cloud Resource Allocation: A Hybrid Machine Learning Ensemble Approach - Configuration Manual

Abhishek Malik
x22165843

1 Introduction

This configuration manual outlines the process that can be used to recreate the development of a hybrid machine-learning ensemble to predict the allocation of cloud resources. The basic configuration steps are included and screenshots are given for easy step by step setup. The System Requirements and Code specifications are also mentioned which can be imported/configured at the time of setup. This Configuration Manual is based on AWS SageMaker. *Set Up Amazon SageMaker Prerequisites* (n.d.)

2 System configuration requirements

- Hardware :- AWS SageMaker ml.t3.medium Instance with 2 CPUs and 4 GB RAM.
- Software :- Python 3.7.9 version, Amazon Sagemaker.
- Python library :- Tensorflow, keras, scikit learn, pandas, numpy, matplotlib, plotly, seaborn etc.
- Dataset Link:- `http://gwa.ewi.tudelft.nl/datasets/gwa-t-12-bitbrains`

3 Project Recreation

The following steps indicate how to steup and configure the Project on AWS SageMaker:

3.1 Environment Setup

AWS SageMaker ml.t3.medium instance is used to develop and evaluate the project. Python 3.7.9 is used to develop the code.

- Step 1: Login to AWS SageMaker and Create an Instance. 1
- Step 2: Choose a name of your Instance. Choose ml.t3.medium instance type. Platform as Amazon Linux 2, Jupyter Lab 3 and create or choose existing IAM Role. 2

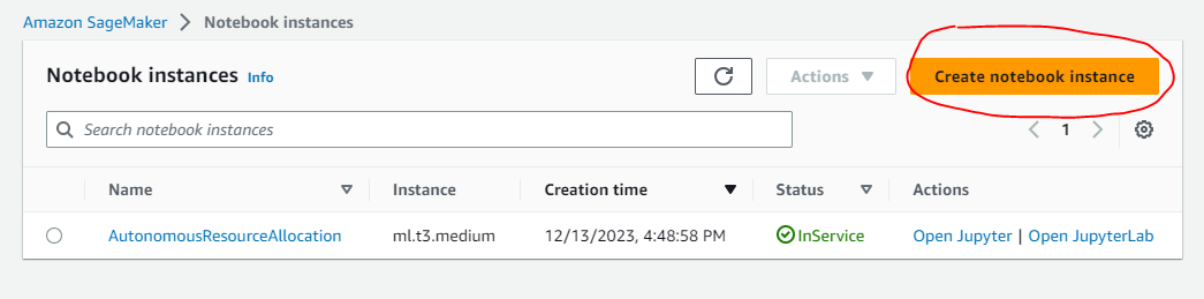

Figure 1: Create AWS SageMaker Instance

- Step 3: Wait for the instance to show status as "InService". Then click on "Open JupyterLab". *Set Up JupyterLab in Amazon SageMaker* (n.d.) 3
- Step 4: Create a folder named "Data" and upload the Code File in the root where Data Folder is created. 4
- Step 5: Inside Data Folder, Upload the CSV Files. 5

The Environment setup is complete with these steps.

3.2 Packages and libraries

The following libraries can be installed (if not already installed in AWS SageMaker) by using "pip" command as "pip install <libraryname>":

- Tensorflow
- keras
- scikit learn
- pandas
- numpy
- matplotlib
- plotly
- seaborn

4 Phases

To run the code, click on the code file and select "Run All Cells" from the "Run" menu. 6

- Data Collection - 7
- Data Description - 8

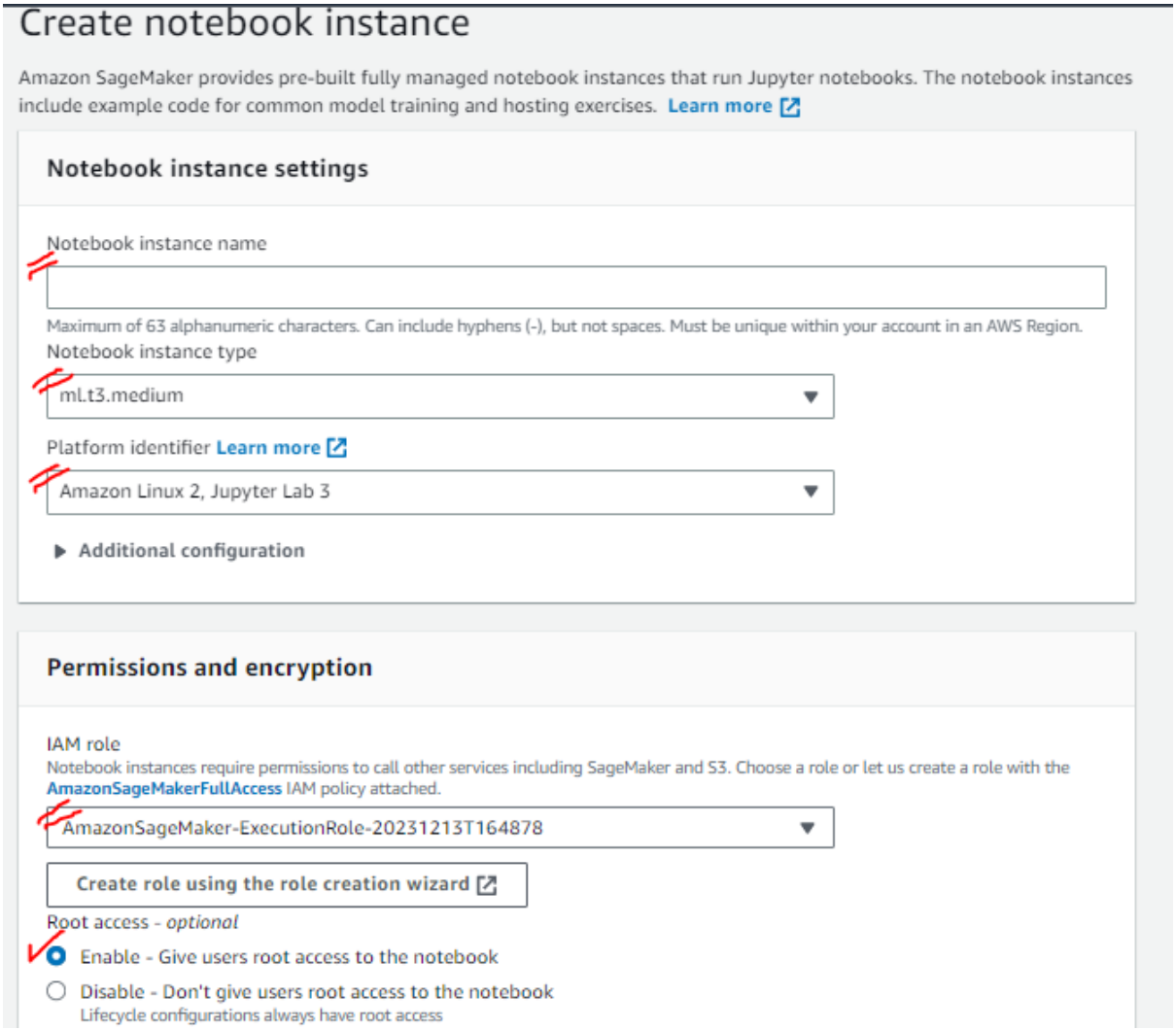

Figure 2: Configure AWS SageMaker Instance

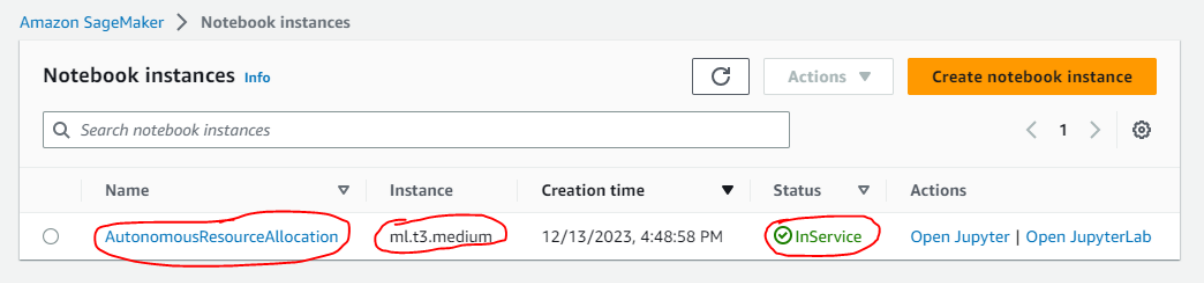

Figure 3: Created AWS SageMaker Instance

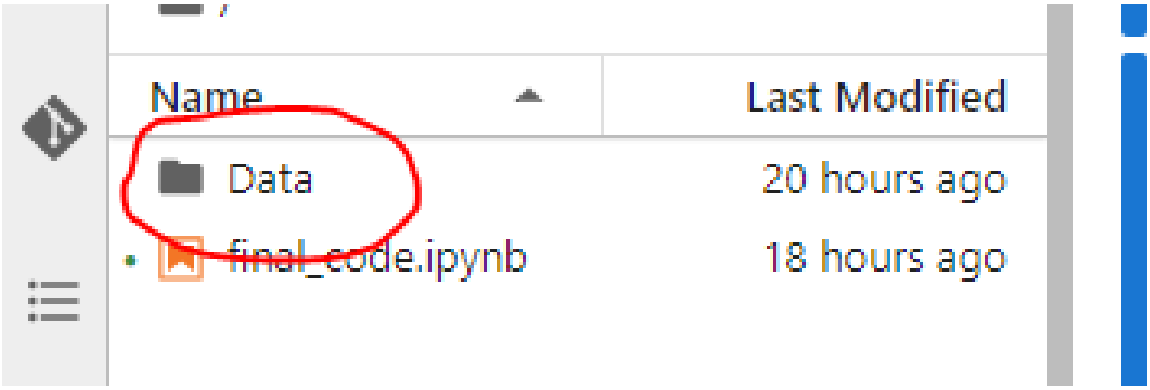

Figure 4: Data Folder

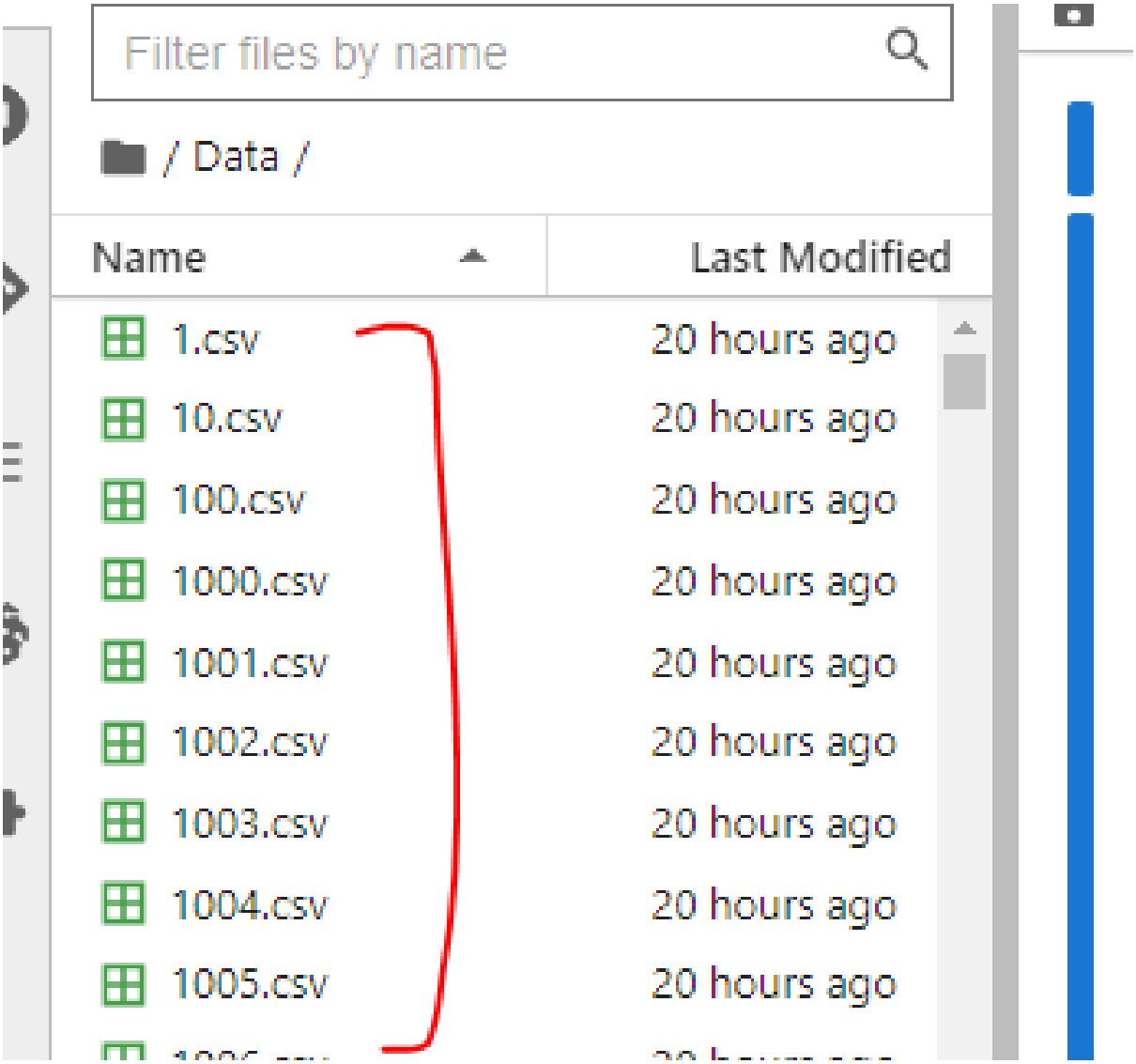

Figure 5: Files Upload

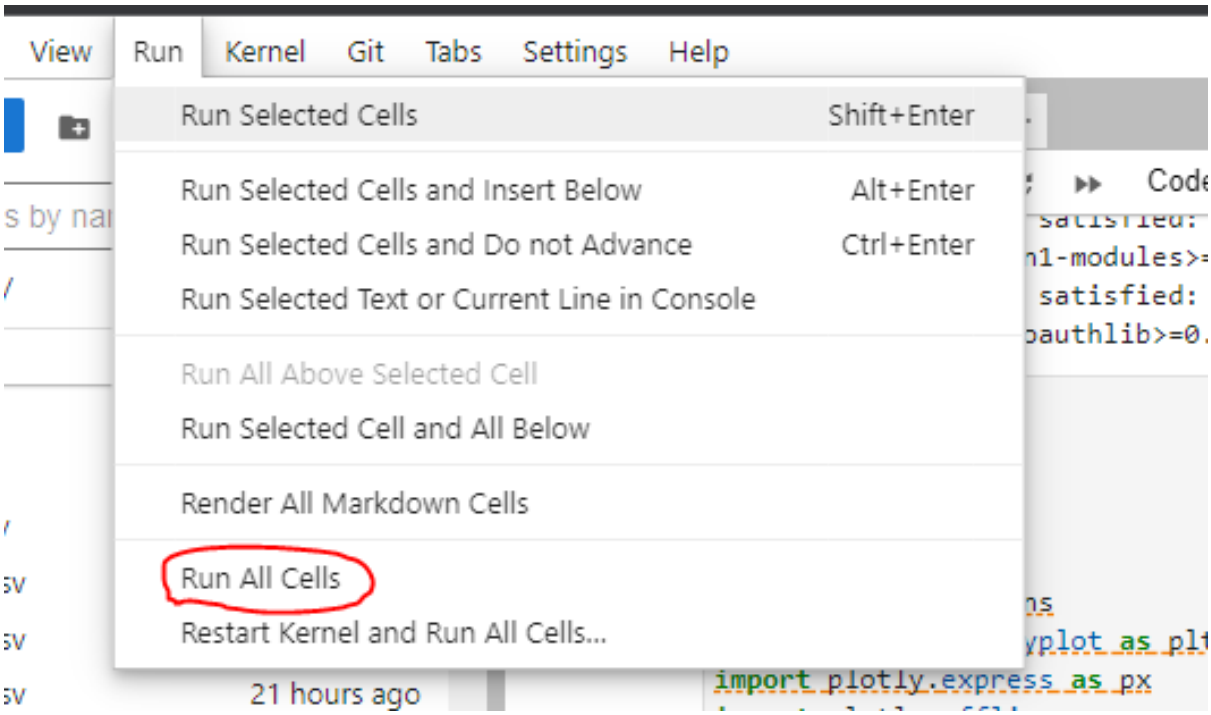

Figure 6: Running the Code

Figure 7: Data Collection

```
[8]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 919080 entries, 0 to 8272
Data columns (total 11 columns):
 #   Column                                Non-Null Count   Dtype
---  ------                                --------------   -----
 0   Timestamp [ms]                        919080 non-null  int64
 1   CPU cores                             919080 non-null  int64
 2   CPU capacity provisioned [MHZ]        919080 non-null  float64
 3   CPU usage [MHZ]                       919080 non-null  float64
 4   CPU usage [%]                         919080 non-null  float64
 5   Memory capacity provisioned [KB]      919080 non-null  float64
 6   Memory usage [KB]                     919080 non-null  float64
 7   Disk read throughput [KB/s]           919080 non-null  float64
 8   Disk write throughput [KB/s]          919080 non-null  float64
 9   Network received throughput [KB/s]    919080 non-null  float64
 10  Network transmitted throughput [KB/s] 919080 non-null  float64
dtypes: float64(9), int64(2)
memory usage: 84.1 MB
```

Figure 8: Data Description

```
[13]: #Feature Engineering
      df['Weekday'] = df['Timestamp'].dt.dayofweek
      df['Weekend'] = ((df.Weekday) // 5 == 1).astype(float)
      df['Month']=df.Timestamp.dt.month
      df['Day']=df.Timestamp.dt.day
      df['Year']=df.Timestamp.dt.year
```

Figure 9: Data Pre-Processing/Engineering

```
[16]: data = df[['Day', 'CPU usage [MHZ]']].groupby(['Day']).mean().sort_values('Day')
data = data.reset_index()
data.head()
fig = px.bar(data, x='Day', y='CPU usage [MHZ]',color='Day',title='Day Wise Average CPU usage [MHZ]')
fig.show()
```

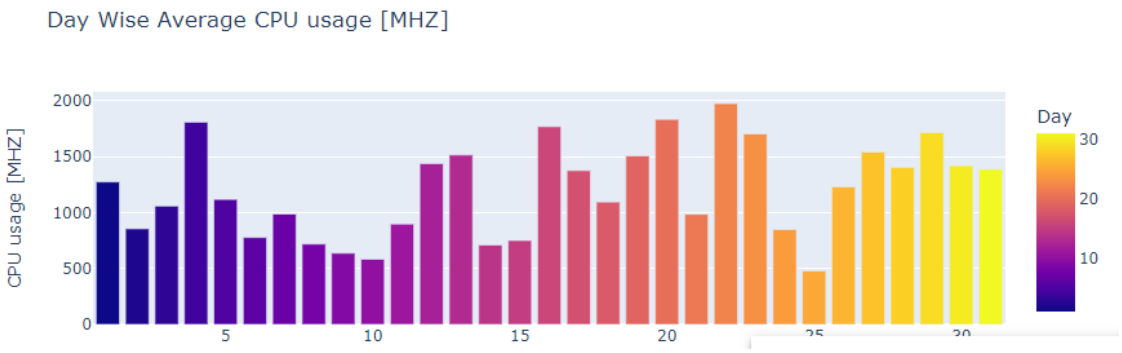

Figure 10: Data Visualisation

STACKED LSTM GRU MODEL

```
[35]: # Define the stacked LSTM GRU model.
model = Sequential()

number_units = 9
dropout_fraction = 0.2
#Stacked LSTM with GRU
model.add(GRU(
    units=number_units,
    return_sequences=True,
    input_shape=(X_train.shape[1], 1))
    )
# Layer 2
model.add(LSTM(
    units=number_units,
    return_sequences=True,
    input_shape=(X_train.shape[1], 1))
    )

model.add(Dropout(dropout_fraction))
# Layer 3
model.add(LSTM(units=number_units, return_sequences=True))
model.add(Dropout(dropout_fraction))
# Layer 4
model.add(LSTM(units=number_units))
model.add(Dropout(dropout_fraction))
```

Figure 11: Stacked LSTM GRU Model

#BILSTM Model

```
[42]: #BILSTM model
model = Sequential()
model.add(Bidirectional(LSTM(units=200, input_shape=(X_train.shape[1], 1),return_sequences=True)))
model.add(Dropout(0.2))
model.add(LSTM(units=100,return_sequences=True))
model.add(Dropout(0.2))
model.add(LSTM(units=50, return_sequences=True))
model.add(Dropout(0.2))
model.add(LSTM(units=50))
model.add(Dropout(0.5))
model.add(Dense(units=50, kernel_initializer='uniform', activation='tanh'))
model.add(Dense(units=1, kernel_initializer='uniform', activation='linear'))

# Compiling the model
model.compile(optimizer='adam',loss='mean_squared_error')
```


Figure 12: BILSTM Model

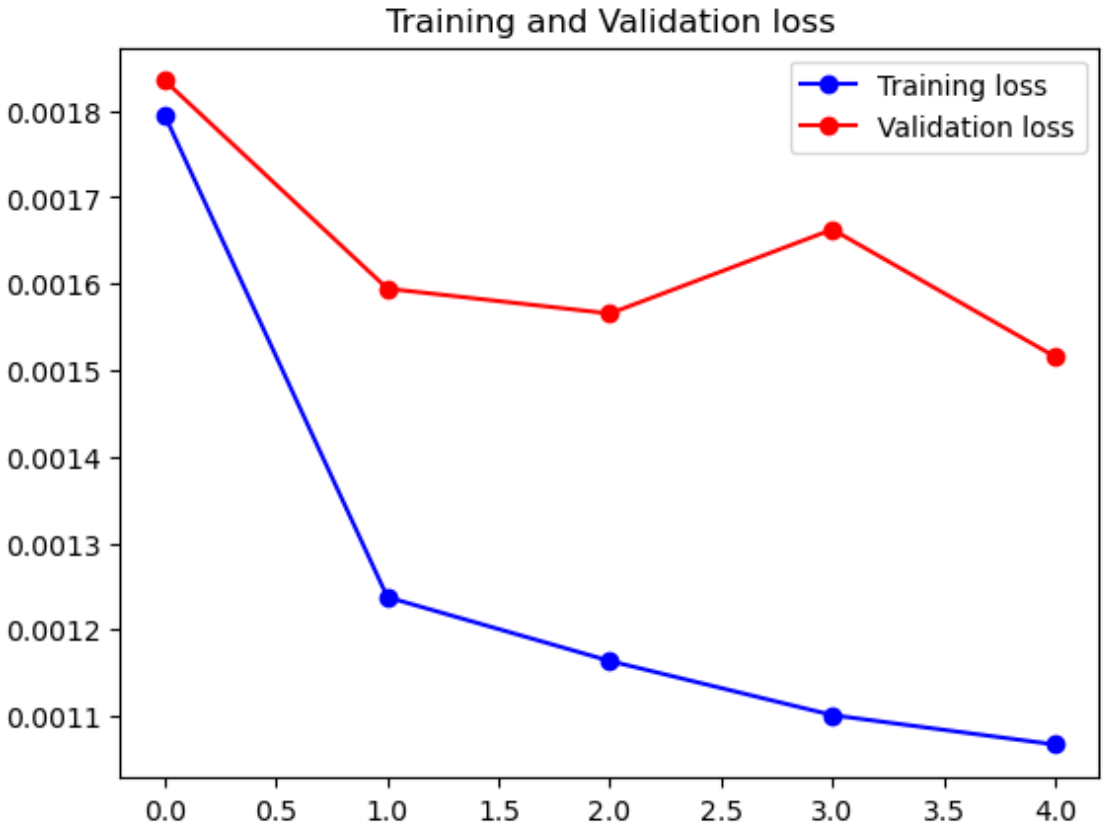

Figure 13: Model Validation

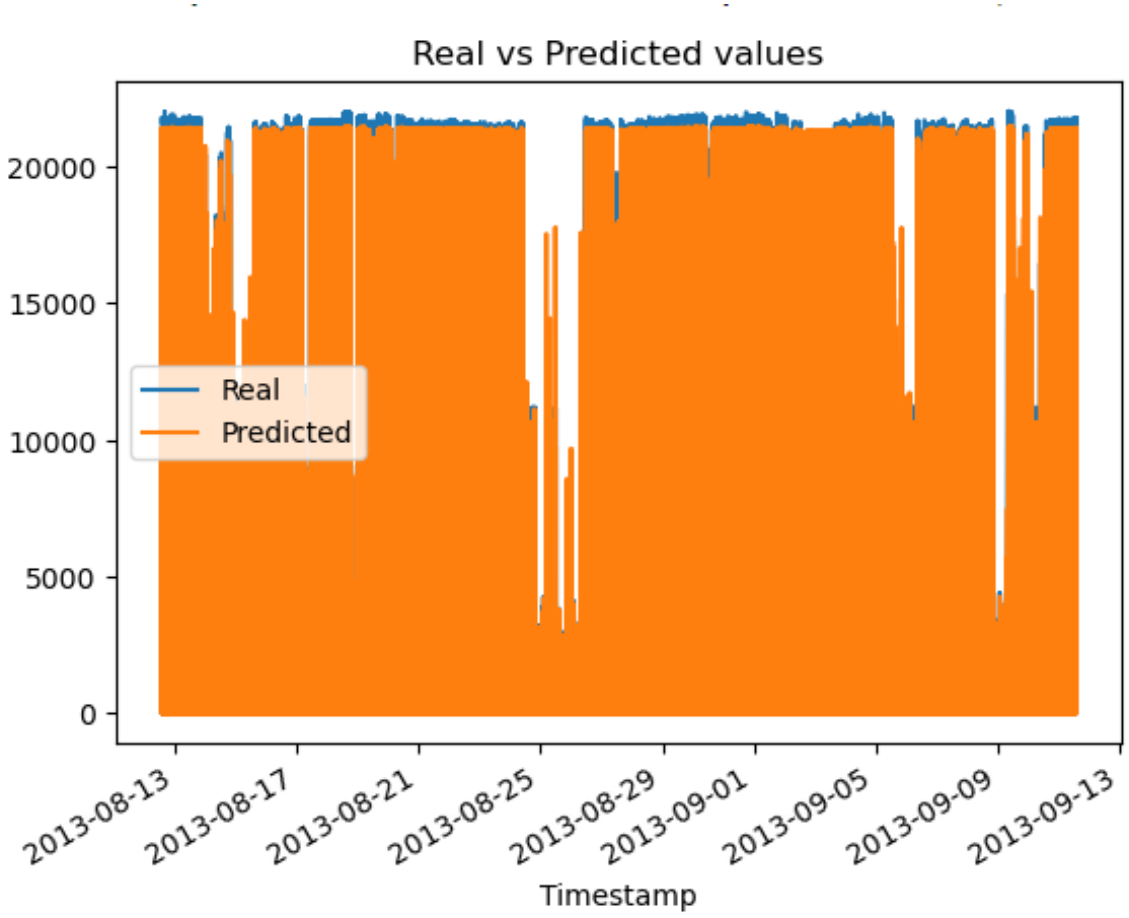

Figure 14: Model Evaluation

- Data Pre-Processing/Engineering - 9
- Data Visualisation - 10
- Stacked LSTM GRU Model Training - 11
- BILSTM Model Training - 12
- Model Validation - 13
- Model Evaluation - 14

5 Result

Results are shown in the table below 1:

Table 1: Comparison of Performance Metrics for BILSTM and STACKED LSTM GRU Models in Autonomous Cloud Resource Allocation.

Model	**Validation Loss**	**RMSE**	**R-squared**
BILSTM	8.6205e-04	0.0294	0.9743
STACKED LSTM GRU	9.1896e-04	0.0303	0.9726

References

Set Up Amazon SageMaker Prerequisites (n.d.).
URL: *https://docs.aws.amazon.com/sagemaker/latest/dg/gs-set-up.html*

Set Up JupyterLab in Amazon SageMaker (n.d.).
URL: *https://docs.aws.amazon.com/sagemaker/latest/dg/studio-jl.html*