National College of Ireland

Configuration Manual

MSc Research Project
Data Analytics

Karim Ladouari
Student ID: 0168195

School of Computing
National College of Ireland

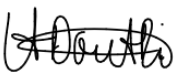
Supervisor: Mr Zahid Iqbal

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Karim Ladouari
Student ID:	20168195
Programme:	Data Analytics
Year:	2022
Module:	MSc Research Project
Supervisor:	Mr Zahid Iqbal
Submission Due Date:	15/12/2022
Project Title:	Configuration Manual
Word Count:	1420
Page Count:	4

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:	
Date:	12th December 2022

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):	

Configuration Manual

Karim Ladouari
0168195

1 Introduction

The research question for this survey is defined as what is, based on a selection of commonly known methods, the most optimal forecasting technique combined with feature selection to predict cloud resource workload. The configuration manual steps match the experimentation steps defined to observe the research experiment outcome or effect of the variables and data manipulation. The experimentation scenario uses genuine data from CompanyA. The configuration manual is designed as follows. Section 2 describes the sequence of execution of the experimentation notebooks. Section 3 outlines the experimentation system and software specifications, and Section 4 describes the data extraction steps. Section 5 discusses the data preprocessing, Section 6 the implementation of the feature selection and Section 7 models fitting, predictions and the associated evaluation.

2 Experimentation Notebooks

Dedicated designed Jupyter notebooks represent each implementation step. The sequence of execution of the notebooks is described in Figure 1.

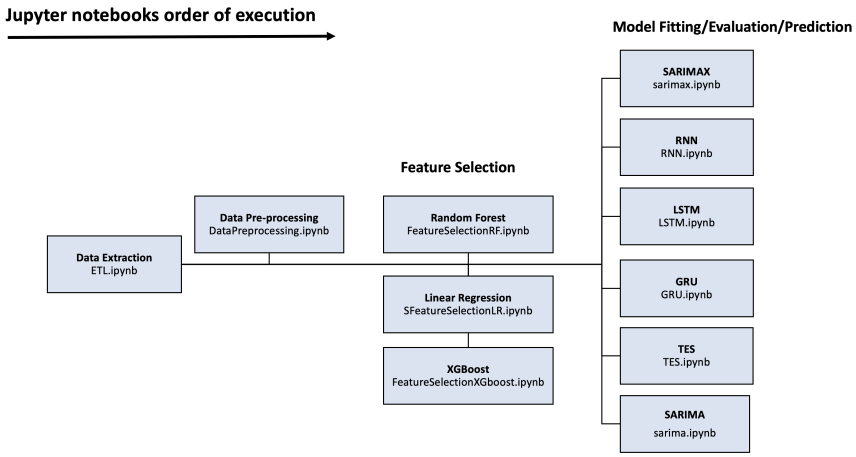


Figure 1: Jupyter Notebooks order of execution

3 System specifications

The laptop's hardware used for this experiment is an Apple MAC with a 2.6GHz 6 Intel Cores i7 processor, a 512GB SSD hard disk and 16GB of 2400MHz DDR4 memory.

The laptop operating system is MacOS Ventura version 13.0.1 and the web browser is Google Chrome version 107.0.5304.110. The experiment is developed using Python programming language version 3.9.12 on the Jupyter notebook web-based interactive development platform running on the Anaconda Navigator development framework 2.2.

4 Data Extraction

The Cloud systems resources (CPU and RAM) and application data are recorded on CompanyA's monitoring system hosted by New Relic SaaS. The data is stored hourly and categorised by customers, application services and resources workload. The notebook ETL.ipynb is designed to extract the data required for the experimentation from the monitoring system. The python code from the ETL.ipynb notebook is constructed in two parts. The first part is dedicated to extracting the application data and consists of the etlapp1, etlapp2 and etlapp3 functions and their associated function calls. The second part is dedicated to the infrastructure data and, more specifically, the cloud systems consumption (CPU and memory) and consists of the etlcpu function and its associated function call. etlapp1, etlapp2 and etlapp3 were designed to extract integration, integration2, service, API, agent, total and transaction calls.
Each function calls require a set of variables to run and complete, as described in Figure 2.

Function variable	quer	file_ext	selected_month	selected_days	dir_ext	totalvalue	val
Description	NRQL query required to extract specific data	CSV file extension for type of extraction (e.g. agent, api..). The format is a string.	Integer between 1 and 12 as function extract data on monthly basis	Integer for days in the selected month plus 1	The directory name where csv file is saved . Directory need to exist on hardisk prior running function. The format is a string	Only applicable to gen_etl_app1 and gen_etl_app2. totalvalue is the saved CSV file total column name.The format is a string.	Only applicable to etl_cpu. val is the variable used to specify on what CPU or RAM sum should be based on. Values are "max", "min" or "average". The format is a string.
Example	service_call_query	"service_calls"	10	3	"ServiceCalls"	"service"	"max"

Figure 2: Python functions variables requirement

After pulling the data and converting the JSON data format into Pandas data frame, the code programmatically saves the extracted application and infrastructure data as a CSV file for each customer on corresponding laptop hard disk directories named AgentCalls, TransactionCalls, IntegrationCalls, Integration2Calls, ApiCalls, ServiceCalls, TotalCalls and CpuMax. Additionally, each function from ETL.ipynb requires, to run and complete successfully, an Nrql Rest API key (api_key), a new relic account id (account_id) and an Nrql query, including a start and end date. However, to ensure confidentiality and data privacy, the NewRelic API keys, the account details, the queries and extracted data cannot be shared.

5 Data Pre-processing

Data pre-processing involves cleaning and preparing the data for feature selection and model fitting. The notebook DataPreprocessing.ipynb was designed to aggregate all previously extracted files, split them into individual files per server and impute any missing data points. All programming sections from DataPreprocessing.ipynb can be run in

sequence starting from the first. However, as a preliminary task, the experimentation selected customer-extracted files need to be gathered together in one directory and specified in the following first line of code.

```
#Laptop Directory where extracted files for one customer are gathered
os.chdir(r"/Users/karimladouari/MyDocuments/Master/Research Project/data_files")
```

Figure 3: Selected customer extracted files location code

6 Data Selection - Feature Selection

The data selection or feature selection is the process of isolating the most influential and relevant features for the selected server data set model fitting. Three Jupyter notebooks were created for this section, one of each feature selection technique used for the experiment. FeatureSelectionLR.ipynb is the notebook dedicated to Linear regression feature selection, FeatureSelectionRF.ipynb for Random Forest and FeatureSelectionXGboost.ipynb for XGBoost. Executing each feature selection notebook will produce the list of the most contributing features to be then used to fit each time series forecasting technique. Code sections from FeatureSelectionLR.ipynb are executed sequentially, starting with loading the BestNormalize class, which is used in the primary function GenFeatSelect for applying the best normalizing transformation method to the dataset. Executing then GenFeatSelect generates the list of the most contributing features but requires assigning to the field serv the experiment selected server CSV file name. Finally, the last part from FeatureSelectionLR.ipynb produced linear regression diagnostics plots for the selected features. FeatureSelectionRF.ipynb and FeatureSelectionXGboost.ipynb require only the function GenerateSelection to run to generate the best features, and like FeatureSelectionLR.ipynb requires assigning to the field serv the experiment selected server CSV file name.

7 Data Modeling, Evaluation and Prediction

Data Modelling involves fitting and training models using selected algorithms. Six Jupyter notebooks were designed to fit the selected models, build the prediction and evaluate the forecasting and predictions. neural networks.

7.1 Neural Networks

Three notebooks RNN.ipynb, LSTM.ipynb and GRU.ipynb are dedicated to neural networks. The structure of the three notebooks is the same, with only the algorithms changing. The notebooks are divided into four sections, starting with the section where all the functions are declared and initiated. For each selected feature selection, the following section (2.Model Fitting) is where the hyperparameter is compiled, and the most optimal models are fitted and evaluated. It is required to assign values to the variables server, col, and loopback before running the hyperparameter and the model fitting. The variable server is the selected server filename for analysis, and col is the aggregation of the dependent variables and the variables selected by the feature selection. Finally,

the loopback is set to three for the number of previous observations combined with the current one as the input. The assignment is performed just before each hyperparameter execution section and only once per feature selection. Additionally, it is required to fit and plot the two or three best models from the hyperparameter to eliminate any with over or underfitting.
The following section (3.Optimal Mode) is the part from the notebook where the most optimal model among the best models from the three feature selection techniques is selected (best RMSE and MAE scores and negligible over and underfitting), saved and loaded again. The requirements for this section are to select a directory location to save the model and, like the previous sections, assign the associated values to the variables server, col, and loopback.
Finally, the last section (C. Predictions) is the part of the notebook where the predictions are compiled and evaluated using unseen data. The variable col is the optimal model dependent and independent features, and the server variable is the unseen data time series filename, Server1_predict.csv for the experiment.

7.2 SARIMAX and SARIMA

The notebook Sarimax.ipynb was designed for SARIMAX model fitting and prediction. Like the neural network notebooks, Sarimax.ipynb programming sections are sequentially executed. The first section is dedicated to declaring and loading all the required functions. The following section (2.Model Fitting) is where the hyperparameter is compiled, and the most optimal model is fitted and evaluated. The variables dataset, col, lag, dependant_var, independant_var, train and test require values to be assigned before execution. The variable server is the filename for the selected server for analysis, and col is the aggregation of the dependent variables and the variables selected by the feature selection. The lag variable is the length of time steps for time series forecasting. Dependant_var, independant_var, train and test are datasets made of the dependent and the selected independent variables. Finally, for the last section (3.Predictions using optimal model), server2 is the unseen data time series filename, and col is the best model independent variable.
The notebook Sarima.ipynb designed for SARIMA model has the same structure as Sarimax.ipynb. The distinction between the two is that Sarima.ipynb contains only one model, as SARIMA applies to univariate time series and subsequently has no independent variable or associated feature selection. The server file name and the dependent variable are the unique variables for this notebook.

7.3 Triple Exponential smoothing

The notebook TES.ipynb was designed for Triple Exponential Smoothing model fitting and prediction. The notebook can be sequentially executed but requires variables server, depvar, lar and server2. The variable server is the selected server filename, depvar is the independent variable, lag is the number of forecasted time steps, and server2 is the server unseen data filename.