# Configuration Manual

MSc Research Project
MSc Fintech

Alexandra Oluwaseun Olaitan Scott
Student ID: x21211809

School of Computing
National College of Ireland

Supervisor: Theo Mendonca

**National College of Ireland**

**MSc Project Submission Sheet**

**School of Computing**

**Student Name:** …….ALEXANDRA OLUWASEUN OLAITAN SCOTT ……………………………….

**Student ID:** ……21211809……………………………………………………………………………..……

**Programme:** ……MSC FINTECH……………………………… **Year:** …2022 / 2023.

**Module:** …RESEARCH PROJECT………………………………………………………….………

**Lecturer:** …THEO MENDONCA…………………………………………………………….………

**Submission Due Date:** …MONDAY, AUGUST 14 2023………………………………………….………

**Project Title:** …… EXPLORING THE APPLICATION OF NEURAL NETWORK WITH BACK PROPAGATION IN DETECTING FRAUDULENT TRANSACTIONS WITHIN THE ETHEREUM NETWORK …………………………………………………….………

**Word Count:** ……652…………………… **Page Count:** ………6……………………………......

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:** ……ALEXANDRA SCOTT…………………………………………………………………

**Date:** ……14 / 08 / 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): | |

# Configuration Manual

Alexandra Oluwaseun Olaitan Scott
Student ID: 21211809

## 1 Introduction

This configuration manual contains extensive details of all techniques and technical details which were employed in carrying out the research analysis

## 2 System Requirements

### 2.1. Hardware Requirements:

- Mac MacBook Pro (Retina, 13-inch, Early 2015)
- Processor: 2.7 GHz Dual-Core Intel Core i5
- Memory 8 GB 1867 MHz DDR3
- Storage: 128gb
- Graphics: Intel Iris Graphics 6100 1536 MB

### 2.2. Software Requirements:

- macOS Monterey Version 12.6.5 (21G531)
- R 4.2.1 GUI 1.79 High Sierra build
- R Studio Version 2023.06.1+524
- Microsoft Word version 16.75
- Oracle Java Version 20.0.2, 2023-07-18

## 3 Data

The data used is called 'Ethereum Fraud detection dataset' and was obtained from Kaggle – https://www.kaggle.com/datasets/vagifa/ethereum-frauddetection-dataset/code?datasetId=1074447&sortBy=voteCount&searchQuery=r

## 4 Analysis

### 4.1. Package installation on RStudio

The following packages were installed on RStudio to enable the user carry out all the techniques subsequently discussed.

| Package Name | Application |
|---|---|
| mlbench | To compare performance |
| dplyr | Enables data manipulation and transformation |
| ggplot2 | For data visualisation |
| lattice | For visualisation purposes |
| caret | Training and tuning model and building the confusion matric |
| tidyverse | Data Manipulation and visualisation |
| e1071 | To perform the Support Vector Machine |
| plotly | Data visualisation - To create box plots |

| | |
|---|---|
| smotefamily | To handle the class imbalance using SMOTE |
| Amelia | To handle and visualise missing values in the dataset |
| matrixStats | Applied to rows and columns |
| neuralnets | To implement the neural network with back propagation algorithm |
| keras | Training neural networks |
| h2o | To perform the XGBoost model |

### 4.2. Data Preparation

### 4.2.1 Importing the dataset

The dataset which was stored as a .csv file on the users desktop was imported into the RStudio Environment and saved as ‘Ethereum’

```
Ethereum <- read.csv("transaction_dataset.csv", header = T , na.strings =  c(""), stringsAsFactors = T)
```

### 4.2.2. Handling and visualising any Missing values

- The sapply function is used to indicate how many missing values were observed in each of the columns within the Ethereum dataset
- These missing values were then visually represented using the missmap function and the output is shown in figure 1.

```
sapply(Ethereum, function(x) sum(is.na(x)))

missmap(Ethereum, main = "Missing values vs Observed")
```

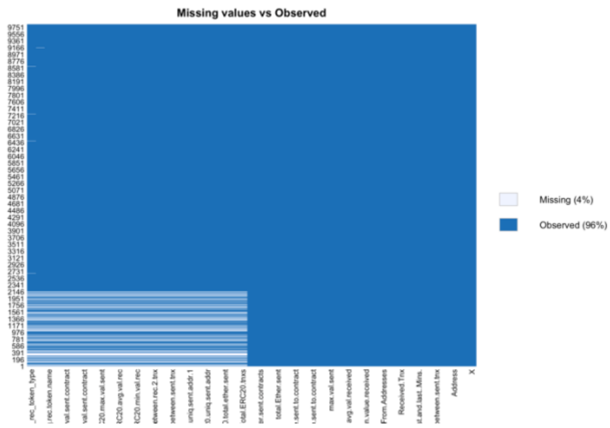


Figure 1: missmap of missing values within the Ethereum dataset

The missing values were then replaced with their median. To do these the last two categorical values were dropped from the dataset as the replacing with median was not feasible. These variables do not have any significant impact on the dataset.

They were put into a dataframe called ‘factor_columns_to_drop and then removed from the Ethereum dataset. Figure 2 shows the dataset without any missing values

```
factor_columns_to_drop <- c("ERC20.most.sent.token.type","ERC20_most_rec_token_type")

Ethereum <- Ethereum %>%
  select(-one_of(factor_columns_to_drop))
```

```
medians <- sapply(Ethereum[, c("Total.ERC20.tnxs","ERC20.total.Ether.received",
"ERC20.total.ether.sent","ERC20.total.Ether.sent.contract","ERC20.uniq.sent.addr",
"ERC20.uniq.rec.addr","ERC20.uniq.sent.addr.1","ERC20.uniq.rec.contract.addr","ERC20.avg.time.betwe
en.sent.tnx","ERC20.avg.time.between.rec.tnx","ERC20.avg.time.between.rec.2.tnx","ERC20.avg.time.betw
een.contract.tnx","ERC20.min.val.rec","ERC20.max.val.rec","ERC20.avg.val.rec","ERC20.min.val.sent","E
RC20.max.val.sent","ERC20.avg.val.sent","ERC20.min.val.sent.contract","ERC20.max.val.sent.contract",
"ERC20.avg.val.sent.contract","ERC20.uniq.sent.token.name","ERC20.uniq.rec.token.name")], median,
na.rm = TRUE)
```

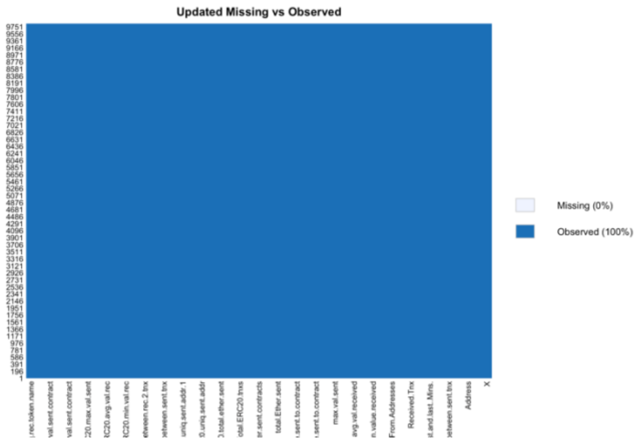

Figure 2: missmap showing no missing values within the Ethereum dataset

### 4.3. Data Visualisation

Using the plotly, dplyr, ggplot2 and DataExplorer libraries the following data visualisations were implemented

- Boxplots
- Heatmap of correlation matrix
- Barchat showing the distribution of fraud vs valid transactions

```
fig <- plot_ly(y = ~Ethereum$Avg.min.between.sent.tnx, type = "box")

DataExplorer::plot_correlation(Ethereum)

FraudBarChart <- ggplot(bar_data, aes(x = Transaction_Type, y = Count, fill = Transaction_Type))
+
  geom_bar(stat = 'identity', width = 0.6) +
  theme_minimal() +
  labs(title = "Distribution of Fraud vs. Non-fraud Transactions",
       x = "Transaction Type",
      y = "Count")
```

### 4.4 Handling Data imbalance

Using Synthetic Minority Over Sampling Technique (“SMOTE”), the data imbalance was handled shown in figures 3 and 4 which contain the data distribution before and after balancing respectively.

```
smotetrain <- SMOTE(TrainingData[,-1], TrainingData$FLAG)

BalancedTrainingData <- smotetrain$data

table(BalancedTrainingData$class)
```

# 5 Modelling

### 5.1. Splitting the dataset into a training and testing

The Ethereum datasets was split into a training and testing dataset named: 'TrainingData' and 'TestingData' respectively using an 80/20 ratio

```
TrainIndex <- createDataPartition(Ethereum$FLAG, p = 0.8, list = FALSE)

TrainingData <- Ethereum[TrainIndex, ]

TestingData <- Ethereum[-TrainIndex, ]
```

The data was further split into the feature (x) and the target variable (y)

```
X <- BalancedTrainingData[, !names(BalancedTrainingData) %in% c("FLAG")]

y <- BalancedTrainingData[, "FLAG"]

test_X <- TestingData[, !names(TestingData) %in% c("FLAG")]
```

Once these steps have been carried out, the proposed machine learning algorithms can be developed.

5.2.1. Support Vector Machine (SVM)
5.2.2 K-Nearest Neighbour ("KNN")
5.2.3. XGBoost
5.2.4. Neural Network with back propagation