National College of Ireland

Configuration Manual

MSc Research Project

MSc in Cybersecurity

Sujay Hegde

Student ID: 20174217

School of Computing

National College of Ireland

Supervisor: Liam McCabe

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Sujay Hegde
Student ID:	20174217
Programme:	MSc Cybersecurity **Year:** 2021
Module:	Research Project
Lecturer:	Liam McCabe
Submission Due Date:	16/12/2021
Project Title:	Identification of Dominant Spam Email Features to Improve Detection Accuracy of Machine Learning Algorithms
Word Count:	572 **Page Count:** 9

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: 16/12/2021

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

Sujay Hegde
Student ID: 20174217

1. Introduction

The configuration manual's goal is to assist users in installing the research project code on their system so that they can use it to evaluate the study or modify it to meet their specific needs. The prerequisites and environment set up section provides comprehensive guidance for creating a project environment as well as a list of requirements for replicating the research results.

2. Requirements

2.1 System Requirement

The process of machine learning involves overhead of resources on the host machine.
Hence, it is critical that the hardware configuration on the employed machine be capable of doing such tasks. The following are the system's minimum requirements:

- CPU: Intel i5 6th Gen or Intel i7 5th Gen Processor with 2.4 GHZ
- RAM: 8gb DDR4
- Storage: 15 GB of free space HDD or SSD

2.2 Software Requirements

- Python 3.x recommended
- Web Browser

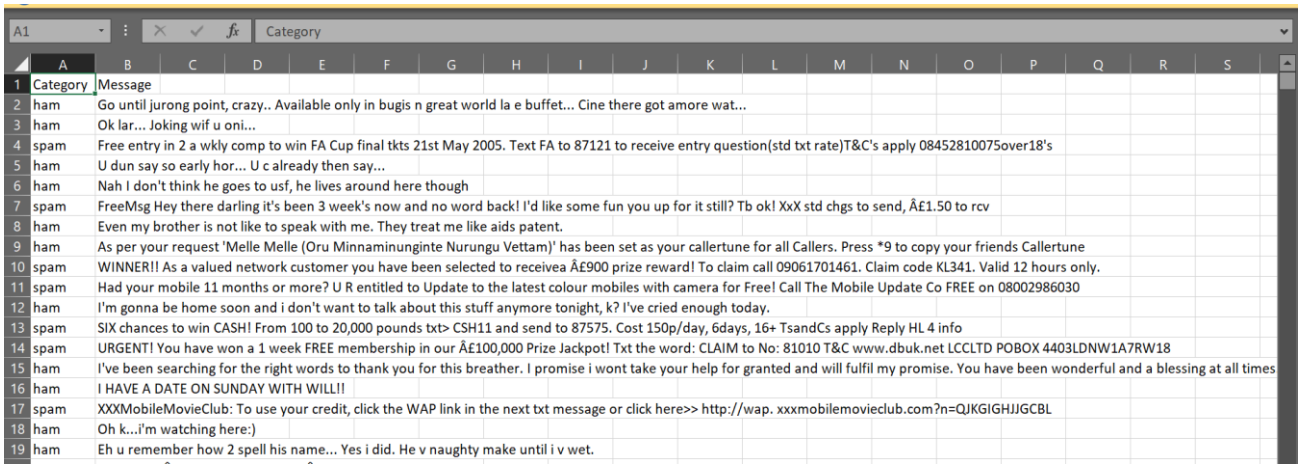
2.3 Imports and Packages

- Nltk 2.0.2
- Scikit-learn 0.24
- Numpy 1.19.2
- Pandas
- Seaborn
- Tqdm
- Matplotlib 3.5.1
- Tensorflow 2.7.0
- Re
- Keras
- Sklearn.metrics
- Sklearn.tree

The project makes use of google colab for runtime environments.
The .ipynb file needs to imported.

3. Dataset Collection

The project uses two datasets is CSV format. The dataset found at https://archive.ics.uci.edu/ml/datasets/spambase contains spam or ham dictionary words. The screenshot below shows a preview of the dataset in excel.



	A	B
1	Category	Message
2	ham	Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat...
3	ham	Ok lar... Joking wif u oni...
4	spam	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question(std txt rate)T&C's apply 08452810075over18's
5	ham	U dun say so early hor... U c already then say...
6	ham	Nah I don't think he goes to usf, he lives around here though
7	spam	FreeMsg Hey there darling it's been 3 week's now and no word back! I'd like some fun you up for it still? Tb ok! XxX std chgs to send, Â£1.50 to rcv
8	ham	Even my brother is not like to speak with me. They treat me like aids patent.
9	ham	As per your request 'Melle Melle (Oru Minnaminunginte Nurungu Vettam)' has been set as your callertune for all Callers. Press *9 to copy your friends Callertune
10	spam	WINNER!! As a valued network customer you have been selected to receivea Â£900 prize reward! To claim call 09061701461. Claim code KL341. Valid 12 hours only.
11	spam	Had your mobile 11 months or more? U R entitled to Update to the latest colour mobiles with camera for Free! Call The Mobile Update Co FREE on 08002986030
12	ham	I'm gonna be home soon and i don't want to talk about this stuff anymore tonight, k? I've cried enough today.
13	spam	SIX chances to win CASH! From 100 to 20,000 pounds txt> CSH11 and send to 87575. Cost 150p/day, 6days, 16+ TsandCs apply Reply HL 4 info
14	spam	URGENT! You have won a 1 week FREE membership in our Â£100,000 Prize Jackpot! Txt the word: CLAIM to No: 81010 T&C www.dbuk.net LCCLTD POBOX 4403LDNW1A7RW18
15	ham	I've been searching for the right words to thank you for this breather. I promise i wont take your help for granted and will fulfil my promise. You have been wonderful and a blessing at all times
16	ham	I HAVE A DATE ON SUNDAY WITH WILL!!
17	spam	XXXMobileMovieClub: To use your credit, click the WAP link in the next txt message or click here>> http://wap. xxxmobilemovieclub.com?n=QJKGIGHJJGCBL
18	ham	Oh k...i'm watching here:)
19	ham	Eh u remember how 2 spell his name... Yes i did. He v naughty make until i v wet.

The Enron 1 dataset is around 364 megabytes and expands to 1.32 Gigabytes once unzipped by the colab notebook. This dataset contains email header information. The dataset can be found at http://www.cs.columbia.edu/~rambow/enron/

4. Code Evaluation

Deep Learning Model

Defining Patterns(grouping all data using regex function)

```
# Define the patterns:
xfn_pattern = re.compile(r'%s.*$'%metadata_names[-1], re.M) # X-FileName: is the last one in the metadata_names list
content_pattern = re.compile('[^\n].*', flags=re.S) # not a newline followed by any number of any characters

# Test the patterns:
xfn_end = xfn_pattern.search(data.message[10]).end() # Find the X-FileName information and take the index of its last character
match = content_pattern.search(data.message[10], pos=xfn_end) # Start searching from this index
print(match.group())
```

Tokenizing and defining neural network parameters

```
# Maximum number of words, we are going to embed
max_features = 2048

# Number of embedding dimensions in the word embedding space constructed by the embedding layer
embed_dim = 256

# The length of the sequence (message) - The longer messages will be cropped to 256, while the shorter ones
maxlen = 256

# Take messages from both classes and shuffle them before feeding to tokenizer
messages_all = messages_0+messages_1
np.random.shuffle(messages_all)

# The tokenizer ascribes a number to each token (word) in the sequence
tokenizer = Tokenizer(num_words=max_features)
tokenizer.fit_on_texts(messages_all)
word_index = tokenizer.word_index # This dictionary translates each word to its index (corresponding number)
```

Defining model parameters

```
model = models.Sequential(layers=[
    layers.Embedding(input_dim=max_features, output_dim=embed_dim, input_length=maxlen),
    layers.Bidirectional(layers.GRU(32, activation='relu', return_sequences=True, dropout=.1, recurrent_dropout=.1)),
    layers.Bidirectional(layers.GRU(32, activation='relu', return_sequences=False, dropout=.1, recurrent_dropout=.1)),
    layers.Dense(64, activation='relu', kernel_regularizer='l2'),
    layers.BatchNormalization(),
    layers.Dropout(.2),
    layers.Dense(32, activation='relu', kernel_regularizer='l2'),
    layers.BatchNormalization(),
    layers.Dropout(.1),
    layers.Dense(1, activation='sigmoid')
])
model.summary()
```

Training the model; Epochs = 24

```
# Model training with the data
callbacks_list = [
    callbacks.ModelCheckpoint('best_model.h5', monitor='val_loss', save_best_only=True, save_freq='epoch'),
    callbacks.ReduceLROnPlateau(monitor='val_loss', factor=.2, patience=5), # Reduce the learning rate by a
    callbacks.EarlyStopping(patience=10) # Stop training after 10 epochs of no validation loss reduction
]

model.compile(
    optimizer='rmsprop',
    loss='binary_crossentropy',
    metrics=['acc']
)

EPOCHS = 24

history = model.fit(
    train_X, train_y,
    validation_data = (val_X, val_y),
    epochs = EPOCHS, batch_size=64,
    shuffle = True,
    verbose = 1,
    callbacks = callbacks_list
```

Classifier Models

Naïve Bayes and Decision Tree

Label Encoding and Tokenizing

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
train_y = le.fit_transform(target_train.values)
test_y = le.transform(target_test.values)

train_y
```

```
from keras.preprocessing.text import Tokenizer
tokenizer = Tokenizer(num_words=max_feature)

tokenizer.fit_on_texts(x_train)

x_train_features = np.array(tokenizer.texts_to_sequences(x_train))
x_test_features = np.array(tokenizer.texts_to_sequences(x_test))

x_train_features[0]
```

Model Training

```
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(x_train_features, train_y)

# Predicting the Test set results
y_pred = classifier.predict(x_test_features)
```

```
from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier(criterion = 'entropy', random_state = 0)
classifier.fit(x_train_features, train_y)
import matplotlib.pyplot as plt

# Predicting the Test set results
y_pred = classifier.predict(x_test_features)
```