National College of Ireland

Configuration Manual

Industry Internship
MSc Cybersecurity

Oluwafunsho John Alabi
Student ID: X22126899

School of Computing
National College of Ireland

Supervisor: Vikas Sahni

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Oluwafunsho John Alabi

Student ID: X22126899

Programme: MSc. Cybersecurity **Year:** 2023

Module: Industry Internship

Lecturer: Vikas Sahni

Submission Due Date: 5th January 2024

Project Title: A Resilient NLP-Based Detection System of Phishing Emails leveraging Deep Learning Techniques

Word Count: 884 **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: Oluwafunsho Alabi

Date: 5th January 2024

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):	

A Resilient NLP-Based Detection System of Phishing Emails leveraging Deep Learning Techniques

Oluwafunsho Alabi
X22126899

1.0 INTRODUCTION

Project Overview: This manual guides the setup and execution of a Python-based Natural Language Processing (NLP) system for detecting phishing emails in Google Colab.

Intended Audience: Individuals interested in replicating the phishing email detection analysis, irrespective of technical background.

2.0 System Requirements

Hardware: A typical PC with internet connectivity is required.

Software: A computer with a browser is required to access Google Colab.

3.0 Configuration of the Environment

Launch Google Colab on your web browser.

Import the Python notebook file provided: Select File > Upload Notebook.

4.0 Data Configuration

Adjust folders and load data by following the directions in the notebook.

Execute the data setup code blocks.

5.0 Code Execution

Run the code blocks in the order shown in the notebook.

Before proceeding to the next block, ensure that the previous one running has been completed.

6.0 Data Analysis and Visualization

Run sections for data cleaning, transformation, and analysis.

Within the notebook, use visualisation tools such as Matplotlib and Seaborn to depict data.

Before moving forward, ensure that you understand each step and outcome.

7.0 Model Evaluation and Results Analysis

Look at the outputs closely and compare evaluation scores on metrics like F1 recall, precision etc. from the various models.

8.0 Tools

- ***Pandas****:* Employed for data manipulation and analysis, as widely recognized in the data science community (McKinney and Team, 2015).
- ***NumPy****:* Integral for numerical computations, its utility in handling large arrays and matrices is well-documented (Taye, 2023).
- ***Scikit-learn***: A comprehensive tool for implementing various machine learning algorithms and metrics (Bischl *et al*., 2023).
- ***Matplotlib and Seaborn****:* These libraries facilitated data visualization, essential for interpreting complex data and models (Park *et al*., 2020).
- ***NLTK****:* Utilized for NLP tasks, especially stop words removal, aligning with the techniques discussed in Egozi and Verma (2018).

9.0 Screenshots and Visual Guide

How to Run Code Blocks Below: Click the play button next to each code block to execute it.

Import python modules

```
import itertools
import time
import numpy as np
import pandas as pd
import os
import matplotlib.pyplot as plt
from matplotlib import pyplot
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import GridSearchCV

from sklearn.preprocessing import RobustScaler

from sklearn import tree
from sklearn.ensemble import AdaBoostClassifier
from sklearn.neural_network import MLPClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC

from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score
from sklearn.metrics import roc_curve
```

Load data as pandas dataframe

```
data_df = pd.read_csv('/content/fraud_email_.csv') # change file path as needed!
```

```
[ ]
data_df = data_df.dropna()

data_df['Text'] = data_df['Text'].apply(lambda x: " ".join(x.lower() for x in x.split()))
data_df['Text'] = data_df['Text'].str.replace('[^\w\s]','')

stop = stopwords.words('english')
data_df['Text'] = data_df['Text'].apply(lambda x: " ".join(x for x in x.split() if x not in stop))


from sklearn.feature_extraction.text import TfidfVectorizer
corpus = data_df['Text']
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(corpus)


print(X.shape)
```

```
<ipython-input-3-d6bd2072152a>:4: FutureWarning: The default value of regex will change from True to False in a future version.
  data_df['Text'] = data_df['Text'].str.replace('[^\w\s]','')
(11928, 130424)
```

Exploratory Data Analysis

```
1 data_df.head()
```

	Text	Class
0	supply quality chinas exclusive dimensions unb...	1
1	sidlet know thx	0
2	dear friendgreetings youi wish accost request ...	1
3	mr cheung puihang seng bank ltddes voeux rd br...	1
4	surprising assessment embassy	0

```
[ ]  1 data_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 11928 entries, 0 to 11928
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Text    11928 non-null  object
```

Decision Trees

```
1 import time
2 from sklearn import tree
3 from sklearn.metrics import roc_auc_score
4 from sklearn.model_selection import train_test_split
5 import numpy as np
6
7
8 max_depth = 15
9 tree_auc_train, tree_auc_test = np.zeros(max_depth), np.zeros(max_depth)
10 training_time, prediction_time = np.zeros(max_depth), np.zeros(max_depth)
11
12 for i in range(1, max_depth):
13     clf_decision_tree = tree.DecisionTreeClassifier(max_depth=i, criterion='entropy', random_state=1)
14     t0 = time.perf_counter()
15     clf_decision_tree = clf_decision_tree.fit(train_X, train_y)
16     training_time[i] = round(time.perf_counter() - t0, 3)
17     tree_auc_train[i] = roc_auc_score(train_y, clf_decision_tree.predict_proba(train_X)[:, 1])
18     t1 = time.perf_counter()
19     tree_auc_test[i] = roc_auc_score(test_y, clf_decision_tree.predict_proba(test_X)[:, 1])
20     prediction_time[i] = round(time.perf_counter() - t1, 3)
21
```

Decision tree pruning

```python
from sklearn.tree._tree import TREE_LEAF

def is_leaf(inner_tree, index):
    # Check whether node is leaf node
    return (inner_tree.children_left[index] == TREE_LEAF and
            inner_tree.children_right[index] == TREE_LEAF)

def prune_index(inner_tree, decisions, index=0):
    # Start pruning from the bottom - if we start from the top, we might miss
    # nodes that become leaves during pruning.
    # Do not use this directly - use prune_duplicate_leaves instead.
    if not is_leaf(inner_tree, inner_tree.children_left[index]):
        prune_index(inner_tree, decisions, inner_tree.children_left[index])
    if not is_leaf(inner_tree, inner_tree.children_right[index]):
        prune_index(inner_tree, decisions, inner_tree.children_right[index])

    # Prune children if both children are leaves now and make the same decision:
    if (is_leaf(inner_tree, inner_tree.children_left[index]) and
        is_leaf(inner_tree, inner_tree.children_right[index]) and
        (decisions[index] == decisions[inner_tree.children_left[index]]) and
        (decisions[index] == decisions[inner_tree.children_right[index]])):
        # turn node into a leaf by "unlinking" its children
        inner_tree.children_left[index] = TREE_LEAF
        inner_tree.children_right[index] = TREE_LEAF
        print("Pruned {}".format(index))
```

Boosting

```python
import time
import numpy as np
from sklearn.ensemble import AdaBoostClassifier
from sklearn import tree
from sklearn.metrics import roc_auc_score
from sklearn.model_selection import train_test_split


max_depth = 15
adaboost_auc_train, adaboost_auc_test = np.zeros(max_depth), np.zeros(max_depth)
training_time, prediction_time = np.zeros(max_depth), np.zeros(max_depth)
for i in range(1, max_depth):
    clf_adaboost = AdaBoostClassifier(
        base_estimator=tree.DecisionTreeClassifier(max_depth=i, criterion='entropy'),
        n_estimators=10,
        random_state=1
    )
    t0 = time.perf_counter()
    clf_adaboost = clf_adaboost.fit(train_X, train_y)
    training_time[i] = round(time.perf_counter() - t0, 3)
    adaboost_auc_train[i] = roc_auc_score(train_y, clf_adaboost.predict_proba(train_X)[:, 1])
    t1 = time.perf_counter()
    adaboost_auc_test[i] = roc_auc_score(test_y, clf_adaboost.predict_proba(test_X)[:, 1])
    prediction_time[i] = round(time.perf_counter() - t1, 3)
```

Neural Networks

```python
import time
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import roc_auc_score
from sklearn.model_selection import train_test_split


learning_rate = [0.1, 1, 2, 3]
training_time, prediction_time = [], []
nn_auc_train, nn_auc_test = [], []

for rate in learning_rate:
    clf_nn = MLPClassifier(learning_rate_init=rate, random_state=1)
    t0 = time.perf_counter()
    clf_nn = clf_nn.fit(train_X, train_y)
    training_time.append(round(time.perf_counter() - t0, 3))
    nn_auc_train.append(roc_auc_score(train_y, clf_nn.predict_proba(train_X)[:, 1]))
    t1 = time.perf_counter()
    nn_auc_test.append(roc_auc_score(test_y, clf_nn.predict_proba(test_X)[:, 1]))
    prediction_time.append(round(time.perf_counter() - t1, 3))

```

k-Nearest Neighbors

```python
import time
import numpy as np
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import roc_auc_score
from sklearn.model_selection import train_test_split


max_k = 5
knn_auc_train, knn_auc_test = np.zeros(max_k), np.zeros(max_k)
training_time, prediction_time = np.zeros(max_k), np.zeros(max_k)

for i in range(1, max_k):
    clf_knn = KNeighborsClassifier(n_neighbors=i, algorithm='auto', leaf_size=30, metric='minkowski',
                                   metric_params=None, n_jobs=-1, p=2, weights='uniform')
    t0 = time.perf_counter()
    clf_knn = clf_knn.fit(train_X, train_y)
    training_time[i] = round(time.perf_counter() - t0, 3)
    pred_train = clf_knn.predict_proba(train_X)[:, 1]
    t1 = time.perf_counter()
    pred_test = clf_knn.predict_proba(test_X)[:, 1]
    prediction_time[i] = round(time.perf_counter() - t1, 3)
    knn_auc_train[i] = roc_auc_score(train_y, pred_train)
    knn_auc_test[i] = roc_auc_score(test_y, pred_test)

```

Evaluation and Model Comparison

```python
# Assuming you have already trained the following models:
# clf_decision_tree, clf_adaboost, clf_nn, clf_knn, clf_svm_linear, clf_svm_rbf

# Define the models in a list for easy iteration
models = [
    ('Decision Tree', clf_decision_tree),
    ('AdaBoost', clf_adaboost),
    ('MLP', clf_nn),
    ('KNN', clf_knn),
    ('SVM Linear', clf_svm_linear),
    ('SVM RBF', clf_svm_rbf)
]

# Initialize dictionaries to store metrics
accuracy, precision, recall, f1, auc = {}, {}, {}, {}, {}

# Evaluate each model
for name, model in models:
    y_pred = model.predict(test_X)
    y_pred_proba = model.predict_proba(test_X)[:, 1]

    accuracy[name] = accuracy_score(test_y, y_pred)
    precision[name] = precision_score(test_y, y_pred)
```

9.0 Troubleshooting and Common Issues

- If problems come up, double-check code syntax and make sure all needed libraries are installed correctly.
- For persistent issues, use Google Colab's support resources or relevant Python programming communities.

10.0 Security Considerations

- Use only trustworthy data sources and libraries, as specified in the notebook.
- Uphold data privacy and security vigilance while working with any confidential data.

References

Beloglazov, A. and Buyya, R. (2015). Openstack neat: a framework for dynamic and energy-efficient consolidation of virtual machines in openstack clouds, *Concurrency and Computation: Practice and Experience* 27(5): 1310–1333.

Bischl, B., Binder, M., Lang, M., Pielok, T., Richter, J., Coors, S., ... & Lindauer, M. (2023). Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *13*(2), e1484.

Egozi, G., & Verma, R. (2018, November). Phishing email detection using robust nlp techniques. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)* (pp. 7-12). IEEE.

Feng, G. and Buyya, R. (2016). Maximum revenue-oriented resource allocation in cloud, *IJGUC* 7(1): 12–21.

Gomes, D. G., Calheiros, R. N. and Tolosana-Calasanz, R. (2015). Introduction to the special issue on cloud computing: Recent developments and challenging issues, *Computers & Electrical Engineering* 42: 31–32.

Kune, R., Konugurthi, P., Agarwal, A., Rao, C. R. and Buyya, R. (2016). The anatomy of big data computing, *Softw., Pract. Exper.* 46(1): 79–105.

McKinney, W., & Team, P. D. (2015). Pandas-Powerful python data analysis toolkit. *Pandas—Powerful Python Data Analysis Toolkit*, *1625*.

Park, H., Nam, Y., Kim, J. H., & Choo, J. (2020). Hypertendril: Visual analytics for user-driven hyperparameter optimization of deep neural networks. *IEEE Transactions on Visualization and Computer Graphics*, *27*(2), 1407-1416.

Taye, M.M., 2023. Understanding of Machine Learning with Deep Learning: Architectures, Workflow, Applications and Future Directions. *Computers*, *12*(5), p.91.

Monthly Internship Activity Report

The Internship Activity Report is a 1-page monthly summary of the activities performed by you and what you have learned during that month. The Internship Activity Report must be signed off by your Company and included in the configuration manual as part of the portfolio submission.

Student Name: Oluwafunsho Alabi Student number: x22126899

Company: APB Ireland & Poland Month Commencing: 11th September, 2023

My role as an Intern for the first month in the IT department of ABP Ireland & Poland, included offering IT security knowledge to aid in creating, analysing, and implementing information security systems and frameworks that would safeguard the company operations.

Responsibilities of the job function entailed:

- Assisting in identifying and mitigating IT security risks and ensuring compliance with regulatory requirements
- Analyzing, implementing and executing security controls proactively to preventexternal threat actors from infiltrating
- Maintaining and reviewing security vulnerabilities and systems
- company information systems.
- Research more advance and complex attempts/efforts to compromise security protocols.
- Identifying and managing vulnerabilities through thorough assessments to minimize potential entry points for threats.
- Software asset and patch management
- Supporting Incidence response operations
- Assisting in identifying and mitigating IT security risks and ensuring compliance with regulatory requirements

Employer comments

Oluwafunsho has fitted seamlessly into the infrastructure team, showing a keen willingness to learn and engage with various projects and tasks. He has displayed a firm grasp of IT principles and concepts and have applied them diligently to each assignment given to him. Notably, his ability to pick up new technologies and software quickly has been truly impressive.

Student Signature:  Date: 10th October, 2023.

Industry Supervisor Signature: Alan Farrelly Date: 10th October, 2023.

Monthly Internship Activity Report

The Internship Activity Report is a 1-page monthly summary of the activities performed by you and what you have learned during that month. The Internship Activity Report must be signed off by your Company and included in the configuration manual as part of the portfolio submission.

Student Name: Oluwafunsho Alabi Student number: x22126899

Company: APB Ireland & Poland Month Commencing: 11th September, 2023

My role as an Intern in the second month at ABP Ireland & Poland, included helping the organization protect its sensitive data from the constantly evolving cyber threats. This was done by applying the skills and knowledge gained through my education and previous experiences.

Responsibilities of the job function entailed:

- Proactively identify and investigate potential issues and patterns in security controls and recommend mitigation strategies, while also surfacing opportunities for automation to improve efficiency and effectiveness across the network.
- Assist in monitoring and maintaining the security of the organization's network and systems.
- Provide day-to-day administration and maintenance of Windows and Microsoft office 365 systems
- Provide regular reports to management on the state of the organization's cybersecurity posture, including metrics on threats detected, incidents responded to, and vulnerabilities remediated
- Monitor and troubleshoot system performance issues
- Consistently assess established procedural methods and protocols to ensure the SOC remains optimally responsive to operational demands.
- Identify gaps in security policy and administration, recommend solutions, and implement new and revised security standards.

Employer comments

Oluwafunsho contributions to our recent project on Crowdstrike PC updates, where he showed great initiative in troubleshooting technical issues physically and remotely also providing innovative solutions to speedup the process was amazing. Additionally, his involvement in our cybersecurity campaign has been invaluable, displaying a keen eye for detail in identifying potential vulnerabilities and suggesting effective measures to mitigate risks.

Student Signature:  Date: 10th November, 2023.

Industry Supervisor Signature: Alan Farrelly Date: 10th November, 2023.

Monthly Internship Activity Report

The Internship Activity Report is a 1-page monthly summary of the activities performed by you and what you have learned during that month. The Internship Activity Report must be signed off by your Company and included in the configuration manual as part of the portfolio submission.

Student Name: Oluwafunsho Alabi Student number: x22126899

Company: APB Ireland & Poland Month Commencing: 11th September, 2023

In the last month as an intern at ABP Ireland & Poland in the IT department, i was engaged in the following activities:

- Support the vulnerability management process by assisting with vulnerability scans, analyzing results, and tracking remediation efforts.
- Assist in keeping thorough records of security testing outcomes, vulnerability evaluations, and actions taken to address them.
- Assist in developing and updating cybersecurity policies and procedures, including conducting risk assessments and gap analyses to ensure compliance with industry regulations and standards.
- Identifying and managing vulnerabilities through thorough assessments to minimize potential entry points for threats.
- Software asset and patch management
- Assist in managing the organization's network infrastructure, which includes configuring routers, switches, firewalls, and other network devices.
- Assisting in identifying and mitigating IT security risks and ensuring compliance with regulatory requirements
- Server Backups (veeam, Avarma and Druva).
- Memory increase for drives.
- Windows Server Updates
- Participating in troubleshooting and developing network systems under the supervision of more experienced colleagues

Employer comments

Beyond Oluwafunsho technical skills, he has been an exemplary team player, displaying excellent communication and interpersonal skills that has enabled him work effectively with colleagues at all levels. He has been eager to assist wherever possible, whether providing support for client projects or participating in internal initiatives. We anticipate witnessing his continued growth and development.

Student Signature:  Date: 15th December, 2023.

Industry Supervisor Signature: Alan Farrelly Date: 15th December, 2023.