International Journal of Engineering Applied Sciences and Technology, 2022
Vol. 7, Issue 10, ISSN No. 2455-2143, Pages 09-21
Published Online February 2023 in IJEAST (http://www.ijeast.com)

AN OPTIMISED PHISHING EMAIL DETECTION AND PREVENTION USING CLASSIFICATION MODELS

Usman Abdullahi Mohammed & Muhammad Sanusi
Postgraduate School, Department of Computer Science,
University of Abuja, FCT, Nigeria

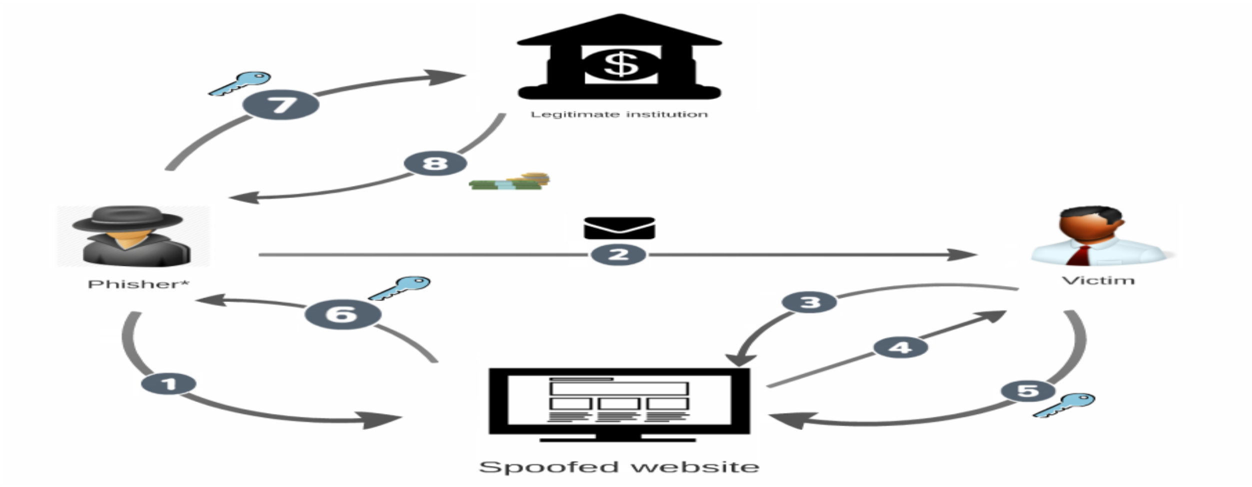
Abstract: **The act of sending a fake e-mail to a user is known as phishing. It involves imitating a legitimate financial institution or organization in order to trick the recipient into providing their personal information. Due to the harmful effects of phishing emails, the development of classification models that can help identify and prevent fraudulent emails has been considered. Four of the most prominent models in the literature for analyzing phishing emails are K-Nearest Neighbor, Support Vector Machine, Random Forest and Nave Bayes. A model that combines three of the models with better performance metrics using 47 features was developed. It was tested against various existing models and performed well in comparison to them. Finally, the comparison analysis of the model is conducted to archive a realistic accuracy rate of 99 percent.**

Keywords: **Phishing email, Data mining, Classification models, Optimized classification Model.**

I. INTRODUCTION

With the advancement of information technology in the modern generation, the evolution of the digital era has become more mature in the sense of effectiveness and ease for societies. People can sell and buy goods, conduct banking activities and even participate in political activities such as elections online. Trusted entities such as financial institutions generally offer their products and services to the public through the internet. Furthermore, modern technology has greatly impacted our society in different ways, such as the way we communicate with each other. Nowadays, we no longer need to use a computer to send an email. With just use of smartphone, which we carry every day in our pockets. As a result, society has been utilizing technological means such as emails, websites, online payment systems and social networks to achieve their tasks efficiently, affordably and in a more focus way. However, the advancement in information and communication technology has been a double-edged sword. As the internet increasingly becomes more accessible, people tend to share more about themselves and as a consequence, it becomes easier to get personal information about someone on the internet. Cyber criminals see this opportunity as a way to manipulate consumers and exploit their confidential information such as usernames, passwords, bank account information, credit card or social security numbers. Personalized information about someone such as email addresses, phone numbers, birthdates, relationships or workplaces can be obtained from the internet. Consequently, cyber criminals can compose an attack in a personalized way to persuade intended victims to grant their malicious requests. One particular type of cybercrimes is called phishing. [7] and [11].

Phishing is a type of social engineering problem where an attacker sends a fraudulent (spoofed) message designed to trick a human victim into revealing sensitive information to the attacker orto deploy malicious software on the victim's infrastructure like ransom ware. Phishing attacks have become increasingly sophisticated and often transparently mirror the site being targeted by allowing the attacker to observe everything while the victim is navigating the site, and transverse any additional security boundaries with the victim.[13] phishing is by far the most common attack performed by cyber criminals, with the FBI's internet crime complaint centre recording over twice as many incidents of phishing than any other type of computer crime [3].



II. RELATED WORK

2.1. Introduction

Following the harmful effect of phishing attacks, it's sporadically trend and danger posed on the privacy of internet users, recent research into detection and prevention of phishing emails has evolved tremendously. Thus, several techniques have been designed to detect phishing emails ranging from communication-oriented techniques, such as authentication protocols, blacklisting, and white-listing, to content-based filtering techniques [9]. The blacklisting and white-listing techniques have not proven to be adequately efficient when used in different environments and with this demerit they are not commonly used. Meanwhile, the content-based phishing filters have been widely adopted and have proven to be of high efficiency. In the light of this, researchers have focused on content-based mechanism and on developing classification techniques based on the header and body of emails.

In 2007, a study was conducted to measure the efficiency of the existing tools for phishing detection. This study showed that even the best phishing detection toolbars missed over 20% of the phishing websites [5]. Another study, which was conducted in 2009 concluded that most anti-phishing tools did not start blocking phishing sites before several hours or days have passed after these phishing emails sent luring users [10]. Thus in conclusion, the currently implemented detection tools do not detect these phishing emails and websites completely [6].We briefly discussed and summarised different classification models used for the detection and prevention of phishing Emails.

2.2. Phishing Detection Methods

A wide range of filters have been formulated by professionals to detect and prevent phishing email attacks and control occurring menace depending on either conventional methods such as authentication protection, or on modern methods of learning machines such as classification models Naïve Bayes, KNN, Random Forest and Support Vector Machine. [14].

2.2.1 Conventional Methods

Conventional methods of detection are divided into two categories, the network-level protection and the authentication protection. The network level protection includes: blacklist filters and white-list filters which prevent phishing by blocking suspected IP addresses or domains from accessing the network. Moreover, there are Pattern Matching filters and Rule-based filters which rely on manually entered and updated fixed rules for detection [12].

2.2.2 Automated Methods

These methods apply automated classifiers or classification algorithms which are the suppervised machine learning techniques. These classifiers work beside the server and filter the received emails into phishing or legitimate by examining different features of the email's header and body [1].

III. METHODOLOGY

3.1 The Adopted Method and Network Design

Following the highlight from the literature review on feature selection scenarios and classification algorithms, we used the complete 47 email features [4] and [14] to construct some classification models such as Random Forest, K-Nearest Neighbour, Spport Vector Machine and Naïve Bayes to examine phishing emails detection accuracy. Moreover an Optimised classification model is developed by integrating three best aforementioned models for better performance metrics. Lastly comparison assessment of the algorithms was evaluated.

The flow diagram and network design depicts the actual flow of the whole system.

3.1 Data Collection

Fig 3.1 illustrates this. In principle, data is collected from two sources; Monkey website for phishing emails [8] and Spam Assassin website [2]. The collected data is then cleaned using various pre-processing techniques, stop words, null values, duplicates are removed, and outliers are removed. The network is then built, trained and tested.

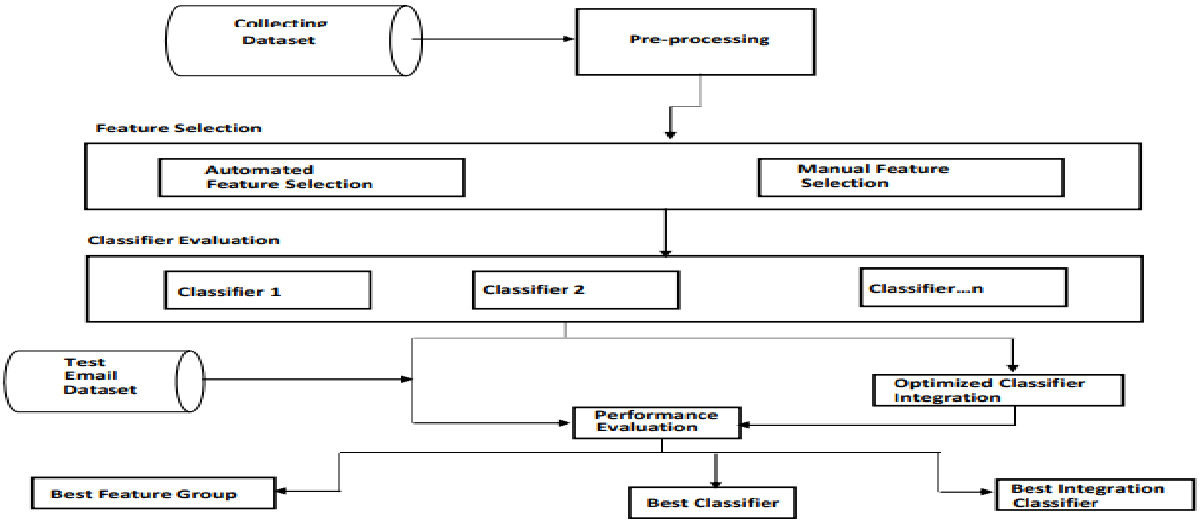


Figure (3.1) Network Flow Chart Design

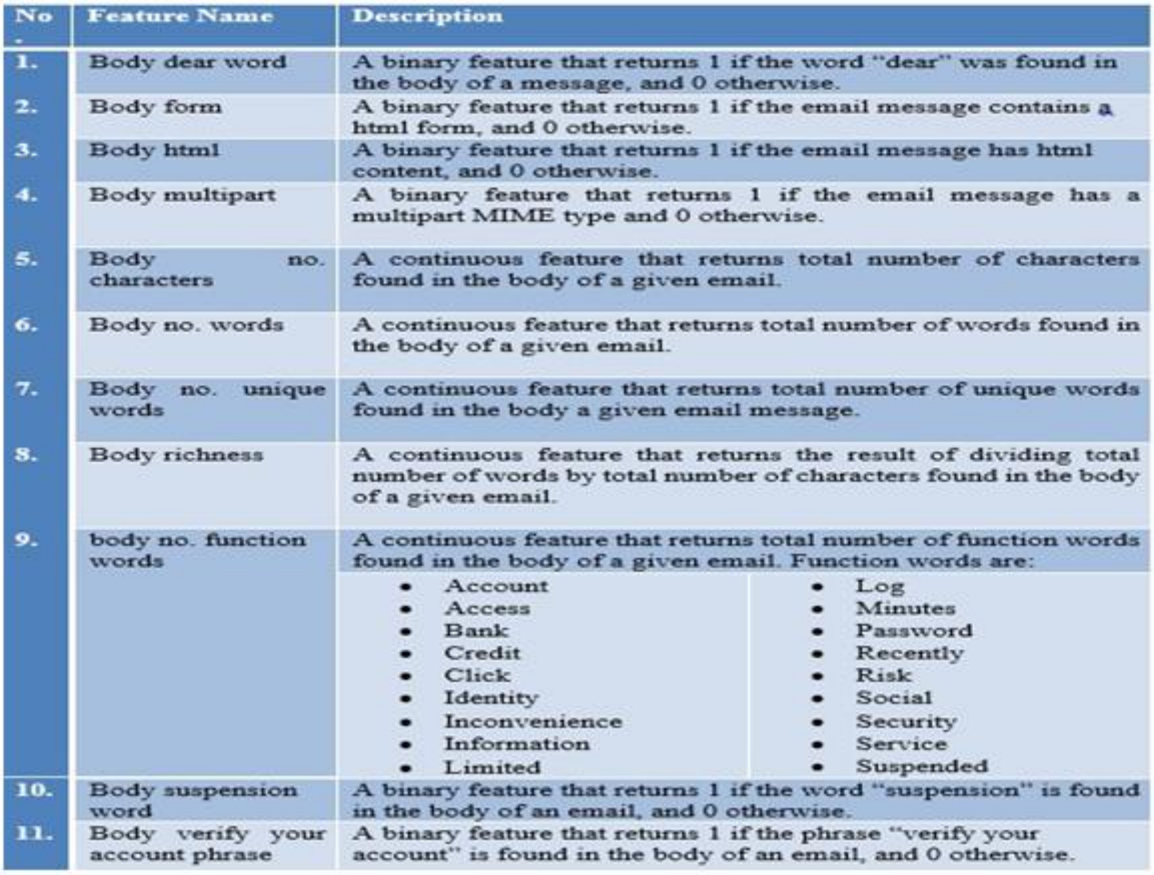
This is the first step in building the proposed phishing email classifier model. The dataset used in this research was streamed from Spam Assassin dataset website for legitimate emails [2] and Monkey website for phishing emails [8] ; the number one open source anti-spam and phishing platform that gives system administrators a filter to classify emails as phishing or legitimate. Sample phishing and legitimate emails of 4800 consisting of 2400 phishing emails and 2400 legitimate emails were downloaded from theses websites. Our sample dataset is then divided into 70:30. Where 70 percent of the total dataset is used for training and 30 percent is for testing.

3.2 Data Preprocessing

In this step the emails in the training data set are prepared and filtered such that they can be transformed into a data format that is easily and effectively processed in subsequent steps of building the classifier. The emails in our chosen training data set are available in plain text format which needs to be pre-processed, cleansed and transformed into Comma-Separated Values format (CSV format) that is interoperable with the python mail package that will be used to extract the email features. The preprocessing step is known to removing irrelevant, redundant inconsistent and noisy data from the dataset. To enhance performance accuracy of the classification model and reducing execution time.

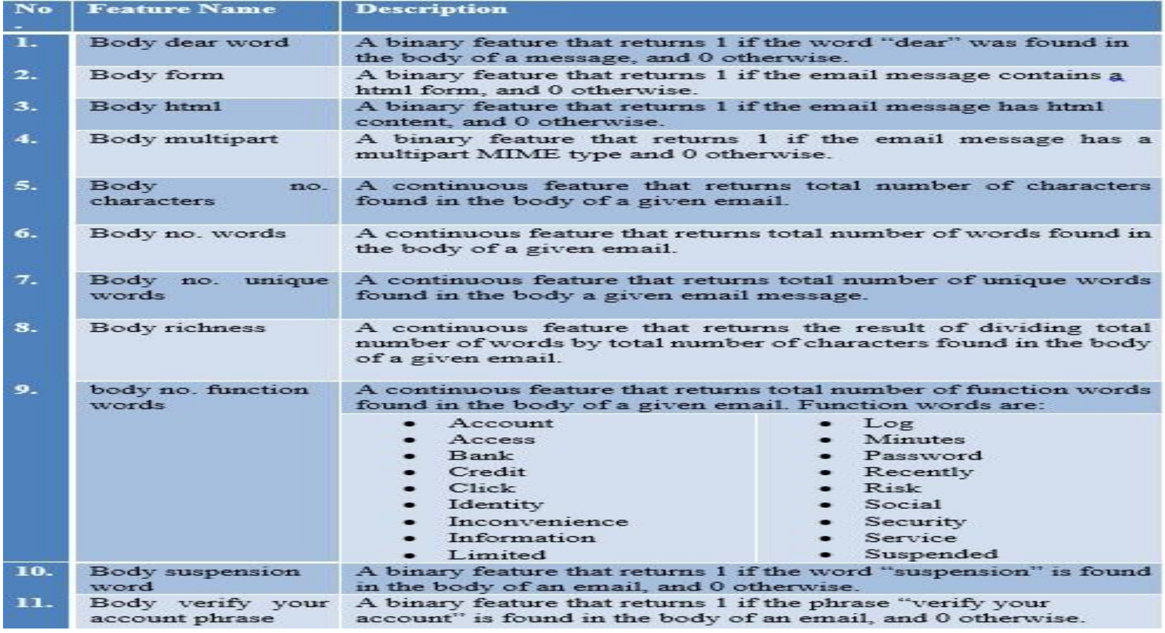
The 47 Features were extracted from each email of the dataset, each raw depict one email along with columns corresponding to 47 selected features, in addition to a column that show the class of the email (whether it is phishing or legitimate email) as depicted in Figure 3.2, 3.3, 3.4 and 3.5. The extracted features were classified into four groups: (Email Body group contains 11 features, Email Header group contains 11 features, URL group contains 18 features and Java script & external group contains 7 features). The 47 features are depicted in Table 3.1, 3.2, 3.3and 3.4respectively.

Table (3.1) The Selected Body features



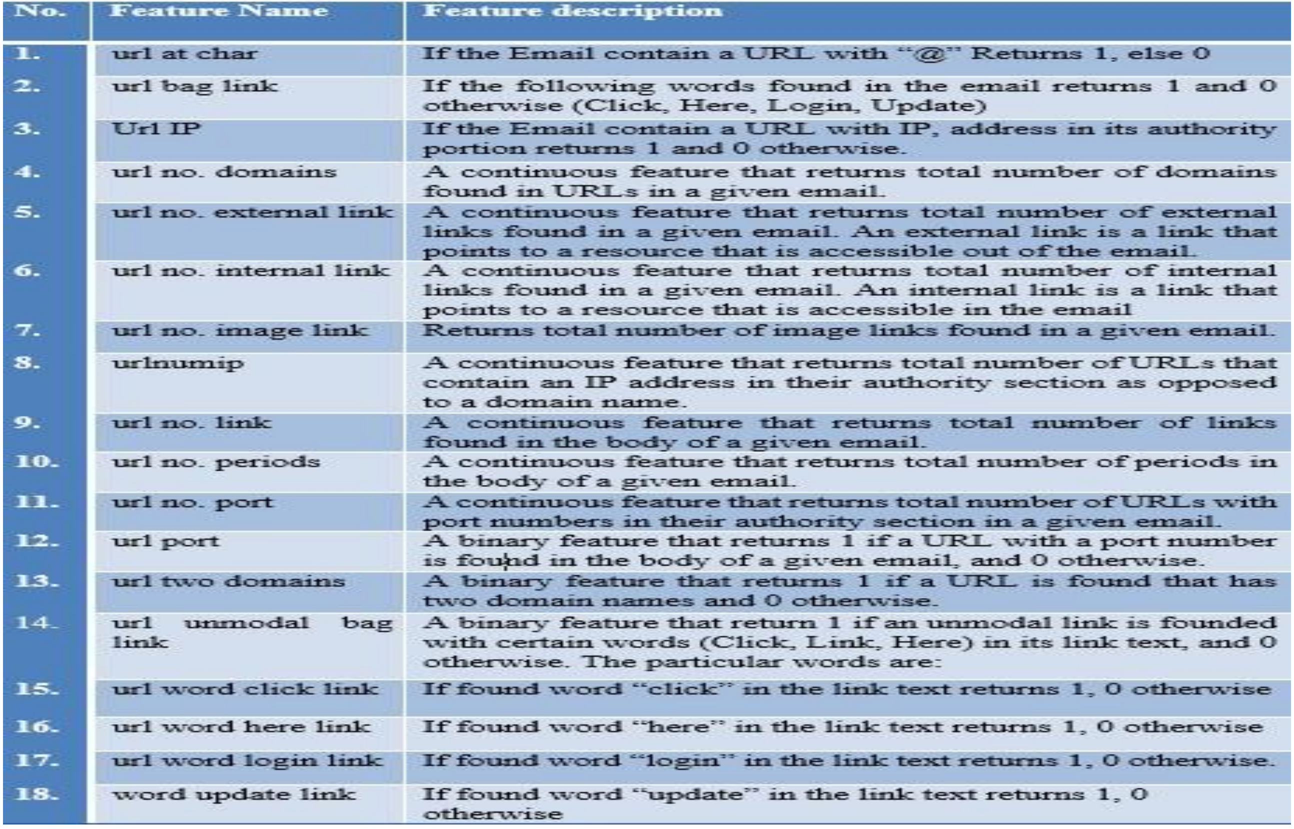
No.	Feature Name	Description
1.	Body dear word	A binary feature that returns 1 if the word "dear" was found in the body of a message, and 0 otherwise.
2.	Body form	A binary feature that returns 1 if the email message contains a html form, and 0 otherwise.
3.	Body html	A binary feature that returns 1 if the email message has html content, and 0 otherwise.
4.	Body multipart	A binary feature that returns 1 if the email message has a multipart MIME type and 0 otherwise.
5.	Body no. characters	A continuous feature that returns total number of characters found in the body of a given email.
6.	Body no. words	A continuous feature that returns total number of words found in the body of a given email.
7.	Body no. unique words	A continuous feature that returns total number of unique words found in the body a given email message.
8.	Body richness	A continuous feature that returns the result of dividing total number of words by total number of characters found in the body of a given email.
9.	body no. function words	A continuous feature that returns total number of function words found in the body of a given email. Function words are: • Account • Access • Bank • Credit • Click • Identity • Inconvenience • Information • Limited • Log • Minutes • Password • Recently • Risk • Social • Security • Service • Suspended
10.	Body suspension word	A binary feature that returns 1 if the word "suspension" is found in the body of an email, and 0 otherwise.
11.	Body verify your account phrase	A binary feature that returns 1 if the phrase "verify your account" is found in the body of an email, and 0 otherwise.

Table (3.2) The Selected Email Header Features



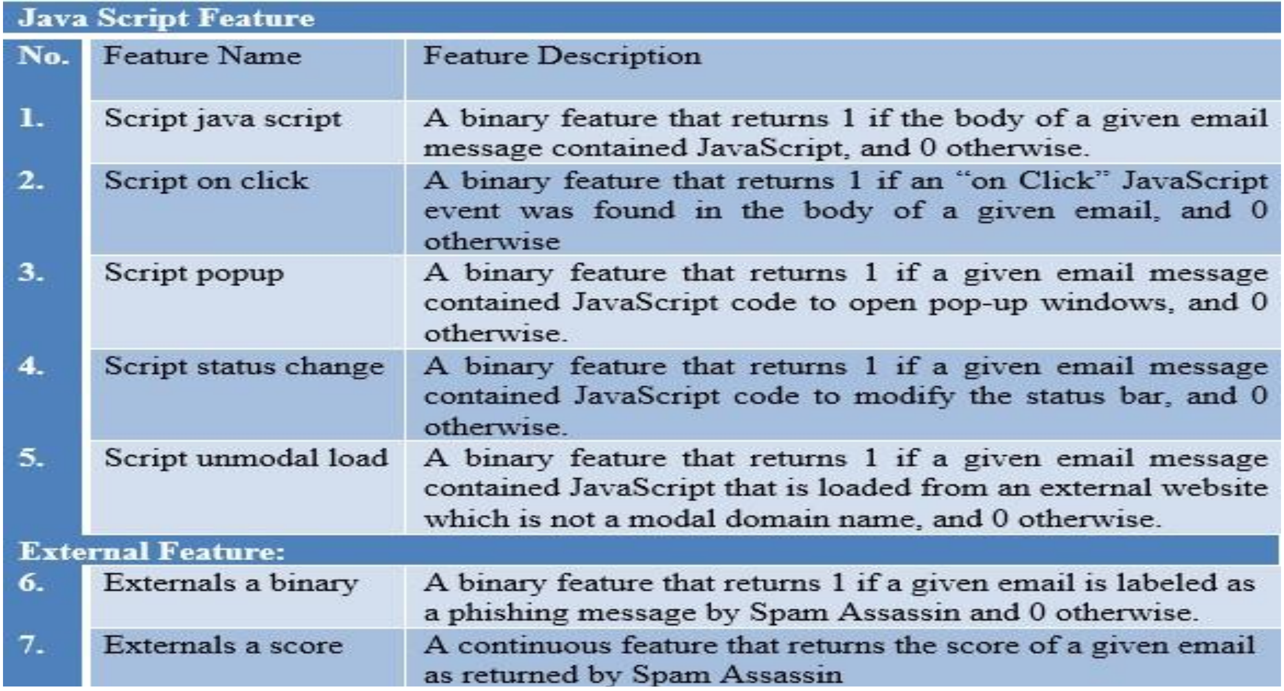
No.	Feature Name	Description
1.	Body dear word	A binary feature that returns 1 if the word "dear" was found in the body of a message, and 0 otherwise.
2.	Body form	A binary feature that returns 1 if the email message contains a html form, and 0 otherwise.
3.	Body html	A binary feature that returns 1 if the email message has html content, and 0 otherwise.
4.	Body multipart	A binary feature that returns 1 if the email message has a multipart MIME type and 0 otherwise.
5.	Body no. characters	A continuous feature that returns total number of characters found in the body of a given email.
6.	Body no. words	A continuous feature that returns total number of words found in the body of a given email.
7.	Body no. unique words	A continuous feature that returns total number of unique words found in the body a given email message.
8.	Body richness	A continuous feature that returns the result of dividing total number of words by total number of characters found in the body of a given email.
9.	body no. function words	A continuous feature that returns total number of function words found in the body of a given email. Function words are: • Account • Access • Bank • Credit • Click • Identity • Inconvenience • Information • Limited • Log • Minutes • Password • Recently • Risk • Social • Security • Service • Suspended
10.	Body suspension word	A binary feature that returns 1 if the word "suspension" is found in the body of an email, and 0 otherwise.
11.	Body verify your account phrase	A binary feature that returns 1 if the phrase "verify your account" is found in the body of an email, and 0 otherwise.

Table (3.3) The Selected URL Features



No.	Feature Name	Feature description
1.	url at char	If the Email contain a URL with "@" Returns 1, else 0
2.	url bag link	If the following words found in the email returns 1 and 0 otherwise (Click, Here, Login, Update)
3.	Url IP	If the Email contain a URL with IP, address in its authority portion returns 1 and 0 otherwise.
4.	url no. domains	A continuous feature that returns total number of domains found in URLs in a given email.
5.	url no. external link	A continuous feature that returns total number of external links found in a given email. An external link is a link that points to a resource that is accessible out of the email.
6.	url no. internal link	A continuous feature that returns total number of internal links found in a given email. An internal link is a link that points to a resource that is accessible in the email
7.	url no. image link	Returns total number of image links found in a given email.
8.	urlnumip	A continuous feature that returns total number of URLs that contain an IP address in their authority section as opposed to a domain name.
9.	url no. link	A continuous feature that returns total number of links found in the body of a given email.
10.	url no. periods	A continuous feature that returns total number of periods in the body of a given email.
11.	url no. port	A continuous feature that returns total number of URLs with port numbers in their authority section in a given email.
12.	url port	A binary feature that returns 1 if a URL with a port number is found in the body of a given email, and 0 otherwise.
13.	url two domains	A binary feature that returns 1 if a URL is found that has two domain names and 0 otherwise.
14.	url unmodal bag link	A binary feature that return 1 if an unmodal link is founded with certain words (Click, Link, Here) in its link text, and 0 otherwise. The particular words are:
15.	url word click link	If found word "click" in the link text returns 1, 0 otherwise
16.	url word here link	If found word "here" in the link text returns 1, 0 otherwise
17.	url word login link	If found word "login" in the link text returns 1, 0 otherwise.
18.	word update link	If found word "update" in the link text returns 1, 0 otherwise

Table (3.4) The Selected Java Script and External Features



Java Script Feature		
No.	Feature Name	Feature Description
1.	Script java script	A binary feature that returns 1 if the body of a given email message contained JavaScript, and 0 otherwise.
2.	Script on click	A binary feature that returns 1 if an "on Click" JavaScript event was found in the body of a given email, and 0 otherwise
3.	Script popup	A binary feature that returns 1 if a given email message contained JavaScript code to open pop-up windows, and 0 otherwise.
4.	Script status change	A binary feature that returns 1 if a given email message contained JavaScript code to modify the status bar, and 0 otherwise.
5.	Script unmodal load	A binary feature that returns 1 if a given email message contained JavaScript that is loaded from an external website which is not a modal domain name, and 0 otherwise.
External Feature:		
6.	Externals a binary	A binary feature that returns 1 if a given email is labeled as a phishing message by Spam Assassin and 0 otherwise.
7.	Externals a score	A continuous feature that returns the score of a given email as returned by Spam Assassin

3.3 Feature Selection

Following the work of the researchers[14], [4] and others on feature selection scenarios and classification algorithms, two features were considered: Manual and Automated features, the complete 47 email features are extracted from the manual feature, categorized into four groups: Email Header Features, Email Body Features, URL Features and Java script &External Features each containing 11,11,18 and 7 features respectively. This is summarized in Table 3.1, 3.2, 3.3and 3.4respectively. The Automated feature is categorized into three group: Correlation-based Feature Selection, **Figure (3.2) Manual and automated Feature Groups**
Consistency Subset Evaluator and Principle Component Evaluator. Figure 3.2 shows Manual and automated feature groups:

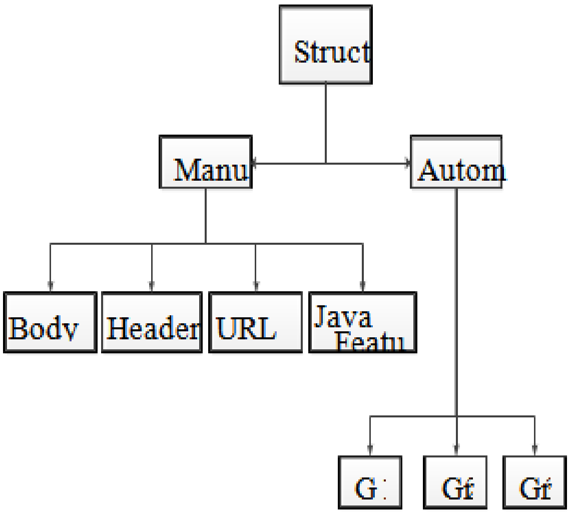


Figure (3.2) Manual and automated Feature Groups

3.4 Classification Algorithms

Our research study uses different well-known classifiers, such as Random Forest (RF), k-Nearest Neighbors (k-NN), Naive Bayes (NB) and Support Vector Machine (SVM) for training, testing and validating the accuracy of the phishing email on the Dataset. . The selected algorithms are:1. KNN
2. Random Forest
3. Support Vector Machine4. Naïve Bayes

3.5 Optimised Classifier Model

Three best techniques have been used to build the Optimised- classifier model: KNN, Random forest and Naïve Bayes, the first two algorithms (KNN and RF) will test email in order to classify it as phishing or legitimate mail by evaluating the decision box if the label is equal to phishing or legitimate otherwise it will be tested by the third algorithm (NB) to classify as a phishing or legitimate email as depicted in Figure 3.4.

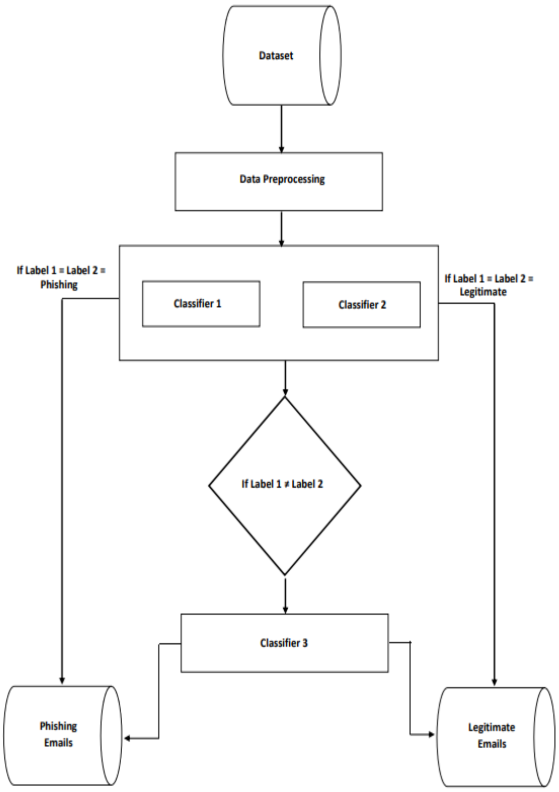


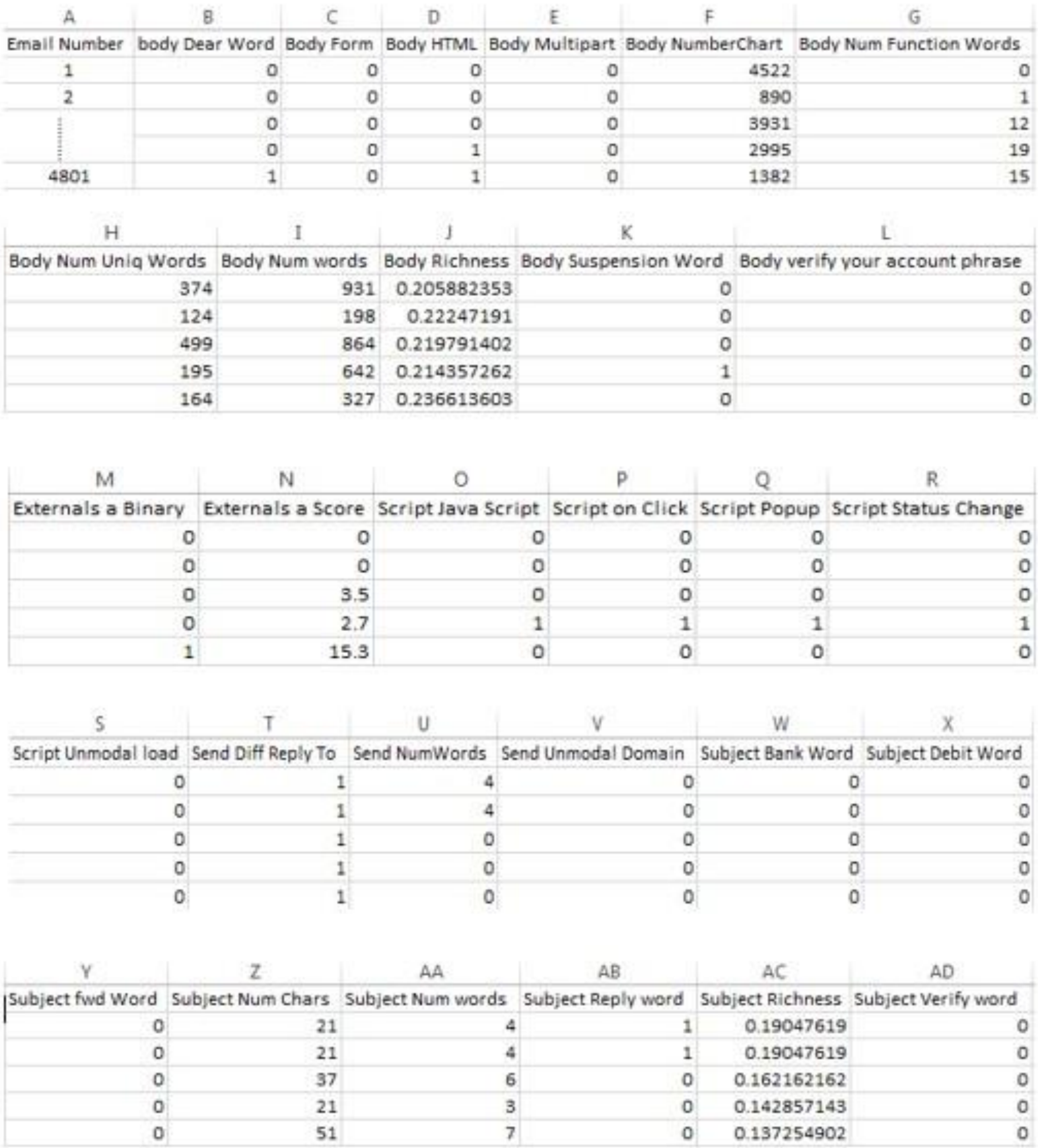
Figure (3.4) Optimized-Classifier Model

IV. RESULTS AND IMPLEMENTATIONS

4.1 Data Set

The research data set all together is 4800 emails, with phishing emails totally 2400 and the legitimate emails is 2400. The emails are available from two sources, firstly, is the monkey website for phishing emails [8], While, the legitimate emails were collected from the spam Assassin website for the data mining competition [2].

Feature extraction is carry out by converting the 47 feature selection by changing them to csv format that is the email selected features and a column which represent the type of the email (whether it is phishing or legitimate email) as depict in Figure 4.1 and 4.2.



A	B	C	D	E	F	G
Email Number	body Dear Word	Body Form	Body HTML	Body Multipart	Body NumberChart	Body Num Function Words
1	0	0	0	0	4522	0
2	0	0	0	0	890	1
	0	0	0	0	3931	12
	0	0	1	0	2995	19
4801	1	0	1	0	1382	15

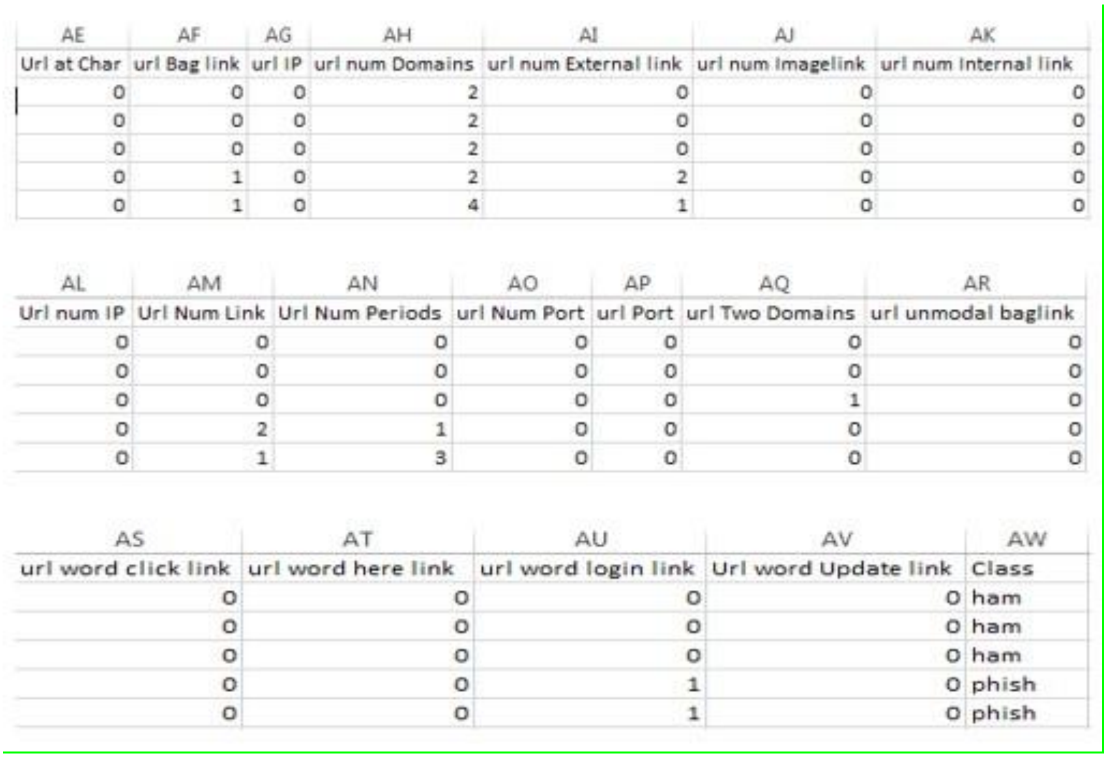
H	I	J	K	L
Body Num Uniq Words	Body Num words	Body Richness	Body Suspension Word	Body verify your account phrase
374	931	0.205882353	0	0
124	198	0.22247191	0	0
499	864	0.219791402	0	0
195	642	0.214357262	1	0
164	327	0.236613603	0	0

M	N	O	P	Q	R
Externals a Binary	Externals a Score	Script Java Script	Script on Click	Script Popup	Script Status Change
0	0	0	0	0	0
0	0	0	0	0	0
0	3.5	0	0	0	0
0	2.7	1	1	1	1
1	15.3	0	0	0	0

S	T	U	V	W	X
Script Unmodal load	Send Diff Reply To	Send NumWords	Send Unmodal Domain	Subject Bank Word	Subject Debit Word
0	1	4	0	0	0
0	1	4	0	0	0
0	1	0	0	0	0
0	1	0	0	0	0
0	1	0	0	0	0

Y	Z	AA	AB	AC	AD
Subject fwd Word	Subject Num Chars	Subject Num words	Subject Reply word	Subject Richness	Subject Verify word
0	21	4	1	0.19047619	0
0	21	4	1	0.19047619	0
0	37	6	0	0.162162162	0
0	21	3	0	0.142857143	0
0	51	7	0	0.137254902	0

Figure (4.1) Sample of the Dataset 47 Feature



AE	AF	AG	AH	AI	AJ	AK
Url at Char	url Bag link	url IP	url num Domains	url num External link	url num Imagelink	url num Internal link
0	0	0	2	0	0	0
0	0	0	2	0	0	0
0	0	0	2	0	0	0
0	1	0	2	2	0	0
0	1	0	4	1	0	0

AL	AM	AN	AO	AP	AQ	AR
Url num IP	Url Num Link	Url Num Periods	url Num Port	url Port	url Two Domains	url unmodal baglink
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	1	0
0	2	1	0	0	0	0
0	1	3	0	0	0	0

AS	AT	AU	AV	AW
url word click link	url word here link	url word login link	Url word Update link	Class
0	0	0	0	ham
0	0	0	0	ham
0	0	0	0	ham
0	0	1	0	phish
0	0	1	0	phish

Figure (4.2) Sample of the Dataset 47 Feature

4.2 Performance Metrics

In order to evaluate our proposed phishing email classification model using different classification techniques, we applied a set of evaluation metrics for each algorithm to compute Precision, Recall, F1-Score, and Accuracy of the Models. Below is the formula for calculating each metrics:

$$\Pr ecision = \frac{TP}{TP + FP} \qquad 1$$

$$\text{Re}\,call = \frac{TP}{TP + FN} \qquad 2$$

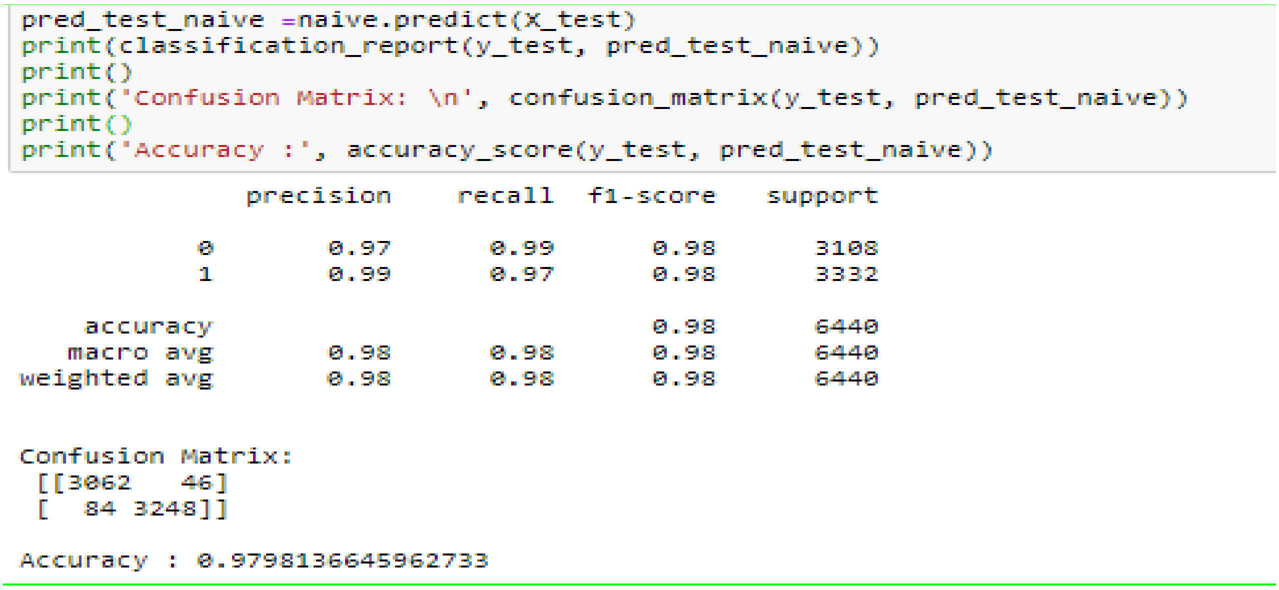
$$F1\_Measure = \frac{2 * \Pr ecision * \text{Re}\,call}{\Pr ecision + \text{Re}\,call} \qquad 3$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \qquad 4$$

Where TP , TN , FP and FN are the True Positive, True Negative, False Positive and False Negative rates.

4.3 Comparison of the Results

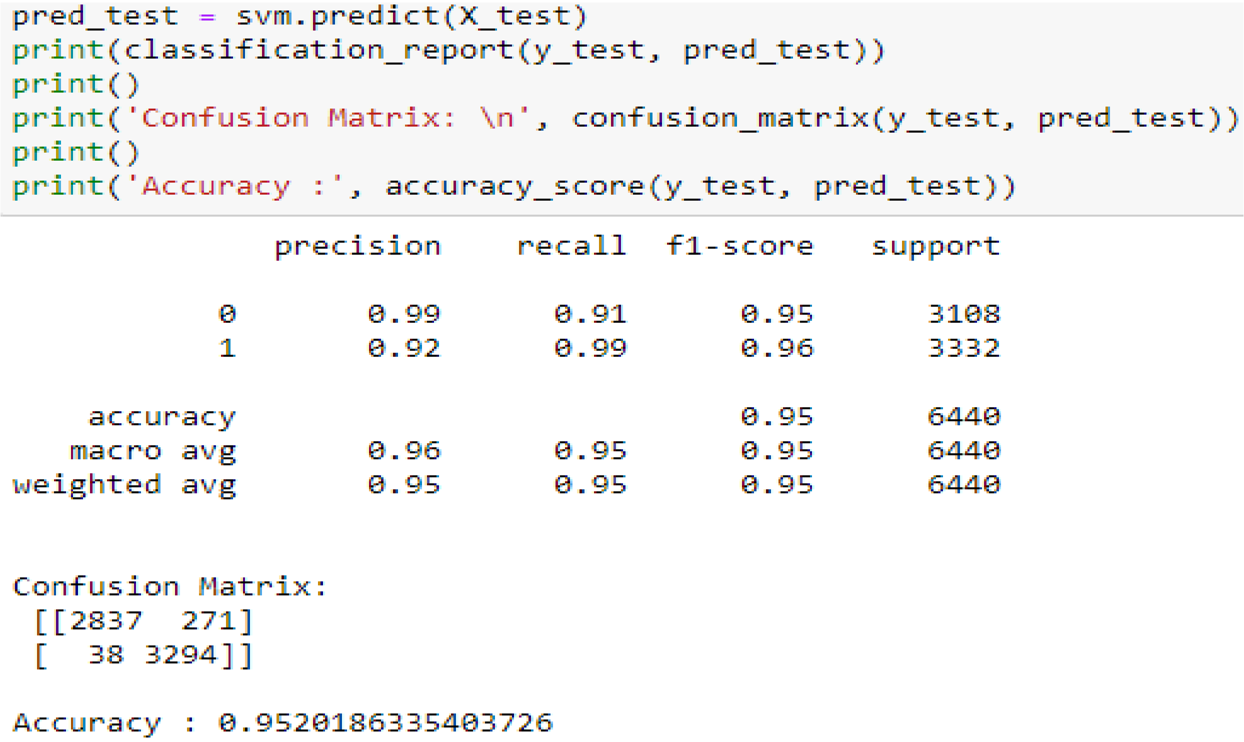
Using python code, we compute the performance evaluation of the classifier models: Naïve Bayes, Support Vector Machines, KNN, Random Forests and our Optimised classifier model. The accuracy of each of the model is estimated in figure 4.3,4.4, 4.5,4.6, 4.7 and tabulated in table 4.1&4.2. Finally our proposed model, "Optimised Classifier Model of Usman (OCM)" is compared with Multi-classifier Method of Sa'id (MCM),2018) and Intelligent classification model of Adwan & Abdelmunem (ICM), 2016:



```
pred_test_naive =naive.predict(X_test)
print(classification_report(y_test, pred_test_naive))
print()
print('Confusion Matrix: \n', confusion_matrix(y_test, pred_test_naive))
print()
print('Accuracy :', accuracy_score(y_test, pred_test_naive))
```

```
              precision    recall  f1-score   support

           0       0.97      0.99      0.98      3108
           1       0.99      0.97      0.98      3332

    accuracy                           0.98      6440
   macro avg       0.98      0.98      0.98      6440
weighted avg       0.98      0.98      0.98      6440


Confusion Matrix: 
 [[3062   46]
 [  84 3248]]

Accuracy : 0.9798136645962733
```

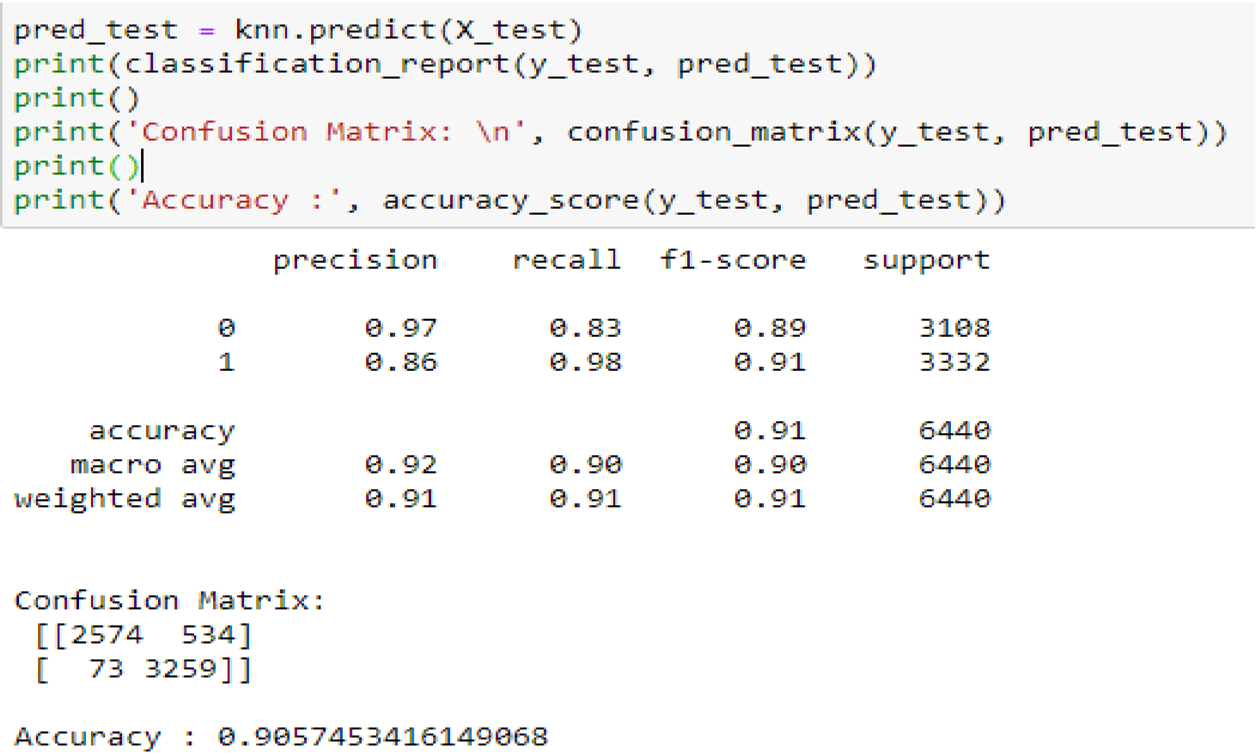
Figure 4.3Python code for performance evaluation of Naïve Bayes algorithm



```
pred_test = svm.predict(X_test)
print(classification_report(y_test, pred_test))
print()
print('Confusion Matrix: \n', confusion_matrix(y_test, pred_test))
print()
print('Accuracy :', accuracy_score(y_test, pred_test))
```

```
              precision    recall  f1-score   support

           0       0.99      0.91      0.95      3108
           1       0.92      0.99      0.96      3332

    accuracy                           0.95      6440
   macro avg       0.96      0.95      0.95      6440
weighted avg       0.95      0.95      0.95      6440


Confusion Matrix: 
 [[2837  271]
 [  38 3294]]

Accuracy : 0.9520186335403726
```

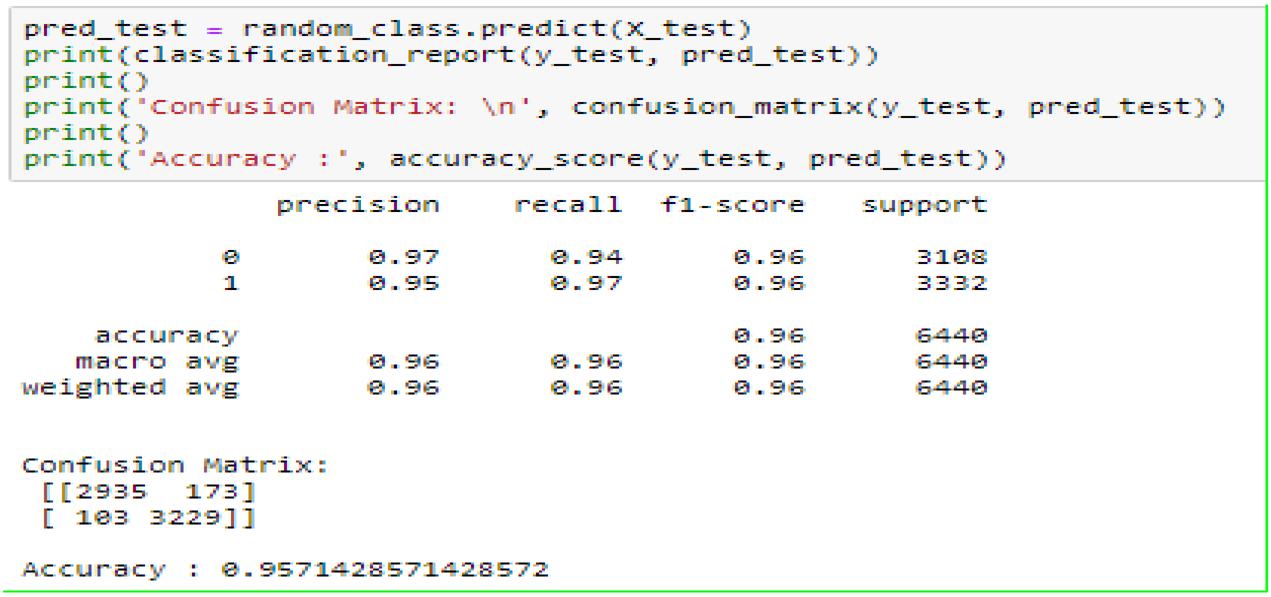
Figure 4.4Python code for performance evaluation of Support Vector Machine algorithm



```
pred_test = knn.predict(X_test)
print(classification_report(y_test, pred_test))
print()
print('Confusion Matrix: \n', confusion_matrix(y_test, pred_test))
print()
print('Accuracy :', accuracy_score(y_test, pred_test))
```

```
              precision    recall  f1-score   support

           0       0.97      0.83      0.89      3108
           1       0.86      0.98      0.91      3332

    accuracy                           0.91      6440
   macro avg       0.92      0.90      0.90      6440
weighted avg       0.91      0.91      0.91      6440


Confusion Matrix:
 [[2574  534]
 [  73 3259]]

Accuracy : 0.9057453416149068
```

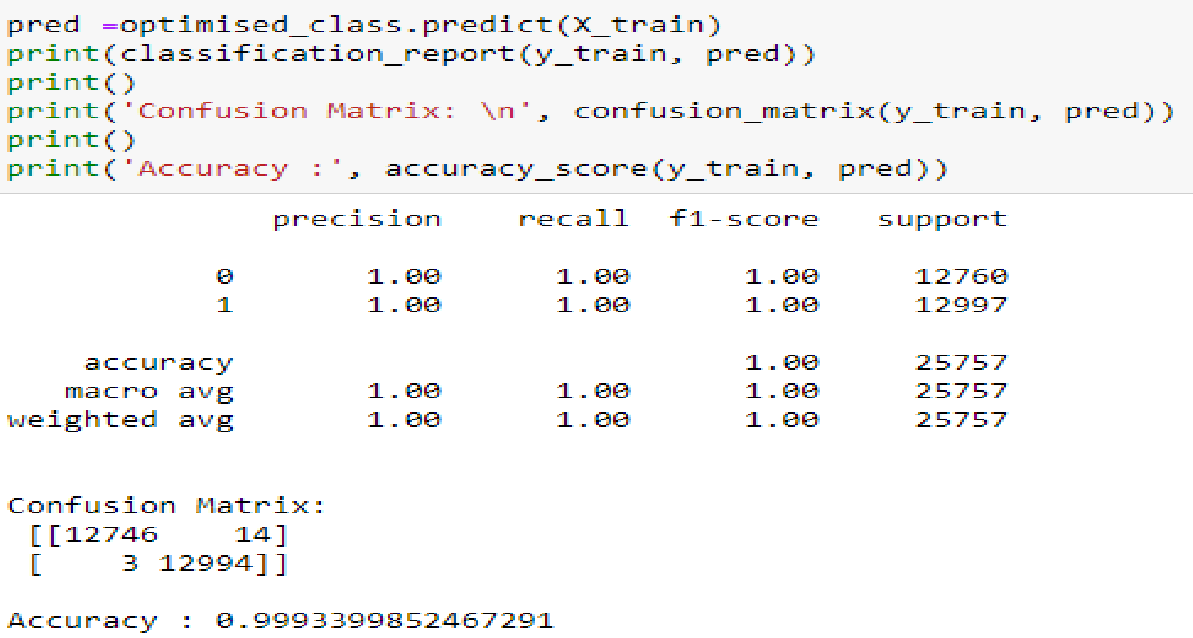
Figure 4.5Python code for performance evaluation of KNN algorithm



```
pred_test = random_class.predict(X_test)
print(classification_report(y_test, pred_test))
print()
print('Confusion Matrix: \n', confusion_matrix(y_test, pred_test))
print()
print('Accuracy :', accuracy_score(y_test, pred_test))
```

```
              precision    recall  f1-score   support

           0       0.97      0.94      0.96      3108
           1       0.95      0.97      0.96      3332

    accuracy                           0.96      6440
   macro avg       0.96      0.96      0.96      6440
weighted avg       0.96      0.96      0.96      6440


Confusion Matrix:
 [[2935  173]
 [ 103 3229]]

Accuracy : 0.9571428571428572
```

Figure 4.6Python code for performance evaluation of Random Tree algorithm



```
pred =optimised_class.predict(X_train)
print(classification_report(y_train, pred))
print()
print('Confusion Matrix: \n', confusion_matrix(y_train, pred))
print()
print('Accuracy :', accuracy_score(y_train, pred))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     12760
           1       1.00      1.00      1.00     12997

    accuracy                           1.00     25757
   macro avg       1.00      1.00      1.00     25757
weighted avg       1.00      1.00      1.00     25757


Confusion Matrix:
 [[12746    14]
 [    3 12994]]

Accuracy : 0.9993399852467291
```

Figure 4.7Python code for performance evaluation ofUsman (OCM).

Table 4.1 Comparison table for the Different Performance Classifiers.

Classifiers	Accuracy (%)	Precision (%)	Recall (%)	F1_score (%)
NB	0.98	0.98	0.98	0.98
SVM	0.95	0.95	0.95	0.95
KNN	0.91	0.91	0.91	0.91
RF	0.96	0.96	0.96	0.96
Usman(OCM)	0.999	1.0	1.0	1.0

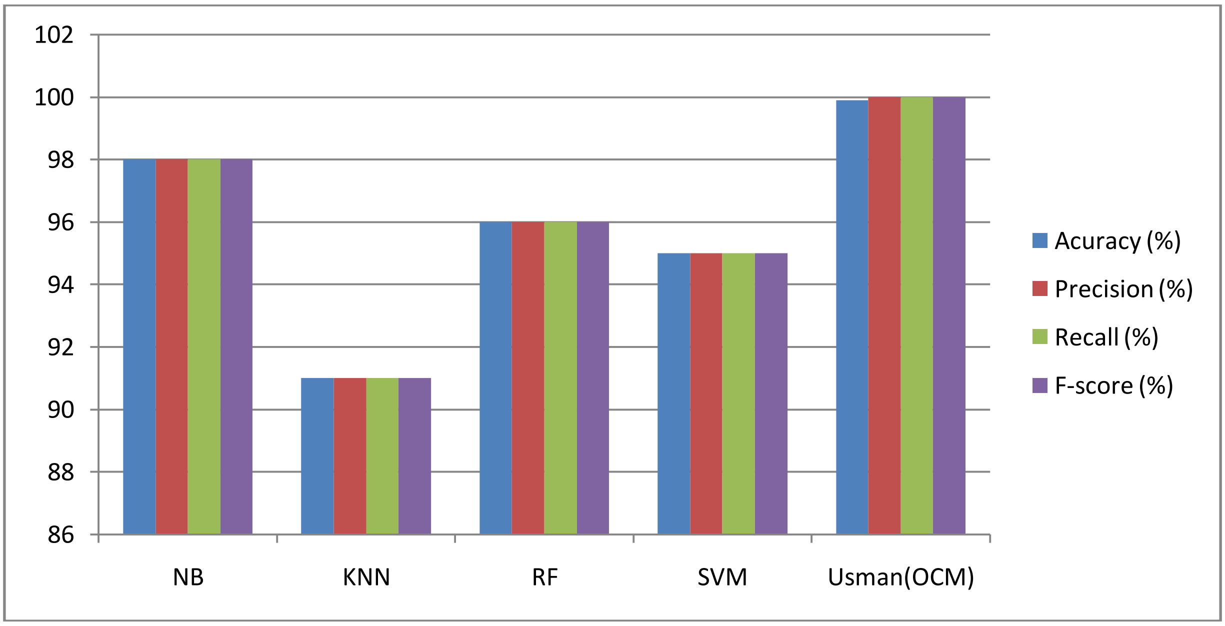


Figure (4.8) Graph plot of the Results

Table 4.2 Comparison for the Different Performance Classifier of Usman, 2020, Sa'id, 2018 and Adwan & Abdelmunem, 2016

Classifiers	Accuracy (%)	Precision (%)	Recall (%)	F1_score (%)
Usman(OCM),2020	0.999	0.999	0.999	0.999
Sa'id(MCM),2018	0.983	0.983	0.983	0.983
Adwan(ICM),2016	0.991	0.991	0.991	0.991

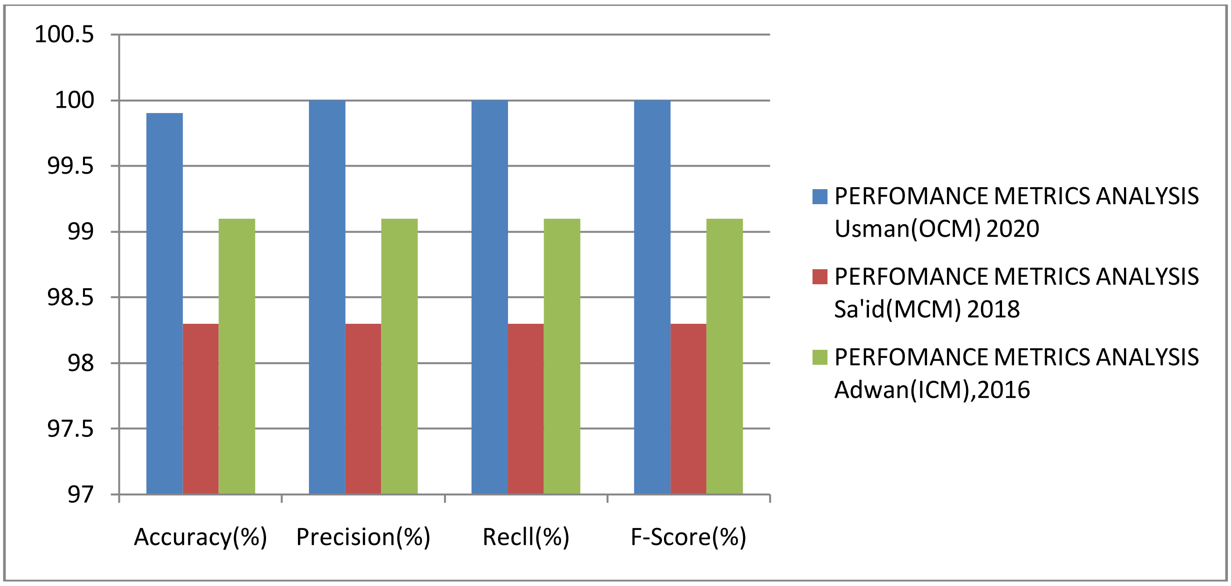


Figure (4.9) Graph plot of the comparison results

V. CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

In this research, manual feature selection of 47 email features were selected from a dataset, grouped into four as Email body features, Email Header features, URL features and Java script features with external features and used as training examples to classified any email as phishing or not.
The accuracy of phishing email detection was evaluated based on manual and automated features selection on four classification algorithms and comparison among the algorithms were conducted. The Naïve Bayes model attained an accuracy of 98% to outperform K-Nearest Nieghbour, Random forest and Support vector Machine models.
The Optimised-classifier model was built by combining the three best classification models to attain an accuracy of 99.9% to beat the other four classification models.
Finally, the advantage of the Optimised-classifier model over some existing methods in literatures is established by comparing their performance metrics.

5.2 Recommendations

Feature selection techniques need more improvement to cope with the over whelming growing new features by the hackers over the period. Therefore, further future works on new automated tool is recommended in order to extract new features from new dataset to improve the accuracy of detecting phishing email and to cope with the expanding phishing techniques.

VI.REFERENCES

[1]. Abu-Nimeh, S., Nappa, D., Wang, X., & Nair, S. (2007, October). A comparison of machine learning techniques for phishing detection. In Proceedings of the antiphishing working groups 2nd annual e-Crime researchers summit (pp. 60-69). ACM.

[2]. Adwan,Y. & Abdelmunem, A.(2016, July).An Intelligent classification model for Phishing email Detection.International Journal of Network Security & Its Applications.Vol,8(no 4). (pp. 55-69)

[3]. Apacheorg. (2020). Apacheorg. Retrieved 18 March, 2020, fromhttp://spamassassin.apache.org/publiccorpus/

[4]. "Internet Crime Report 2020" (https://www.ic3.gov/Media/PDF/AnnualReport/2020_IC 3Report.pdf)

[5]. (PDF). FBI Internet Crime Complaint Centre. U.S. Federal Bureau of Investigation. Retrieved 21 March2021.

[6]. Khonji, A. M., and Iraqi, Y. (2011). A Brief Description of 47 Phishing Classification Features.In GCC conference and exhibition IEE, (pp. 19-24).

[7]. Kumaraguru, P., Sheng, S., Acquisti, A., Cranor, L. F., and Hong, J. (2010). Teaching Johnny not to fall for phish. ACM Transactions on Internet Technology (TOIT), Vol. 10(no.2), (pp. 7-16).

[8]. Kumaraguru, P., Rhee, Y., Acquisti, A., Cranor, L. F., Hong, J., andNunge,E.(2011,April). Protecting people from phishing: the design and evaluation of an embedded training email system. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 905-914). ACM.

[9]. Markus Jakobsson and Steven Myers. (2010).Phishing and countermeasures:understanding the increasing problem of electronic identity theft. John Wiley & Sons, (pp. 1, 9–12, 18, 20).

[10]. Monkeyorg. (2020). Monkeyorg. Retrieved 18 November, 2020, from http://monkey.org

[11]. Paaß, G.,and Bergholz, A. (2009). Project Exhibition:AntiPhish-Machine Learning for Phishing Detection. (pp. 1-7).

[12]. Parmar, B. (2012). Protecting against spear-phishing. Computer Fraud & Security, Vol 10(no 7)(pp. 8-11).

[13]. Rachna Dhamija, J Doug Tygar, and Marti Hearst. (2009). “Why phishing works.” In: Proceedings of the SIGCHI conference on Human Factors in computing systems,ACM. (pp. 1-11).

[14]. Ramanathan, V., and Wechsler, H. (2012). PhishGILLNET—phishing detection methodology using probabilistic latent semantic analysis, AdaBoost, and cotraining. EURASIP Journal on Information Security, (pp. 1-22).

[15]. Ramzan, Zulfikar (2010)."Phishing attacks and countermeasures" (https://books.google.com/books?id=I-9P1EkTkigC&pg=PA433).In Stamp, Mark; Stavroulakis, Peter (eds.). Handbook of Information andCommunication Security. Springer. ISBN 978-3-642-04117-4.

[16]. Sa”id, A. A. (2018). Detecting Phishing Emails Using Machine Learning Techniques,Master thesis, Middle East University.