Phishing Emails Detection using Machine Learning Algorithms

1. Methodology:

To build Email detection system, algorithms should be used for email classification. Moreover, those algorithms require data and this data should be processed to extract features and handle the errors. In addition, features selection techniques should be adapted to get the best results. Finally, the models should be tested and evaluated. Those steps are illustrated in figure 1.

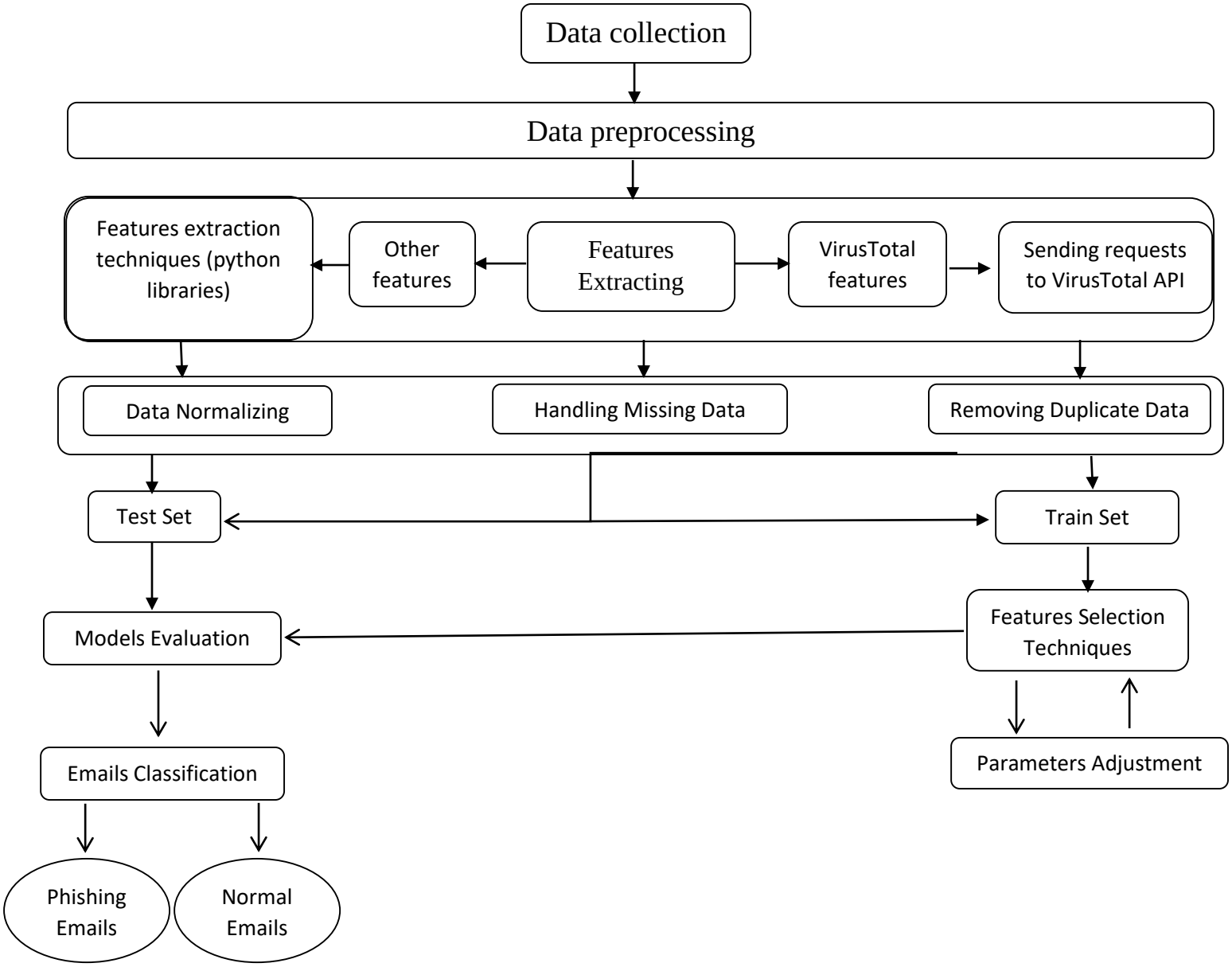

Figure 1: Class Diagram for Work Methodology

2. Data Collecting and Features classification:

2.1. Data collecting:

To perform the process, four raw datasets were used to form the final dataset. First, three raw datasets of phishing Email were combined to obtain 10308 sample which is a sufficient number to be used in the classification process. On the other hand, the **Enron dataset** were used to get 10155 normal sample. As a result, the total number of elements in the dataset is 20307 and the ratio of positive samples is around 50 % which is considered an acceptable value in the area of machine learning. To convert the raw data into an appropriate format to be used in the classification, a python code was written to extract the features from the emails and rearrange the samples in a CSV file. The code was implemented for both phishing and normal datasets and the resulting files were combined to get the final CSV file.

2.2. Features collecting:

A group of features were selected to be extracted from the raw data. The features were extracted from the emails and from the inner urls of those emails using python coding techniques. In addition, external data sources like **VirusTotal** were used to extract some features about urls in the email's text. The features were selected from **VirusTotal** to be convenient and effective for classification algorithms. Those features are displayed in rows 13 to 34 of Table 3

2.3. Features classification:

The features were classified into three main groups: Email-id features (Table 1), Url-text features (Table 2) and Url-content features (Table 3). Some of Url features were exported from **VirusTotal** platform because it is very popular in the area of phishing detection. In addition, the number of attachments was selected as a feature and grouped with Email-id features.

Table 1: Email-id features

Feature Number	Feature Name	Feature Description
1	lengthOfEmailId	an integer column that represents the length of the email ID before the "@" symbol
2	noOfDotsInEmailId	an integer column that represents the number of dots (.) in the email ID before the "@" symbol
3	noOfDashesInEmailId	an integer column that represents the number of dashes (---) in the email ID before the "@" symbol

4	noOfSpecialCharsInEmailId	an integer column that represents the number of special characters in the email ID before the "@" symbol
5	noOfDigitsInEmailId	an integer column that represents the number of digits in the email ID before the "@" symbol
6	noOfSubdomainsInEmailId	an integer column that represents the number of subdomains in the email ID before the "@" symbol
7	noof_urls	an integer column that represents the number of URLs in the email.
8	noof_attachments	an integer column that represents the number of attachments (e.g., files) included in the email

Table 2:URL-Text features

Feature Number	Feature Name	Feature Description
1	length_url	an integer column that represents the length of the URL in the email
2	dots_number	an integer column that represents the number of dots in the URL
3	hyphens_number	an integer column that represents the number of dashes (---) in the email ID before the "@" symbol
4	at_number	an integer column that represents the number of "@" symbols in the URL
5	qm_number	an integer column that represents the number of question marks (?) in the URL
6	and_number	an integer column that represents the number of "&" symbols in the URL
7	or_number	an integer column that represents the number of "|" symbols in the URL
8	eq_number	an integer column that represents the number of "=" symbols in the URL
9	tildes_number	an integer column that represents the number of "~" symbols in the URL
10	percent_number	an integer column that represents the number of "%" symbols in the URL
11	slash_number	an integer column that represents the number of "/" symbols in the URL
12	star_number	an integer column that represents the number of "*" symbols in the URL
13	colon_number	an integer column that represents the number of ":" symbols in the URL
14	comma_number	an integer column that represents the number of "," symbols in the URL
15	semicolon_number	an integer column that represents the number of ";" symbols in the URL

16	dollar_number	an integer column that represents the number of "$" symbols in the URL
17	space_number	an integer column that represents the number of spaces in the URL
18	www_number	an integer column that represents the number of "www" substrings in the URL
19	com_number	an integer column that represents the number of ".com" substrings in the URL
20	http_in_path	a boolean column that indicates whether the string "http" appears in the path of the URL
21	https_token	a boolean column that indicates whether the URL in the email contains the "https" token
22	ratio_digits_url	a float column that represents the ratio of digits to the total number of characters in the URL in the email
23	ratio_digits_host	a float column that represents the ratio of digits to the total number of characters in the hostname in the URL
24	punycode	a boolean column that indicates whether the hostname in the URL is in punycode format
25	port	an integer column that represents the port number in the URL in the email
26	tld_in_path	a boolean column that indicates whether the top-level domain (TLD) appears in the path of the URL in the email
27	tld_in_subdomain	a boolean column that indicates whether the TLD appears in a subdomain of the hostname in the URL
28	abnormal_subdomain	a boolean column that indicates whether the hostname in the URL has an abnormal (i.e., non-standard) subdomain
29	subdomains_number	an integer column that represents the number of subdomains in the hostname in the URL
30	prefix_suffix	a boolean column that indicates whether the hostname in the URL has a prefix or suffix (e.g., "www." or ".com")
31	random_domain	a boolean column that indicates whether the hostname in the URL appears to be randomly generated
32	path_extension	a boolean column that indicates whether the URL in the email has a path extension (e.g., ".html")
33	length_words_raw	an integer column that represents the total number of words in the raw text of the URL
34	char_repeat	an integer column that represents the number of repeated characters in the URL
35	shortest_word_host	an integer column that represents the length of the shortest word in the hostname in the URL
36	longest_word_host	an integer column that represents the length of the longest word in the hostname in the
37	shortest_word_path	an integer column that represents the length of the shortest word in the path of the URL in the email
38	longest_word_path	an integer column that represents the length of the longest word in the path of the URL in the email
39	shortest_words_raw	an integer column that represents the length of the shortest word in the raw text of the URL

40	longest_words_raw	an integer column that represents the length of the longest word in the raw text of the URL
41	average_word_raw	a float column that represents the average length of words in the raw text of the URL
42	average_word_host	a float column that represents the average length of words in the hostname in the URL
43	average_word_path	a float column that represents the average length of words in the path of the URL in the email
44	phish_hints	an integer column that represents the total number of phishing hints in the URL (e.g., "Login")
45	shortening_service	a boolean column that indicates whether the URL in the email points to a URL shortening service
46	path_extension	a boolean column that indicates whether the URL in the email has a path extension (e.g., ".html")

Table 3:URL-Content features

Feature Number	Feature Name	Feature Description
1	hyperlinks_number	an integer column that represents the total number of hyperlinks in the URL
2	ratio_intHyperlinks	a float column that represents the ratio of internal hyperlinks (i.e., hyperlinks that point to the same domain as the URL.) to the total number of hyperlinks in the URL
3	ratio_extHyperlinks	a float column that represents the ratio of external hyperlinks (i.e., hyperlinks that point to a different domain than the URL.) to the total number of hyperlinks in the URL
4	ratio_nullHyperlinks	a float column that represents the ratio of null hyperlinks (i.e., hyperlinks with no target URL) to the total number of hyperlinks in the URL
5	extCSS_number	an integer column that represents the number of external CSS files linked to in the URL
6	submit_emails	a boolean column that indicates whether the URL content contains any submit buttons that could potentially submit data to a remote server
7	ratio_intMedia	a float column that represents the ratio of internal media files (e.g., images, videos) to the total number of media files in the URL content
8	ratio_extMedia	a float column that represents the ratio of external media files to the total number of media files in the URL content
9	sfh	a boolean column that indicates whether the URL content contains any "same origin" links (i.e., links that point to a different page on the same domain as the URL)

10	external_redirections_number	an integer column that represents the number of external redirections in the URL in the email.
11	redirections_number	an integer column that represents the number of redirections in the URL in the email
12	ip	a Boolean column that indicates whether the IP address found in the URL or not
13	ip_asn	an integer column that represents the autonomous system number (ASN) of the IP address found in the URL
14	ip_last_analysis_date	an integer column that represents the date of the last analysis of the IP address found in the URL
15	ip_harmless	<*integer*> number of reports saying that is harmless n integer column that represents the number of ";" symbols in the URL
16	ip_malicious	<*integer*> number of reports saying that is malicious
17	ip_suspicious	<*integer*> number of reports saying that is suspicious
18	ip_timeout	<*integer*> number of timeouts when checking this URL
19	ip_undetected	<*integer*> number of reports saying that is undetected
20	ip_last_modification_date	<*integer*> date when any of the IP's information was last updated. UTC timestamp
21	ip_harmless_votes	<*integer*> number of positive votes
22	ip_malicious_votes	<*integer*> number of negative votes
23	first_submission_date	<*integer*> UTC timestamp of the date where the URL was first submitted to VirusTotal
24	last_analysis_date	<*integer*> UTC timestamp representing last time the URL was scanned
25	harmless	<*integer*> number of reports saying that is harmless
26	malicious	<*integer*> number of reports saying that is malicious
27	suspicious	<*integer*> number of reports saying that is suspicious
28	timeout	<*integer*> number of timeouts when checking this URL
29	undetected	<*integer*> number of reports saying that is undetected
30	last_http_response_code	an integer column that represents the last HTTP response code received from the URL
31	last_http_response_content_length	<*integer*> length in bytes of the content received
32	last_modification_date	<*integer*> UTC timestamp representing last modification date
33	harmless_votes	<*integer*> number of positive votes
34	malicious_votes	<*integer*> number of negative votes

2.4. Data Preprocessing:

After extracting features from the raw data, preprocessing is necessary before implementing algorithms. Preprocessing involves removing duplicate and zero elements, converting string values to numerical values (Boolean, integer, and double). Additionally, preprocessing may include normalization to ensure that all values are within the same range. However, in this particular dataset, normalization was not applied as it contains numerous features with distinct numerical data types.

3. Algorithms and features selection:

3.1. Algorithms:

Five supervised classification algorithms were chosen for training and evaluating the accuracy of phishing email detection using grouped features. These algorithms were selected due to their distinct training strategies, rule discovery mechanisms, and learning/testing approaches. The following well-known algorithms were included in the selection:

- K-Nearest Neighbors (KNN)
- Decision Tree (DT)
- Logistic Regression (LR)
- Support Vector Machine (SVM)
- Random Forest (RF)

3.2. Features:

Due to the high number of features in the dataset, an appropriate feature selection method should be used. To optimize the selection process, two approaches were adapted:

3.2.1. Manual features selection:

It is based on selecting random features from each group and then testing the group selected and discuss the results. After some experiments, 5 groups were selected with a high performance.

3.2.2. Automated features selection:

Since manual selection is based on experiments, it may have some problems. Two automated features selection algorithms were used:

- SelectKBest and ANOVA F-value: SelectKBest is a filter method that selects the top K features based on a statistical test. The statistical test can be chosen from a variety of options, such as chi-squared, mutual information, and ANOVA F-value. (Saturn Cloud, 2023)
- Principal Component Analysis (PCA): PCA belongs to a set of methods designed to handle high-dimensional data by leveraging the relationships between variables to transform it into a more manageable and lower-dimensional representation, while minimizing the loss of relevant information. PCA stands out as a straightforward and resilient approach for achieving effective dimensionality reduction. (Carnegie Mellon University, 2010).

4. Tools and Metrics:

4.1. Tools:

All the processes were performed using python on google colab platform. Many libraries were used for feature extraction and data preprocessing including: mailbox, chardet, os, whois, requests, urllib, numby and pandas. In addition, some libraries like sklearn and matplotlib were used for algorithms implementation and results representation.

4.2. Metrics:

Accuracy, Precision, Recall and f1 score were selected as metrics to evaluate the algorithms performance. The confusion matrix (fig 1) help to understand them. Accuracy represents the ratio between true (T) classified items to the total classified items:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

On the other hand, precision is calculated as the ratio between true positive (TP) classified items to the total number of positive classified items (T and F):

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

	Classified Phishing	Classified legitimate
Actual Phishing	TP	FN
Actual Legitimate	FP	TN

Figure 2: Confusion Matrix

Moreover, recall measures the model's ability to detect positive samples and it is calculated as the ratio between true positive (TP) classified items to the total number of positive items (TP and FN):

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

Finally, f1 score can be defined as the harmonic mean of precision and recall. It is important to adjust precision and recall to get the optimal model. It can be calculated using the following equation:

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} = \frac{TP}{TP + (FP + FN)/2} \quad (4)$$

(UNIVERSITY of WISCONSIN–MADISON, 2003)

5. Results and Discussion:

After applying the above classification algorithms form many groups of features, the results varied between 90.25 % as a minimum and 99.98 % as a maximum for the **accuracy** metric. For the **precision** metric, the maximum precision was around 99.96 % and 86.01 % for the minimum. In addition, the **recall** metric was calculated with 87.7% as a minimum and 100 % as a maximum. The detailed results are illustrated in the appropriate tables and figures for each feature group:

5.1. All Feature together:

The Algorithms were applied on the overall features in the dataset (88 features). The obtained results are illustrated in Table 4 and figures 3, 4, 5,6,7,8.

Table 4: Features 1 results

	Accuracy	Precision	Recall	F1 score
Random Forest	0.991136	0.988091	0.994010	0.991041
SVM	0.956076	0.974615	0.935304	0.954555
Logistic Regression	0.930668	0.976950	0.880192	0.926050
KNeighborsClassifier	0.980303	0.974724	0.985623	0.980143
Decision Trees	0.983455	0.978261	0.988419	0.983313

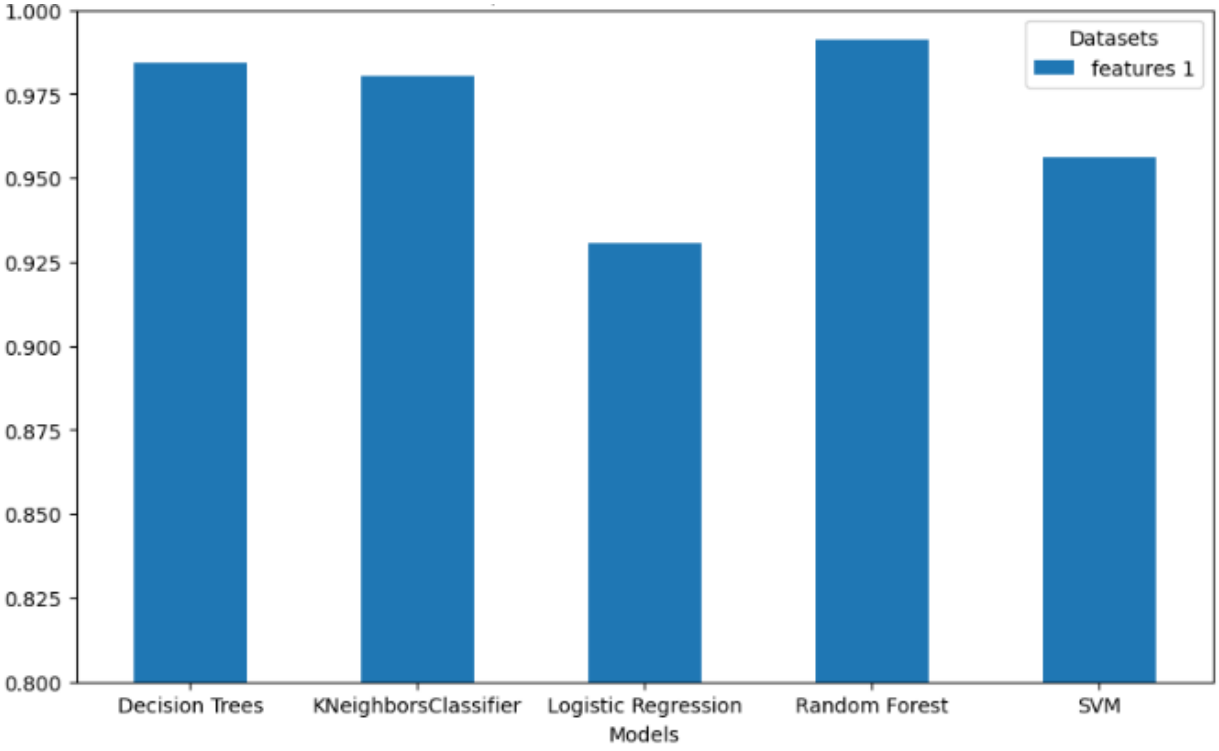

Figure 3: Comparasion of Models Performance for Features 1

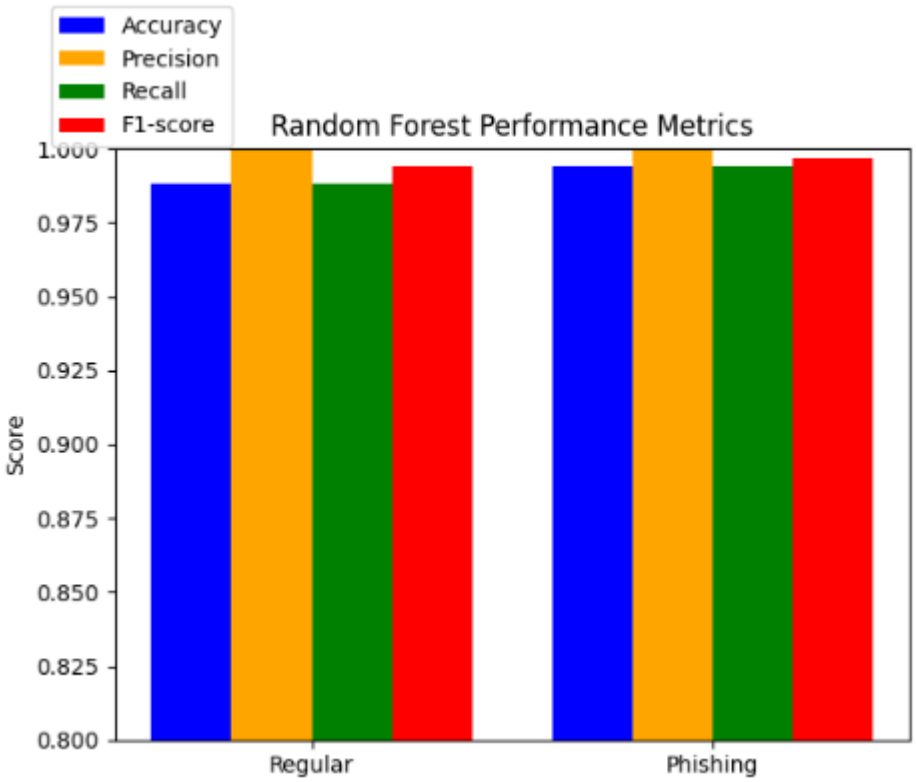

Figure 4: Random Forest Classification metrics (Features 1)

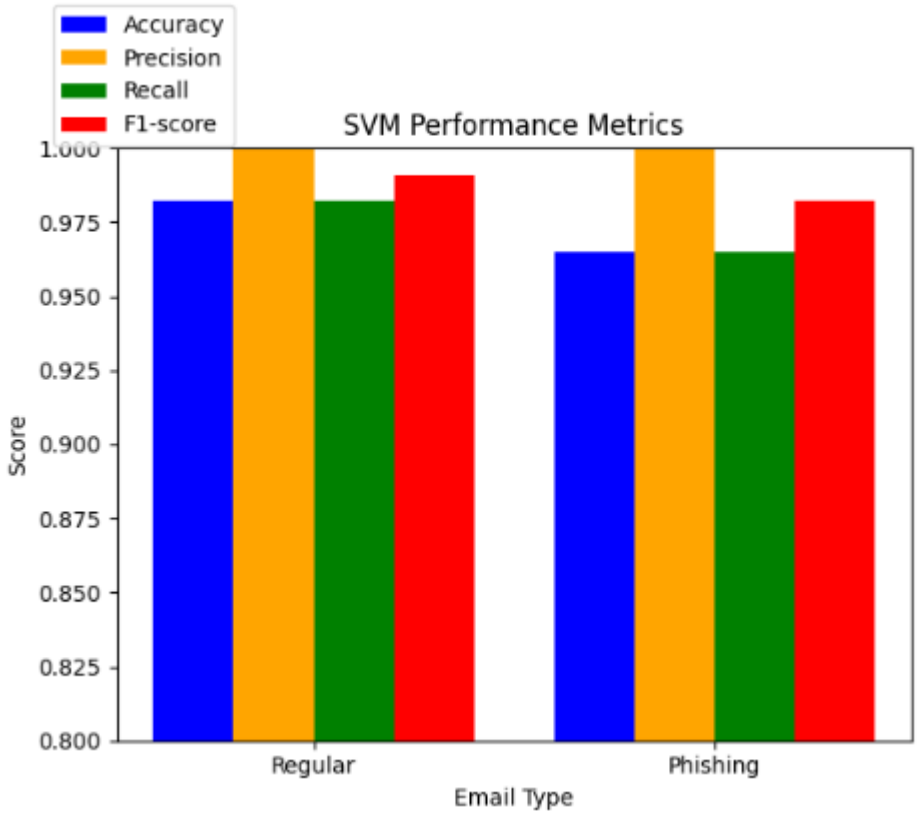

Figure 5: SVM Classification metrics (Features 1)

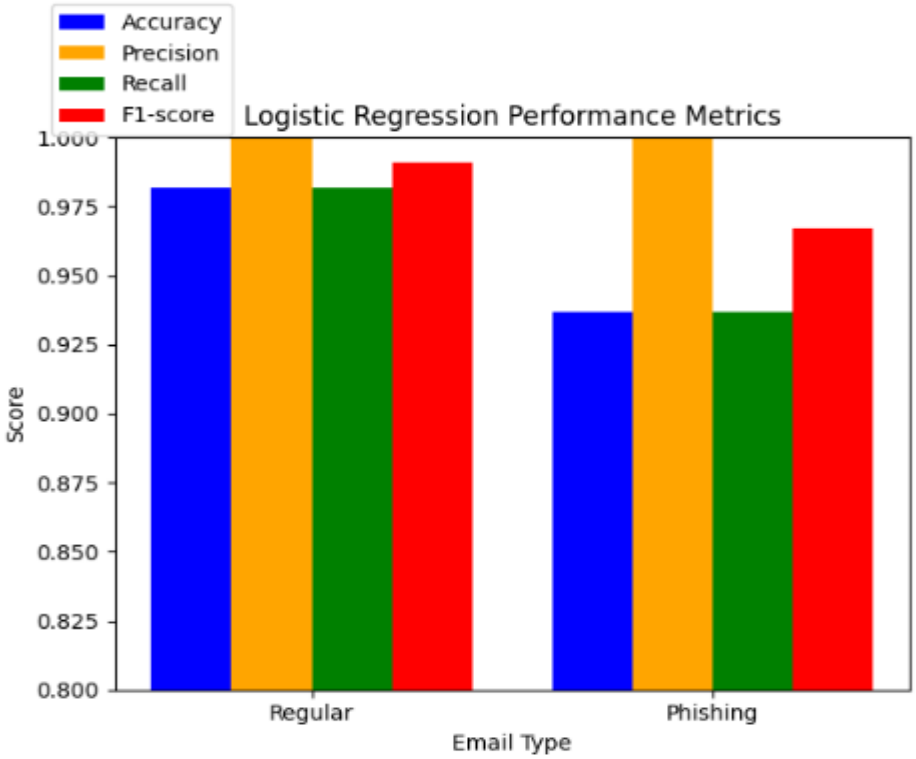

Figure 6: Logistic Regression Classification metrics (Features 1)

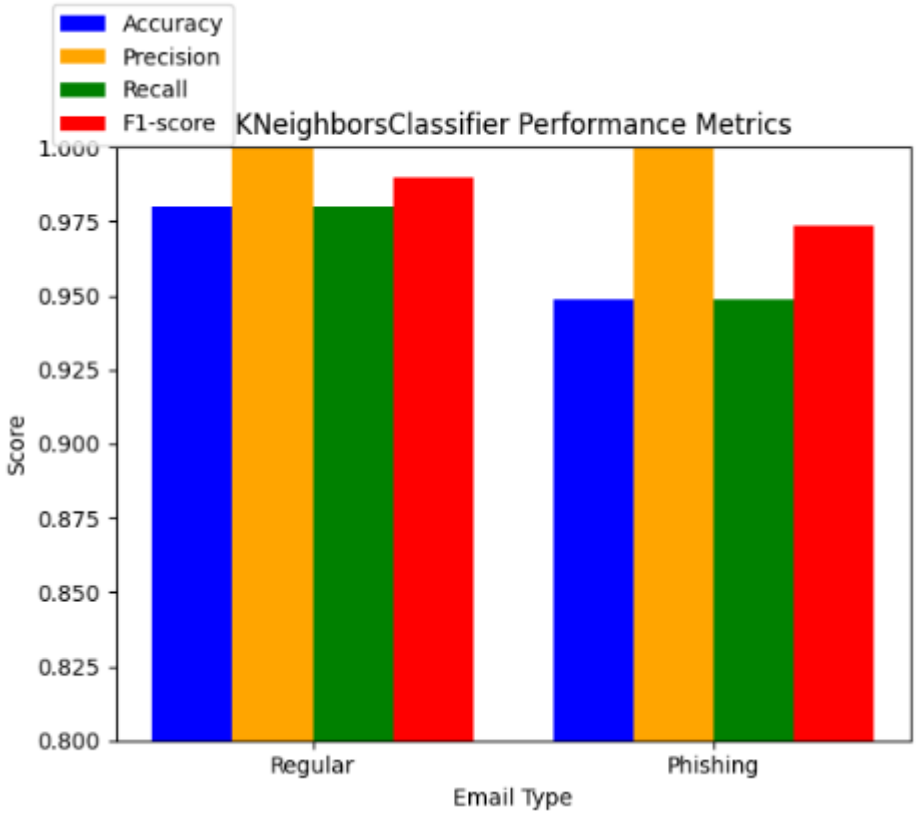

Figure 7: KNeighborsClassifier Classification metrics (Features 1)

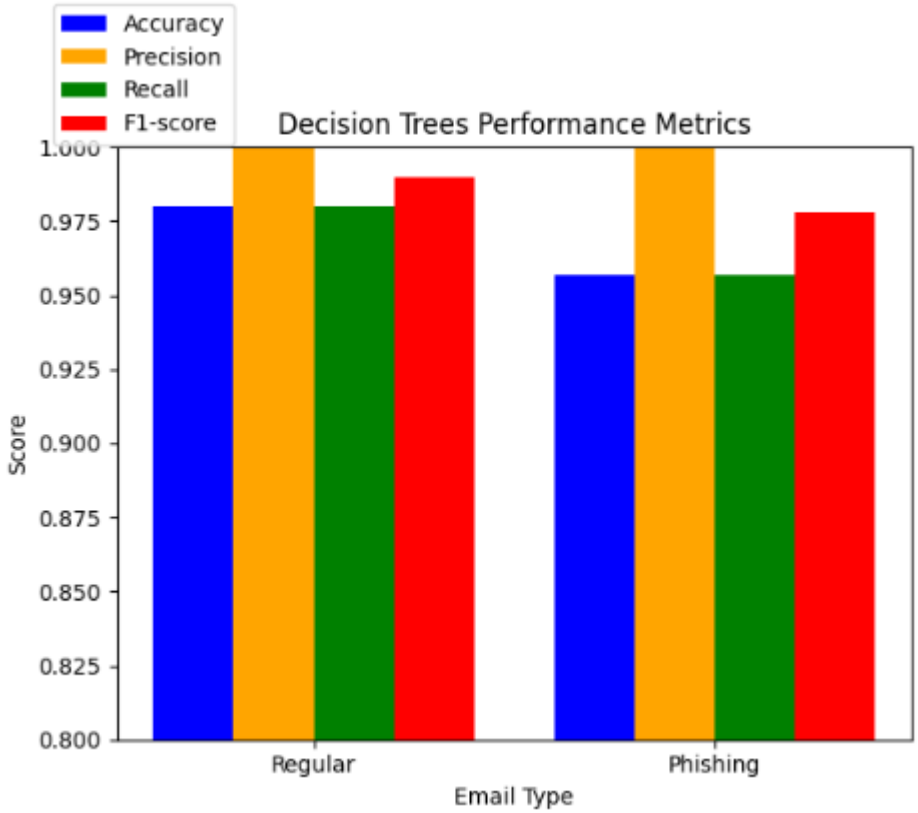

Figure 8: Decision Tress Classification metrics (Features 1)

As shown, the highest results were achieved by RF with accuracy (0.991136), precision (0.988091), recall (0.994010) and f1 score (0.991041).

5.2. Email-id features:

The algorithms were applied for the Email-id features excluding (noof_attachments) and (noof_urls). These two features were excluded because they concern the email text itself not just the Email-id. The obtained results are illustrated in Table 4 and figure 9,10,11,12,13,14.

Table 5: Features 2 results

	Accuracy	**Precision**	**Recall**	**F1 score**
Random Forest	0.918062	0.881579	0.963259	0.920611
SVM	0.896002	0.853109	0.953275	0.900415
Logistic Regression	0.897380	0.855759	0.952476	0.901531
KNeighborsClassifier	0.897380	0.891433	0.901757	0.896565
Decision Trees	0.916880	0.881040	0.961262	0.919404

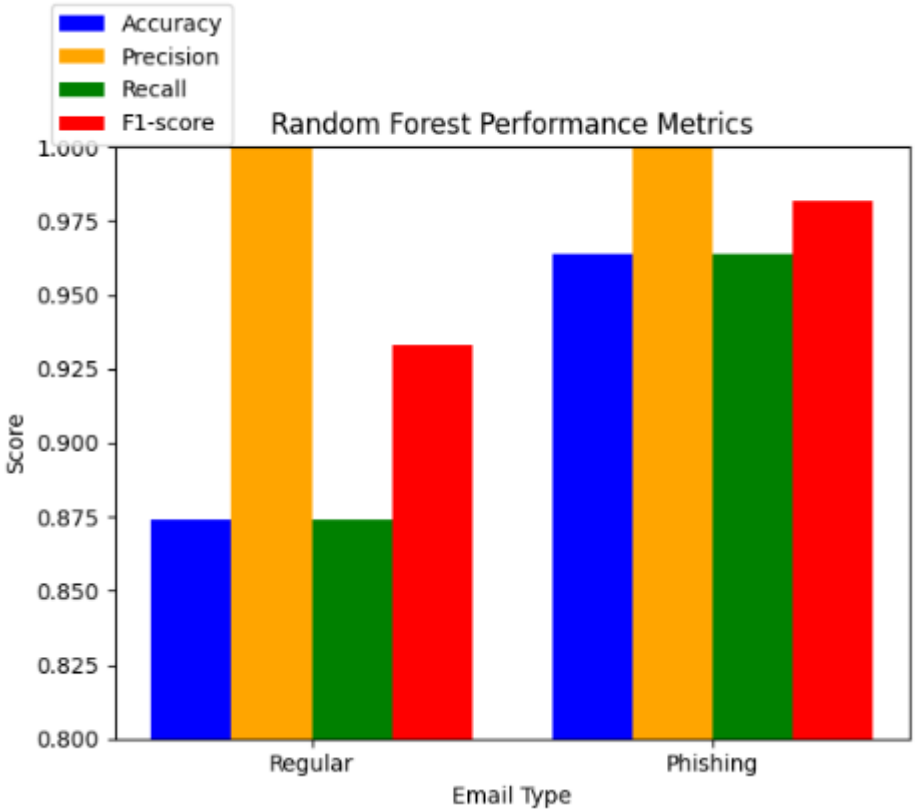

Figure 9: Random Forest Classification metrics (Features 2)

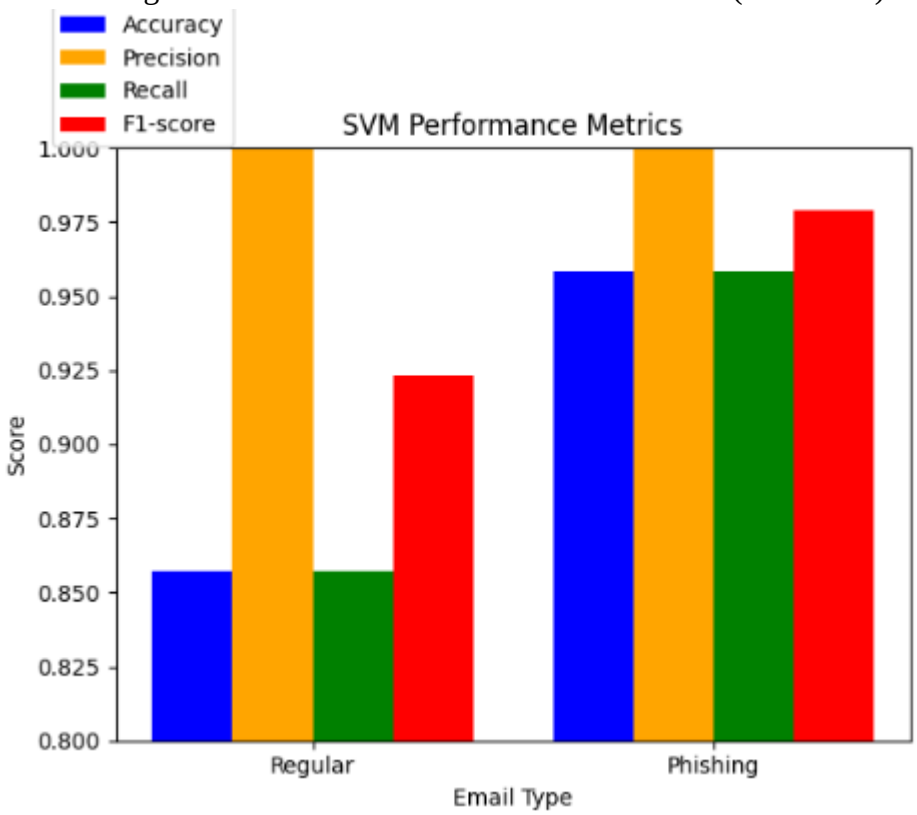

Figure 10: SVM Classification metrics (Features 2)

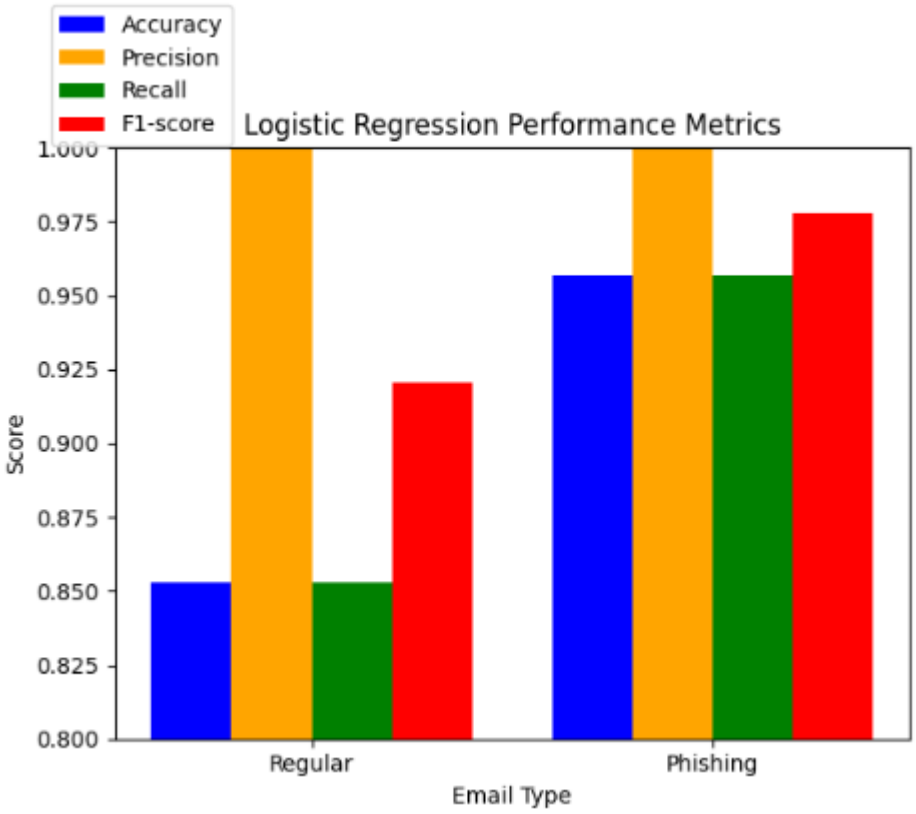

Figure 11: Logistic Regression Classification metrics (Features 2)

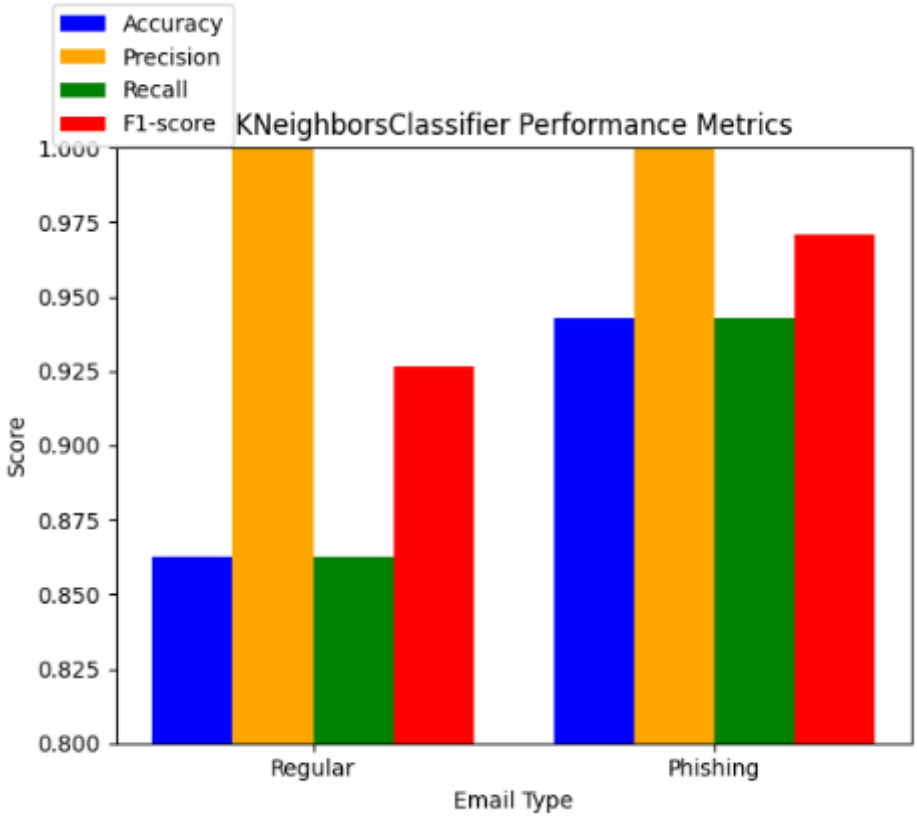

Figure 12: KNeighborsClassifier Classification metrics (Features 2)

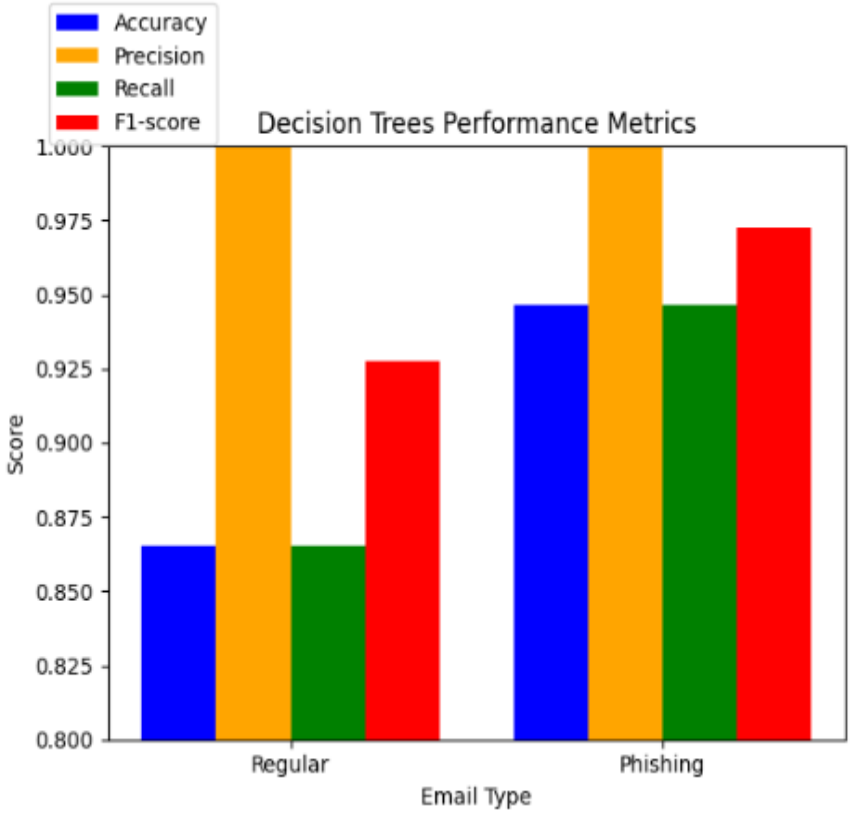

Figure 13: Decision Tress Classification metrics (Features 2)

As shown, the highest accuracy (0.918062), recall (0.963259) and f1 score (0.920611) were reached by Random Forest (RF). While the highest Precision was around 89.14 % by KNeighborsClassifier (KNN). In addition, the models achieved a high performance in detecting phishing emails (around 95 % for accuracy and recall, around 99.5 % for precision, 97% for f1 score). On the other hand, the model does not have the same ability to recognize regular emails (just 86 % for accuracy and recall, 925 for f1 score).

5.3. Email-id features combined with some URL features:

The algorithms were applied for the Email-id features excluding (noof_attachments) group (7) combined with 27 features from URL-text features and the IP feature, so the total number of features is 35. The obtained results are illustrated in Table 5 and figures 15, 16, 17,18,19,20.

Table 5: Features 3 results

	Accuracy	**Precision**	**Recall**	**F1 score**
Random Forest	0.943668	0.927196	0.961262	0.943922
SVM	0.901517	0.863043	0.951278	0.905015
Logistic Regression	0.904865	0.874954	0.941693	0.907098
KNeighborsClassifier	0.922395	0.907336	0.938498	0.922654
Decision Trees	0.939334	0.928237	0.950479	0.939227

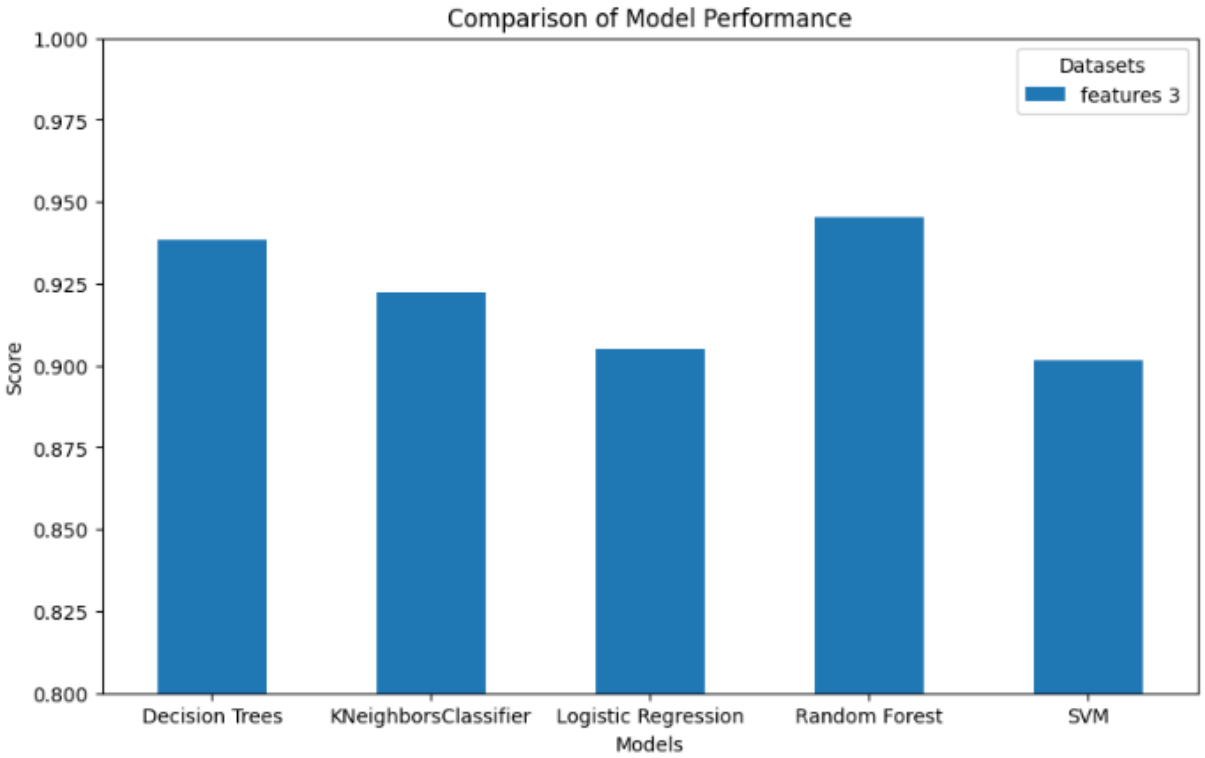

Figure 14: Comparison of Models Performance for Features 3

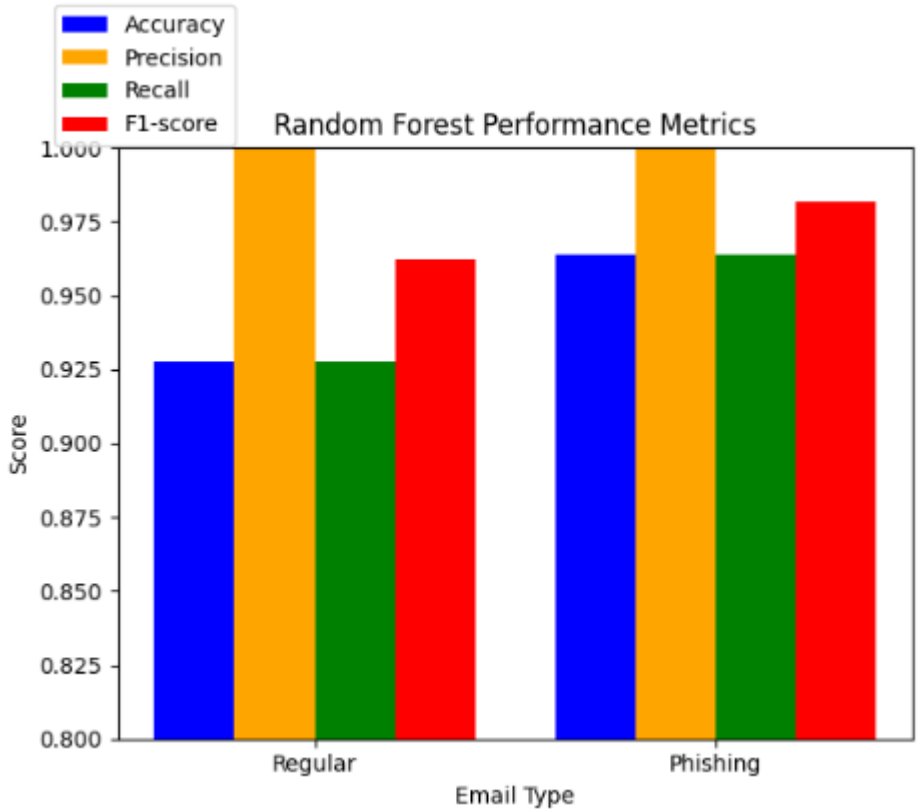

Figure 15: Random Forest Classification metrics (Features 3)

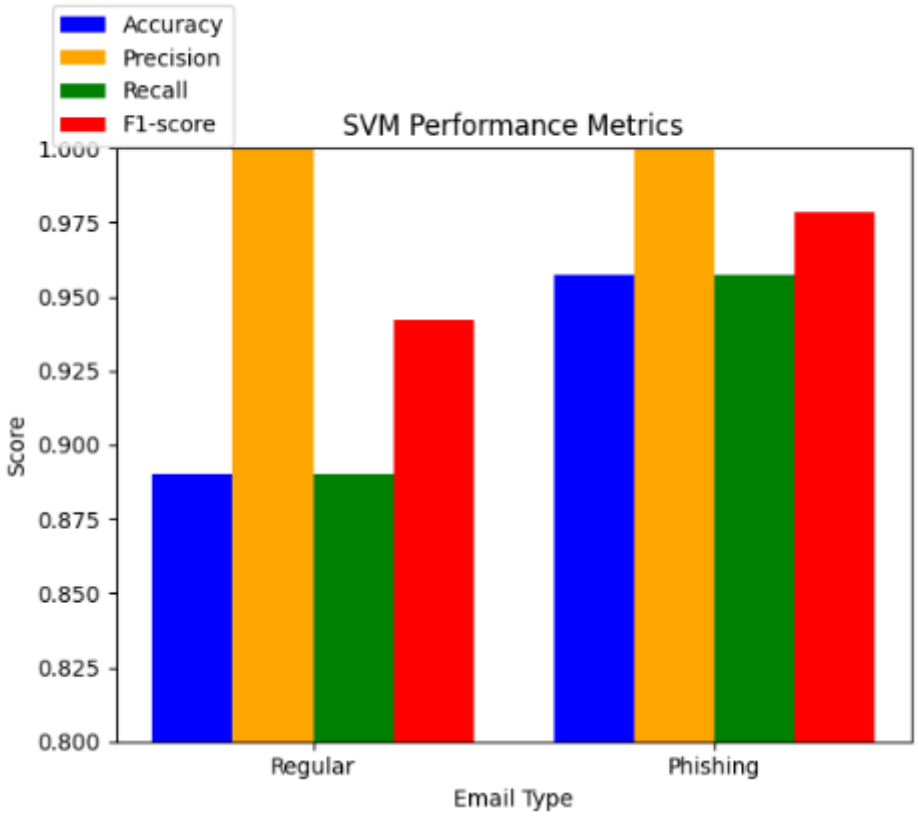

Figure 16: SVM Classification metrics (Features 3)

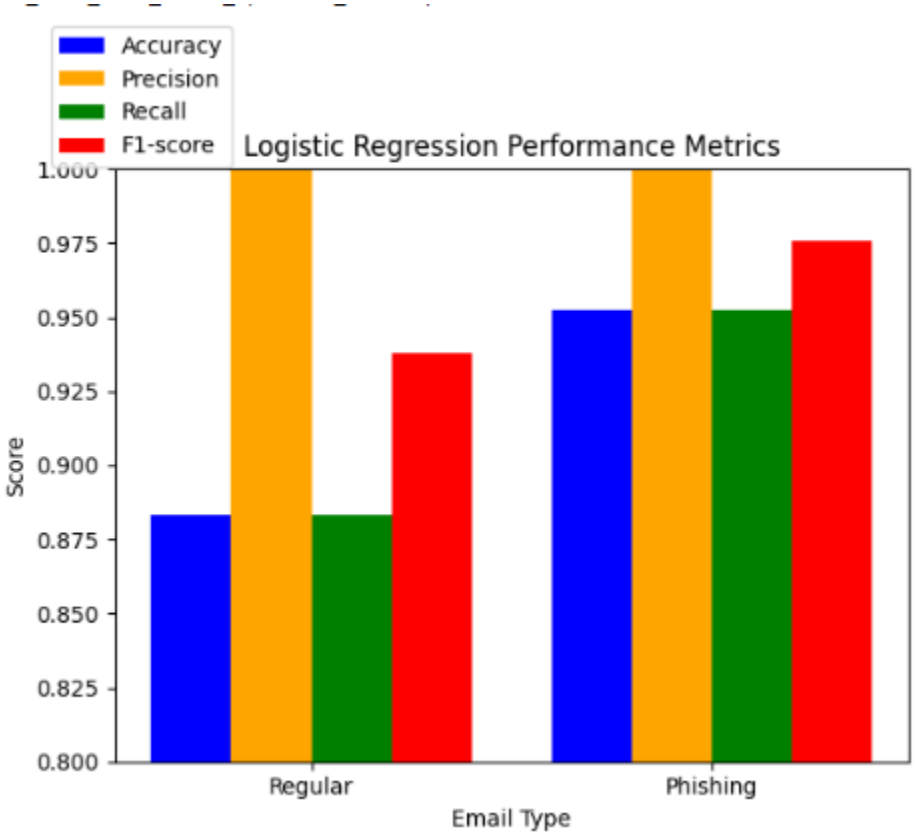

Figure 17: Logistic Regression Classification metrics (Features 3)

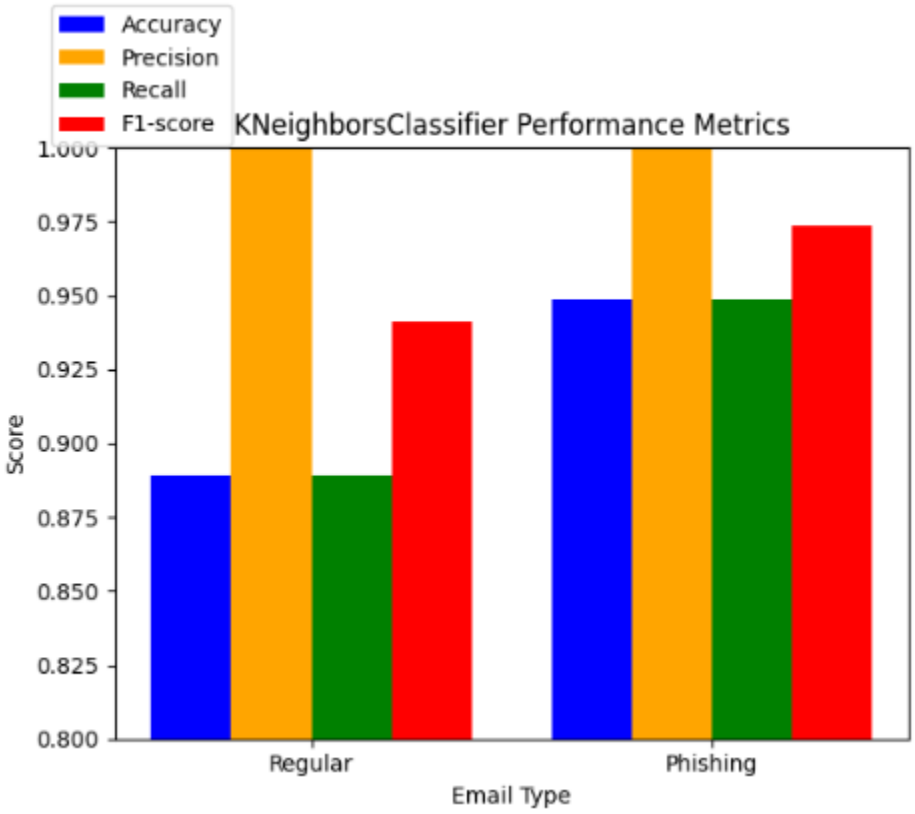

Figure 18: KNeighborsClassifier Classification metrics (Features 3)

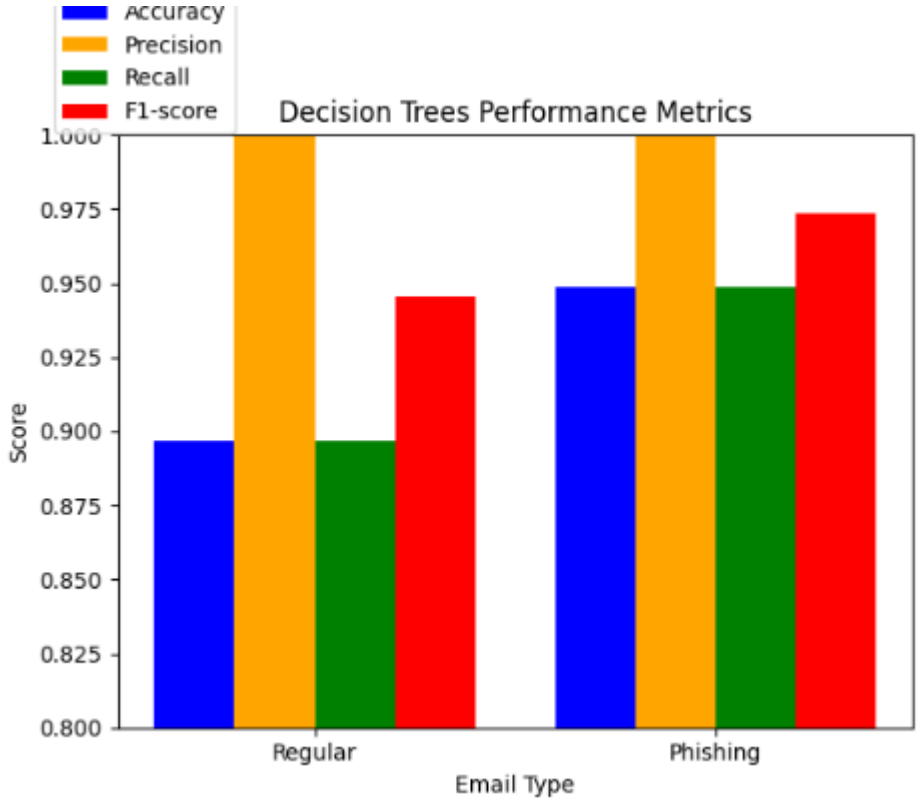

Figure 19: Decision Tress Classification metrics (Features 3)

As shown, the highest accuracy (0.943668), recall (0.961262) and f1 score (0.943922) were reached by Random Forest (RF). While the highest Precision was around 92.82 % by Decision Trees (DT). Moreover, the models are better than (features 1) in recognizing regular emails (around 93 % accuracy for RF).

5.4. URL features:

The algorithms were implemented for the URL features including VirusTotal features. The obtained results are illustrated in Table 6 and figure 21, 22, 23,24,25,26.

Table 6: Features 4 results

	Accuracy	**Precision**	**Recall**	**F1 score**
Random Forest	0.990152	0.986133	0.994010	0.990056
SVM	0.956273	0.974626	0.935703	0.954768
Logistic Regression	0.930668	0.976528	0.880591	0.926081
KNeighborsClassifier	0.982076	0.974813	0.989217	0.981962
Decision Trees	0.980894	0.974006	0.987620	0.980765

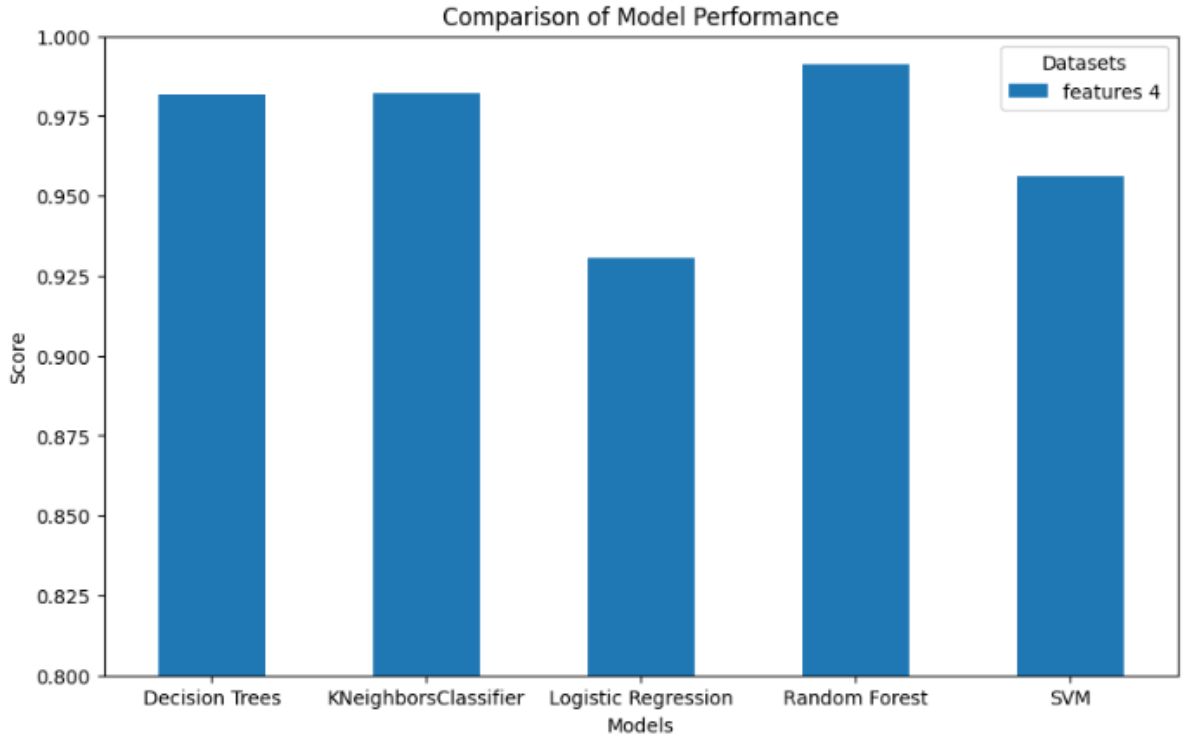

Figure 20: Comparison of Models Performance for Features 4

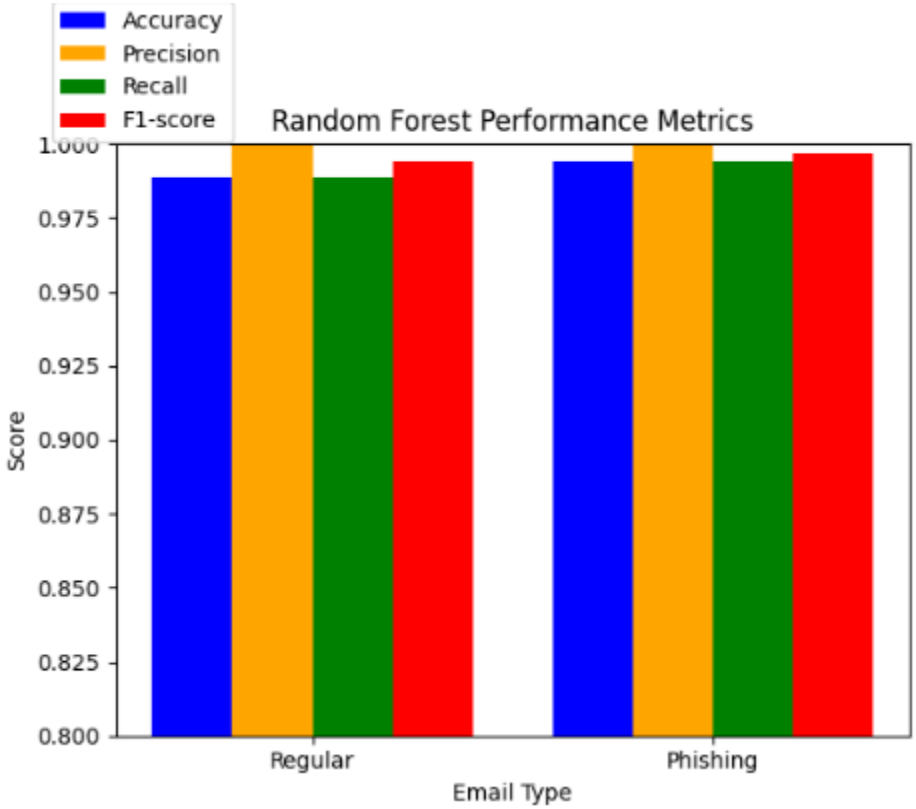

Figure 21: Random Forest Classification metrics (Features 4)

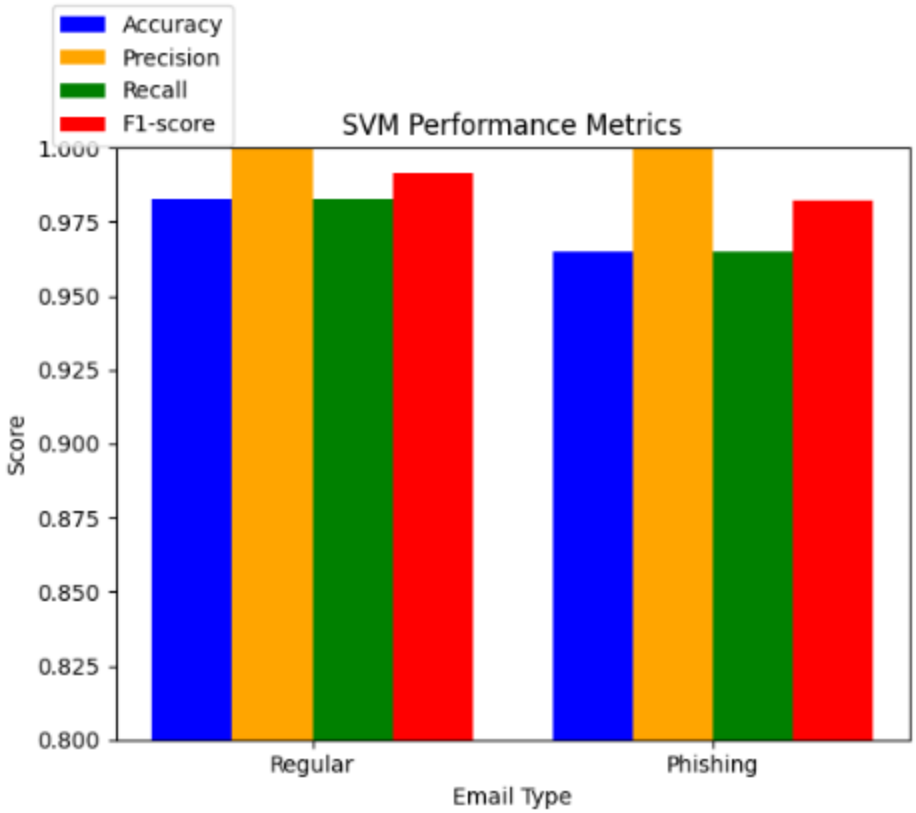

Figure 22: SVM Classification metrics (Features 4)

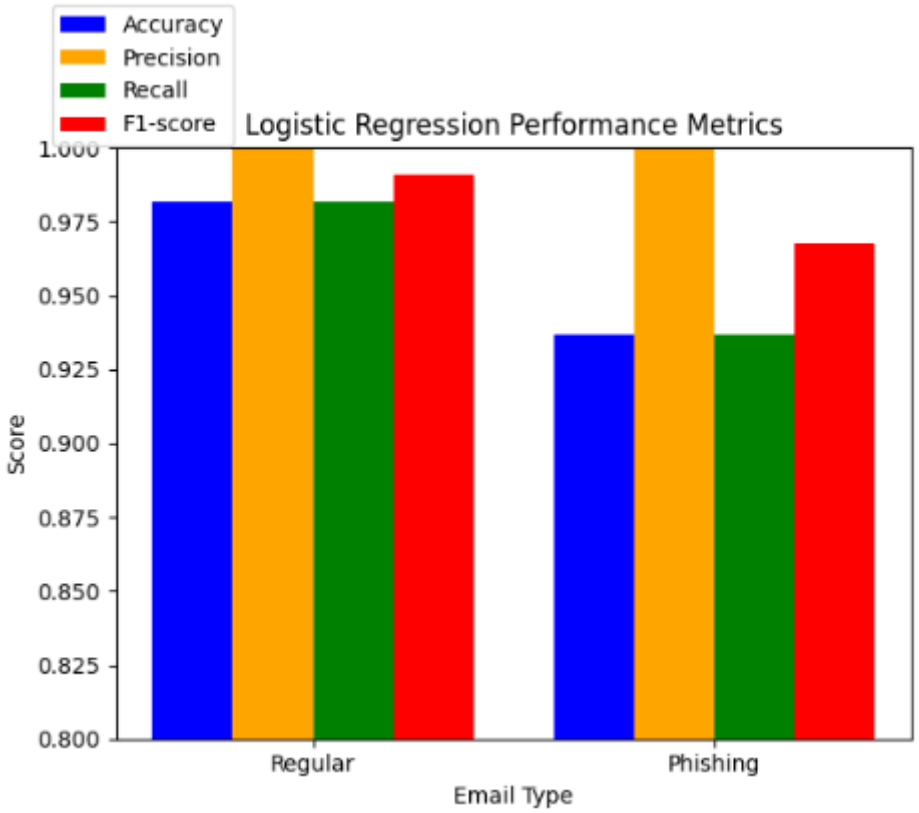

Figure 23: Logistic Regression Classification metrics (Features 4)

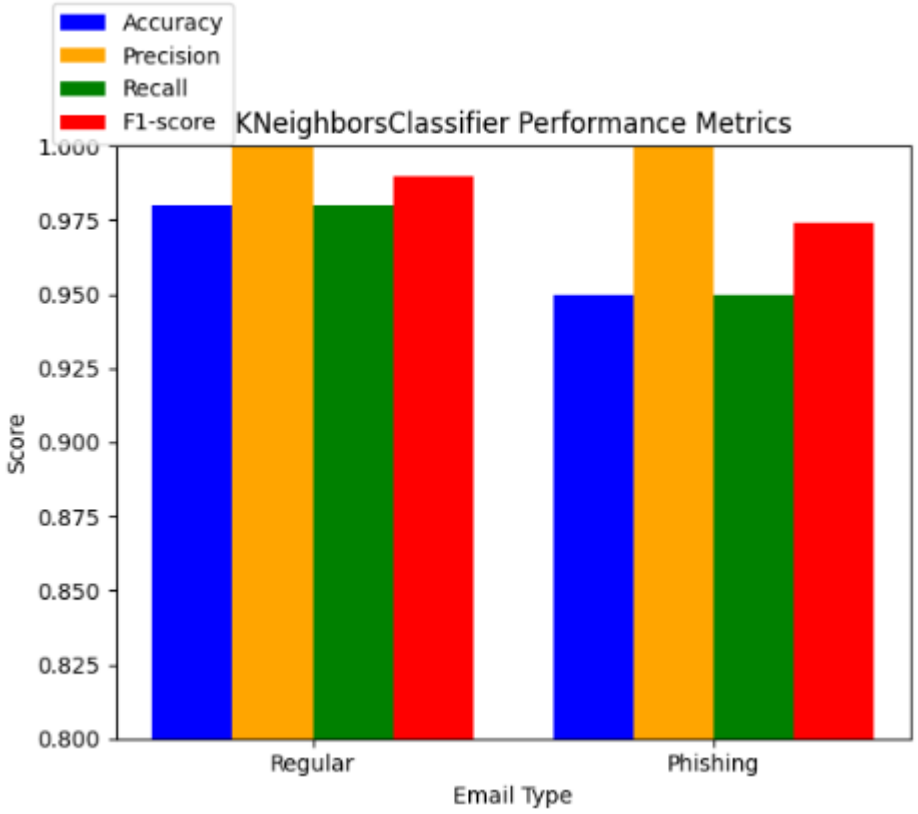

Figure 24: KNeighborsClassifier Classification metrics (Features 4)

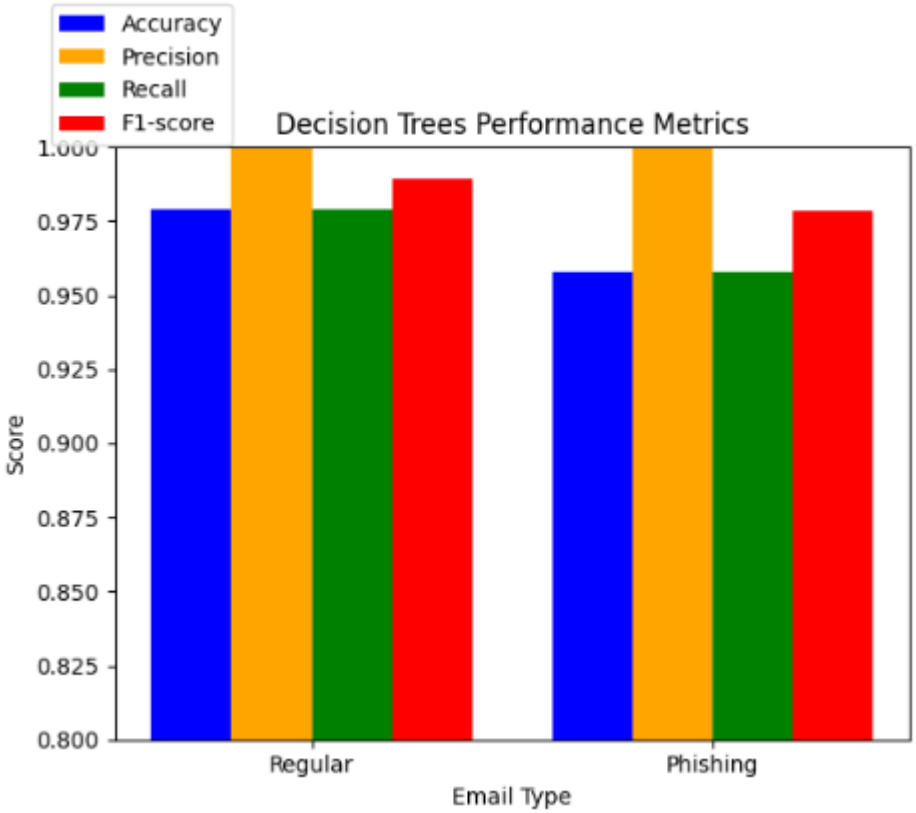

Figure 25: Decision Tress Classification metrics (Features 4)

As shown, the highest accuracy (0.990152), precision (0.986133), recall (0.994010) and f1 score (0.990056) were reached by Random Forest (RF). In addition, the models have significant results for detecting both phishing and regular emails

5.5. Email-id features combined with VirusTotal features:

The algorithms were implemented for Email-id features combined with the features exported from VirusTotal platform. The total number of features is 30. The obtained results are illustrated in Table 7 and figures 27,28,29,30,31,32.

Table 7 : Features 5 results

	Accuracy	Precision	Recall	F1 score
Random Forest	0.985227	0.975342	0.995208	0.985175
SVM	0.956667	0.974647	0.936502	0.955193
Logistic Regression	0.930668	0.976528	0.880591	0.926081
KNeighborsClassifier	0.982076	0.974813	0.989217	0.981962
Decision Trees	0.973213	0.967615	0.978435	0.972994

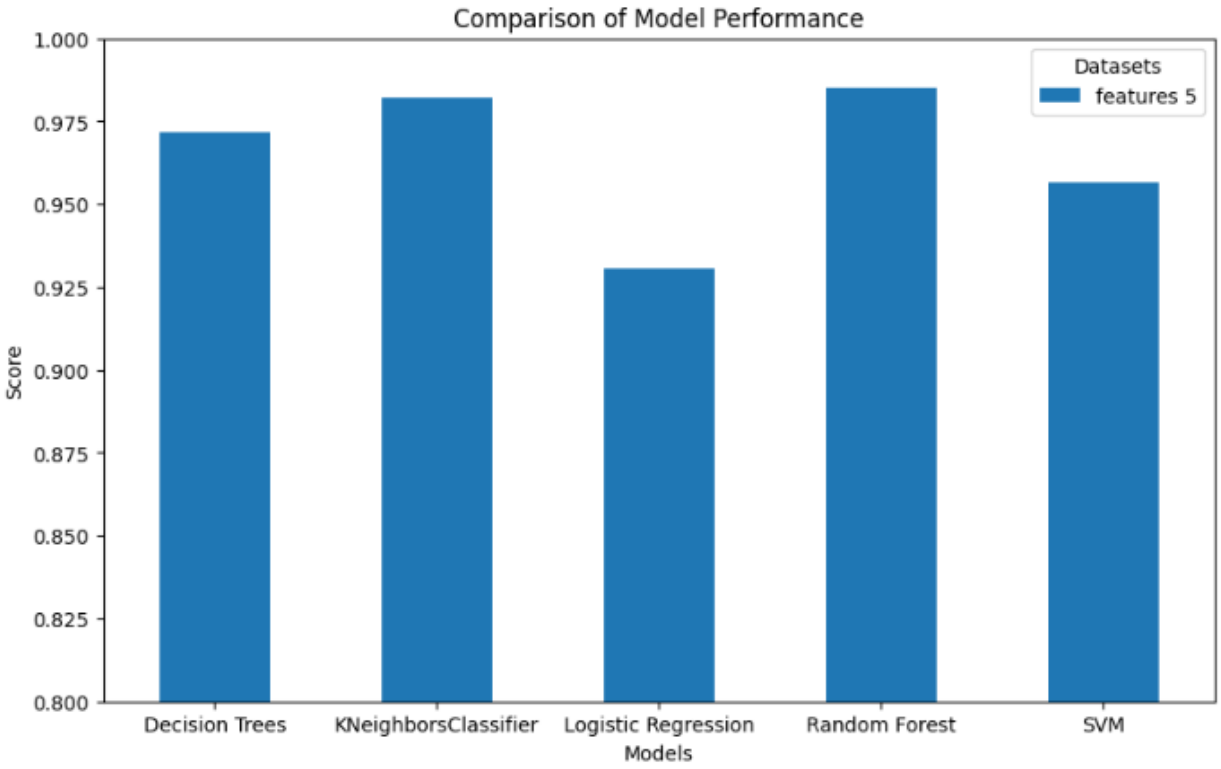

Figure 26: Comparison of Models Performance for Features 5

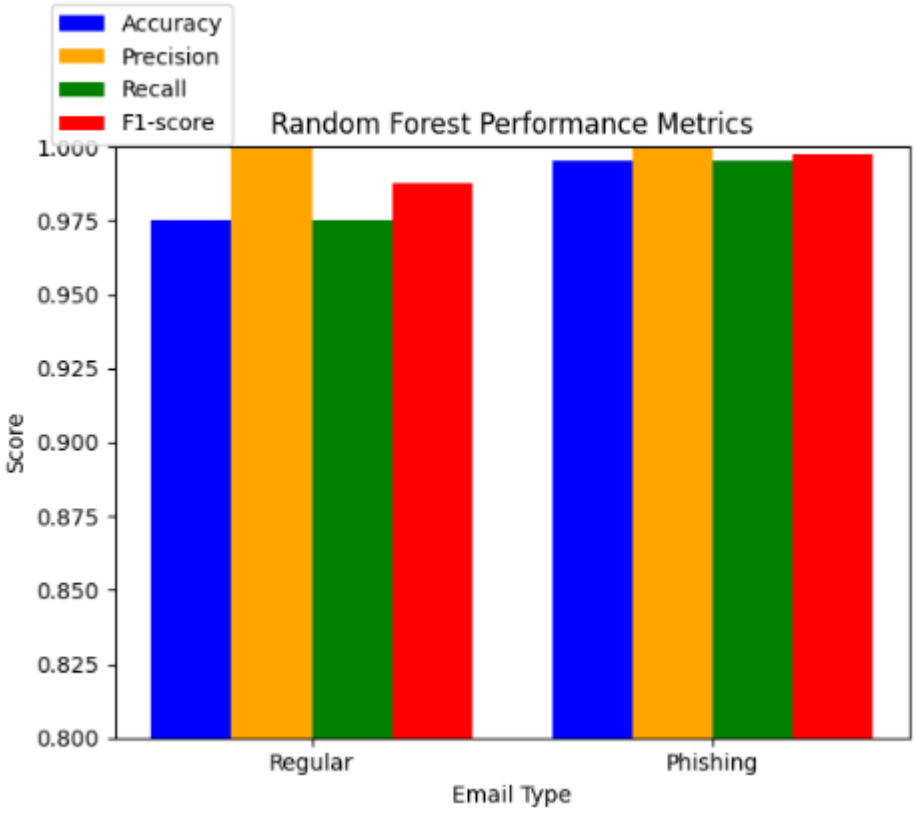

Figure 27: Random Forest Classification metrics (Features 5)

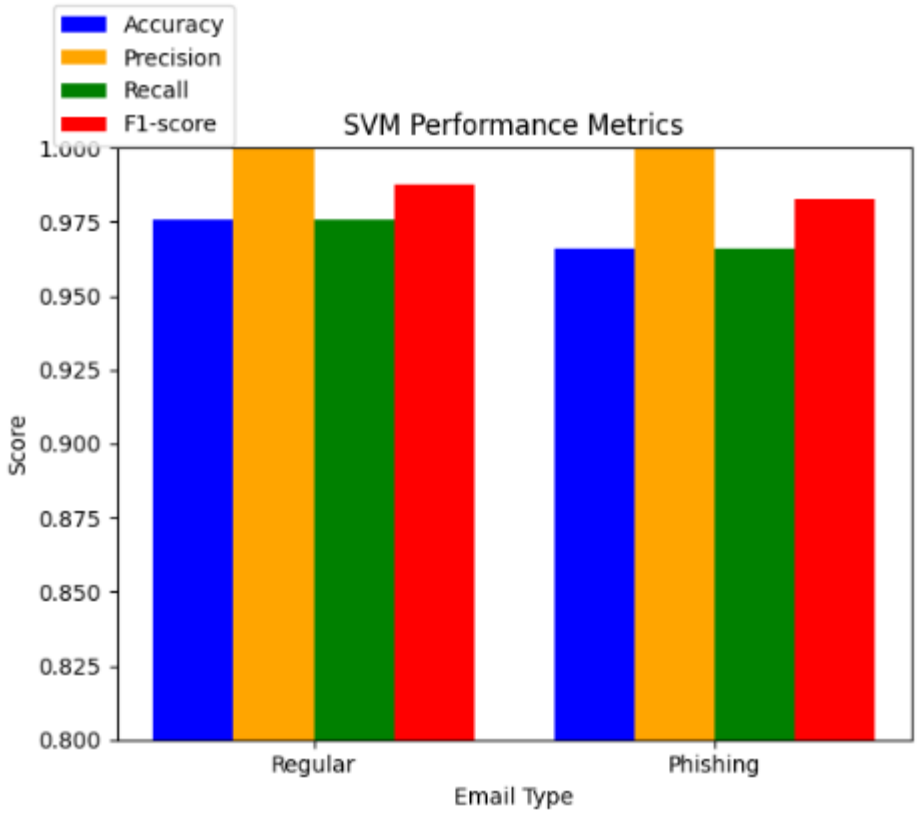

Figure 28: SVM Classification metrics (Features 5)

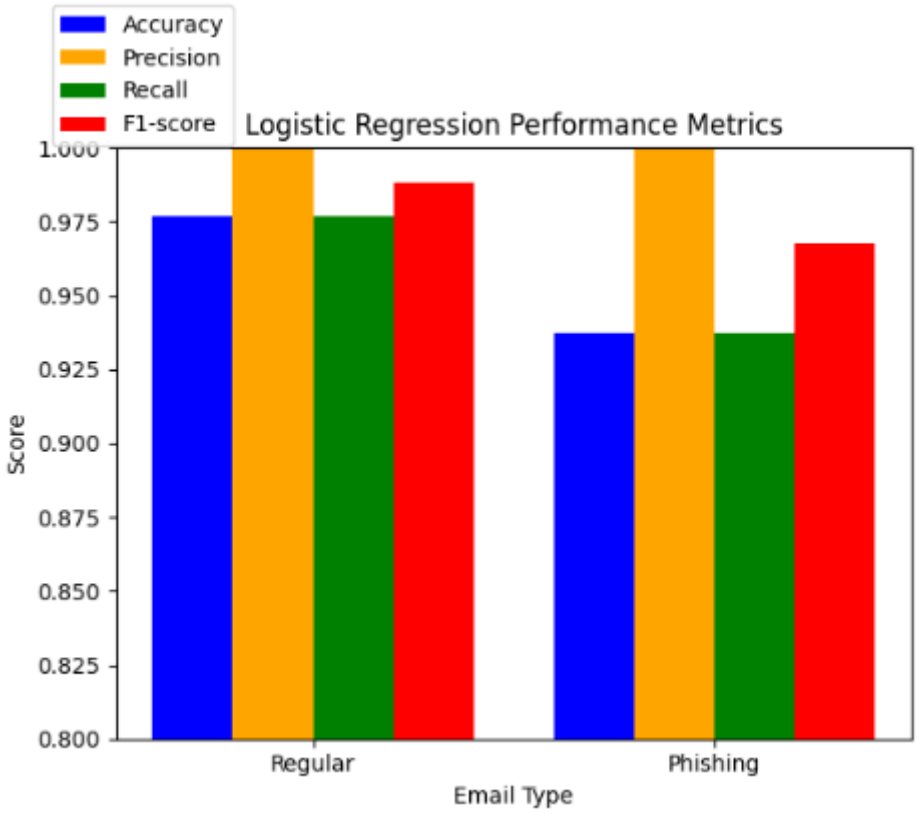

Figure 29: Logistic Regression Classification metrics (Features 5)

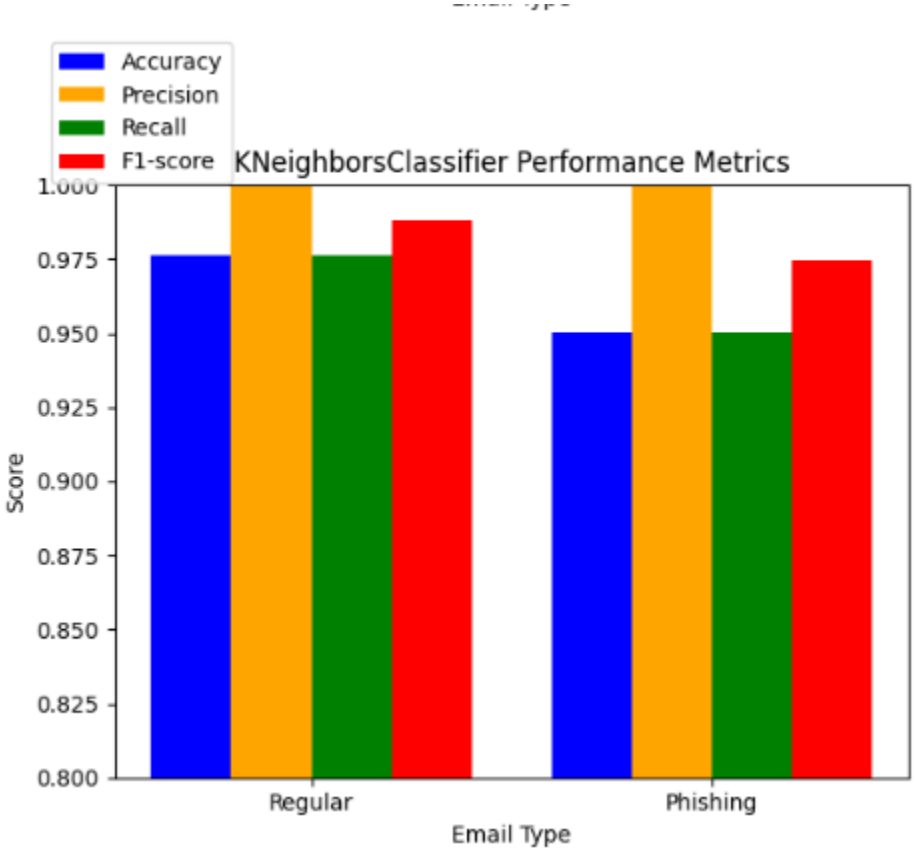

Figure 30: KNeighborsClassifier Classification metrics (Features 5)

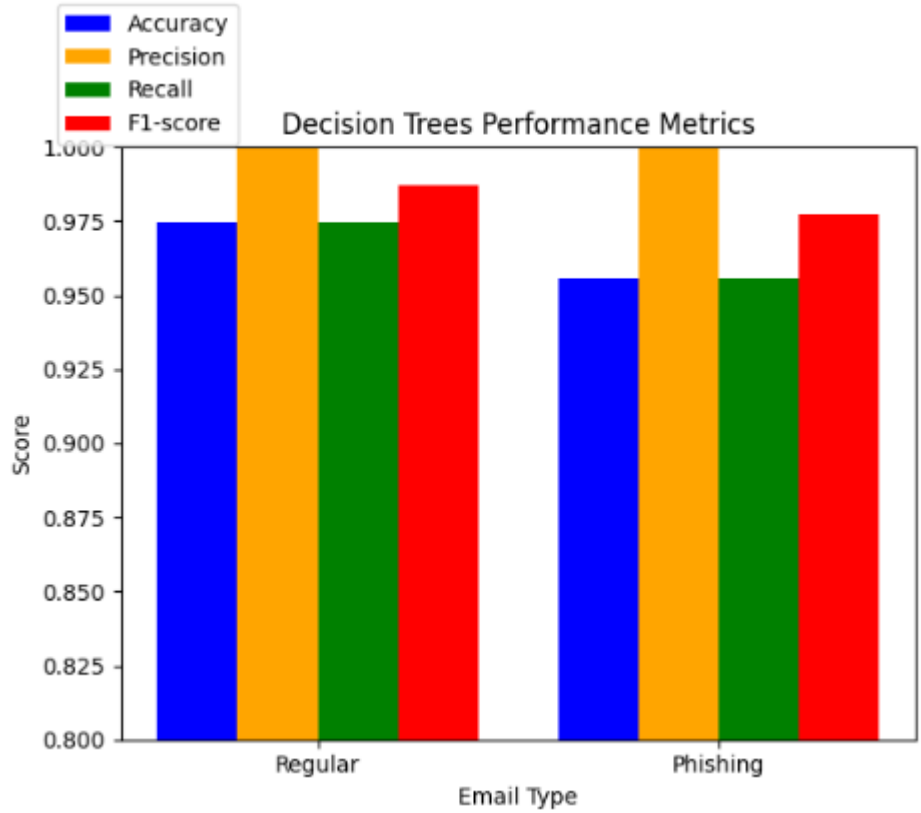

Figure 31: Decision Tress Classification metrics (Features 5)

As shown, the highest accuracy (0.991628) and recall (100%) were reached by Random Forest (RF). While the highest Precision was around 98.62 % by Decision Trees (DT). The models perform better for detecting regular emails (around 98% accuracy), but some models do not have optimal results for recognizing phishing emails.

5.6. SelectKBest and ANOVA F-value:

By applying SelectKBest algorithm with ANOVA F-value to the dataset, 10 features were selected to be used in the classification process. One of them belongs to Email-id features and the rest belong to VirusTotal urls features. The obtained results are illustrated in Table 9 and figures 33, 34, 35,36,37,38.

Table 8: Features 6 results

	Accuracy	Precision	Recall	F1 score
Random Forest	0.985031	0.974961	0.995208	0.984980
SVM	0.957061	0.974668	0.937300	0.955619
Logistic Regression	0.926925	0.976755	0.872604	0.921746
KNeighborsClassifier	0.982076	0.974813	0.989217	0.981962
Decision Trees	0.970258	0.965204	0.974840	0.969998

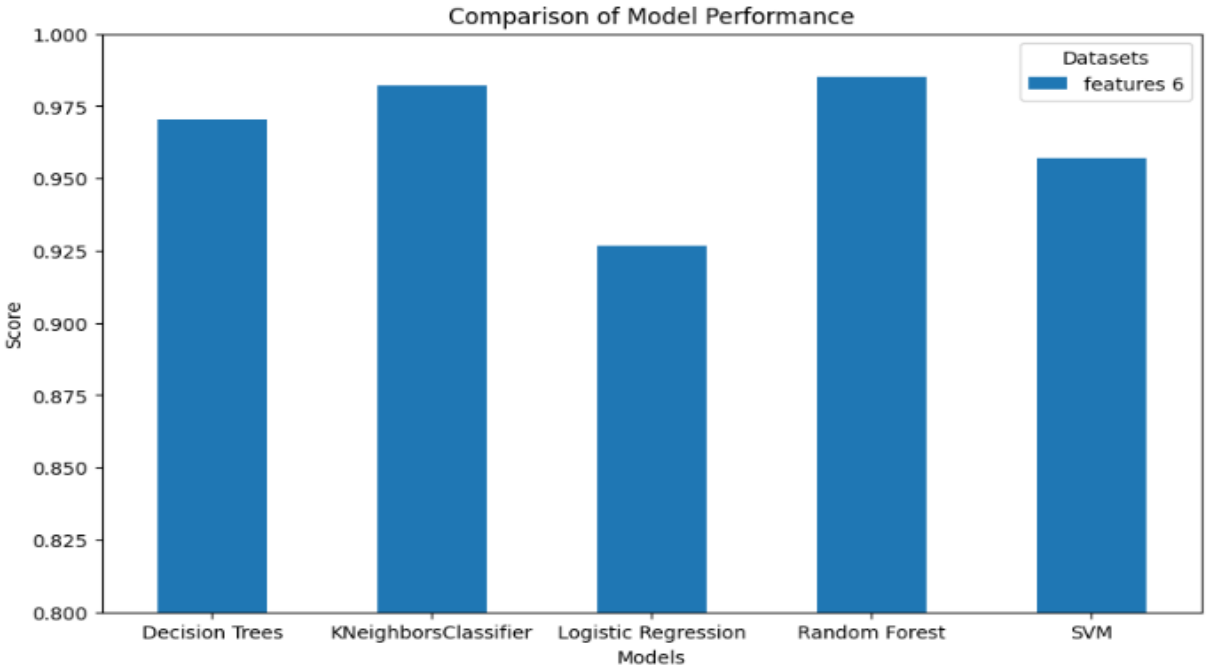

Figure 32: Comparison of Models Performance for Features 6

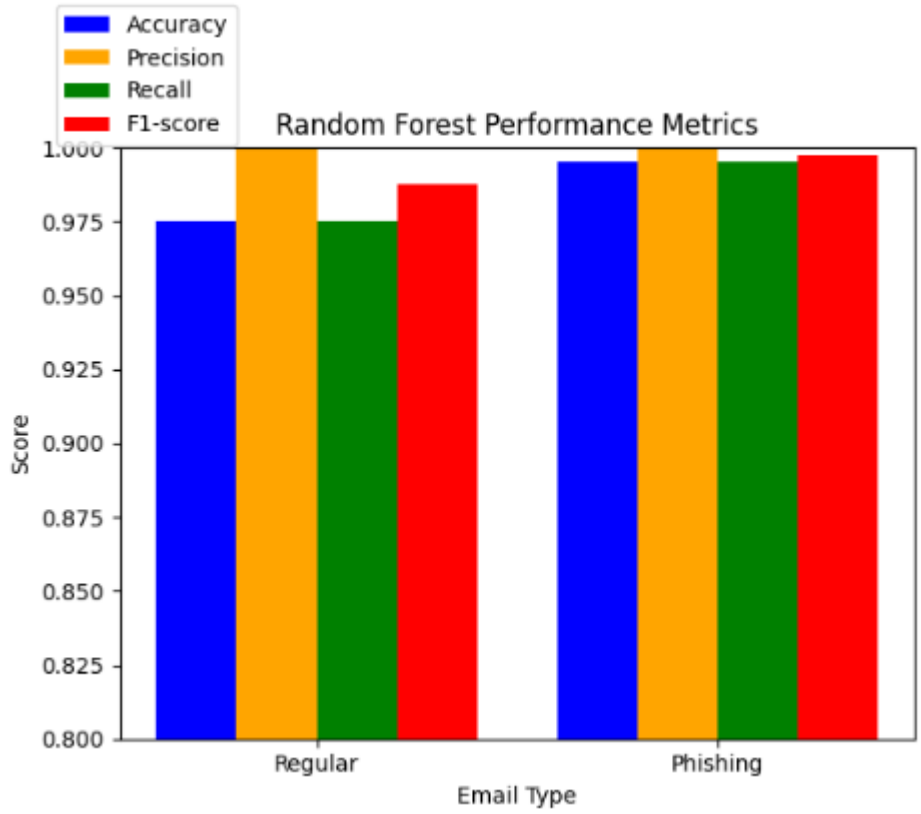

Figure 33: Random Forest Classification metrics (Features 6)

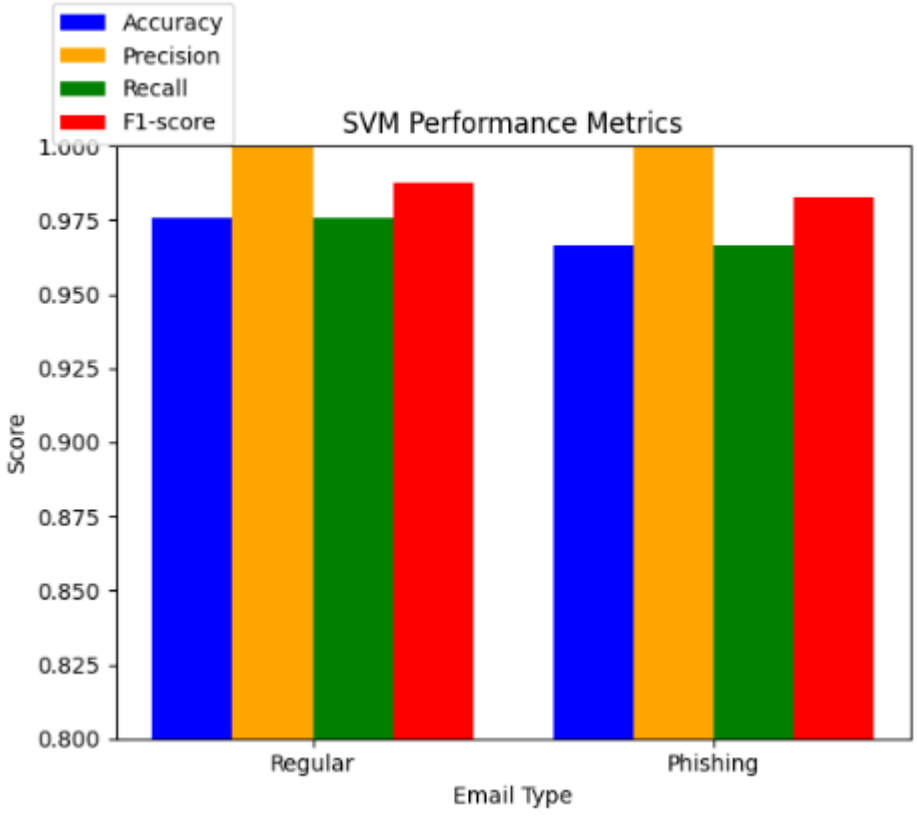

Figure 34: SVM Classification metrics (Features 6)

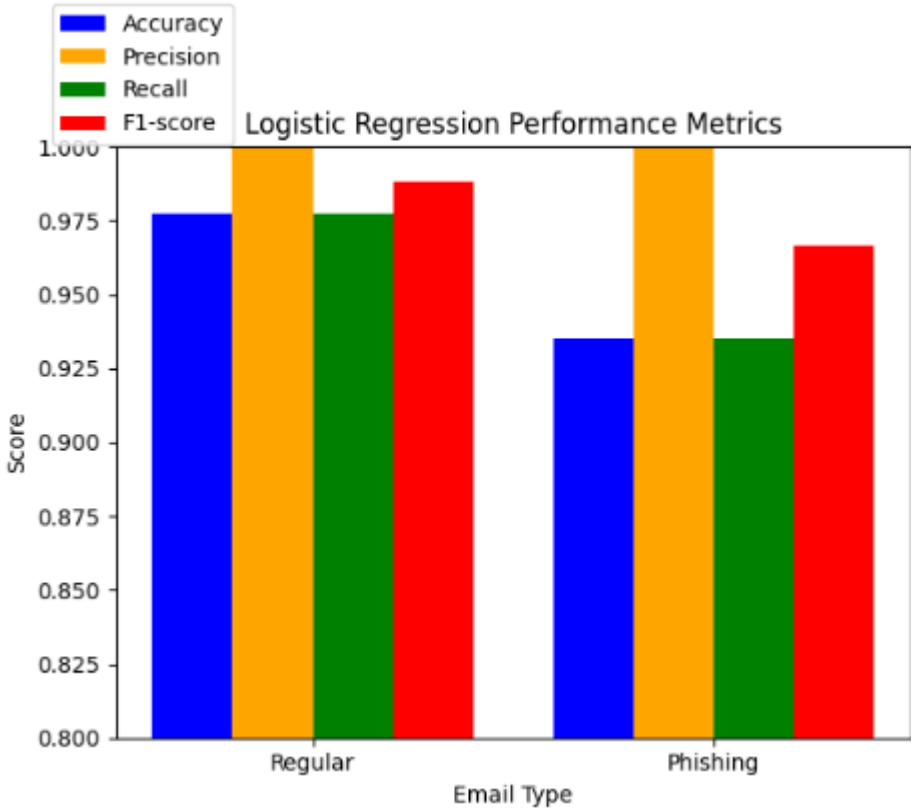

Figure 35: Logistic Regression Classification metrics (Features 6)

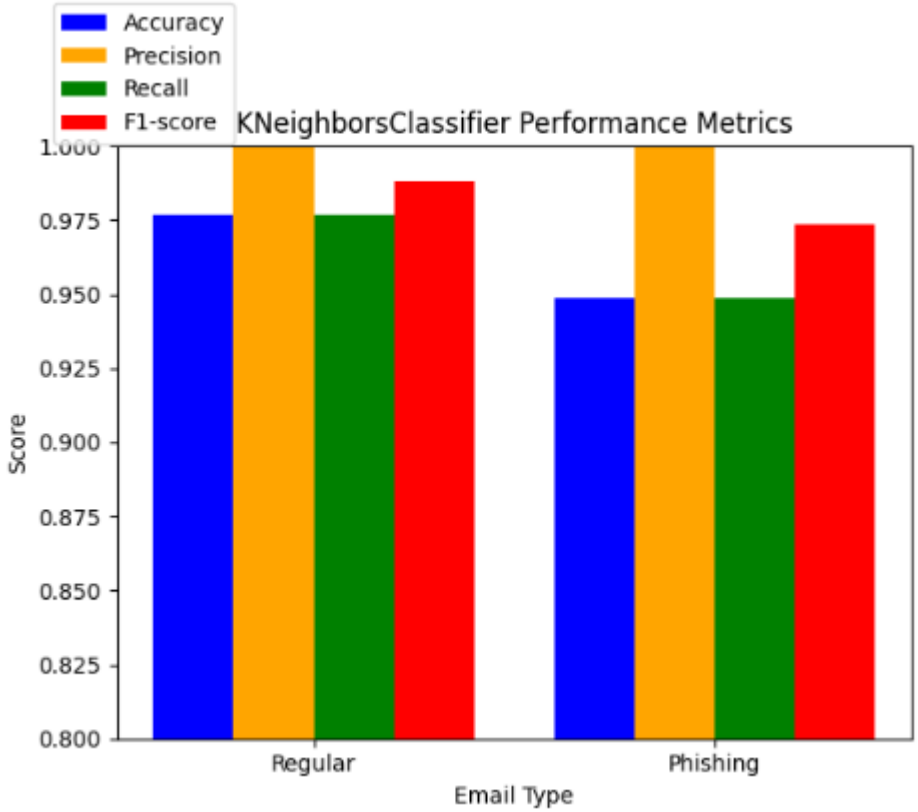

Figure 36: KNeighborsClassifier Classification metrics (Features 6)

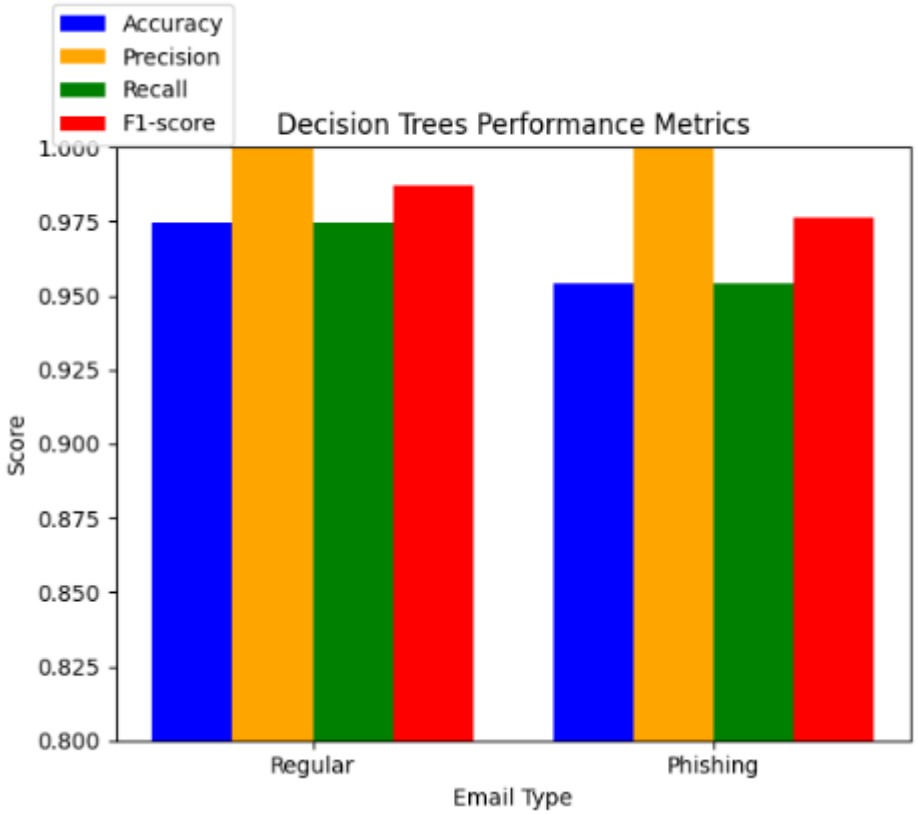

Figure 37: Decision Tress Classification metrics (Features 6)

As shown, the highest accuracy (0.991464), recall (0.995208) and f1 score (0.984980) were reached by Random Forest (RF). While the highest Precision was around 97.67 % by logistic Regression (LR). The results for RF is acceptable for both classes (around 97.5% for normal and 99% for phishing).

5.7. Principal Component Analysis:

PCA method was used to select features from the dataset. The total number of selected features is 30, which included the Email-id features, some of urls features and most of VirusTotal features. The obtained results are illustrated in Table 10 and figure 39, 40, 41,42,43,44.

Table 9: Features 7 results

	Accuracy	Precision	Recall	F1 score
Random Forest	0.985227	0.975342	0.995208	0.985175
SVM	0.956667	0.974647	0.936502	0.955193
Logistic Regression	0.930668	0.976950	0.880192	0.926050
KNeighborsClassifier	0.982076	0.974813	0.989217	0.981962
Decision Trees	0.973016	0.966864	0.978834	0.972812

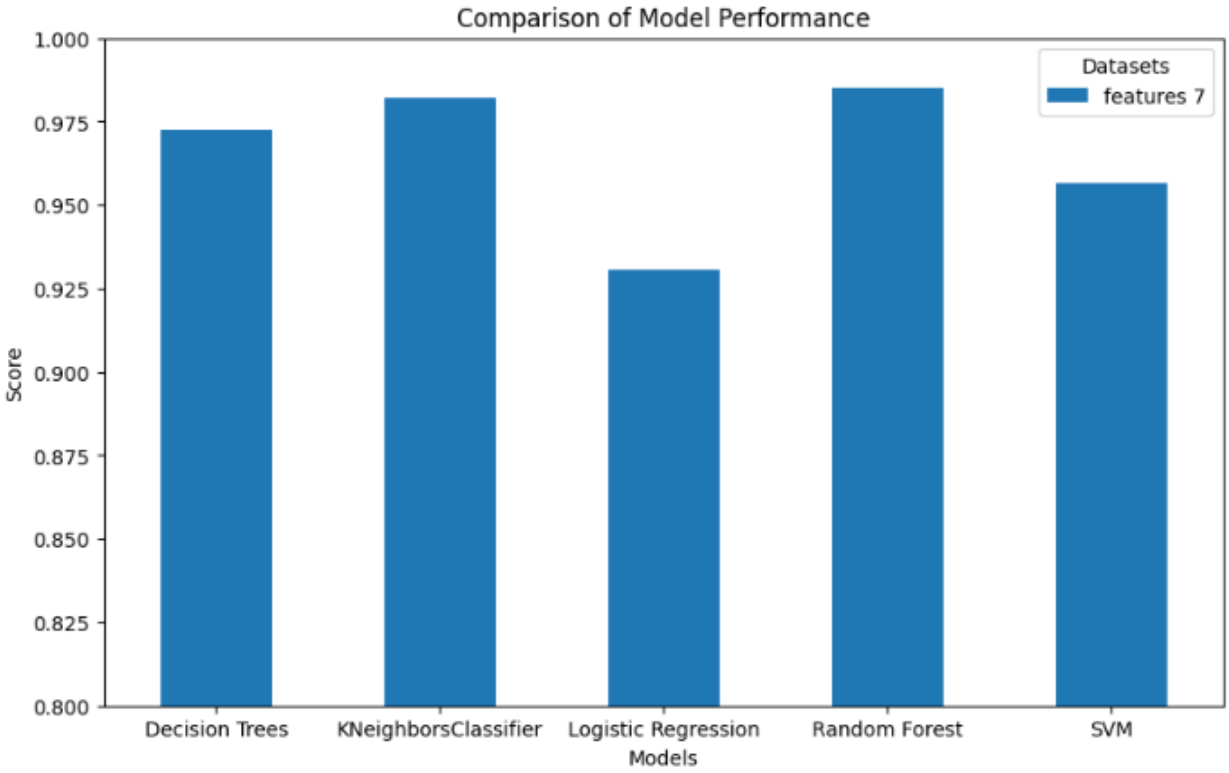

Figure 38: Comparison of Models Performance for Features 7

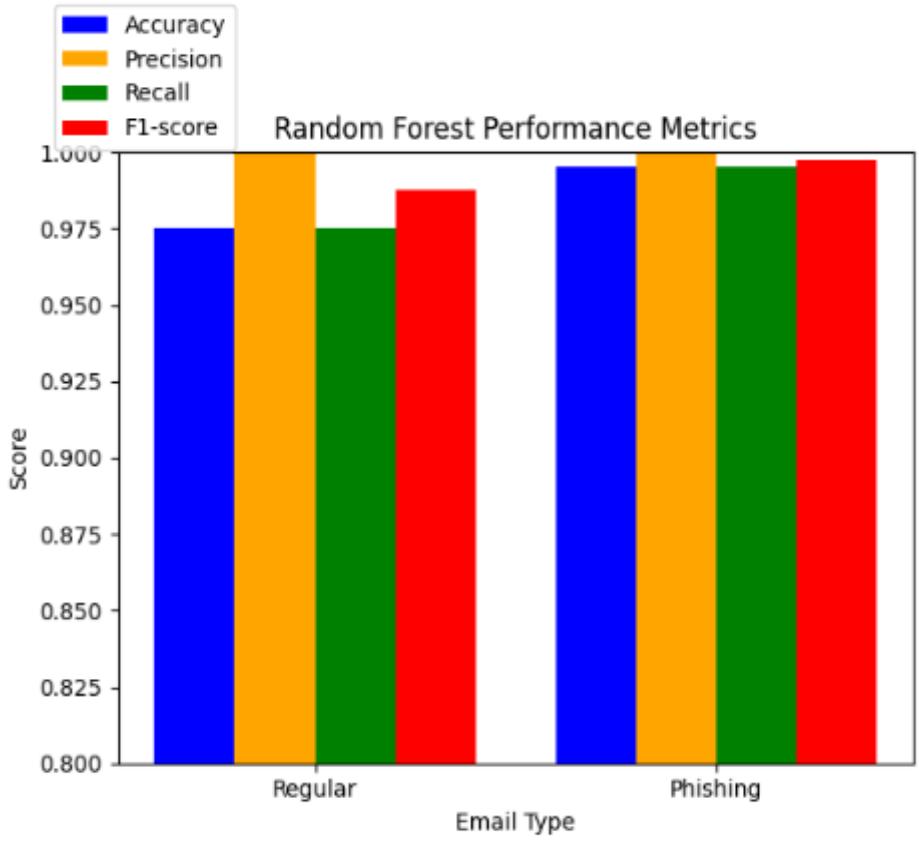

Figure 39: Random Forest Classification metrics (Features 7)

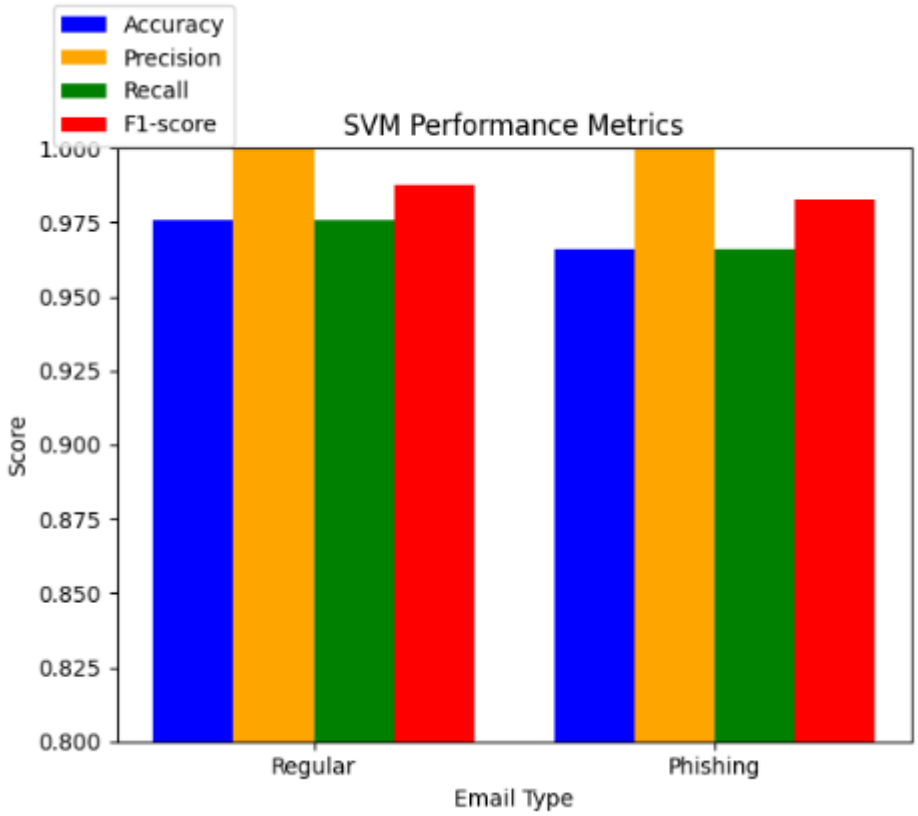

Figure 40: SVM Classification metrics (Features 7)

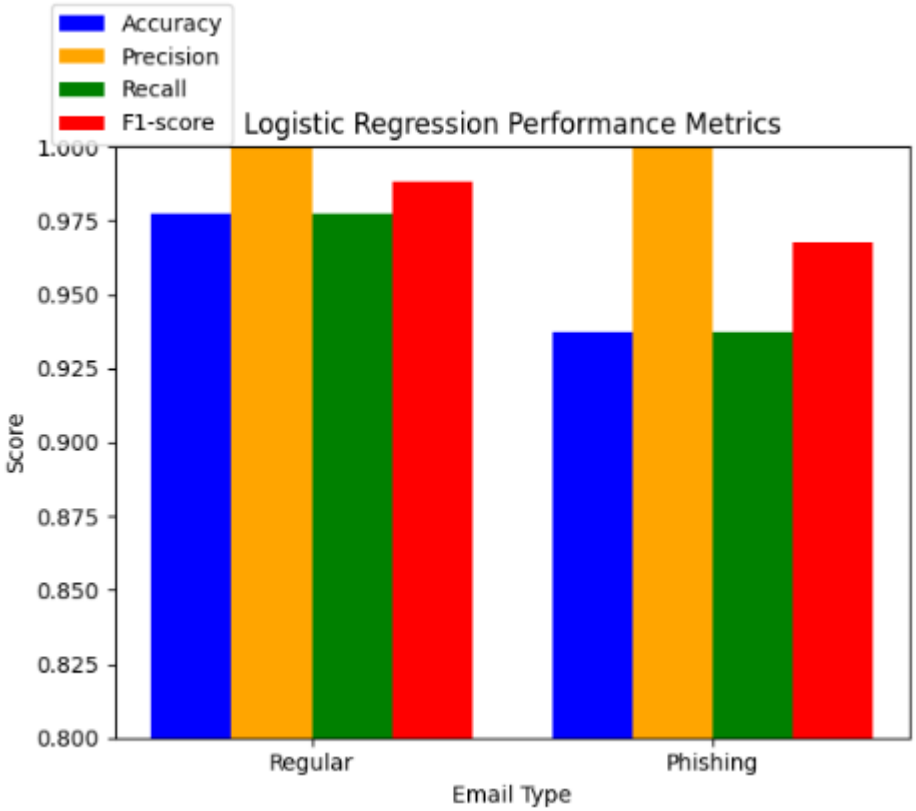

Figure 41: Logistic Regression Classification metrics (Features 7)

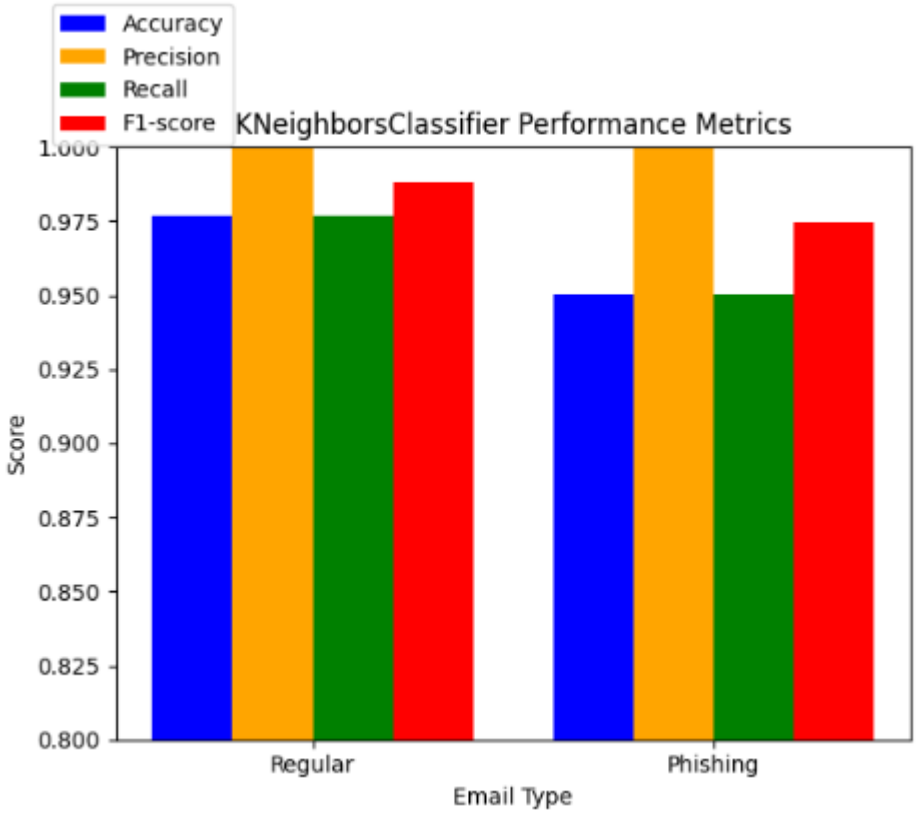

Figure 42: KNeighborsClassifier Classification metrics (Features 7)

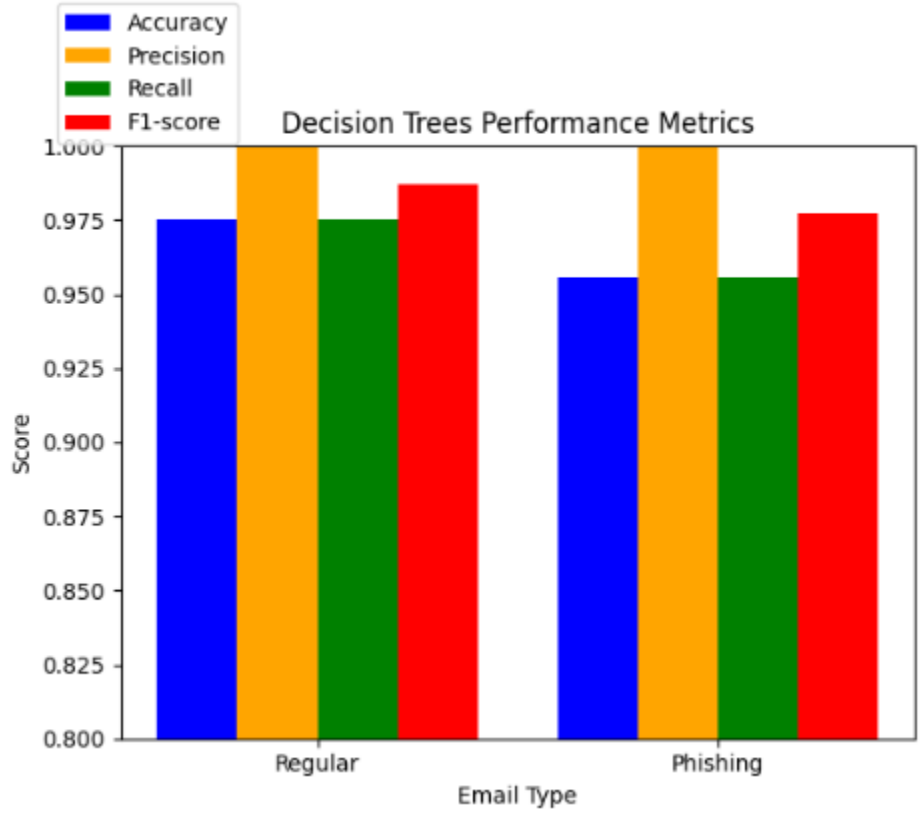

Figure 43: Decision Tress Classification metrics (Features 7)

As shown, the highest accuracy (0.991464), recall (100%) and f1 score (0.985227) were reached by Random Forest (RF). While the highest Precision was around 97.69 % by Decision Trees (DT). There are not crucial changes in models performance comparing to the previous group in terms of classes recognition (features 6).

5.8. Discussion:

As shown in table 1 for the performed experiments, most of the algorithms showed a significant performance. Especially, Random forest (RF) and Decision Trees (DT) reached the highest values for all the metrics. The highest accuracy was reached in experiments 1 and 4 because all urls features including VirusTotal features were used. Moreover, algorithms achieved significant results in experiments 5, 6 and 7 where some of VirusTotal features were used. On the other hand, the classifiers had not performed in experiments 2 and 3 like the other experiments because the number of features is not enough and features do not include urls features. In terms of classes, most of the models in all features groups performed better in detecting regular emails (most accuracy values are around 97 %) than phishing emails (around 95 % for most experiments). This problem occurs because of limited data and the features selected. As a result, more features should be extracted and used in the classification process.

Table 10: Comparison of experiments results

	Ex 1	**Ex 2**	**Ex 3**	**Ex 4**	**Ex 5**	**Ex 6**	**Ex 7**
Random Forest	0.991136	0.918062	0.943668	0.990152	0.985227	0.985031	0.9852
SVM	0.956076	0.896002	0.901517	0.956273	0.956667	0.957061	0.9566
Logistic Regression	0.930668	0.897380	0.904865	0.930668	0.930668	0.926925	0.9306
KNeighborsClassifier	0.980303	0.897380	0.922395	0.982076	0.982076	0.982076	0.9820
Decision Trees	0.983455	0.916880	0.939334	0.980894	0.973213	0.970258	0.9730

References:

Carnegie Mellon University. .(2010 ,9 4) *Principal Components Analysis - Statistics & Data Science.* تم الاسترداد من Carnegie Mellon University: https://www.stat.cmu.edu/~cshalizi/uADA/12/lectures/ch18.pdf

Saturn Cloud. 10) July, 2023). *How to Perform Feature Selection in Multiclass Logistic Regression in Python* تم الاسترداد من .SaturnCloud: https://saturncloud.io/blog/how-to-perform-feature-selection-in-multiclass-logistic-regression-in-python/

UNIVERSITY of WISCONSIN–MADISON. .(2003) *Evaluating Machine Learning Methods.* تم الاسترداد من Computer Sciences,School of Computer, Data & Information Sciences: https://pages.cs.wisc.edu/~dpage/cs760/evaluating.pdf

Dataset documentation:

1. Phishing corpus - Nazario. phishingcorpus homepage, Apr. 2006. http://monkey.org/%7Ejose/wiki/doku.php?id=PhishingCorpus ↩
2. https://www.kaggle.com/code/riyapatel1697/phishing-email-detection-ai-ml/input?select=emails-phishing-nazario.mbox
3. https://www.kaggle.com/datasets/rtatman/fraudulent-email-corpus
4. https://www.cs.cmu.edu/~./enron/

Figure 1: Class Diagram for Work Methodology 1
Figure 2: Confusion Matrix 9
Figure 3: Comparasion of Models Performance for Features 1 10
Figure 4: Random Forest Classification metrics (Features 1) 11
Figure 5: SVM Classification metrics (Features 1) 11
Figure 6: Logistic Regression Classification metrics (Features 1) 12
Figure 7: KNeighborsClassifier Classification metrics (Features 1) 12
Figure 8: Decision Tress Classification metrics (Features 1) 13
Figure 10: Random Forest Classification metrics (Features 2) 14
Figure 11: SVM Classification metrics (Features 2) 14
Figure 12: Logistic Regression Classification metrics (Features 2) 15
Figure 13: KNeighborsClassifier Classification metrics (Features 2) 15
Figure 14: Decision Tress Classification metrics (Features 2) 16
Figure 15: Comparison of Models Performance for Features 3 17
Figure 16: Random Forest Classification metrics (Features 3) 17
Figure 17: SVM Classification metrics (Features 3) 18
Figure 18: Logistic Regression Classification metrics (Features 3) 18
Figure 19: KNeighborsClassifier Classification metrics (Features 3) 19
Figure 20: Decision Tress Classification metrics (Features 3) 19
Figure 21: Comparison of Models Performance for Features 4 20
Figure 22: Random Forest Classification metrics (Features 4) 21

Figure 23: SVM Classification metrics (Features 4)..... 21
Figure 24: Logistic Regression Classification metrics (Features 4) 22
Figure 25: KNeighborsClassifier Classification metrics (Features 4) 22
Figure 26: Decision Tress Classification metrics (Features 4) 23
Figure 27: Comparison of Models Performance for Features 5..... 24
Figure 28: Random Forest Classification metrics (Features 5)..... 24
Figure 29: SVM Classification metrics (Features 5)..... 25
Figure 30: Logistic Regression Classification metrics (Features 5) 25
Figure 31: KNeighborsClassifier Classification metrics (Features 5) 26
Figure 32: Decision Tress Classification metrics (Features 5) 26
Figure 33: Comparison of Models Performance for Features 6..... 27
Figure 34: Random Forest Classification metrics (Features 6) 28
Figure 35: SVM Classification metrics (Features 6)..... 28
Figure 36: Logistic Regression Classification metrics (Features 6) 29
Figure 37: KNeighborsClassifier Classification metrics (Features 6) 29
Figure 38: Decision Tress Classification metrics (Features 6) 30
Figure 39: Comparison of Models Performance for Features 7..... 31
Figure 40: Random Forest Classification metrics (Features 7)..... 31
Figure 41: SVM Classification metrics (Features 7)..... 32
Figure 42: Logistic Regression Classification metrics (Features 7) 32
Figure 43: KNeighborsClassifier Classification metrics (Features 7) 33
Figure 44: Decision Tress Classification metrics (Features 7) 33

Table 1: Email-id features..... 2
Table 2:URL-Text features..... 3
Table 3:URL-Content features..... 5
Table 4: Features 1 results 10
Table 6: Features 3 results 16
Table 7: Features 4 results 20
Table 8 : Features 5 results 23
Table 9: Features 6 results 27
Table 10: Features 7 results 30
Table 11: Comparison of experiments results..... 34