Available online at www.qu.edu.iq/journalcm
JOURNAL OF AL-QADISIYAH FOR COMPUTER SCIENCE AND MATHEMATICS
ISSN:2521-3504(online) ISSN:2074-0204(print)

Phishing Attacks Detection by Using Support Vector Machine

Majeed Jasim Nabet [a*], ***Loay E. George*** [b]

[a] *Informatics Institute for Postgraduate Studies. Email:ms202110662@iips.icci.edu.iq*

[b] *University of Information Technology and Communications. Email:Loayedwar57@cs.uobaghdad.edu.iq*

ARTICLE INFO

Article history:
Received: 30 /04/2023
Rrevised form: 10 /06/2023
Accepted : 13 /06/2023
Available online: 30/06/2023

Keywords:

Keywords:
Phishing
Attacks Detection
Kmeans
Support Vector Machine

ABSTRACT

Today's world is heading towards complete digital transformation, and with all its advantages, this transformation involves many risks, the most important of which is phishing. This article proposes a system that extracts features from all parts of the email, initially brought from different data sets, and uses one of the machine learning algorithms (K-means algorithm) to extract the valuable features, as used four methods to calculate the distance in the K-means algorithm. This work used SVM as a classifier to classify emails into phishing and legitimate and tuned its parameters to obtain a high percentage of accuracy. The proposed model gave accuracy equal to 98.8 %.

MSC..

https://doi.org/ 10.29304/jqcm.2023.15.2.1246

1. Introduction

With the ubiquity of the Internet today, society uses Internet products for various things, such as sharing knowledge, socializing, and conducting multiple financial activities, including purchases, advertising, and sending money [1][2]. This state has led to the emergence of cybercrime; Cybercrime is using computers, communication devices, or networks as a tool for illicit purposes. Such as phishing, ransomware attack, identity fraud, theft of financial or credit card payment data, robbery and sale of corporate data, crypto-jacking (hackers use resources they don't own to mine cryptocurrency), or cyber espionage (hackers accessing government or corporate data), cyber extortion [3][4]. Phishing is a crime where perpetrators send fake emails that appear to be from popular and trusted brands or organizations and ask for personal credentials such as bank passwords, usernames, phone numbers, addresses, credit card details, etc. [5][6]. Developments in phishing have led to numerous phishing methods such as Spear-phishing, mobile malware, Man in the Middle, Spam messages, Vishing, Tabnabbing, Xbox Live, Chat in the Middle, etc. Phishing detection is classified according to different methods [7], such as List-based approaches; the response time and detection accuracy are very high. If the URLs are considered phished sites, they are saved in the database. When a new URL is used, it compares with the URLs in the database, and if it matches, it is prevented by the browser and is saved in the database for future use. This technique is called blacklisting, and whitelisting is a technique for keeping legitimate URLs in a database and checking for new ones [8].

∗Corresponding author: Majeed Jasim Nabet

Email addresses: ms202110662@iips.icci.edu.iq

Communicated by 'sub etitor'

Heuristic-based approaches, Detection uses common qualities of phishing sites, such as domain name information and The URL string. Antivirus or intrusion detection systems create and use a database of signatures of known attacks to scan websites [9]. In visual Similarity approaches, a list of regularly attacked pages (domain names and screenshots) is kept to protect users against the possible impersonation of such pages. If a user accesses a page not on the trusted list, its content is evaluated against the trusted ones [10]. Machine learning approach, Uses a large labeled dataset as input to train a classification model, which then categorizes each input data point into a predefined number of classes [11][12]. Because there are only two classifications in a phishing detection problem (benign and phishing), the trained models are binary classifiers [12][13]. Deep learning approaches, Neural Networks are increasingly being used in phishing detection due to their ability to analyze complex patterns and identify anomalies in data. NNs can be trained on large datasets of phishing and non-phishing samples [14].

The literature has extensively covered detecting phishing emails, but there have always been some shortcomings [15][3]. For example, some articles use a data set that contains emails of only one class, or it consists of two classes. Still, the difference between them is very large, or the data set is relatively small, or the data set is from one source, and this source may be biased. As for extracting features, some works extract features from only one part of the email; the features are extracted and represented digitally without giving the actual values of the features. In the stage of selecting features, some articles choose the features based on experience or depending on some criteria, such as associating features with persuasion without working with features in an abstract way. When the stages that precede classification suffer from weakness, the classification will not give the desired results. This work aims to build a model with the following characteristics: Reliance on data sets from different sources and the samples are close and not of one class. Extract the features from all email parts and give them actual values. Abstractly, they select the features based on machine learning techniques. Executing classifier, tuning its parameters to provide good results. This work used the K-means algorithm to determine features significantly affecting the classification process. And to calculate the distance in the K-means algorithm, the Euclidean and Manhattan methods and two other methods were the Euclidean method divided by the standard deviation and the Manhattan method divided by the standard deviation. Then applied the support vector machine and tuned its parameters to get the best values. The article's organization is as follows: Section 2 reviews related works; Section 3 explains the proposed model. Section 4 discusses the results of training and testing. Finally, section 5 concludes the contribution of this article.

2. Related Works

Niu et al., 2017, introduced a hybrid classifier, making the kernel function's parameter selection more efficient using cuckoo search (CS) [16]. CS is integrated with a support vector machine to construct a hybrid classifier. In this work, the researchers selected features based on experience, and the focus was on the URL address, and this selection has a significant human error factor. Kumar et al., 2020, The researchers used hybrid classification, including SVM and PNN. The researchers have operated an XOR between SVM and PNN outputs[17]. The data set used in this work is small, consisting of only 1,705 samples, and such a data set may give biased results. Rastenis et al., 2021, This work distinguished the researchers' use of different data sets (the Nazario dataset, the SpamAssassin dataset, and Vilnius Gediminas Technical University have compiled a dataset of individual spam and phishing emails.) and merged them. These data sets use different languages. Then the researchers used the TF-IDF method to represent the features and choose the most suitable one, while the methods used for classification are (SVM, RF, DT, NB, LR, and K-NN) [18]. The researchers used the private data set for training and the public data set for testing, and they divided the data into 90% for training and 10% for testing. Working in this way does not give correct results. The researchers also tested the model using a translated dataset, not in its original language. Mughaid et al., 2022, this article used three different types of data sets. The first consisted of 8,351 phishing emails and 517,402 legitimate emails. To avoid the over-fitting case, the researchers randomly selected 8,400 legitimate emails as samples for balance. The second data set consisted of 5,000 phishing samples and as many legitimate emails. The third dataset contains 3,000 non-phishing samples but contains 500 spam samples. The researchers used seven methods to classify the samples: a Locally-deep SVM, Logistic regression, an SVM, Boosted decision tree, a Neural network, a Decision forest, and Averaged perceptron [19]. The proposed model in this work divided the data set into 70% for training and 30% for testing, and the division was not tuned to find the best division. Butt et al., 2022, The researchers proposed (in this work) three methods for classifying SVM, LSTM, and Naive Bayes samples where the accuracy percentage of the SVM method was 99.62%, the accuracy percentage of the LSTM method was 98%. The accuracy percentage of the NB method was 97% [20]. The complexity of this work is high and disproportionate to the given results, and this complexity increases the time required for implementation. Livara et al., 2022, This article uses a dataset called Phishing Email Collection, updated by Akashsurya156 in 2020. It consists of 22 features. The model of work chose twenty-one features to assess the samples as phishing or not. 90% of phishing emails are merged with 90% of legitimate emails for the training dataset after the division of emails into legitimate and phishing categories. The testing dataset is related to the remaining 10% of phishing and legitimate emails. The researchers used SVM with the parameters: a kernel function is RBF, Regression Loss is 0.10, One hundred for Iteration Limit, and 0.0010 for Numerical Tolerance. The SVM accuracy of this article is 16.85% which is very poor [2]. The dataset has predefined attributes. These features may not be valuable. They also divided the data into 90% for training and 10% for testing, and this percentage is inefficient.

3. Materials and Methods

This work aggregated two common datasets, SpamAssassin and Naziro. Because SpamAssassin does not contain phishing emails, have merged it with the Naziro dataset, which only contains phishing emails. Figure 1 illustrates our proposed model, which consists of three stages:

- Features extraction: At this stage, extracted features from all parts of emails.
- Features selection stage: At this stage, used the K-means algorithm to select the features through which samples can be classified as phishing or not.
- Classification stage: This stage classified the samples into phishing and legitimate samples using SVM.

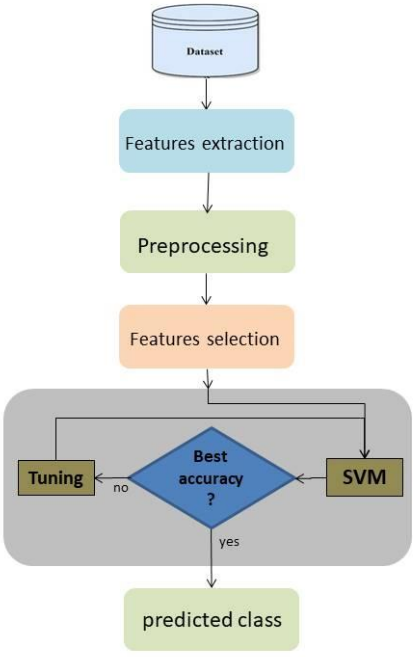

Fig. 1- The proposed model

3.1. Features extraction stage

Data sets in the form of actual emails were utilized to extract features from every aspect of the emails using the tool (Email Features Extraction). It is an open-source tool that allows users to extract features from any email set containing emails, its extension EML. The tool will then divide the emails into components such as (To, From, BCC, CC, Subject, HTML Body, and Text Body) and extract the desired features into a separate output file, a CSV file. An error file will also be generated if errors are made during the extraction procedure. One hundred thirty-eight features are extracted by this tool. Visit (https://github.com/WadeaHijjawi/EmailFeaturesExtraction) to access the tool. Appendix A contains the features extracted at this stage according to their part of the email.

3.2. Features selection stage

Kmeans algorithm is an unsupervised clustering algorithm, but the proposed model used it in the feature selection process because it will cluster the features depending on the Similarity and proximity of the features. Where took the features as pairs (the first feature with the second feature, then the first feature with the third, and so on) and implemented the algorithm considering k = 2. After the end of the algorithm, examined the centroids of the two clusters were. The cluster whose centroid is a phishing sample; is considered a phishing cluster, and the cluster whose centroid is a legitimate sample; is considered a legitimate cluster, then calculated the Matching between the actual class and the predicted class, after choosing the top twenty pairs with the highest percentage of Matching and re-work by adding the features to the twenty features so that the assembly becomes a combination of three features and so on.

This work used four methods to calculate the distance in the Kmeans algorithm:

I. Euclidean method: where used the Euclidean equation to calculate the distance, which is Equation (1)

$$\boldsymbol{d(C,F) = (F_x - C_x)^2 + \left(F_y - C_y\right)^2} \quad (1)$$

Where C= centroid; F= feature
Noticed that this method stopped improving at the quadruple combinations, so stopped forming new combinations.

II. Manhattan method: This method replaces the square with the absolute; it uses Equation (2)

$$d(C,F) = |F_x - c_x| + \left|F_y - c_y\right| \quad (2)$$

Where C= centroid; F= feature
Like the previous one, this method also stopped improving with the quadruple combination.

III. Euclidean divided by standard deviation method: in this method, the Euclidean value is divided by the standard deviation to adjust the values more, and this method used Equation (3)

$$d(C,F) = \left(\frac{F_x - C_x}{STD1}\right)^2 + \left(\frac{F_y - C_y}{STD2}\right)^2 \quad (3)$$

Where C= centroid; F= feature; STD1= standard deviation of the first feature; STD2= standard deviation of the second feature.
This method is unlike the previous methods, as the improvement continued until the Sixth combination.

IV. Manhattan divided by standard deviation method: As in the previous method, but in this method, the Manhattan value is divided by the standard deviation by using Equation (4)

$$d(C,F) = \left|\frac{F_x - C_x}{STD1}\right| + \left|\frac{F_y - C_y}{STD2}\right| \quad (4)$$

Where C= centroid; F= feature; STD1= standard deviation of the first feature; STD2= standard deviation of the second feature.
This method also continued improvement until the sixth combination.

All previous methods assumed that the first initial centroid is the Mean of the 1st feature and the Standard Deviation of the 1st feature, and the second initial centroid is the Mean of the 2nd feature and the standard deviation of the 2nd feature. This is in the binary combination, while the triple combination has added the Mean of the third feature to the first initial centroid and added the standard deviation of the third feature to the second initial centroid.

3.3. Classification stage

This stage consists of two steps, carried out repeatedly to obtain the best results.

A. Tuning

In this step, tuned the parameters of the support vector machine after each execution of the SVM and these parameters are:

- **Input**: in the features selection stage explained that used four methods to calculate the distance in the K-means algorithm, and each method gave us different results. Also formed binary, triple, and quadruple combinations in the first two methods and Pentagonal and hexagonal combinations in the other two methods. Therefore, used binary and quadruple combinations for all methods and hexagonal combinations for the other two methods in tuning the inputs of the SVM. And every time use one of the combinations, tuning the number of features used, ten, twenty, or thirty features, as another tuning the inputs of the SVM.
- **The ratio of training samples to test samples**: use the following proportions to tune the SVM; splitting the data set begins with half of the data for training and the other half for testing. The proposed model increases the training ratio to sixty, the testing ratio to forty, and so on, up to eighty to twenty.

B. Support Vector Machine

To classify, the proposed model used two methods in the SVM, Quadratic, and Cubic. The 5-Fold Cross Validation method was used to obtain high Accuracy. The training data set is divided into five folds. In the 1st iteration, the 1st fold is the validation data, and the other four folds are the train data. In the 2nd iteration, the 2nd fold is the validation data, the 1st, 3rd, 4th, and 5th are the training data, and so on. The Accuracy is calculated for each fold, and the Totally Accuracy is the Mean of it. The proposed model will also perform a Standardization operation for the data to work on one scale.

4. Results and discussion

The results will be discussed based on the four methods used in the Kmeans algorithm for calculating the distance, which was explained in the features selection section. Then the results will be compared with each other to choose the method and its tuning that gave the best result.

The evaluation criteria used in this work is total accuracy, which can be obtained from Equation (5).

$$\text{Total accuracy} = \left(\frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}\right) * 100 \quad (5)$$

Where TP= the number of samples that were correctly classified phishing in training and test, TN= the number of samples that were correctly classified ham in training and test, FP= the number of samples that were incorrectly classified phishing in training and test, FN= the number of samples that were incorrectly classified ham in training and test.

In Figures 2, 3 and 4, the y-axis represents the totally accuracy, which is the Mean of training accuracy and test accuracy. The x-axis represents the execution groups that were used. Each execution group represents the type of combination used in the feature selection stage (binary, quadruple, hexagonal), and the kind of Support Vector Machine, whether Quadratic or Cubic, implemented on the features. And the split ratio between training and testing. Each line is one of the methods of calculating the distance previously used in the feature selection stage.

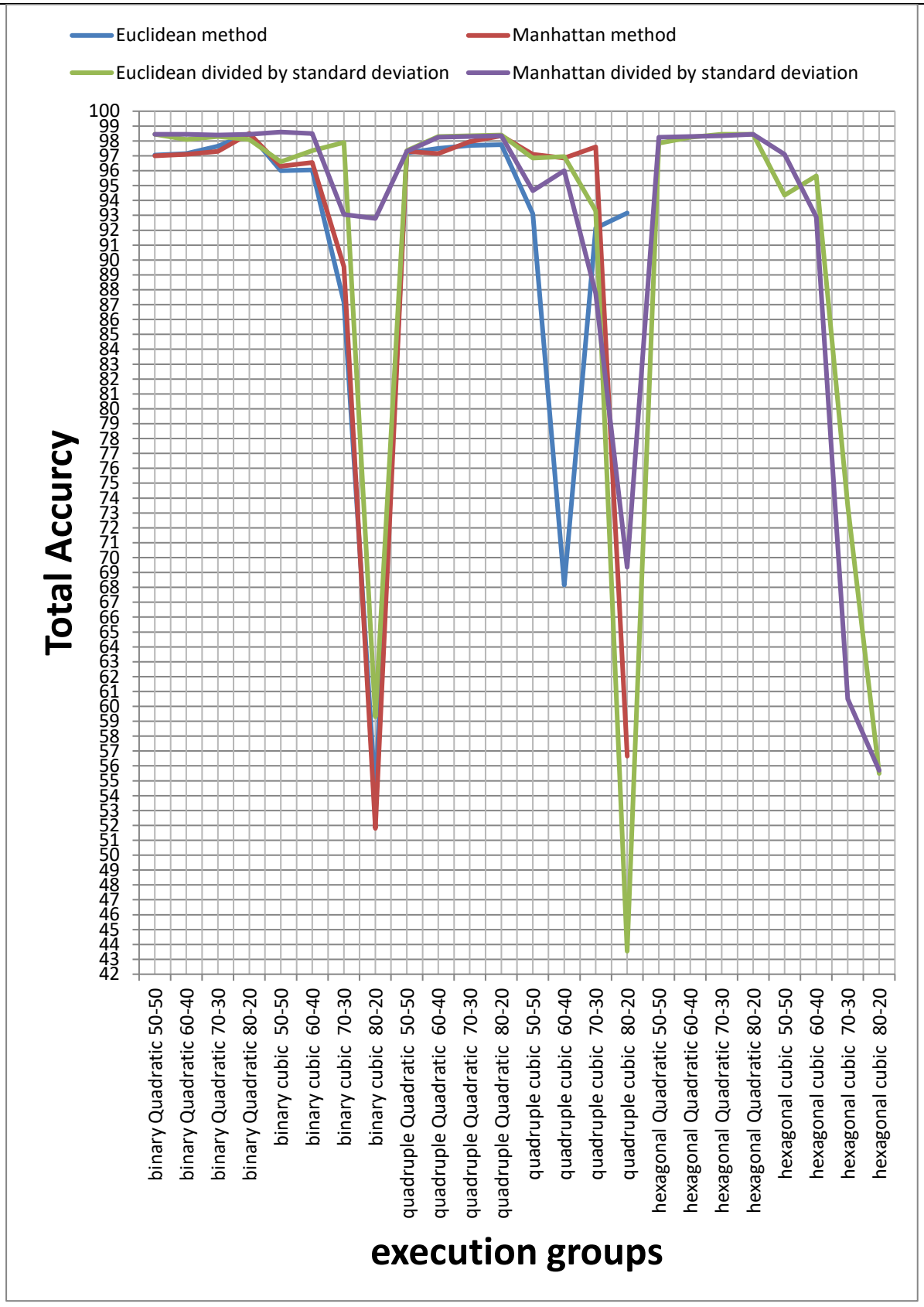

Fig. 2- The results when selecting ten features as inputs

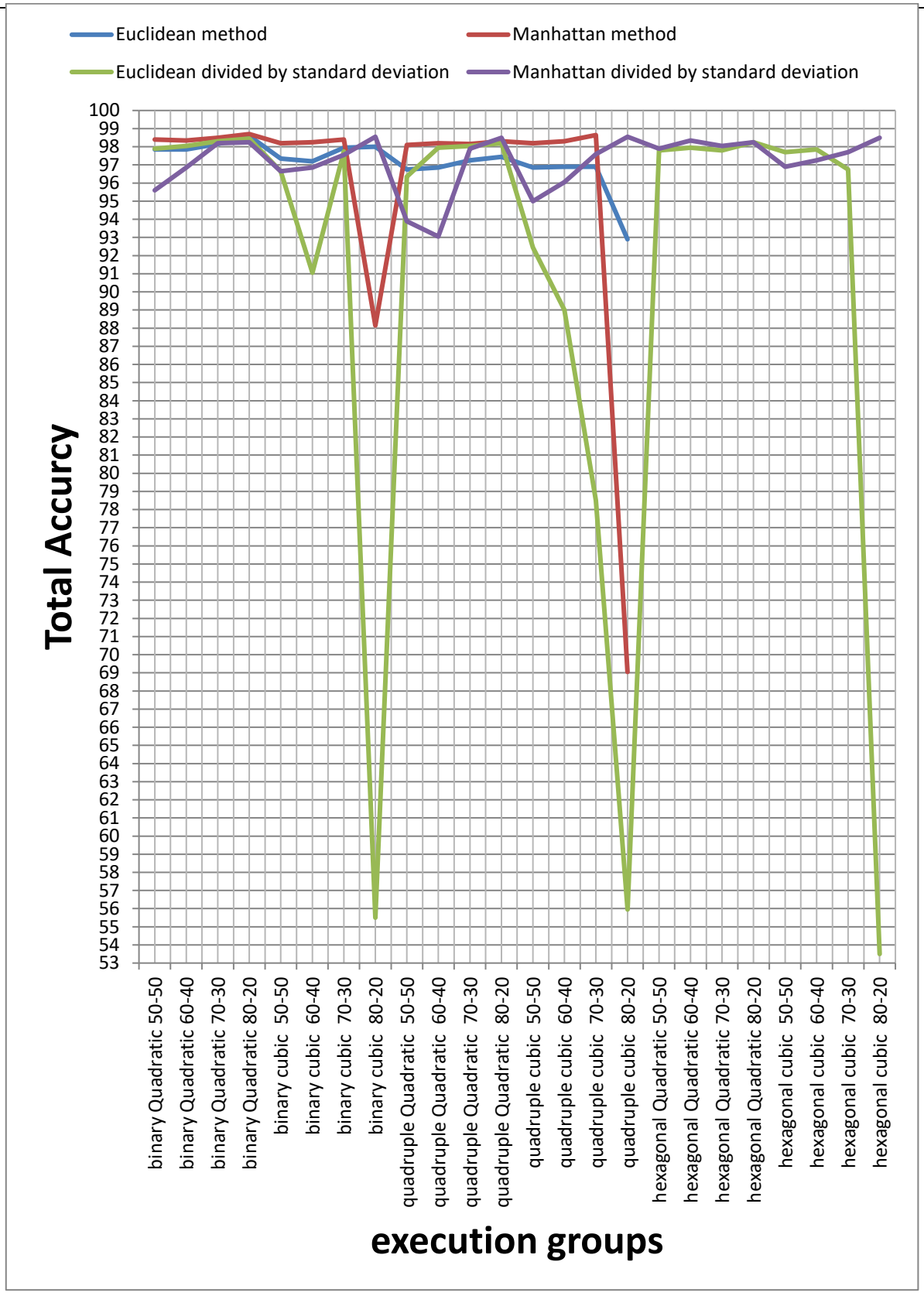

Fig. 3- The results when selecting twenty features as inputs

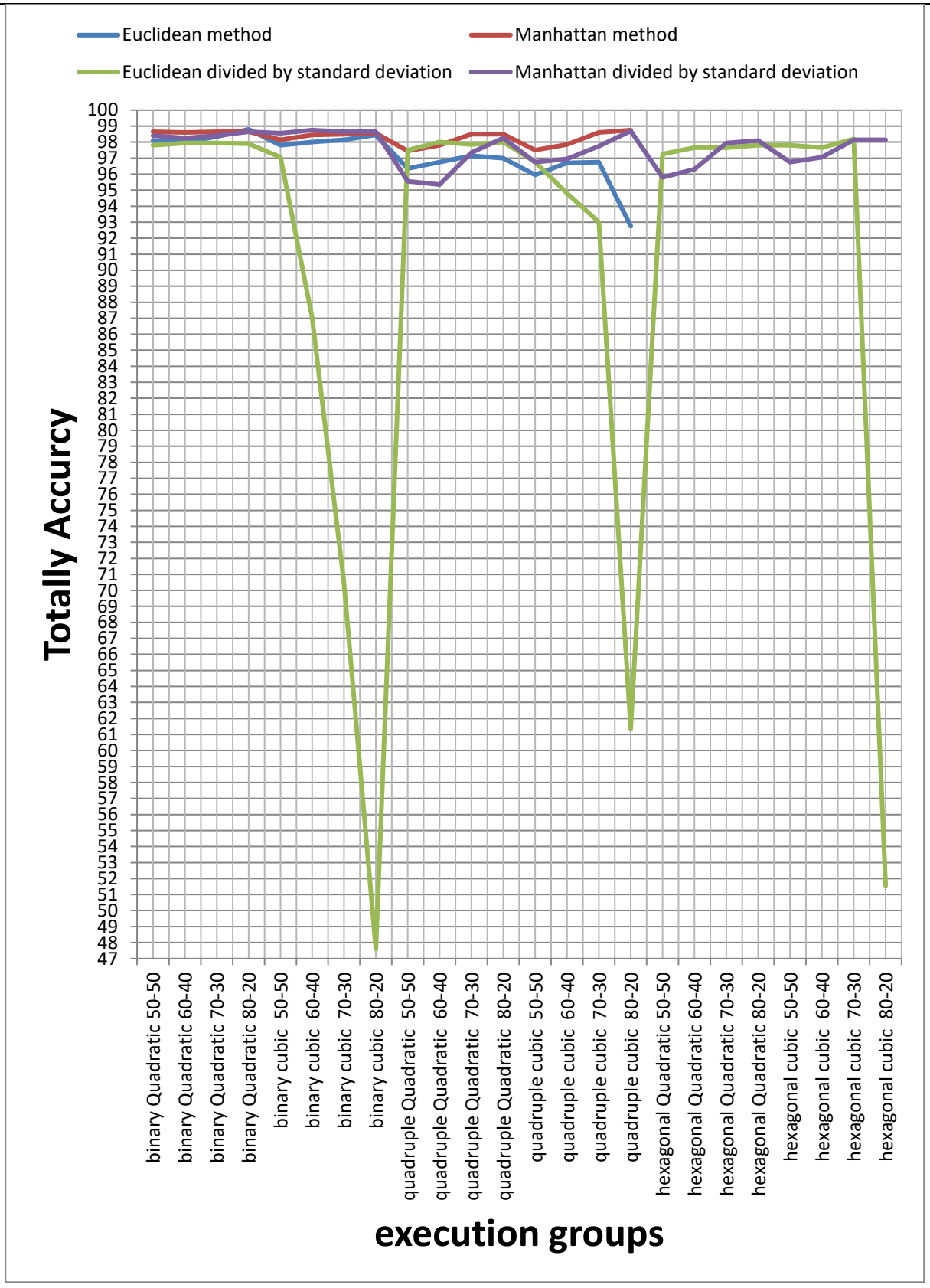

Fig. 4- The results when selecting thirty features as inputs

Figure 2 shows the results of the Support Vector Machine when using ten influential features specified in the features selection stage. Analyzing it showed that the execution group (binary cubic 50-50) in the Manhattan method gives the best result, as its Total Accuracy reaches 98.6%.

Figure 3 shows the results of the Support Vector Machine when using twenty influential features; the execution group (binary Quadratic 80-20) in the Manhattan method gives the best result, as its Total Accuracy reaches 98.7%.

Figure 4 shows the results of the Support Vector Machine when using thirty influential features; the execution group (binary Quadratic 80-20) in the Euclidean method gives the best result, as its Total Accuracy reaches 98.8%.

Table 1- The details of the best results

No. Features and method	Execution Group	Accuracy (Validation) %	Total cost (Validation)	Prediction speed (obs/sec) ~	Training time(sec)	Accuracy (Test)%	Total cost (Test)
Ten Manhattan	binary cubic 50-50	98.3	119	120000	30.562	94.3	402
Twenty Manhattan	binary Quadratic 80-20	98.9	120	91000	20.783	98.5	43
Thirty Euclidean	binary Quadratic 80-20	99.1	99	130000	8.3637	98.5	41

Table 1 shows that the third case was the best in all details. Although the number of features was more than the rest, the prediction speed was better, which qualified the third case as the best. Table 2 shows the 30 features used in the best case.

Table 2- The features of the best case

Feature	Description of Feature	Feature	Description of Feature
Minute	Email time	Hour	Email time
Second	Email time	Day	Email date
Ellipsis	No. of ellipsis	Entropy	Value of Entropy
Inverse FI	The value of the Inverse fog index, including stopwords	SMOG Without Stop Words	Simple Measure Of Gobbledygook index, excluding stopwords
Min Character Diversity	Minimum character diversity of each word in the body	Fog Index	The value of The fog index
Text Plain Unique	Max Number of unique words	Ratio Upper To All Subject	Max ratio of uppercase letters to All in subject
Ratio NonAlphaNum To All	Max ratio of non-alphanumerics to all characters of each word in the body	SMOG	Simple Measure Of Gobbledygook index, including stopwords
HTML Anchor	Count of HTML Anchor	Ratio NonAlphaNum To All Subject	Max ratio of non-alphanumerics to all characters of each word in the subject
X_Mailman Version	The version number of X Mailman	Avg Sentence in Paragraphs	Average No. of sentences per paragraph in the body
Word Length Without Stop words	Max Word lengths, excluding stop words	Words Max Char Subject	No. of words with maximum character in the subject
Word Length	Average word length in	SMOG-I	It is used to measure the

	the body		difficulty of the text writing
Longest Capital	Max The longest capital word	All Count Cap Word Subject	No. of capital words in the subject
Ratio Upper Lower Subject	Max ratio of uppercase letters to lowercase letters of each word in the subject	Vocabulary Richness	the index, which represents the number of distinct words in a text
Words AVG Char	Average No. of characters per word	Min Character Diversity Subject	Minimum character diversity of each word in the subject
Inverse FI Without Stop words	The value of the Inverse fog index, excluding stopwords	Ratio Digit To All	Max ratio of digit characters to all characters of each word. In the body

5. Conclusion

This Paper aggregated two data sets from different sources, containing phishing mail and legitimate spam email, for the data to be convergent to know whether the proposed system is effective. The proposed model extracted 138 features from all parts of the email and subjected them to the K-means algorithm (one of the machine learning algorithms) to reduce the features that will be input to the Support Vector Machine and used SVM as a classifier to classify the samples into phishing and legitimate. The proposed model tuned the parameters of the Support Vector Machine to obtain the highest possible accuracy, which obtained Reliable high accuracy. In the future, Using one of the machine learning algorithms instead of the K-means algorithm to improve accuracy and another type of kernel function in SVM can increase accuracy.

References

[1] M. Almseidin, A. A. Zuraiq, M. Alkasassbeh, and N. Alnidami, "Phishing Detection Based on Machine Learning and Feature Selection Methods," International Journal of Interactive Mobile Technologies, vol. 13, no. 12, p. 171, Dec. 2019, doi: 10.3991/ijim.v13i12.11411.

[2] A. Livara and R. M. Hernandez, An Empirical Analysis of Machine Learning Techniques in Phishing E-mail detection. 2022. doi: 10.1109/iconat53423.2022.9725434.

[3] Kathiravan, Rajasekar, Parvez, Durga, Meenakshi, and Gowsalya, "Detecting Phishing Websites using Machine Learning Algorithm," 7th International Conference on Computing Methodologies and Communication (ICCMC), pp. 5–270, 2023.

[4] P. Bountakas and C. Xenakis, "HELPHED: Hybrid Ensemble Learning PHishing Email Detection," Journal of Network and Computer Applications, vol. 210, p. 103545, Jan. 2023, doi: 10.1016/j.jnca.2022.103545

[5] A. Jain and B. B. Gupta, "Phishing Detection: Analysis of Visual Similarity Based Approaches," Security and Communication Networks, vol. 2017, pp. 1–20, Jan. 2017, doi: 10.1155/2017/5421046.

[6] M. Somesha and AR. Pais, "Classification of Phishing Email Using Word Embedding and Machine Learning Techniques," J Cyber Secur Mobil, pp. 279–320, 2022.

[7] D. N. Quang, A. Selamat, O. Krejcar, E. Herrera-Viedma, and H. Fujita, "Deep Learning for Phishing Detection: Taxonomy, Current Challenges and Future Directions," IEEE Access, vol. 10, pp. 36429–36463, Jan. 2022, doi: 10.1109/access.2022.3151903.

[8] P. Kalaharsha and BM. Mehtre, "Detecting Phishing Sites--An Overview," arXiv Prepr arXiv210312739, 2021.

[9] M. Hara, A. Yamada, and Y. Miyake, Visual similarity-based phishing detection without victim site information. 2009. doi: 10.1109/cicybs.2009.4925087.

[10] S. Abdelnabi, K. Krombholz, and M. Fritz, VisualPhishNet: Zero-Day Phishing Website Detection by Visual Similarity. 2020. doi: 10.1145/3372297.3417233..

[11] S. A. Salloum, T. Gaber, S. Vadera, and K. Shaalan, "A Systematic Literature Review on Phishing Email Detection Using Natural Language Processing Techniques," IEEE Access, vol. 10, pp. 65703–65727, Jan. 2022, doi: 10.1109/access.2022.3183083.

[12] D. M. Divakaran and A. Oest, "Phishing Detection Leveraging Machine Learning and Deep Learning: A Review," arXiv Prepr arXiv220507411, vol. 20, no. 5, pp. 86–95, Sep. 2022, doi: 10.1109/msec.2022.3175225.

[13] A. A. Nafea, N. Omar, and M. Q. Al-Ani, "Adverse Drug Reaction Detection Using Latent Semantic Analysis," Journal of Computer Science, vol. 17, no. 10, pp. 960–970, Oct. 2021, doi: 10.3844/jcssp.2021.960.970.

[14] M. Q. Al-Ani, N. Omar, and A. A. Nafea, "A Hybrid Method of Long Short-Term Memory and Auto-Encoder Architectures for Sarcasm Detection," Journal of Computer Science, vol. 17, no. 11, pp. 1093–1098, Nov. 2021, doi: 10.3844/jcssp.2021.1093.1098.

[15] R. Eckhardt and S. Bagui, "A User-centric Focus for Detecting Phishing Emails," AI, Mach Learn Deep Learn a Secur Perspect, 2023.

[16] N. Weina, X. Zhang, G. Yang, Z. Ma, and Z. Zhuo, Phishing Emails Detection Using CS-SVM. 2017. doi: 10.1109/ispa/iucc.2017.00160

[17] A. Kumar, J. M. Chatterjee, and V. García-Díaz, "A novel hybrid approach of SVM combined with NLP and probabilistic neural network for email phishing," International Journal of Electrical and Computer Engineering, vol. 10, no. 1, p. 486, Feb. 2020, doi: 10.11591/ijece.v10i1.pp486-493.

[18] J. Rastenis, S. Ramanauskaitė, I. Suzdalev, K. Tunaitytė, J. Janulevičius, and A. Čenys, "Multi-Language Spam/Phishing Classification by Email Body Text: Toward Automated Security Incident Investigation," Electronics, vol. 10, no. 6, p. 668, Mar. 2021, doi: 10.3390/electronics10060668.

[19] A. Mughaid, S. AlZu'bi, A. A. Hnaif, S. Taamneh, A. Alnajjar, and E. A. Elsoud, "An intelligent cyber security phishing detection system using deep learning techniques," Cluster Computing, vol. 25, no. 6, pp. 3819–3828, May 2022, doi: 10.1007/s10586-022-03604-4.

[20] U. Butt, R. Amin, H. Aldabbas, S. Mohan, B. Alouffi, and A. Ahmadian, "Cloud-based email phishing attack using machine and deep learning algorithm," Complex & Intelligent Systems, vol. 9, no. 3, pp. 3043–3070, Jun. 2022, doi: 10.1007/s40747-022-00760-3.