Journal of Academic Research Vol 30, Issu 1, 2026 "09-20"

Enhanced Phishing Detection : A Hybrid SVM-Genetic Algorithm Approach

Mohammad M Elsheh and Sarah Al-mabrouk Ebayou

Department of Computer Science, Libyan Academy. Misurata-Libya
m.elsheh@lam.edu.ly

Article information

Key words

Genetic Algorithm, Machine Learning, Phishing Website Detection, Support Vector Machine.

Received **06 01 2026**,
Accepted **20 01 2026,**
Available online **21 01 2026**

Abstract

The majority of public and financial institutions have recently upgraded and enhanced the direct online services they offer to their clients due to the rise in internet applications and users. However, the majority of web users are unaware of internet security measurements. Hence, attacks on various online platforms are gradually increasing. Attackers use various methods to steal users' sensitive information; one of the most common scams is phishing websites. Therefore, there is a need to fight these attacks and constantly improve detection technologies, including machine learning (ML) methods. ML methods classify whether a site is phishing or not based on a number of pieces of data obtained from other webpages. Therefore, this paper aims to present a model for detecting and classifying phishing websites using the Support Vector Machine (SVM) model optimized using the Genetic Algorithm (GA) to obtain the best classification accuracy. The collected dataset consists of 12,000 samples. The phishing URLs were collected from the PhishTank website, while the legitimate ones were from the Kaggle website. Furthermore, accuracy, precision, recall, and the F1-score were used to evaluate the performance of the presented method. The obtained results were compared to the results of previous research, which was conducted using SVM algorithms with Ant Colony Optimization (ACO). The attained results showed that the classification accuracy of the presented approach achieved 97.62%, which is higher than the traditional SVM model by 9.29% and almost equal to the SVM-ACO model.

I. Introduction

Nowadays, with the growth and development of information and communication technology, AI plays a major role in almost all fields. Institutions, companies, and even governments depend on its technologies in order to speed up and automate operations, as well as reduce efforts and costs. AI is the set of systems or devices that simulate human intelligence to perform tasks that can be improved based on the information they collect. Also, it can be considered the ability to think about and analyze the data about

a particular form or function. AI presents images of human-like high-performance robots that aim to greatly enhance human capabilities and contributions, which makes it a very valuable business asset [1]. AI can also be considered as an umbrella term for applications that perform complex tasks which used to require human input, for instance communicating with customers over the Internet. The term AI is often used interchangeably with its subfields, which include ML, deep learning (DL) and data mining (DM) and many others. However, there are several differences, including that ML focuses on creating systems that learn or improve their performance based on the data they consume [2], while DM focuses on the process of discovering patterns from large sets of data based on methods at the intersection of ML, statistics, and database systems. It can also be seen as the process of analyzing data from different perspectives, discovering patterns and correlations in datasets that are useful for predicting results in making the right decision. In DM, models are seen as implementations of algorithms for searching, identifying, and displaying any patterns in the data. In addition, there are two types of models: predictive and descriptive, which can be categorized into classification, prediction, association, and clustering.

There are many techniques that are used for solving the same problem for the same task in data mining. Some technologies have specific requirements based on the data format. Therefore, a return to the data preparation stage is often necessary. Some of the popular DM algorithms are SVM, Decision Tree (DT), Random Forest (RF), K-nearest Neighbor KNN, Naïve Bayes (NB) and GA[3] .

Furthermore, the number of web applications and users increases dramatically due to the fact that the majority of financial and public institutions have recently upgraded and enhanced the direct online services provided to their customers. As a result, the attacks on various online platforms are gradually increasing. These attacks include e-commerce sites, Online Social Networks (OSNs), e-learning and online banking [4]. These activities had an effect on the economy worldwide, as the great dependency on online financial services has increased the security risk for clients as well as financial institutions. The attackers use different methods to steal the private information of the users; one of the most common tricks is social engineering. In addition, numerous communication techniques are used to trick users, including messaging, emails and social media. Yet, the most common crime is phishing websites [5]. Phishing is a type of cyber threat in which attackers impersonate legitimate authentic websites to steal sensitive information such as credentials, credit cards, passwords, bank account information, financial details, and other behavioral data. Phishing attempts can be made through various mediums, including the internet, short message service, Email, smishing (short message phishing), and vishing (voice phishing) [6]. However, the phishing detection mechanism involves user awareness and technology-based approaches. Only a careful and knowledgeable user can detect fake webpages by looking into the Uniform Resource Allocator (URL) of a webpage, and the other examination techniques such as HTML tags, URL addresses, and JavaScript source codes [7].

More consumers are being drawn to actual phishing sites as a result of the vastly increased number of page redirections employed by phishers. Users who click on

phishing links are transferred from original websites to phishing websites where their credentials are sought. Phishers use this obfuscation technique to hide the phishing URL, most particularly from detection, via web server log referrer field monitoring. Furthermore, half of the phishing sites are currently using HTTPS and SSL certificates to confuse users.

The user should be aware enough and informed of the typical tactics used by attackers to avoid falling for a phishing email. Among the most popular tactics are:

- Asking for personal or sensitive information.
- Using spoofed email addresses.
- Including attachments or links.
- Creating a sense of urgency.

Currently, Multiple research studies have been carried out to prevent or detect phishing, such as studies with Blacklists and other ML techniques. Among those that apply machine learning, there are several types of research using multiple ML methods. These perform feature decomposition, obtaining URL resources, and text processing, as well as the use of dictionaries to recognize common characters in URLs. All this can be done, but it represents a fairly large computational load and complexity. Moreover, multiple researchers used SVM with other ML techniques that have proved a high accuracy in terms of URLs.

II. Related Works

As more people use online services, it has become easier for cybercriminals to steal users' confidential information through phishing attacks. These attacks can be prevented by educating users on how to distinguish between phishing and legitimate websites. However, if the user does not have sufficient awareness or cannot detect them, the greatest burden falls on the technologies and applications to protect them. Consequently, many different studies and methods have been conducted to address this issue. Henceforth, some approaches based on the SVM model with other ML models are presented.

The fundamental point of the research conducted by M. Elsheh and K. Swayeb is to develop an approach to detect phishing websites. Their approach combined the SVM model with the ACO algorithm. In addition, they applied the Deep Belief Network (DBN) to select the best features from extracted features. Their dataset contained 12,000 URL websites, with 50% phishing and 50% legitimate websites. The experimental results showed that the SVM model's classification accuracy was 87.96%. When ACO is applied as an optimizer of SVM, it achieved a 97.54% accuracy result. This means that the SVM-ACO is 9.58% higher than traditional SVM [8].

In 2020, Pandey et al., designed an architecture that integrates the source code and a webpage's URL to detect phishing websites. They used Levenstein Distance as the algorithm for determining string similarity, and the ML algorithm model in their system (SVM). Their dataset has 10 Attributes and 1353 instances recorded, with 548 legitimate

websites, 702 phishing URLs, and 103 Suspicious URLs. The dataset has three values in it (−1, 0, and 1). The system was designed to provide high accuracy and low false positive rate detection results for unknown phishing webpages. After the model had taken place, the accuracy of detecting phishing webpages reached 89.3%, while the false positive rate was 6.2%. This means that there are 6.2% of legal webpages were considered phishing webpages [9].

Research in 2023 proposed a novel phishing detection architecture that combines a Deep Neural Network (DNN) and a Bidirectional Long Short-Term Memory (BiLSTM) network. This approach leverages both sequential patterns and semantic information within URLs. Through the fusion of NLP-based features and character-level embeddings, the model achieved notable accuracies of 99.21% and 98.79% on the PhishTank and Ebbu2017 benchmarks, respectively [10].

Building on this progress, a novel architecture termed ResMLP was introduced the following year. This model combined residual pipelining with multi-layer perceptron networks and was trained on a large-scale dataset of over 500,000 URLs from Kaggle. It achieved robust results of 98.29% accuracy, 98.10% precision, and a 98.94% F1-score highlighting its potential for effective real-time phishing detection [11].

In late 2025, Elsheh and Abolawaifa introduced a hybrid stacking ensemble model designed for phishing URL detection. This architecture strategically integrates Logistic Regression (LR), Artificial Neural Networks (ANN), and Random Forest (RF) to leverage their complementary strengths. Utilizing Principal Component Analysis (PCA) for feature selection and training on a dataset of over 11,000 labeled URLs from Kaggle, the model achieved an accuracy of approximately 99.55%. This result surpasses the performance of its constituent individual models and underscores the approach's strong potential for deployment in real-time detection systems [12].

III. Research Design

The methodology of the framework that is utilized in this paper, consisting of six main phases, which are illustrated in the Fig. 1:

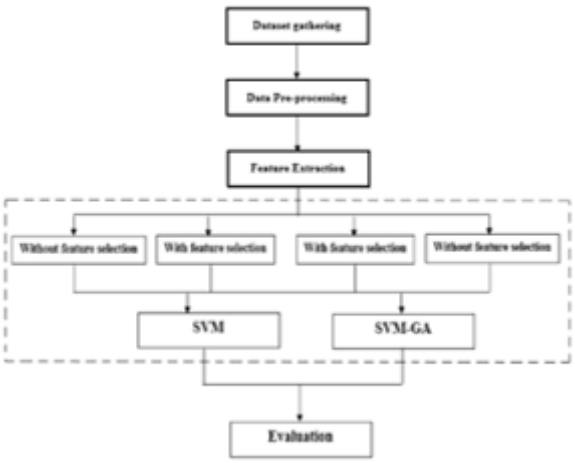

Figure 1: Structure of Research Design.

1. Collection of Phishing and Legitimate Websites Datasets

This is the process of collecting data that is relevant to the objectives of the used model. During this phase, the datasets that need to be processed are searched for. To ensure the legitimacy of the data, it can be sourced from reputable websites such as Alexa and Kaggle. In addition, for phishing related data, PhishTank and OpenPhish are considered reliable sources.

1.1 Data Gathering

The phishing URLs are gathered from Phish Tank [13], an open-source tool that offers a range of phishing URLs in a variety of forms, including .csv,. Json, and others, which are updated on an hourly basis. Meanwhile, the legitimate URLs are collected from the Kaggle website. It is a public data platform that contains a variety of related datasets that are available for developers and researchers for free use [14].

2. Data Pre-processing

Data pre-processing is a cleaning operation that transforms unstructured raw data into well-structured and neat data, which can be used for further research. During this phase, the data is examined for any missing values, any creaky data is smoothed out, and any outliers are identified and deleted. Also, anomalies are fixed to ensure it is clean.

3. Features Extraction and Selection

Feature extraction (FE) refers to the process of transforming raw data into a numerical format that can be processed while maintaining all the information in the original dataset. FE defines and/or aggregates variables into features, effectively reducing the amount of data that must be processed, while still accurately and completely describing the original dataset [15]. On the other hand, feature selection (FS) is the process that involves identifying and selecting the most relevant subset of features from the original features in a dataset to be used as input to a model.

The goal of FS is to enhance model performance by reducing the number of irrelevant or redundant features that may confuse or bias the model. The significance of feature selection lies in its ability to improve model accuracy and efficiency by reducing the dimensions of the dataset and selecting only the most important features. The model can focus on the key variables that have the greatest impact on the outcome and ignore features that may add noise to the data. This can lead to faster training time, improved accuracy, and reduced generalization error [16].

3.1 Features Extraction

A phishing URL and its corresponding website have various characteristics that distinguish them from genuine URLs. For instance, to hide the true domain name, an attacker can create a long and complicated domain name [17]. However, in some cases, using certain features may not be feasible, such as using Content-Based features to create a rapid detection mechanism capable of analyzing a large number of domains. Similarly, Page-Based Features are not very effective when analyzing registered domains. As a result, it is preferable to focus on the URL-Based features that are determined by the detection mechanisms, according to previous literature.

3.2 Features Selection

In the FS, various search algorithms are used heuristically to find an optimal subset of features in order to maximize the classification performance and minimize consuming time in execution (training time). In this case, the GA is used, which is considered a widely used feature reduction algorithm to remove the least significant features in the training process leading to an optimized model [18].

4. Model Classification

In this stage, the phishing websites data is classified using the SVM model with linear, RBF, and sigmoid kernel types. The default 11 parameters, gamma (set to automatic value) and C (set to random values), are used. The GA is used in two phases: the first phase tunes the hyper-parameters (C and gamma) of the SVM model, while the second phase selects the features from the dataset.

5. Implementation

The hardware and software requirements to implement the SVM-GA model are explained in Table I.

Table I: HARDWARE AND SOFTWARE REQUIREMENTS

Hardware/software	Type
Operating system	Windows 8.0
Programing language	Python 3.7.0
IDE	Google Colab.
RAM	4.00 GB
Processor	Intel Core i5

- **Obtaining Dataset**

The phishing URLs dataset was obtained from the open-source platform; Phishing Tank, which offers multiple online datasets in different data formats, such as i.e. .csv, Jason, and xml [13]. Whereas the legitimate URLs were gained from the Kaggle ML repository [14]. To assess the ML model, the URLs are collected and saved in a .csv file. After that, the features are extracted from URLs, and the pre-processing of the data is performed to eliminate null, infinite, and replicated values. The first dataset is named D1, while another dataset made by a study in [8], is named D2.

- **SVM Model Classifier**

The two datasets D1 and D2, are used to run the SVM model. A training set of 80% and a testing set of 20% are created from each dataset. SVM parameter default values were employed, such as (C=1, gamma ='scale', kernel= ‘rbf’). It uses the SVC class for fitting the model, which is a typical classifier used for classification tasks. SVC maps data points to a high-dimensional space and then finds the optimal hyper-plane that divides the data into two classes. SVC is provided by the popular ML library Scikit-learn [19]. The accuracy and other performance results are obtained from the model.score () and classification_report () functions in sequence.

- **The Model Classifier**

SVM's classification process is mostly dependent on the C & gamma parameters, which need to be adjusted in order for SVM to achieve the highest classification accuracy. Based on data and recommendations from [20] [21] [22], we employed the GA optimization technique to maximise the SVM hyper-parameters, C and gamma. The SVM-GA model obtains the optimal parameters, which are used to perform the classification of phishing websites. The model is trained on the two datasets (D1 and D2). The dataset was passed (80% training set, 20% testing set) to the EvolutionaryAlgorithmSearchCV () class, which is within the evoluntionary_search Python package. The main functionality of this class is to tune the hyper-parameters based on genetic evolutionary theory. Once the hyper-parameters are obtained, SVM classifies the website as phishing or legitimate. The parameters of EvolutionaryAlgorithmSearchCV () are set empirically and based on the guidance available in [23], Table II illustrate the values of the parameters.

Table II: Parameters values of genetic algorithm class

The parameters	Value
estimator	SVC()
Params	C=[0,1000], gamma =['linear', 'poly', 'rbf', 'sigmoid']
Scoring	'accuracy'
Cv	5
Population size	10
Generation number	100
Verbose	1
gene_muataion_prob	0.10
gene_crossover_prob	0.5

At this phase, the SVM-GA model and the performance results—accuracy, recall, precision, and F1—of each classified step are explained below:

1. **Results of SVM-GA without Features Selection**

The experimental results of the training model on D1 show that the SVM-GA obtained the best performance when the best values of SVM parameters were: *C* =50, *gamma* = 0.1, and kernel ='rbf', with an accuracy of 96.25%, precision of 96.89%, recall of 96.15%, and F1-score of 96.52. Fig. 2 illustrates the classification report of the results.

```
Best individual is: {'C': 50, 'gamma': 0.01, 'kernel': 'rbf'}
with fitness: 0.9651114351176838
{'C': 50, 'gamma': 0.01, 'kernel': 'rbf'}
-------------------------------------------------
----- Evaluation on Test Data SVM-GA, with-OUT features selection-----
              precision    recall  f1-score   support

          -1       0.95      0.96      0.96      1101
           1       0.97      0.96      0.97      1300

    accuracy                           0.96      2401
   macro avg       0.96      0.96      0.96      2401
weighted avg       0.96      0.96      0.96      2401

------------------------------------------------------------
Accuracy : 96.25156184922949
Prescision : 96.89922480620154
recall: 96.15384615384616
F1-score: 96.52509652509652
Execution time: 00:00:39
```

Figure 2: SVM-GA results without feature selection on D1

In contrast, the best values of SVM parameters when implementing SVM-GA on D2 were: *C*= 0.91, *gamma*= 38.87, and kernel = 'linear', with accuracy of 97.39%, precision of 98.96%, recall of 96.96%, and F1-score of 97.95%. Fig. 3 shows the results and classification report.

```
Best individual is: {'C': 0.9193137641867022, 'gamma': 38.872754828459584, 'kernel': 'linear'}
with fitness: 0.973973973973974
-------------------------------------------------
Accuracy: 0.973973973973974
----- Evaluation on Test Data SVM-GA with features selection-----
              precision    recall  f1-score   support

           0       0.97      0.99      0.98       101
           1       0.99      0.97      0.98        99

    accuracy                           0.98       200
   macro avg       0.98      0.98      0.98       200
weighted avg       0.98      0.98      0.98       200

-----------------------------------------------------
Accuracy = 97.3973973973974
Precision = 98.96907216494846
Recall = 96.96969696969697
F1 Score = 97.95918367346938
Execution time: 00:08:45
```

Figure.3: SVM-GA results without feature selection on D2

2. Results of SVM-GA with Feature Selection

After performing the SVM-GA model with all features available in D1, we implemented the GeneticSelectionCV () class to select the optimal ten features from D1. For comparative purposes, we chose the ten features of D2 that were selected by the ACO algorithm, as stated in [47], in addition to the features selected by our algorithm from D2. The best performance results of D1 when features were selected were accuracy = 96.45%, precision = 96.12%, recall = 97.38%, and F1-score = 96.74%. Through that, the best values of the SVM model's parameters were: *C* =250, *gamma*=0.01, kernel ='rbf'. As one can see in Fig. 4.

```
Best individual is: {'C': 250, 'gamma': 0.01, 'kernel': 'rbf'}
with fitness: 0.9554259529264737
{'C': 250, 'gamma': 0.01, 'kernel': 'rbf'}
-----------------------------------------------------
----- Evaluation on Test Data SVM-GA, with features selection-----
              precision    recall  f1-score   support

          -1       0.97      0.95      0.96      1103
           1       0.96      0.97      0.97      1298

    accuracy                           0.96      2401
   macro avg       0.96      0.96      0.96      2401
weighted avg       0.96      0.96      0.96      2401

-----------------------------------------------------
Accuracy : 96.45980841316118
Prescision : 96.12167300380227
recall: 97.38058551617874
F1-score: 96.7470340604669
Execution time: 00:00:22
-----------------------------------------------------
```

Figure. 4: SVM-GA results with feature selection on D1

The best performance results of D2 using the feature selection function were: accuracy = 97.62%, precision = 98.91%, recall = 98.00%, and F1-score = 98.45%, as shown in Fig. 5.

```
Best individual is: {'C': 1000, 'gamma': 0.001, 'kernel': 'rbf'}
with fitness: 0.976248020668389
-----------------------------------------------------
Accuracy: 0.976248020668389
----- Evaluation on Test Data SVM-GA with features selection-----
              precision    recall  f1-score   support

           0       0.98      0.99      0.98      1194
           1       0.99      0.98      0.98      1206

    accuracy                           0.98      2400
   macro avg       0.98      0.98      0.98      2400
weighted avg       0.98      0.98      0.98      2400

-----------------------------------------------------
Accuracy = 97.6248020668389
Precision = 98.9121338912134
Recall = 98.00995024875621
F1 Score = 98.45897542690545
Execution time: 00:01:11
```

Figure 5: SVM-GA results with feature selection on D2

3. Comparative Analysis of The Models:

The performance results of SVM-GA are discussed and compared to those of popular SVM and SVM-ACO. The experiment with the SVM-GA model showed a high level of accuracy in classifying phishing websites. Table III and table IV illustrate the comparison.

Table III: comparative results of implementing all models on D1/D2.

	Model name	Accuracy (%)	Precision (%)	recall (%)	F1-score (%)
Without feature selection	SVM	87.17%	82.53%,	96.24%	88.86%
	SVM-GA	96.25%	96.89%	96.15%	96.52%
With feature selection	SVM	87.26%	82.35%,	96.79%	88.98%.
	SVM-GA	96.45%	96.12%,	97.38%	95.74%

Table IV: Comparative results of models in D2

	Model	Accuracy (%)	Precision (%)	recall (%)	F1-score (%)
Without feature selection	SVM	88.5%	86.21%,	91.79%	88.91%
	SVM-GA	97.39%	98.96%	96.96%	97.95%
With feature selection	SVM	88.33%	85.89%	91.87%	88.78%
	SVM-GA	97.62%	98.91%	98.00%	98.45%
	SVM-ACO	97.54%	98.47%	96.58%	97.51%

The obtained results over the D1 dataset showed that the accuracy enhancements of the SVM-GA model are 96.45% in the case of feature selection. This is about 9.19% more accurate than using the SVM model alone, and achieved higher results in all of the other metrics. On the other hand, the results obtained from the D2 dataset showed that SVM-GA is around 9.29% more accurate than using SVM; thus, SVM-GA is significantly more effective in detecting phishing websites than using SVM individually. Finally, the results obtained from the D2 dataset, in the case of feature selection, showed that SVM-GA achieved classification accuracy around 0.08% more than the SVM-ACO model. However, the two models have approximately the same effectiveness in detecting phishing websites.

Conclusion

This work presents an approach for detecting phishing and benign websites based on URL-based features. In addition to using GA to select the optimal features, the SVM algorithm was utilized as a classifier and the GA as an optimization technique to determine the best values of the SVM parameters (*C and gamma*). There were 12.000 samples in the dataset, which was collected from online websites, PishTank.com for phishing URLs and Kaggle.com for legitimate ones. Features extraction was applied, and fifty features were gained from each URL in the dataset. The feature selection is performed by GA, which selects ten optimal features among the extracted features.

The obtained results demonstrated that there is no significant difference in accuracy whether feature selection is applied or not, but there is variation in execution time. This work was compared to a previous study that used SVM and the ACO algorithm based on their dataset and through the practical application of the URLs dataset. The comparison was carried out using the performance and confusion matrix. The performance results clearly demonstrate that the

SVM-GA model is more effective in detecting phishing websites, with high accuracy reaching 97.62%, which is 9.58% higher than applying the SVM model alone. It also achieved higher results regarding all metrics used, with 98.47% for precision, 96.58% for recall, and 97.51% for F1 score. SVM-GA outperformed SVM-ACO in terms of accuracy enhancement by about 0.08%, and it also produced better results across all metrics that were examined.

References

[1] Haenlein, M., and Kaplan, A.: 'A brief history of artificial intelligence: On the past, present, and future of artificial intelligence', California management review, 2019, 61, (4), pp. 5-14
[2] Helm, J.M., Swiergosz, A.M., Haeberle, H.S., Karnuta, J.M., Schaffer, J.L., Krebs, V.E., Spitzer, A.I., and Ramkumar, P.N.: 'Machine learning and artificial intelligence: definitions, applications, and future directions', Current reviews in musculoskeletal medicine, 2020, 13, (1), pp. 69-76.
[3] Bhatia, P.: 'Data mining and data warehousing: principles and practical techniques' (Cambridge University Press, 2019. 2019).
[4] Ludl, C., McAllister, S., Kirda, E., and Kruegel, C., On the effectiveness of techniques to detect phishing sites, in The Detection of Intrusions and Malware, and Vulnerability Assessment, Springer, 2007, pp. 20–39
[5] Aburrous, M., Mohammed, R., Dahal, K., and Thabtah, F. (2011). Phishing website detection using intelligent data mining techniques, University of Bradford.
[6] Tanaka and J. Suzuki, "Web and Database Technologies", Proc. of ACM SIGMOD, pp. 10-22, 201 APWG, "Phishing activity trends report, 3rd Quarter 2018".
[7] Eint, S., Chaw, T., Hayato, Y., A Survey of URL-based Phishing Detection, DEIM Forum, 2019.
[8] M. M. Elsheh and K. Swayeb, "Phishing Website Detection Using a Hybrid Approach Based on Support Vector Machine and Ant Colony Optimization," in 2023 IEEE 3rd International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA), 2023, pp. 402-406.
[9] Pandey, A., Gill, N., Sai Prasad Nadendla, K., and Thaseen, I.S.: 'Identification of phishing attack in websites using random forest-svm hybrid model', in Editor (Ed.)^(Eds.): 'Book Identification of phishing attack in websites using random forest-svm hybrid model' (Springer, 2020, edn.), pp. 120-128
[10] A. Ozcan, C. Catal, E. Donmez, and B. Senturk, "A hybrid DNN–LSTM model for detecting phishing URLs," Neural Computing and Applications, vol. 35, pp. 4957-4973, 2023.
[11] S. Remya, M. J. Pillai, K. K. Nair, S. R. Subbareddy, and Y. Y. Cho, "An effective detection approach for phishing URL using ResMLP," IEEE access, vol. 12, pp. 79367-79382, 2024.
[12] Elsheh, M.M. and Abolawaifa, E., 2025. Hybrid Stacking Ensemble Model for Phishing URL Detection Using PCA and Machine Learning. Journal of Technology Research, pp.515-525.
[13] PhishTank. (2023). Developer Information. Available: https://www.phishtank.com/developer_info.php. Accessed: Apr. 24, 2023.
[14] Kaggle. (2023). Developer Information. Available: https:// https://www.kaggle.com/datasets. Accessed: Apr. 24, 2023.
[15] M. Aydin and N. Baykal, "Feature extraction and classification phishing websites based on URL," 2015.
[16] A. S. Raja, R. Vinodini, and A. Kavitha, "Lexical features based malicious URL detection using machine learning techniques," Materials Today: Proceedings, vol. 47, pp. 163-166, 2021.
[17] W. Ali and A. A. Ahmed, "Hybrid intelligent phishing website prediction using deep neural networks with genetic algorithm-based feature selection and weighting," IET Information Security, vol. 13, pp. 659-669, 2019
[18] Q. Zou, L. Ni, T. Zhang, and Q. Wang, "Deep learning-based feature selection for remote sensing scene classification," IEEE Geoscience and remote sensing letters, vol. 12, pp. 2321-2325, 2015.

[19] Scikit-learn. (2023). Sklearn.svm.SVC. available: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html. Accessed: Apr. 23,2023.
[20] B. Bischl, M. Binder, M. Lang, T. Pielok, J. Richter, S. Coors, et al., "Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 13, p. e1484, 2023.
[21] J. Brownlee, Optimization for machine learning: Machine Learning Mastery, 2021.
[22] S.-J. Bu and H.-J. Kim, "Optimized URL Feature Selection Based on Genetic-Algorithm-Embedded Deep Learning for Phishing Website Detection," Electronics, vol. 11, p. 1090, 2022.
[23] Sklearn-deap. (2023). Available: https://github.com/rsteca/sklearn-deap/blob/master/evolutionary_search/optimize.py . Accessed: May, 16,2023.

تحسين كشف التصيد الاحتيالي: نهج هجين يجمع بين خوارزمية آلة متجه الدعم والخوارزمية الجينية

د. محمد مصباح الشح سارة المبروك بعيو
علوم الحاسوب، مدرسة العلوم الأساسية، أكاديمية الدراسات العليا - مصراتة

استلمت الورقة بتاريخ ي/ش/س، وقبلت بتاريخ ي/ش/س، ونشرت بتاريخ ي/ش/س

الكلمات المفتاحية:
الخوارزمية الجينية، آلة متجه الدعم، تعليم الألة، صفحات الويب الاحتيالية.

الملخص

نظرًا لأن الإنترنت أصبح جزءًا أساسيًا من حياة البشر، يستمتع عدد متزايد من الأشخاص بالراحة التي توفرها الإنترنت. ومع زيادة عدد مستخدمي الويب وتطبيقاته ونظرًا لأن غالبية المؤسسات المالية والعامة قامت مؤخرًا بترقية وتعزيز الخدمات المباشرة عبر الإنترنت المقدمة لعملائها، بالتالي تتزايد الهجمات على منصات الإنترنت المختلفة بشكل تدريجي. يستخدم المهاجمون أساليب مختلفة لسرقة المعلومات الحساسة للمستخدمين، إحدى الحيل الأكثر شيوعًا هي مواقع التصيد الاحتيالي. لذلك ظهرت الحاجة إلى التصدي لهذه الهجمات والتحسين المستمر للتقنيات التي تعمل على اكتشافها، والتي من ضمنها أساليب التعلم الآلي التي تعمل على تصنيف ما إذا كان الموقع تصيدًا أم لا. وذلك اعتمادًا على عدد من البيانات التي يتم الحصول عليها من صفحات الويب الاخرى. نتيجة لذلك تهدف هذه الورقة إلى اقتراح وسيلة للكشف وتصنيف مواقع الويب الاحتيالية باستخدام خوارزمية **Support Vector Machine** وتحسينها باستخدام الخوارزمية الجينية **Genetic Algorithm** للحصول على أفضل دقة في التصنيف، بالإضافة لاستخدامها كخوارزمية لاختيار أفضل الميزات من الميزات المستخرجة. تتكون مجموعة البيانات التي تم جمعها من 12,000عينة، حيث تم جمع عناوين الويب الغير شرعية من موقع **PhishTank** وتم جمع عناوين الويب الشرعية من موقع **Kaggle** وتم استخدام مقاييس **accuracy, precision, recall and F1-score** لتقييم أداء الطريقة المقترحة. تمت مقارنة النتائج المتحصل عليها مع نتائج دراسة سابقة باستخدام خوارزميتي **SVM** مع **Ant Colony Optimization**. حيث اظهرت هذه النتائج أن دقة التصنيف للنهج المقترح حققت 97.62% وهي أعلى بنسبة 9.29% من نموذج **SVM** التقليدي، وتساوت تقريبا مع النموذج **SVM-ACO** الاخر.