

Web Phishing Detection Using a Modified Spider Monkey Optimization Algorithm

Hanadi G. ALsheyab^{1,*}, Mosleh M. Abualhaj², Mahran Al Zyoud³, Mohamed Yousif⁴, and Mohammad Sh. Daoud⁵

^{1,2,3} Department of Networks and Cybersecurity, Faculty of Information Technology, Al-Ahliyya Amman University, Amman, Jordan.

⁴ School of Technologies, Cardiff Metropolitan University, Cardiff, UK.

⁵ College of Engineering, Al Ain University, Abu Dhabi, United Arab Emirates.

* Corresponding author: handi.alshyab@hotmail.com

Received: 13 May 2026 **Accepted:** 20 August 2026 **Published:** 7 October 2026

DOI: 10.15849/ijasca.125 **License:** Open access under CC BY 4.0. Authors retain copyright.

Publisher: Al-Zaytoonah University of Jordan (ZUJ), Amman, Jordan.

Cite this article: H. G. ALsheyab, M. M. Abualhaj, M. Al Zyoud, M. Yousif, and M. Sh. Daoud, "Web phishing detection using a modified spider monkey optimization algorithm," *Int. J. Advance Soft Compu. Appl.*, vol. 18, no. 3, pp. 317–334, 2026, doi: <https://doi.org/10.15849/ijasca.125>.

ABSTRACT

In today's digital age, technology is both a powerful tool and a potential threat, with its widespread adoption bringing significant advancements as well as new vulnerabilities. As we increasingly rely on digital solutions for everyday tasks, the importance of safeguarding against cyber threats has become paramount. This is particularly true in the context of web security, where cybercriminals are continuously developing more sophisticated methods to exploit online systems. Given the increasing sophistication and frequency of phishing attacks, traditional phishing detection methods remain necessary. However, more advanced methods are needed to detect new attacks. This study presents a web phishing detection approach using machine learning (ML) algorithms and an enhanced feature selection (FS) through the modified Spider Monkey Optimization (SMO) algorithm. The SMO algorithm was modified by adding an effective crossover function to improve the algorithm's performance. The study employs Mendeley phishing datasets containing features extracted from both phishing and legitimate web pages. In binary classification, the proposed detection model consistently demonstrated strong performance using XGBoost, LightGBM, and Random Forest.

Keywords: ML; FS; SMO; metaheuristic algorithm.

1 Introduction

Phishing websites are designed to device users into revealing their personal, financial, or login details. Phishing sites are designed to imitate legitimate and well-known websites such as banks, social media platforms, and email services. The goal of the attackers is to steal personal information and login credential details from users who interact with these services regularly. This leads to accounts being taken, financial losses, and identity theft [1]. There are many types of phishing, including email phishing, web phishing, and spear phishing. Email phishing involves sending malicious messages or links via email. Meanwhile, web phishing involves creating websites imitate legitime ones that are designed to steal user's login credentials and information, while spear phishing targets a specific individual with the phishing attack. In short, each type of phishing attack has its own characteristics, but the goal is always malicious, such as stealing information, infecting users' systems, or gaining unauthorized access. Awareness is the first and one of the most effective lines of defense against these attacks [2,3]. Phishing websites are a serious and growing threat online, deceiving users by impersonating legitimate organizations. The APWG Phishing Activity Trends Report illustrates the increase in phishing attacks from 2015 to 2020 on a quarterly basis, as shown in Fig. 1; during the third quarter of 2019, phishing attacks increased to 266,387, marking the highest point in three years since late 2016. This increased by 46% from 182,465 in the second quarter and was nearly double the 138,328 recorded in the fourth quarter of 2018. During that quarter, there were 118,260 distinct phishing emails reported to APWG. Additionally, it was discovered that 1,283 brands were targeted by phishing campaigns [4,5].

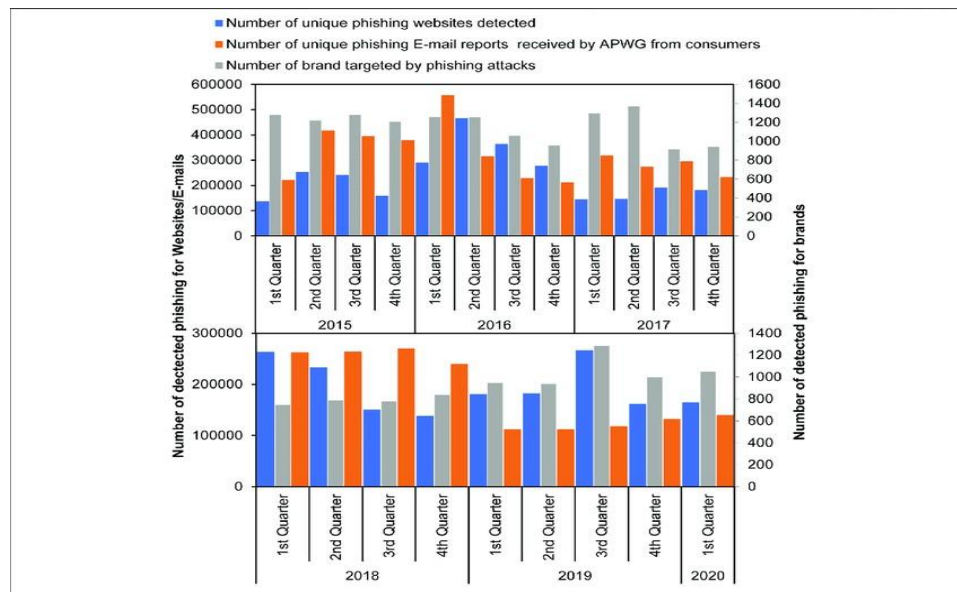


Fig. 1. The growth in phishing attacks 2015–2020 by quarters based on data collected from APWG annual reports [4,5].

Since phishing attacks are always increasing, this demands more effort and tools to combat these attacks and spread more awareness [6]. There are many methods for detecting phishing websites. The most blacklists, content analysis, heuristic-based analysis, and ML. One of the basic methods is the blacklist, which is a regularly updated database of known phishing URLs. It has the lowest real-time accuracy among all methods used [7,8]. The heuristic method uses some website characteristics, such as HTTPS, domain age, usage of certain words in URLs, offering a better but still not real-time response to new phishing sites. It relies on certain strategies and predefined rules [9]. Content analysis leverages natural language processing to inspect webpage text, looking for linguistic characteristics associated with phishing. This method analyses semantics and syntax to search for irregularities, which are commonly found in phishing websites, for example, urgency in language or grammatical mistakes not found in legitimate pages [10].

ML is used in phishing detection by analyzing large data stores called datasets. ML models are then used to predict whether the websites are phishing or legitimate [11]. There are three primary categories of ML: supervised learning, unsupervised learning, and reinforcement learning [12,13]. Supervised learning is the most common form of ML in which algorithms are trained on labeled datasets to accurately classify data or predict outcomes [14]. Models learn by taking input data and mapping it to the correct output; common examples include credit scoring and image recognition [15,16]. FS involves the selection of a subset of relevant features, referred to as predictors, that is necessary for creating a model [17]. The FS process removes irrelevant, redundant, or noisy data that does not contribute to building the model, hence making the model less inefficient and less accurate [18,19]. Metaheuristic algorithms have demonstrated their efficiency in FS [20,21]; among these, the SMO is effective in FS because it selects the best subset of features for the models [22]. The study evaluates the model performance on multiple web phishing datasets utilizing the Mendeley web phishing dataset.

2 Literature Review

The following literature explores the usage of the Mendeley phishing dataset; authors have used ML algorithms to make robust ML models for phishing detection.

Sarma et al. [23] also proposed a robust detection framework that uses various ML algorithms to enhance security against phishing attacks. Focusing on classifiers such as KNN, RF, SVM, and LR, the study uses the Mendeley phishing dataset. The performance of these classifiers was tested with metrics like precision, recall, and F1 scores to accurately identify phishing threats. The results pointed out RF as the best classifier with the highest performance scores, while 98% was a notable accuracy.

Al-Sarem et al. [24] discussed the use of GA to perform FS in web phishing detection. Some of the main classifiers that were studied are RFs, AdaBoost, XGBoost, Bagging, GradientBoost, and LightGBM. This technique was applied to three datasets containing features of phishing and legitimate websites, a comprehensive approach toward identifying harmful and safe internet entities. These classifiers optimized the performance immensely on the task of phishing detection on all datasets. The best performance was on Dataset 2, the Mendeley phishing dataset, with the LightGBM classifier reporting an impressive 98.65% accuracy, high precision, and F-score metrics.

Jayaprakash et al. [25] aimed to identify and mitigate phishing threats across different platforms, including URLs, emails, and websites using ML. The ML models are trained on the datasets used, including URL Phishing, Email Phishing, and Web phishing datasets. The highest accuracy for URL Phishing Detection is 97.2%, and for Email Phishing Detection is 97.4%. On the other hand, the results for the Mendeley phishing dataset used RF, SVM, and NB. The highest results for the ML model were achieved using the RF with an accuracy of 95.3, precision of 0.95, recall of 0.96, and F1-score of 0.96.

The authors of Newaz et al. [26] have also presented a framework for web phishing detection that would correctly identify phishing websites from non-phishing websites. The authors used various datasets for their framework: Dataset-1 (UCI Phishing Websites Dataset), Dataset-2 (Mendeley Phishing Dataset), Dataset-3 (Phishing Websites Dataset), and Dataset-4 (Web page phishing detection). The authors did FS using RFE to find the best subset of features for the ML algorithms. The classification was done with several classifiers such as SVM, KNN, NB, LR, RF, AdaBoost, ET, GBM, LGBM, and XGBoost. The highest accuracies on Dataset-2 for the XGBoost, LGBM, and RF were 98.15, 98.02, and 97.72, respectively. Precision scores were 98.46, 98.32, and 98.22, respectively. F1-scores were 98.14, 98.02, and 97.71, respectively.

On the other hand, Calzarossa & Zieni [27] compared different ML methods for phishing detection on the Mendeley phishing dataset. Focusing on understanding the explanations provided by ML models for detecting phishing websites, the study examined several classifiers, including KNN, LR, RF, SVM, and XGBoost, with an accuracy of 0.639, 0.933, 0.965, 0.934, and 0.965, respectively.

These studies have shown that ML can improve web security by making it easier to spot phishing attacks. Using datasets like Mendeley, researchers have used different ML classifiers to identify phishing websites from normal ones. These studies show the scarcity of using metaheuristics like SMO in FS in web phishing detection on the Mendeley dataset, which could be an area for future work.

3 Proposed SMO Phishing Website Detection

FS is the process of identifying the important features that affect the performance of the ML-based detection model; by removing irrelevant or redundant features from the data, the model's accuracy is improved in detecting phishing websites. Metaheuristic algorithms have demonstrated their efficiency in FS. Among these is the SMO algorithm, as it is effective in navigating the complex search space for FS. SMO is a metaheuristic algorithm used to solve optimization problems and perform FS. It was inspired by the behavior of spider monkeys. Here are the steps of the SMO algorithm:

- 1. Initializing the Population:** The SMO starts by initializing a population of potential solutions, represented by "spider monkeys." Each monkey represents a potential solution to the optimization problem.
- 2. Computing Solution Fitness:** The algorithm evaluates the fitness of each solution; the fitness evaluation helps in identifying the most promising solutions.
- 3. Selecting the Local and Global Leader:** The algorithm selects a local leader for each subgroup of monkeys and a global leader for the entire population. The local leader is the best solution within its subgroup, and the global leader is the best solution among all subgroups. These leaders play a crucial role in guiding the algorithm's process.
- 4. Update the Solutions of Both the Local and Global Leaders:** The entire population of solutions is updated based on local leaders and the global leader using the algorithm's mathematical operations.

5. Update Local and Global Leader Decision Stage: The algorithm enters a decision stage for these leaders, allowing them to explore the solution space for different possible solutions (exploration) and exploit the solution space by searching for better solutions.

6. Stopping Criteria Met: The algorithm continuously runs until a stopping criterion is reached, such as a maximum number of iterations. If the criteria is met, then the algorithm terminates and returns the best solution found so far. The flowchart for the SMO is outlined in Fig. 2.

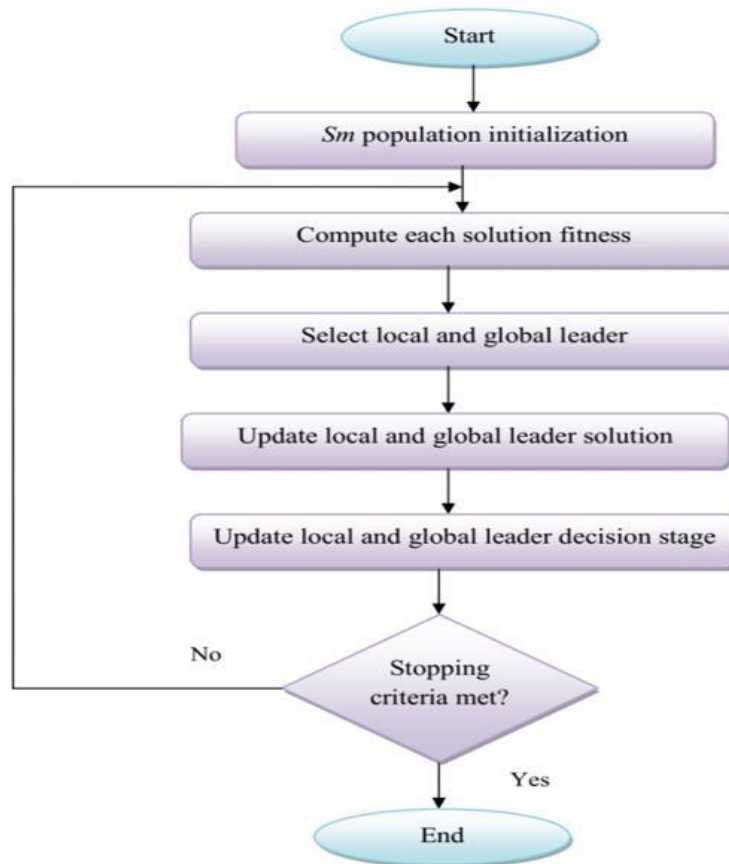


Fig. 2. Flowchart of the spider monkey optimization algorithm.

The aim of this research is to develop an ML model for phishing website detection. The steps for the proposed web phishing detection model are shown in Fig. 3. The Phishing Dataset for Machine Learning (Mendeley) is used for training and evaluation of the model. The dataset undergoes preprocessing steps to improve the performance of the ML classification algorithms. Data normalization and FS are the preprocessing steps applied to the dataset. Normalization was performed using the MinMaxScaler to transform the numerical values into a common range, as most classification algorithms and the SMO require all input variables to be on the same scale (range). FS is the second and an important step in data preprocessing since the irrelevant features used in the ML model for detection can negatively affect the performance of the ML model in identifying phishing websites. The SMO is used for FS to help eliminate the irrelevant features from the model, improving its performance. The last step in the model is classification, where the best subset of features will be applied along with several ML techniques to evaluate the model performance.

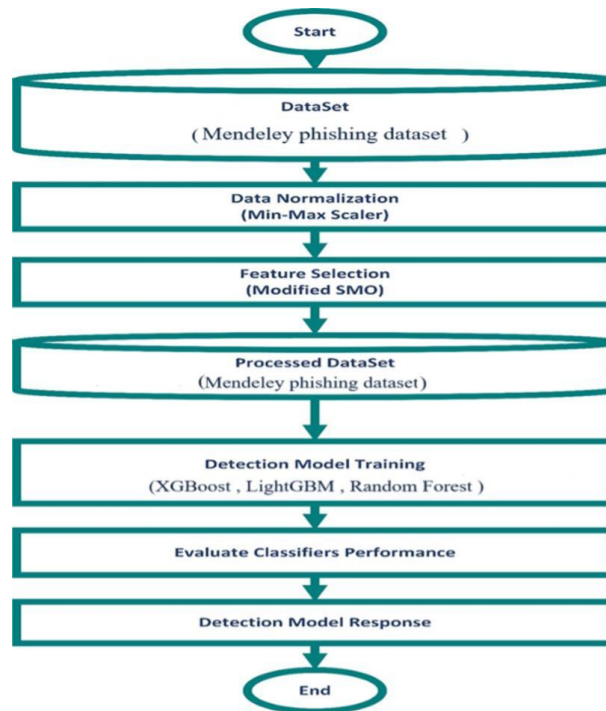


Fig. 3. The proposed web phishing detection model.

3.1 Mendeley Dataset

The Phishing Dataset for Machine Learning: Feature Evaluation, also called the Mendeley phishing dataset, contains 48 features extracted from 5000 phishing webpages and 5000 legitimate webpages, which were downloaded from January to May 2015 and from May to June 2017. The dataset balance of phishing and legitimate websites ensures a realistic evaluation of phishing detection models. The authors of the dataset automated the collection of the webpages using the GNU Wget tool and a Python script, with the complete HTML documents downloaded with their related resources (e.g., images, CSS, JavaScript) to ensure all downloaded webpages can be rendered properly in the browser. The dataset phishing webpage sources were collected as follows: 5000 phishing webpages from PhishTank and OpenPhish and another 5000 legitimate webpages from Alexa and the Common Crawl archive [28]. Table 1 describes the features of the Phishing Dataset for ML.

Table 1. Phishing dataset for ML feature description.

#	Feature	#	Feature	#	Feature	#	Feature
1	NumDots	13	NumHash	25	NumSensitiveWords	37	RightClickDisabled
2	SubdomainLevel	14	NumNumericChars	26	EmbeddedBrandName	38	PopUpWindow
3	PathLevel	15	NoHttps	27	PctExtHyperlinks	39	SubmitInfoToEmail
4	UrlLength	16	RandomString	28	PctExtResourceUrls	40	IframeOrFrame
5	NumDash	17	IpAddress	29	ExtFavicon	41	MissingTitle
6	NumDashInHostname	18	DomainInSubdomains	30	InsecureForms	42	ImagesOnlyInForm
7	AtSymbol	19	DomainInPaths	31	RelativeFormAction	43	SubdomainLevelRT
8	TildeSymbol	20	HttpsInHostname	32	ExtFormAction	44	UrlLengthRT
9	NumUnderscore	21	HostnameLength	33	AbnormalFormAction	45	PctExtResourceUrlsRT
10	NumPercent	22	PathLength	34	PctNullSelfRedirectHyperlinks	46	AbnormalExtFormActionR
11	NumQueryComponents	23	QueryLength	35	FrequentDomainNameMismatch	47	ExtMetaScriptLinkRT
12	NumAmpersand	24	DoubleSlashInPath	36	FakeLinkInStatusBar	48	PctExtNullSelfRedirectHyperlinksRT

3.2 Implementation Operations

The phishing detection model was implemented using a high-performance Microsoft computer running Windows. The Visual Studio Code IDE was used to implement and edit the code. The experiment was conducted using Python (v3.11) with the Anaconda Python package. Anaconda comes with many popular data science and ML libraries.

The implementation required multiple ML libraries to develop the ML model. These some of the libraries are outlined in Table 2.

Table 2. ML libraries used for model implementation.

Library Name	Description and Use
numpy	A library for numerical computing with arrays and matrices used for mathematical operations and array manipulations.
pandas	Data manipulation and analysis library, used for data handling and reading/writing CSV files.
sklearn.ensemble	Ensemble methods module from scikit-learn, used for classification using RF and ET.
sklearn.tree	Decision Tree (DT) module from scikit-learn, used for classification using DT.
sklearn.inspection	Model inspection module from scikit-learn, used for determining permutation importance.
sklearn.metrics	Metrics module from scikit-learn, used for evaluation metrics like accuracy, ROC AUC, etc.
sklearn.preprocessing	Preprocessing module from scikit-learn, used for Label Encoding and MinMaxScaler.
sklearn.pipeline	Pipeline module from scikit-learn, used for building a pipeline for preprocessing.

The implementation of the proposed phishing detection ML model using the Modified SMO consisted of three main steps: data normalization, FS using the modified SMO, and ML classification.

3.3.1 Data Normalization

Normalization rescales features to fall within a certain range, typically between 0 and 1. The MinMaxScaler is one of the most used techniques for normalization. Many classification algorithms, as well as optimization methods like SMO, assume that the input variables are on the same scale, so this step is critical. If the inputs are not on the same scale, the performance of the model can be negatively affected. The MinMaxScaler is necessary to ensure that every feature is on the same scale for the detection model. The MinMaxScaler was utilized to normalize the features (NumDots, SubdomainLevel, PathLevel, UrlLength, NumDash, NumDashInHostname, NumUnderscore, NumPercent, NumQueryComponents, NumAmpersand, NumNumericChars, HostnameLength, PathLength, QueryLength, NumSensitiveWords).

The MinMaxScaler is used with the equation (1):

$$X_{Scaled} = \frac{X - X_{Min}}{X_{Max} - X_{Min}} \quad (1)$$

Where XScaled is the result of the MinMaxScaler, X is the original value, XMin is the minimum value in the column, and XMax is the maximum value in the column. The MinMaxScaler was used for the features in the dataset.

3.3.2 FS Using the Modified SMO

The second stage of the web phishing detection ML model implementation is FS using the modified SMO. The population size for the modified SMO was set to 10, and the maximum number of iterations was 100. The number of iterations and population size are often sufficient for the algorithm to efficiently reach an optimal solution [29,30]. These parameters strike a balance between computational efficiency and the evaluation of the solution to find the error rate. The fitness function is based on the accuracy of the solutions, as well as the number of selected features. The fitness function gives more importance to

the accuracy of the model using the alpha parameter set to (0.99), and beta is assigned to the ratio of the number of selected features to the maximum number of features, with a value of (0.01).

A lower fitness value indicated a better feature subset. The fitness function is shown in the following equation (2).

$$\text{Fitness function} = \alpha * (1 - \text{Accuracy}) + \beta * \left(\frac{\text{number of selected features}}{\text{maximum number of features}} \right) \quad (2)$$

For the Mendeley dataset, the modified SMO for the FS selected 26 important features for the ML model to make predictions on the Mendeley dataset. The two most influential features are “PctExtHyperlinks,” which is the ratio of hyperlinks pointing to an external website, and “PctNullSelfRedirectHyperlinks,” which is the percentage of hyperlink fields containing an empty value with a self-redirect value. Another feature is “FrequentDomainNameMismatch,” which indicates if the most frequent domain name in the HTML source code does not match the domain of the webpage URL. These features are highly ranked in the model, showing their importance to the model for predicting phishing attempts more accurately. This is important to understand which characteristics are most important in differentiating legitimate websites from phishing websites, as shown in Fig. 4.

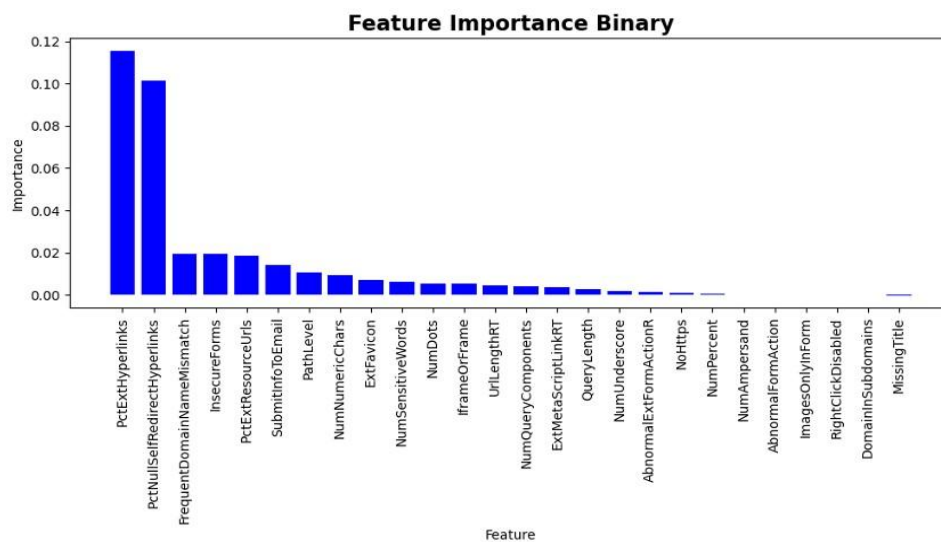


Fig. 4. SMO selected feature importance, Mendeley dataset.

The SMO convergence is shown in Fig. 5, which shows the SMO fitness values over 100 iterations for the FS process on the Mendeley dataset. It shows the algorithm reaching the best solution (global optimal solution) at approximately the 55th iteration out of 100 iterations.

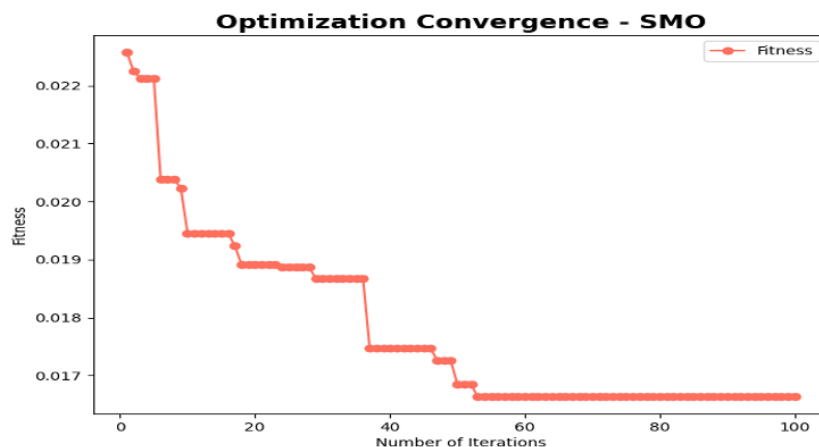


Fig. 5. Modified SMO convergence plot, Mendeley dataset.

FS is the process of identifying the important features that impact the performance of the detection ML model by removing irrelevant or redundant features. The SMO has demonstrated its effectiveness in navigating the complex search space of FS. SMO is utilized for FS for the web phishing detection model. However, the algorithm suffers from a common problem called local optima, as do all metaheuristic algorithms. Local optima refer to a situation where the algorithm finds a solution and considers it the best one, although if the search space were explored more effectively a better solution could have been discovered. The SMO can be trapped in local optima. To address this issue, researchers have developed new methods based on local search schemes and improvements such as crossover functions [31]. In the modified SMO, a crossover function was introduced to improve the algorithm's chances of avoiding local optima. Fig. 6 outlines the placement of the crossover function.

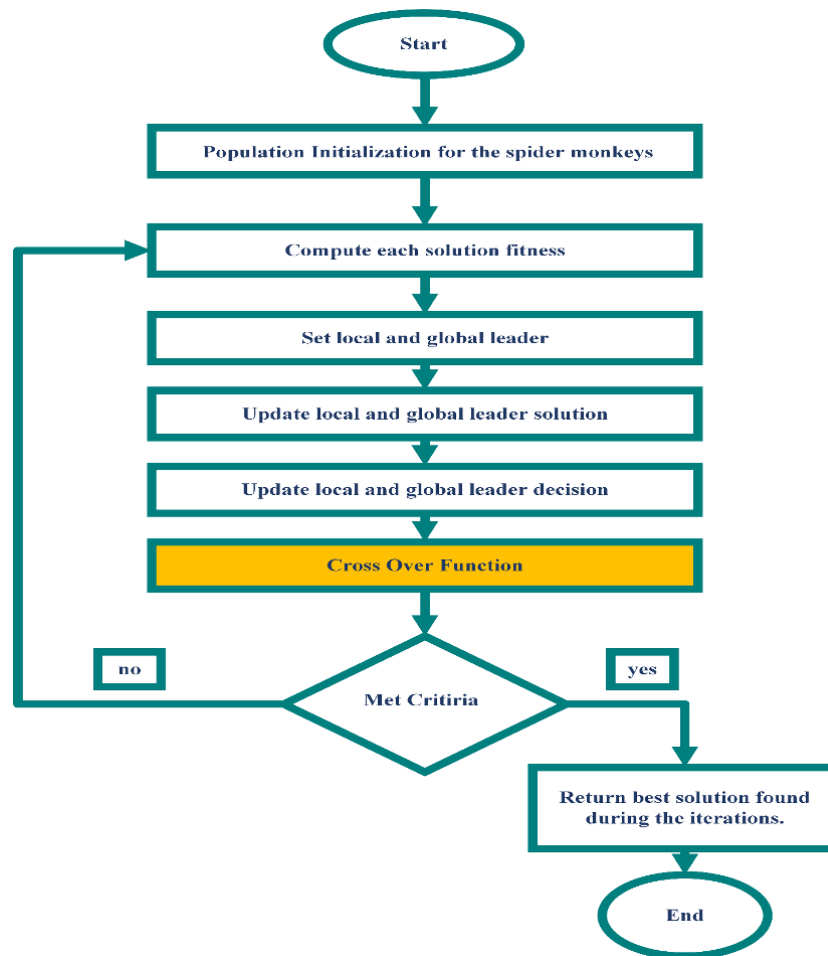


Fig. 6. Flowchart of the modified spider monkey optimization algorithm.

This is done by generating new offspring solutions by combining features from the best-performing solution (global leader) and the local leaders (the best solutions within the subgroups) of the population. This process improves the algorithm's ability to explore the solution space more effectively. The breakdown of the crossover process is presented below:

Selecting Parents for the Crossover: The crossover function selects the global leader, which is the best solution based on fitness, to serve as the first parent, and the second parent is selected from the best solutions of the subgroups in the population. A Gaussian crossover operation is then performed between the two selected parents, which generates two offspring solutions. The crossover is repeated between the global leader and each local leader. The resulting offspring are added to the population.

The Gaussian Crossover: The Gaussian crossover generates two new offspring by mixing the genes of two parents; it uses a blending factor combined with a random value β drawn from a Gaussian distribution to add diversity. This method is applied to each gene (feature) of two parents, producing two new offspring with greater variation.

$$\begin{cases} O_{1i} = \alpha P_{1i} + (1 - \alpha)P_{2i} + \beta \\ O_{2i} = \alpha P_{2i} + (1 - \alpha)P_{1i} + \beta \end{cases} \quad (3)$$

For each gene i :

These are the genes of the first and second offspring, respectively. Are the corresponding genes from parent one and parent two. Is the blending factor set to 0.5, determining how much of each gene is inherited from each parent. β is a random value from a Gaussian distribution, adding unique variations to each gene, thereby enhancing genetic diversity in the offspring.

Fitness Evaluation: The entire population, with the added offspring, undergoes a fitness evaluation. This evaluation is used to determine the most effective solutions.

Selecting Top Individuals: After evaluating fitness, the algorithm selects the top-performing solutions equal to the original population size, where lower-performing solutions are discarded, maintaining the population size and retaining the best solutions. Using the crossover function enhances the performance of the phishing detection model by finding better feature subsets.

3.3.3 Classification

The phishing detection model depends on classification to identify whether a given website is legitimate or phishing by previously training it on labeled data. The phishing detection model is trained and evaluated using the following algorithms: XGBoost, Random Forest (RF), and LightGBM. It uses various classifiers because for each classification algorithm, a different mathematical model is employed for pattern recognition and prediction. Some classification algorithms may perform better in detecting phishing websites. This model improves security by including several classifiers, therefore increasing its likelihood of identifying a successful approach for the detection of phishing websites. To ensure a robust and reliable evaluation, 5-fold cross-validation ($k = 5$) was employed, where the dataset was divided into five subsets and the model was iteratively trained on four folds and evaluated on the remaining fold.

XGBoost has its own implementation of Decision Tree (DT) that helps in making predictions faster and better. It builds each tree one by one, with each new tree fixing mistakes made by the previous trees, as shown in Fig. 7 [32]. RF is a method of building multiple DTs, passing new data through each tree and using voting based on the classification results of each tree, then selecting the result with the most votes as the final classification. Fig. 8 shows an example of the structure of the RF [33]. LightGBM uses a leaf-wise growth strategy with a depth limit to find the leaf node with the largest split gain among the current leaf nodes, then splits, and the process continues, as shown in Fig. 9 [34].

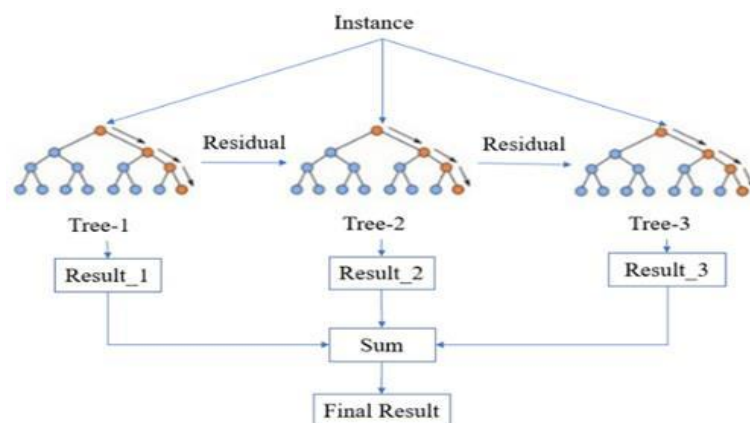


Fig. 7. Extreme gradient boosting [32].

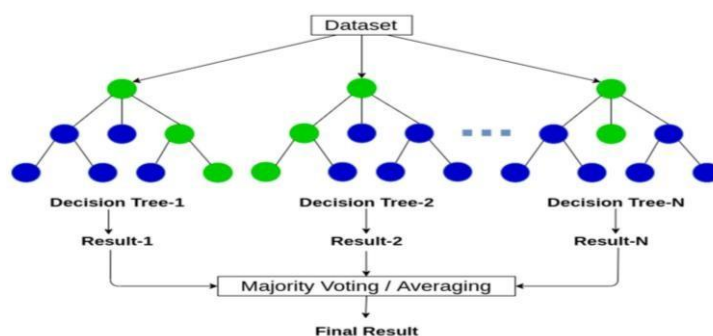


Fig. 8. Structure of the random forest model [33].

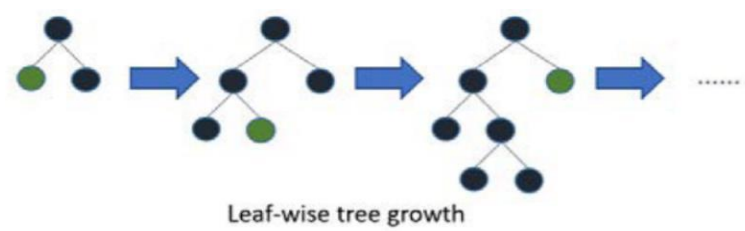


Fig. 9. LightGBM leaf-wise growth strategy in the classification [34].

4 Results and Discussion

4.1 The Modified SMO Binary Results

The phishing detection model was evaluated using multiple ML algorithms, including XGBoost, RF, and LightGBM. The model achieved strong results across all metrics, as summarized in Table 3.

Table 3. Mendeley dataset binary classification results.

Classifier	Accuracy	Precision	F1-score	Recall
XGBoost	98.87	98.87	98.9	99.02
LightGBM	98.77	98.77	98.8	98.89
RF	98.37	98.37	98.4	98.18

Among the classifiers using the modified SMO, XGBoost had the highest accuracy of 98.87%, followed by LightGBM at 98.77% and RF at 98.37%. The ensemble methods consistently achieved high performance across all evaluation metrics on the Mendeley dataset.

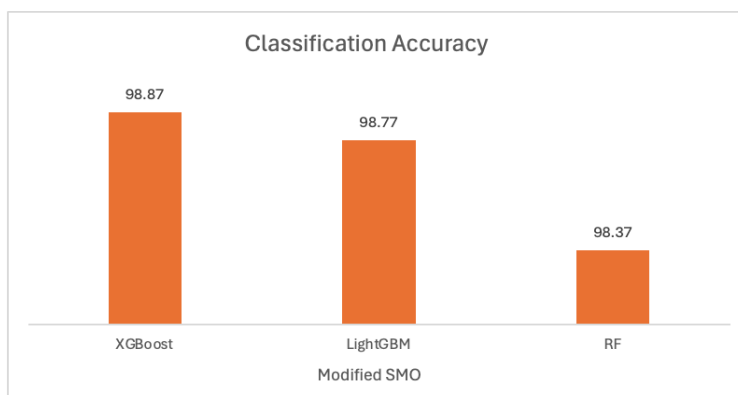


Fig. 10. Mendeley dataset classification accuracy results.

As shown in Fig. 11, the precision results show a similar trend to the accuracy results. XGBoost and LightGBM obtained near-identical precision scores of 98.87% and 98.77%, respectively, while RF achieved 98.37%.

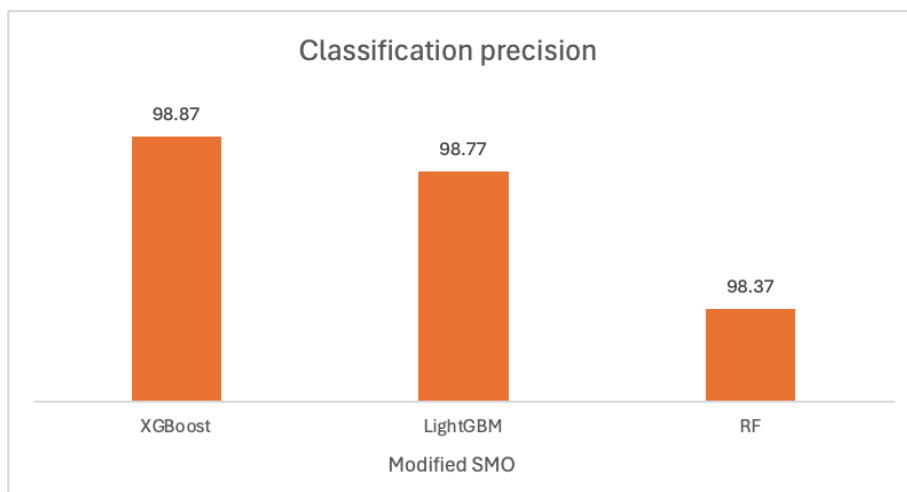


Fig. 11. Mendeley dataset classification precision results.

As shown in Fig. 12, the F1 scores, which represent the balance between precision and recall, show XGBoost at 98.90% and LightGBM at 98.80%. These values indicate a strong balance between precision and recall for both classifiers, with RF close behind at 98.40%.

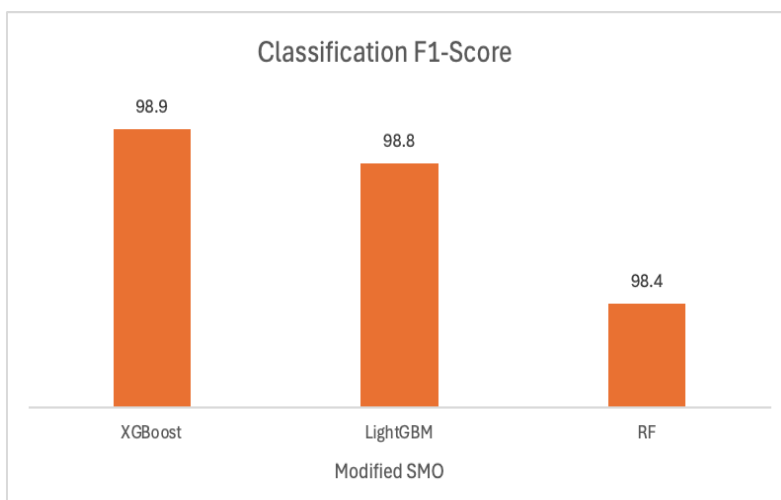


Fig. 12. Mendeley dataset classification F1-score results.

As shown in Fig. 13, the recall results show the best-performing models: XGBoost achieved the highest recall of 99.02%, which indicates that it correctly identifies the largest proportion of actual phishing cases, followed by LightGBM at 98.89% and RF at 98.18%.

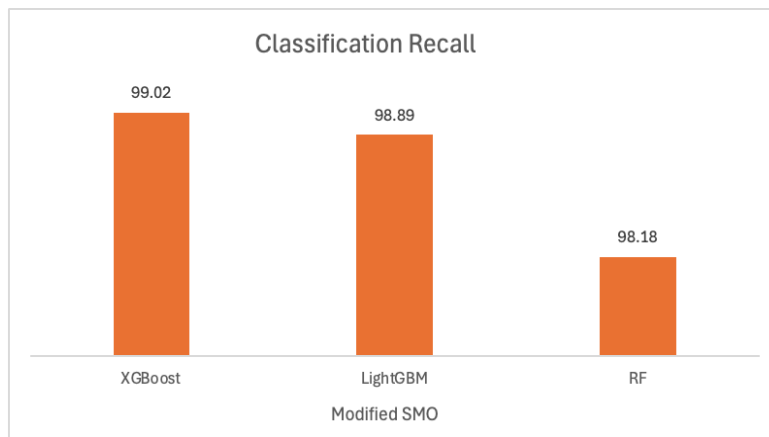


Fig. 13. Mendeley dataset classification recall results.

4.2 XGBoost Results

Fig. 14 shows the CM for the XGBoost classifier, which correctly classified 1,444 normal instances as normal (TN) and 1,522 phishing instances as phishing (TP).

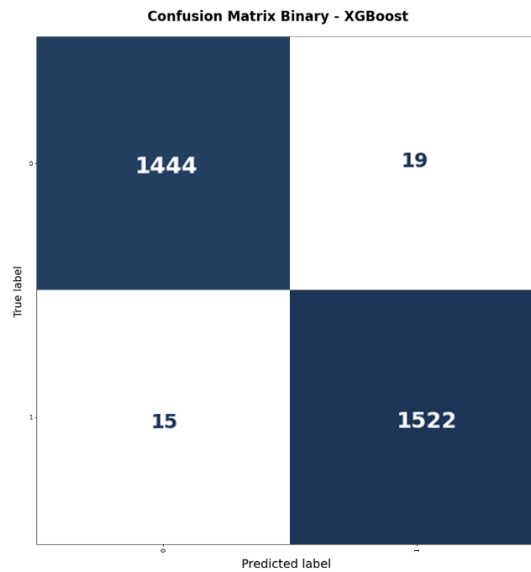


Fig. 14. Mendeley dataset XGBoost confusion matrix.

4.3 LightGBM Results

Fig. 15 shows the CM for the LightGBM classifier. LightGBM correctly classified 1,443 normal instances as normal (TN) and 1,520 phishing instances as phishing (TP).

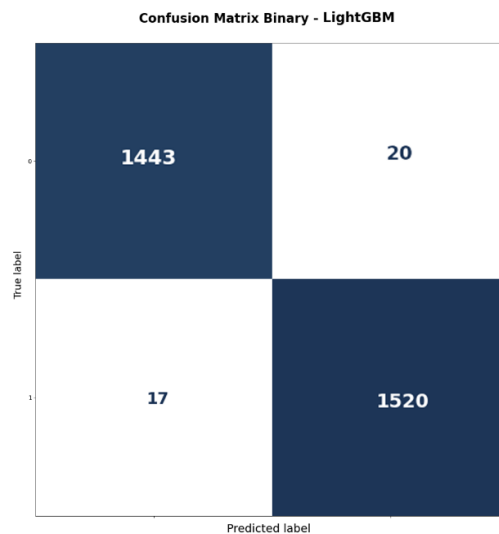


Fig. 15. Mendeley dataset LightGBM confusion matrix.

4.4 Random Forest Results

Fig. 16 shows the CM for the RF classifier, which correctly classified 1,442 normal cases as normal (TN) and 1,509 phishing cases as phishing (TP). The RF misclassified a slightly higher number of cases than XGBoost and LightGBM, consistent with its lower accuracy, precision, F1-score, and recall values reported in Table 3.

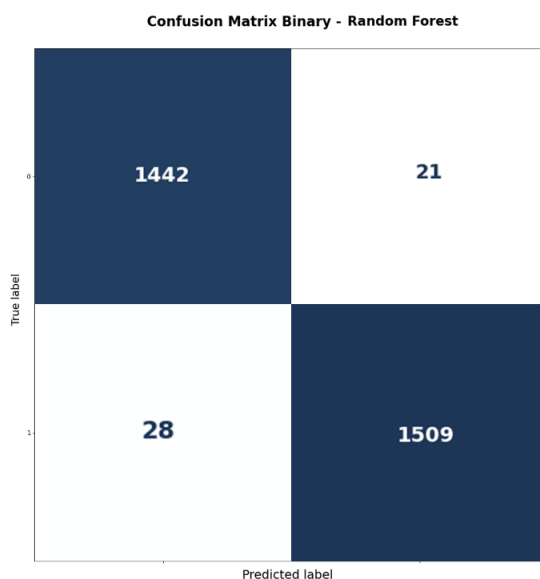


Fig. 16. Mendeley dataset random forest confusion matrix.

4.5 Original SMO and Modified SMO

The modified SMO shows improvements in accuracy. LightGBM improved from 98.70% to 98.77%. RF also improved from 98.37%. XGBoost maintained the same result as the original SMO, indicating that the modification primarily benefits classifiers that were not already at their performance ceiling.

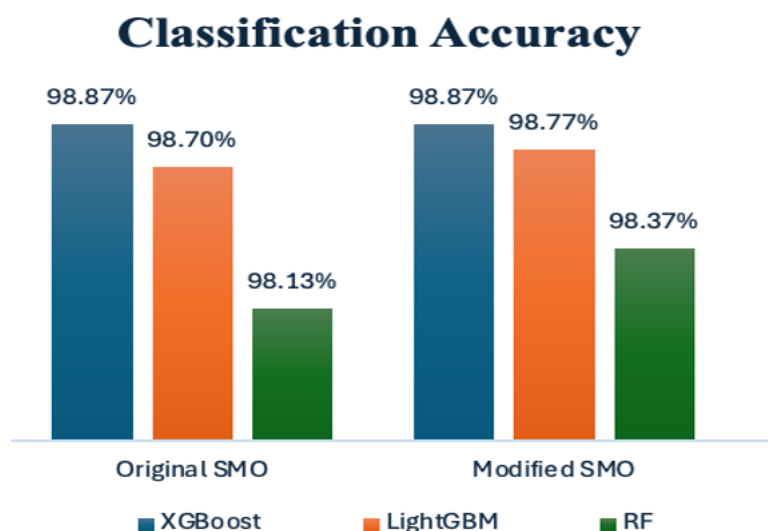


Fig. 17. Mendeley dataset accuracy comparison, original SMO vs. modified SMO.

Additionally, precision also improved with the modified SMO. Specifically, RF shows a significant increase: RF precision increased from 98.13% to 98.37%. LightGBM also showed a modest improvement.

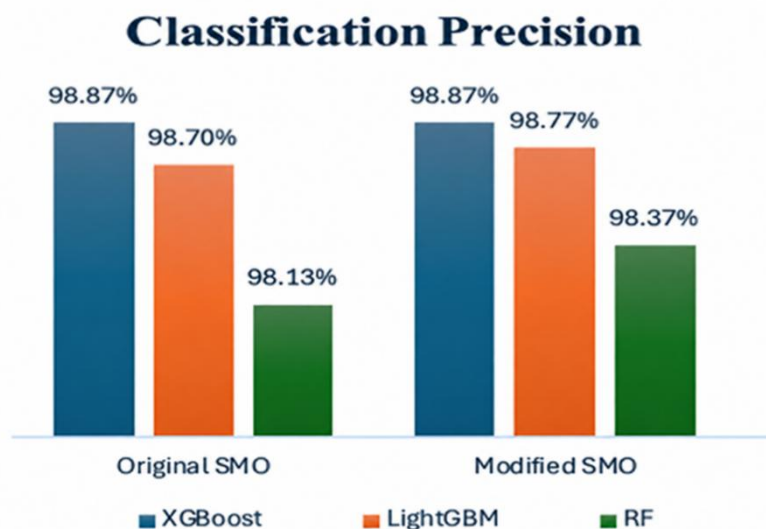


Fig. 18. Mendeley dataset precision comparison, original SMO vs. modified SMO.

Similarly, the F1 score also shows improvements. The F1 score for RF improved from 98.18% to 98.40%. LightGBM also improved from 98.70% to 98.77%. For XGBoost, the F1 score remained essentially unchanged.

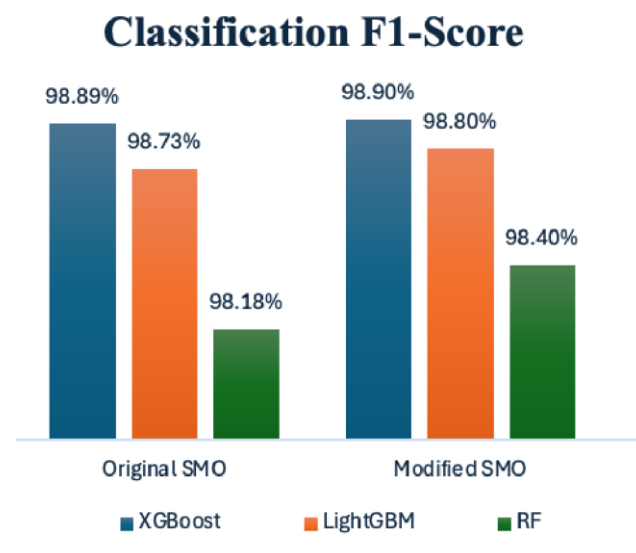


Fig. 19. Mendeley dataset F1-score comparison, original SMO vs. modified SMO.

Finally, recall highlights the improvements achieved with the modified SMO. The recall rates for all classifiers increased, indicating that the number of missed phishing detections was reduced relative to the original SMO.

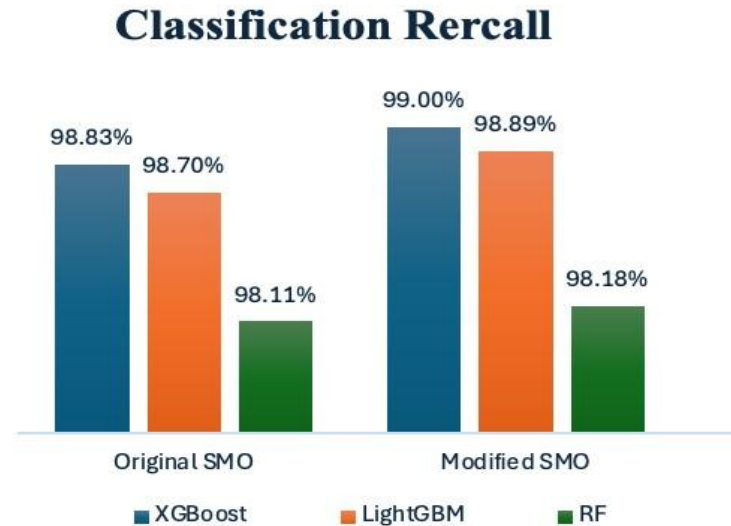


Fig. 20. Mendeley dataset recall comparison, original SMO vs. modified SMO.

4.6 The Modified SMO Model vs. Other Phishing Detection Models

The modified SMO model was tested for web phishing detection using binary classification and compared with other detection models using the Mendeley web phishing dataset, as summarized in Table 4.

Table 4. Comparison of the modified SMO model with other phishing detection models.

Reference	Classifier	Accuracy	Precision	F1-score	Recall
The proposed SMO Model	XGBoost	98.87	98.87	98.9	99.02
	LightGBM	98.77	98.77	98.8	98.89
	RF	98.37	98.37	98.4	98.18
Sarma et al., 2021	RF	98.00	97.70	98.00	98.40
Al-Sarem et al., 2021	RF	98.37	98.54	98.39	-
	XGBoost	97.70	97.57	97.71	-
	LightGBM	98.65	98.56	98.65	-
Jayaprakash et al., 2024	RF	95.30	95.00	96.00	-
Newaz et al., 2024	XGBoost	98.15	98.46	98.14	-
	LightGBM	98.02	98.32	98.02	-
	RF	97.72	98.22	97.71	-
Calzarossa & Zieni, 2024	RF	96.50	-	-	-

These results demonstrate that the proposed SMO model yields superior and consistent performance across various classifiers compared to the other approaches reported in the literature.

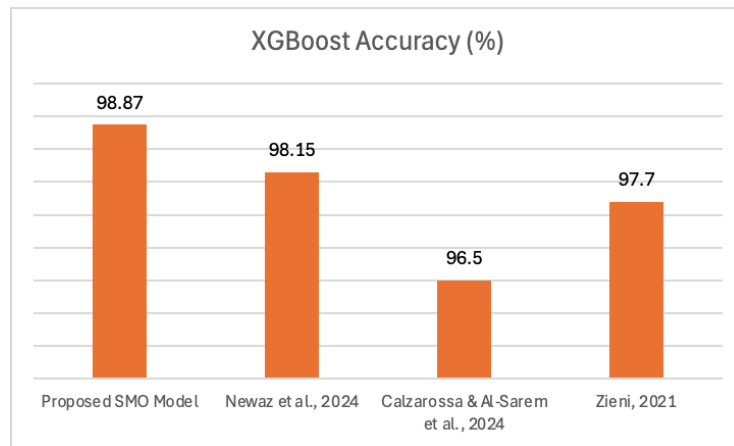


Fig. 21. Ranking XGBoost accuracy, modified SMO detection model vs. other models.

Fig. 22 provides a comparison between the different studies and the proposed SMO in terms of accuracy using LightGBM classifiers. The proposed SMO model again outperformed the compared studies.

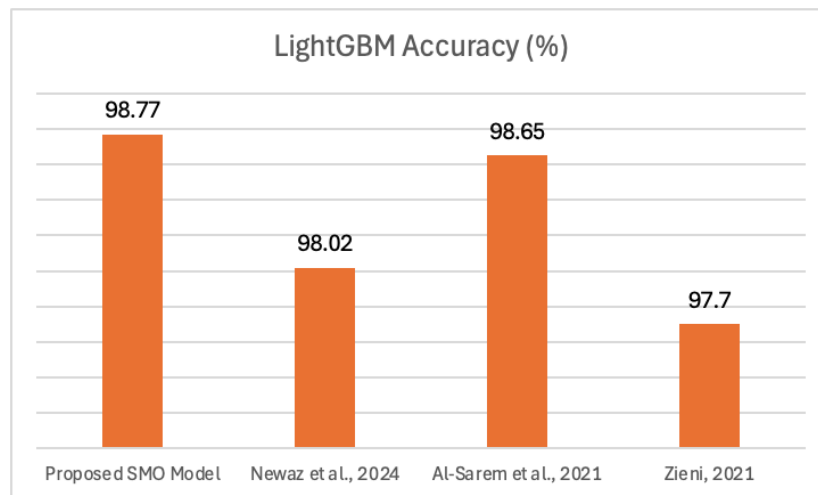


Fig. 22. Ranking LightGBM accuracy, modified SMO detection model vs. other models.

Fig. 23 provides a ranking comparison between the different studies and the proposed SMO in terms of accuracy using RF classifiers. The proposed SMO combined with RF ranked favorably against the compared studies.

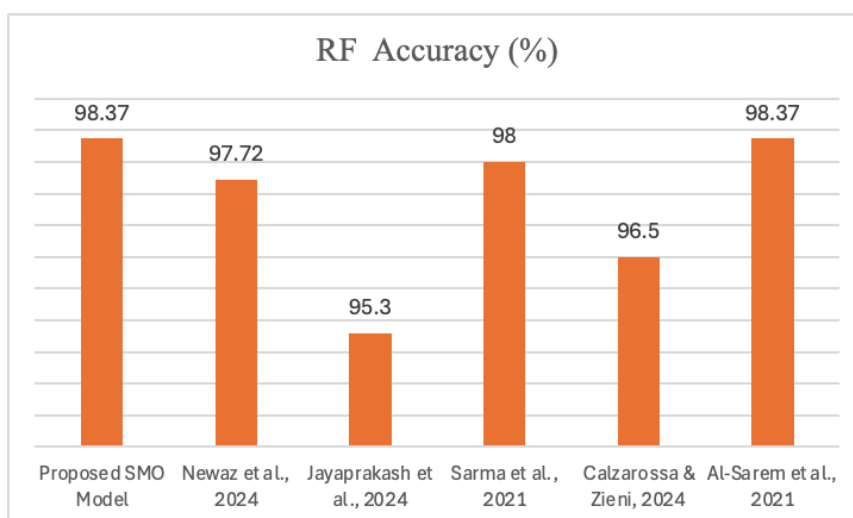


Fig. 23. Ranking random forest accuracy, modified SMO detection model vs. other models.

5 Conclusion

The modified SMO-based models using various ML algorithms show improvements in detecting phishing websites across several performance metrics compared to the original SMO and other state-of-the-art methods. They consistently enhanced the accuracy, precision, F1-score, and recall by applying the modified SMO to several ML classifiers such as XGBoost, LightGBM, and RF. XGBoost and LightGBM produced high-quality results, with XGBoost maintaining the lead at a near-perfect accuracy of 98.87%, while LightGBM improved over its original performance to reach 98.77%. The confusion matrices further revealed that the enhanced models demonstrated strong performance not only in overall accuracy but also in minimizing false positives and false negatives, thereby providing a more reliable layer of protection against phishing attacks. This modified SMO outperformed other models in phishing detection using the Mendeley dataset. The SMO models demonstrate competitive performance in web phishing detection and can contribute to improving cybersecurity defense against phishing attacks.

DECLARATIONS

Conflicts of interest. The authors declare that they have no competing interests.

Data availability. The Mendeley Phishing Dataset for ML used in this study is a publicly available benchmark dataset; see Section 3.1 for its description and source [28].

REFERENCES

- [1] M. M. Abualhaj, "Enhancing Spam Detection Using Hybrid of Harris Hawks and Firefly Optimization Algorithms," *Journal of Applied Data Sciences*, vol. 5, no. 3, pp. 901–911, Aug. 2024, doi: <https://doi.org/10.47738/jads.v5i3.279>.
- [2] Arshad, Attique Ur Rehman, S. Javaid, T. Ali, J. Sheikh, and M. Azeem, "A Systematic Literature Review on Phishing and Anti-Phishing Techniques," *arXiv (Cornell University)*, Apr. 2021, doi: <https://doi.org/10.48550/arxiv.2104.01255>.
- [3] M. M. Abualhaj, S. N. Al-Khatib, A. A. Abu-Shareha, A. Hyassat, and M. S. Daoud, "Smart firewall for phishing detection powered by bio-inspired algorithms," *Journal of Advances in Information Technology*, vol. 16, no. 11, pp. 1529–1539, 2025.
- [4] Z. Alkhalil, C. Hewage, L. Nawaf, and I. Khan, "Phishing Attacks: a Recent Comprehensive Study and a New Anatomy," *Frontiers in Computer Science*, vol. 3, no. 1, pp. 1–23, Mar. 2021. Available: <https://www.frontiersin.org/journals/computer-science/articles>
- [5] APWG, "APWG | Phishing Activity Trends Reports," *Apwg.org*, 2024. <https://apwg.org/trendsreports/>
- [6] A. Hyassat, S. Al-Khatib, M. M. Elshourbagy, M. M. Abualhaj, Q. Y. Shambour, M. Sh. Daoud, M. A. Alsharaiah, M. Anbar, "A Hybrid Metaheuristic Feature Selection Framework for Enhanced Phishing Detection," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 17, no. 10, pp. 1–17, 2026.
- [7] P. Tsigaris and J. A. Teixeira da Silva, "Why blacklists are not reliable: A theoretical framework," *The Journal of Academic Librarianship*, vol. 47, no. 1, p. 102266, Jan. 2021, doi: <https://doi.org/10.1016/j.acalib.2020.102266>.
- [8] L. Al-dabbas, "Early Detection of Female Type-2 Diabetes using Machine Learning and Oversampling Techniques," *Journal of Applied Data Sciences*, vol. 5, no. 3, pp. 1237–1245, Sep. 2024, doi: <https://doi.org/10.47738/jads.v5i3.298>.
- [9] C. M. R. da Silva, E. L. Feitosa, and V. C. Garcia, "Heuristic-based strategy for Phishing prediction: A survey of URL-based approach," *Computers & Security*, vol. 88, p. 101613, Jan. 2020, doi: <https://doi.org/10.1016/j.cose.2019.101613>.
- [10] Rania Zaimi, M. Hafidi, and M. Lamia, "A literature survey on anti-phishing in websites," pp. 1–7, Apr. 2021, doi: <https://doi.org/10.1145/3454127.3456580>.
- [11] D. Ali, M. Abualhaj, A. Abu-Shareha, M. Yousif, M. Daoud, M. Faheem, and H. Soltani, "Securing IoT Networks Against Cyber Threats Using an Improved Whale Optimization Algorithm," *Journal of Artificial Intelligence and Technology*, pp. 1–15, 2026, doi: <https://doi.org/10.37965/jait.2026.1033>
- [12] C. Janiesch, P. Zschech, and K. Heinrich, "Machine learning and deep learning," *Electronic Markets*, vol. 31, no. 31, pp. 685–695, Apr. 2021, doi: <https://doi.org/10.1007/s12525-021-00475-2>.
- [13] Almomani, A. Alsaaidah, A. A. Abu-Shareha, A. Alzaqebah, M. Amin Almaiah, and Q. Shambour, "Enhance URL Defacement Attack Detection Using Particle Swarm Optimization and Machine Learning," *Journal of Computational and Cognitive Engineering*, Feb. 2025.
- [14] H. A. Anabousi, M. M. Abualhaj, A. Abu-Shareha, M. Yousif, M. Sh. Daoud, M. R. Faheem, and H. Soltani, "Malware Detection Using Dynamic Analysis with a Modified Harris Hawk Optimizer," *Discover Artificial Intelligence*, vol. 6, Art. no. 1309, Sep. 2026, doi: <https://doi.org/10.1007/s44163-026-01359-0>.

- [15] T. Jiang, J. L. Gradus, and A. J. Rosellini, "Supervised Machine Learning: A Brief Primer," *Behavior Therapy*, vol. 51, no. 5, pp. 675–687, Sep. 2020, doi: <https://doi.org/10.1016/j.beth.2020.05.002>.
- [16] M. Abualhaj, M. O. Hiari, Adeeb Alsaaidah, Mahran Al-Zyoud, and Sumaya Al-Khatib, "Spam Feature Selection Using Firefly Metaheuristic Algorithm," *Journal of Applied Data Sciences*, vol. 5, no. 4, pp. 1692–1700, 2024, doi: <https://doi.org/10.47738/jads.v5i4>.
- [17] Al Saaidah, A., Abualhaj, M. M., Shambour, Q. Y., Abu-Shareha, A. A., Abualigah, L., Al-Khatib, S. N., and Alraba'nah, Y. H., "Enhancing malware detection performance: leveraging K-nearest neighbors with firefly optimization algorithm," *Multimedia Tools and Applications*, 2025.
- [18] J. Li et al., "Feature Selection: A Data Perspective," *arXiv (Cornell University)*, vol. 50, no. 6, p. 94, Dec. 2017.
- [19] Abualhaj, M. M., Abu-Shareha, A. A., Alkhatib, S. N., Shambour, Q. Y., and Alsaaidah, A. M., "Detecting spam using Harris Hawks optimizer as a feature selection algorithm," *Bulletin of Electrical Engineering and Informatics*, vol. 14, no. 3, pp. 2361–2369, 2025.
- [20] P. Agrawal, H. F. Abutarboush, T. Ganesh, and A. W. Mohamed, "Metaheuristic Algorithms on feature selection: A survey of one decade of research (2009–2019)," *IEEE Access*, pp. 1–1, 2021, doi: <https://doi.org/10.1109/access.2021.3056407>.
- [21] M. Sh. Daoud, M. M. Abualhaj, H. Al-Mimi, and H. Al-Aqrabi, "Stagnation-Aware Phomone Reshuffling for Ant Colony-Based Neural Network Topology Search in System Identification," *Computers, Materials & Continua*, vol. 89, no. 2, 2026.
- [22] H. Sharma, G. Hazrati, and J. C. Bansal, "Spider monkey optimization algorithm," in *Evolutionary and Swarm Intelligence Algorithms*, pp. 43–59, 2019.
- [23] D. Sarma, T. Mittra, R. M. Bawm, T. Sarwar, F. F. Lima, and S. Hossain, "Comparative analysis of machine learning algorithms for phishing website detection," in *Inventive Computation and Information Technologies: Proceedings of ICICIT 2020, Singapore, 2021*.
- [24] M. Al-Sarem et al., "An Optimized Stacking Ensemble Model for Phishing Websites Detection," *Electronics*, vol. 10, no. 11, p. 1285, Jan. 2021, doi: <https://doi.org/10.3390/electronics10111285>.
- [25] Ramprasath Jayaprakash et al., "Heuristic machine learning approaches for identifying phishing threats across web and email platforms," *Frontiers in Artificial Intelligence*, vol. 7, Oct. 2024, doi: <https://doi.org/10.3389/frai.2024.1414122>.
- [26] A. Newaz, F. S. Haq, and N. Ahmed, "A Sophisticated Framework for the Accurate Detection of Phishing Websites," *arXiv (Cornell University)*, Mar. 2024, doi: <https://doi.org/10.48550/arxiv.2403.09735>.
- [27] M. C. Calzarossa, Paolo, and Rasha Zieni, "Explaining Explainable Ai, with Applications to Phishing Detection," Jan. 2024, doi: <https://doi.org/10.2139/ssrn.4790045>.
- [28] K. L. Chiew, C. L. Tan, K. Wong, K. S. C. Yong, and W. K. Tiong, "A new hybrid ensemble feature selection framework for machine learning-based phishing detection system," *Information Sciences*, vol. 484, pp. 153–166, May 2019, doi: <https://doi.org/10.1016/j.ins.2019.01.064>.
- [29] Isaac, A., Nehemiah, H. K., Dunston, S. D., Christo, V. E., and Kannan, A., "Feature selection using competitive coevolution of bio-inspired algorithms for the diagnosis of pulmonary emphysema," *Biomedical Signal Processing and Control*, vol. 72, 2022.
- [30] A. Alzaqebah, I. Aljarah, O. Al-Kadi, and R. Damaševičius, "A Modified Grey Wolf Optimization Algorithm for an Intrusion Detection System," *Mathematics*, vol. 10, no. 6, p. 999, Mar. 2022, doi: <https://doi.org/10.3390/math10060999>.
- [31] M. S. Viana, O. Morandin Junior, and R. C. Contreras, "A Modified Genetic Algorithm with Local Search Strategies and Multi-Crossover Operator for Job Shop Scheduling Problem," *Sensors*, vol. 20, no. 18, p. 5440, Sep. 2020, doi: <https://doi.org/10.3390/s20185440>.
- [32] Z. Xu, X. Huang, X. Zheng, J.-Y. Deng, and B. Sun, "A Performance and Data-Driven Method for Optimization of Traditional Courtyards," *Sustainability*, vol. 16, no. 13, p. 5779, Jul. 2024, doi: <https://doi.org/10.3390/su1613577>.
- [33] A. Jain, "Everything About Random Forest," *Medium*, Aug. 13, 2023. Available: <https://medium.com/@abhishekjainindore24/everything-about-random-forest-90c106d63989>.
- [34] Y. Wang and T. Wang, "Application of Improved LightGBM Model in Blood Glucose Prediction," *Applied Sciences*, vol. 10, no. 9, p. 3227, May 2020, doi: <https://doi.org/10.3390/app10093227>.