ISSN 2723-6471

Spam Feature Selection Using Firefly Metaheuristic Algorithm

Mosleh M. Abualhaj[1,*], Mohammad O. Hiari[2], Adeeb Alsaaidah[3],
Mahran Al-Zyoud[4], Sumaya Al-Khatib[5]

[1,2,3,4,5] *Faculty of Information Technology, Al-Ahliyya Amman University, Amman 19111, Jordan*

(Received: July 11, 2024; Revised: August 17, 2024; Accepted: September 13, 2024; Available online: October 15, 2024)

Abstract

This paper presents a novel method for improving spam detection by utilizing the Firefly Algorithm (FA) for feature selection. The FA, a bio-inspired metaheuristic optimization algorithm, is applied to identify the most relevant features from the ISCX-URL2016 dataset, which contains 72 features. By balancing exploration (searching for new solutions) and exploitation (focusing on the best solutions), FA is able to effectively reduce the feature space from 72 to 31 features. This reduction improves model efficiency without sacrificing performance, as only the most impactful features are retained for the classification task. The selected features were then used to train three machine learning classifiers: Decision Tree (DT), Gradient Boost Tree (GBT), and Naive Bayes (NB). Each classifier's performance was evaluated based on accuracy, with DT achieving the highest accuracy of 99.81%, GBT achieving 99.70%, and NB scoring 90.33%. The superior performance of the DT algorithm is attributed to its ability to handle non-linear relationships and high-dimensional data, making it particularly well-suited for the FA-selected features. This combination of FA for feature selection and DT for classification demonstrates significant improvements in spam detection performance, highlighting the importance of selecting the most relevant features. The results show that by reducing the dimensionality of the dataset, the FA algorithm not only accelerates the classification process but also enhances detection accuracy.

Keywords: Spam, Machine Learning, Feature Selection, Firefly Algorithm

1. Introduction

E-mail is an example of the many useful services that have been made available to the world as a result of advancements in digital world [1]. The use of e-mail is an essential component of everyday life online. Worldwide, the daily volume of e-mails sent and received in 2020 amounted to roughly 306 billion [2]. Many people use e-mails to sign up for websites and newsletters, and they should be prepared for the inevitable torrent of spam that will come their way. Users of a variety of platforms are dealing with significant challenges as a result of spam, which is a significant concern. As a result of the proliferation of e-mail services, spam has become an ideal attack tool that hackers can use to deliver false materials, such as advertisements, which can lead to attacks that are both serious and damaging [3], [4].

Filtering spam in email can be accomplished by the use of machine learning (ML) algorithms and a wide range of software that has been developed as an anti-spam solution using these algorithms. ML is the study of utilizing computers to replicate human learning activities. It is the process by which computers acquire new knowledge, recognize current knowledge, and continuously improve their performance capabilities [5], [6], [7]. In many cases, the two most important considerations related to ML are speed and accuracy. In order to train a successful ML model, high-dimensional data is necessary, which increases the amount of time required. One of the most straightforward approaches to reducing dimensionality is through the use of feature selection [8], [9]. In this method, one chooses spam features that contain the information necessary to identify spam. The Firefly algorithm (FA) is a metaheuristic algorithm that is widely utilized for feature selection. The process of selecting the spam features that most significantly contribute to improving the learning accuracy of the ML algorithms will be carried out with the help of FA in this work. Subsequently, the feature that has been selected will undergo evaluation by employing decision tree (DT), gradient boost tree (GBT), and Naive Bayes (NB) ML algorithms [8], [9], [10].

***Corresponding author: Mosleh M. Abualhaj (m.abualhaj@ammanu.edu.jo)**
DOI: https://doi.org/10.47738/jads.v5i4.336

ISSN 2723-6471

2. Literature Review

Xu et al. [11] propose a spam filtering framework that is built from an automatic thesaurus creation system and redesigned back propagation neural network (BP-NN) components. The typical BP-NN has a modest learning velocity and tends to become stuck at a local minimum, resulting in low performance and proficiency. Experiments have demonstrated that the proposed RBP-NN framework outperforms the typical BP-NN framework in terms of performance.

The hybrid model that I. Idris et al. [12] suggest is a combination of a differential evolution (DE) algorithm and a negative selection algorithm (NSA). The implementation of DE improved the detector generation step of the NSA. Simultaneously, the local outlier factor (LOF) was utilized as a fitness function in order to optimize the distance between the detectors that were formed. In order to resolve the issue of overlap between two detectors, a fitness function was used to calculate the distance between the detectors that came into mutual overlap. Based on the findings, the hybrid model that was suggested for spam detection has an accuracy of 83.06%.

By employing a stochastic distribution to describe the data point through the utilization of particle swarm optimization (PSO), I. Idris et al. and A. Selamat [13] have suggested a unique model that enhances the random generation of a detector in the NSA algorithm. The LOF function is presented as the fitness function in order to ascertain the local best of the candidate detector that provides the best possible solution. For evaluation purposes, the Spambase dataset, which is located in the UCI ML repository, is utilized. The suggested model attained a 91.22% level of accuracy.

K. Debnath and N. Kar [14] construct email spam detection models utilizing machine learning and deep learning methodologies to differentiate spam emails from authentic ones reliably. Researchers have used the Enron email dataset to create advanced deep learning models, including LSTM and BERT, to identify and categorize new instances of email spam. The NLP methodology assessed and prepared data for the email's text. The results are compared to the prior models in email spam detection. The deep learning strategy achieved the highest accuracy rates of 99.14% with BERT, 98.34% with BiLSTM, and 97.15% with LSTM. Python is used for all implementations.

A. Wijaya and A. Bisri [15] suggest merging Logistic Regression (LR) and DT algorithms to detect email spam. LR filters out noisy data or instances before inputting them into DT induction. LR uses false negative thresholds to filter correct predictions and decrease noisy data. This study evaluates the proposed approach using the Spambase dataset. The experiment demonstrates that the suggested approach produces remarkable and encouraging outcomes, with an accuracy rate of 91.67%. LR can enhance DT performance by mitigating the impact of noisy data.

3. Method

3.1. ISCX-URL2016 dataset

The spam samples of the ISCX-URL2016 dataset were used in the evaluation process. The ISCX-URL2016 dataset is widely used as a benchmark for evaluating spam detection systems. It contains a reasonable number of spam and benign samples, as well as diverse features that thoroughly represent spam emails. Therefore, the ISCX-URL2016 dataset is reliable and comprehensive for evaluating the employed feature selection and classification approach in detecting spam. After removing the other attack samples from the ISCX-URL2016 dataset, the number of remaining samples is 14479, and the number of remaining features is 72. The samples are distributed in two types: benign and spam samples. The number of spam samples is 6,698, while the number of benign samples is 7,780. The number of samples is balanced, and therefore, there is no need to implement oversampling or undersampling on the dataset. However, many of the 72 features could have very little impact in identifying the sample as spam or benign. Accordingly, the FA algorithm will be applied to the dataset to select only the features that have a high impact on detecting spam [16], [17], as will discussed in the following section (Section 3.2). However, table 1 shows that the values of the features scale over a large space, which will impact the spam detection operation. Normalization is an ML operation that handles large spaces of values using certain mechanisms, such as Min-max normalization. As a normalization mechanism, the Min-max produces balanced value comparisons between the pre- and post-process data by performing linear transformations on the original data [7], [18]. Table 2 shows sample of the ISCX-URL2016 spam dataset before and after using the Min-max mechanism.

 ISSN 2723-6471

Table 1. Sample of the ISCX-URL2016 Spam dataset

#	Feature	Min Value	Max Value
1	LongestPathTokenLength	0	1393
2	LongestVariableValue	-1	1385
3	URL_Letter_Count	15	1202
4	Extension_LetterCount	-1	1179
5	Query_LetterCount	-1	1173
6	LongestPathTokenLength	0	1393

Table 2. Spam dataset Sample of the ISCX-URL2016 Spam dataset before and after normalization

#	Before Normalization	After Normalization
1	0 ,2 ,4.5 ,2	0 ,0 ,0.227272727 ,0
2	17 ,2 ,6 ,2	0.012274368 ,0 ,0.363636364 ,0
3	0 ,2 ,4.5 ,2	0 ,0 ,0.227272727 ,0
4	0 ,2 ,7 ,2	0 ,0 ,0.454545455 ,0
5	0 ,2 ,8 ,2	0 ,0 ,0.545454545 ,0
6	0 ,2 ,6.5 ,2	0 ,0 ,0.409090909 ,0
7	0 ,3 ,2.6666667 ,3	0 ,0.333333333 ,0.060606064 ,0.333333333
8	0 ,2 ,4.5 ,2	0 ,0 ,0.227272727 ,0
9	0 ,2 ,3.5 ,2	0 ,0 ,0.136363636 ,0
10	0 ,2 ,5 ,2	0 ,0 ,0.272727273 ,0
11	0 ,3 ,3 ,3	0 ,0.333333333 ,0.090909091 ,0.333333333

3.2. Feature Selection Using FA Algorithm

In the realm of metaheuristic swarm optimization, the FA is a robust algorithm that draws inspiration from the natural behavior of fireflies. The phenomenon of bioluminescence serves as the foundation for the natural behavior of fireflies. They do this in order to communicate with other fireflies and to attract possible prey by producing flashes that are both brief and rhythmic. Figure 1 illustrates the nature of the behavior of fireflies. In order to make it possible to create an optimization algorithm, the flashing light of fireflies can be expressed in such a way that it is related to the objective function that needs to be optimized [8], [19], [20].

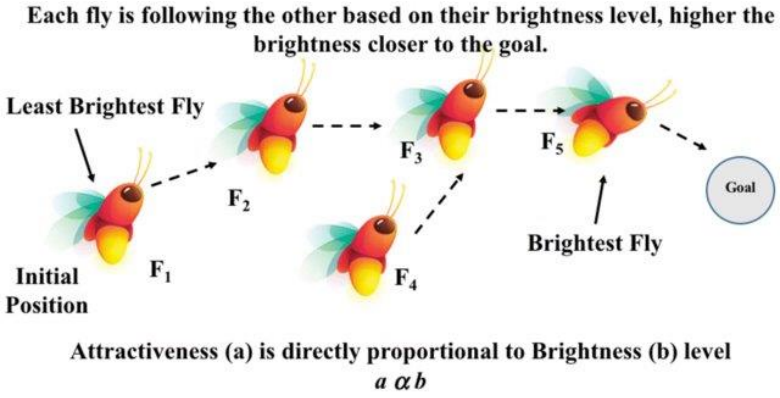

Figure 1. Behavior of fireflies [20]

The FA algorithm is applied to the ISCX-URL2016 dataset to select only the features that greatly impact spam detection. The algorithm iteratively adjusts the feature subset to select the best features that improve classification accuracy. In addition, the FA algorithm considers the interaction between the selected features. This ensures that the chosen features are complementary and non-redundant representations of the spam data. The FA was able to reduce the features from 72 to 31 after identifying the features that provided an optimal balance between model complexity and performance. These features are Querylength, domain_token_count, avgdomaintokenlen, tld, charcompaceldl_domain, ldl_filename, ldl_getArg, domainlength, subDirLenfileNameLen, pathurlRatio, domainUrlRatio, Querylength, isPortEighty, CharacterContinuityRate, host_DigitCount, Directory_DigitCount, Extension_DigitCount, Query_DigitCount, host_letter_count, Filename_LetterCount, Extension_LetterCount, LongestPathTokenLength, Path_LongestWordLength, Arguments_LongestWordLength, delimeter_Count, SymbolCount_FileName, SymbolCount_Extension, Entropy_URL, Entropy_Domain [16], [17].

3.2. Classification

The DT, GBT, and NB ML algorithms were used for the classification task. The DT algorithm is a versatile and interpretable model that splits data into subsets based on feature values, forming a tree structure where each node represents a decision rule. It is easy to visualize and understand, making it useful for both classification task. GBT is an ensemble learning method that builds multiple DTs sequentially, where each tree corrects the errors of its predecessor. By combining the predictions of many weak learners, GBT creates a strong predictive model, offering high accuracy and robustness. NB is a simple, probabilistic classifier based on Bayes' theorem, assuming independence between features. Despite its simplicity, it is highly effective for large datasets and text classification problems, such as spam detection, due to its fast training and prediction times [8], [9], [10].

Several hyperparameters control the performance of these machine learning algorithms. Constructing an effective DT, GBT, and NB algorithms for spam detection presents a significant challenge, primarily revolving around the selection of hyperparameters. Optimal hyperparameter choices are crucial in order to strike the right balance, preventing overfitting while achieving the best results in spam detection. It's worth noting that many hyperparameters selection methods tend to generate highly complex models, which, if not properly controlled, can lead to overfitting. Overfitted hyperparameters not only perform poorly on new, unseen data but can also be challenging to interpret. This interpretability issue poses a significant barrier to the practical application of these models including spam detection. The importance of hyperparameters selection cannot be overstated, as it plays a pivotal role in averting overfitting and enhancing the effectiveness of spam detection [8], [9], [10].

The Random Search (RS) mechanism was used to choose the hyperparameters for the DT, GBT, and NB ML algorithms. Several mechanisms are available to choose the hyperparameters of ML classifiers. RS was used because it is simple and efficient. In addition, it explores large hyperparameter spaces with fewer iterations and is less computationally intensive [21], [22], [23]. The RS is a mechanism that chooses the hyperparameters of an ML classifier that achieve the best performance for that classifier. The RS mechanism defines the space for each hyperparameter of the classifier. Then, random samples from the combination of the defined hyperparameter spaces are chosen and evaluated. The combination that achieves the best performance is selected as the optimal set of hyperparameters for the classifier [24], [25]. Table 3 summarizes the hyperparameters for the DT, GBT, and NB algorithms based on the RS mechanism.

Table 3. Hyperparameters of the DT algorithm

Algorithm	Hyperparameter	Value	Description
DT	criterion	entropy	Measures the quality of a split
	splitter	best	Chooses the split at each node
	max_depth	20	Maximum depth of the tree
	min_samples_split	2	Minimum number of samples to split an internal node
	min_samples_leaf	1	Minimum number of samples at a leaf node

	n_estimators	90	Number of boosting stages to be run
	learning_rate	0.2	Shrinks the contribution of each tree
GBT	max_depth	4	Maximum depth of the individual trees
	min_samples_split	3	Minimum number of samples required to split an internal node
	min_samples_leaf	1	Minimum number of samples required to be at a leaf node
	alpha	1.0	Additive smoothing parameter
NB	fit_prior	True	Whether to learn class prior probabilities.
	class_prior	None	Prior probabilities of the classes

4. Results and Discussion

The DT, GBT, and NB ML algorithms were used for the classification task. The FA algorithm uses the confusion matrix (figure 2) to evaluate the performance of the selected features with the DT, GBT, and NB algorithms. Base of the confusion matrix, FA used the accuracy (1), recall (2), and precision (3), Matthews Correlation Coefficient (MCC) (4), F1-Score (5) metrics to assess the selected features with DT, GBT, and NB algorithms. Accuracy, recall, precision, MCC, and F1-score were chosen to provide a comprehensive evaluation of the model's performance. Accuracy assesses overall correctness, while recall and precision are critical for imbalanced datasets. MCC and F1-score offer balanced insights into model quality, aligning with the study's goal of a thorough performance assessment [10], [26].

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

Figure 2. Confusion matrix

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

$$\text{Recall} = \frac{\text{TP}}{(\text{TP} + \text{FN})} \quad (2)$$

$$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})} \quad (3)$$

$$\text{MCC} = \frac{((\text{TP} * \text{TN}) - (\text{FP} * \text{FN}))}{\sqrt{(\text{TP} + \text{FP}) * (\text{TP} + \text{FN}) * (\text{TN} + \text{FP}) * (\text{TN} + \text{FN})}} \quad (4)$$

$$\text{F1} - \text{score} = 2 \times \frac{\text{Pre} \times \text{Rec}}{\text{Pre} + \text{Rec}} \quad (5)$$

Figure 3 shows the accuracy of the selected features by the FA algorithm. As we can see, with the DT, GBT, and NB algorithms, the FA algorithm achieves accuracy above 90%. However, the DT algorithm outperforms the GBT and NB algorithms. Figure 4 shows the recall of the selected features by the FA algorithm. As we can see, with the DT, GBT, and NB algorithms, the FA algorithm achieves recall above 90%. However, again, the DT algorithm outperforms the GBT and NB algorithms. Figure 5 shows the precision of the selected features by the FA algorithm. Similar to accuracy and recall, with the DT, GBT, and NB algorithms, the FA algorithm achieves precision above 90%. However, again, the DT algorithm outperforms the GBT and NB algorithms. Figure 6 shows the MCC of the selected features by the FA algorithm. The DT algorithm achieved the highest MCC of 99.59%, while the GBT and NB algorithms achieved 99.40% and 80.11%, respectively. Finally, figure 7 shows the F1-Score of the selected features by the FA algorithm.

ISSN 2723-6471

As we can see, with the DT, GBT, and NB algorithms, the FA algorithm achieves F1-Score above 90%. However, again, the DT algorithm outperforms the GBT and NB algorithms. The DT outperforms GBT and NB because it effectively captures complex feature interactions, handles mixed data types, and resists overfitting in the ISCX-URL2016 dataset. GBT's sensitivity to hyperparameter tuning and NB's feature independence assumption limit their performance in this context. DT's robustness and alignment with the dataset's characteristics make it the best-performing algorithm in this study.

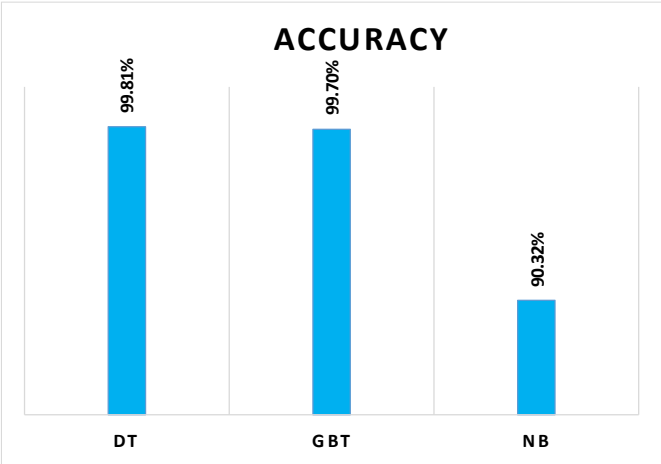

Figure 3. Accuracy of the FA algorithm

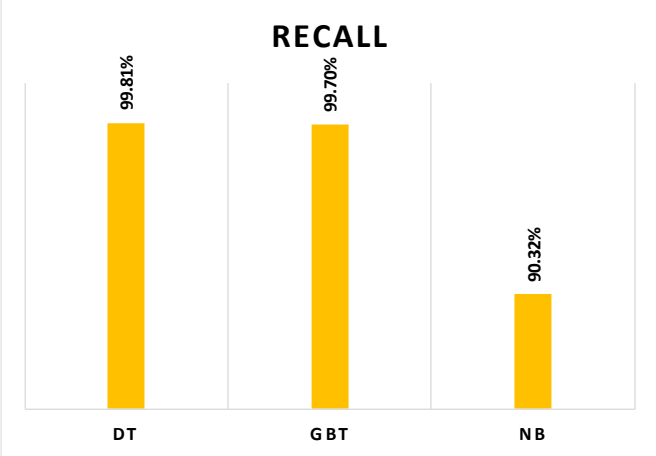

Figure 4. Recall of the FA algorithm

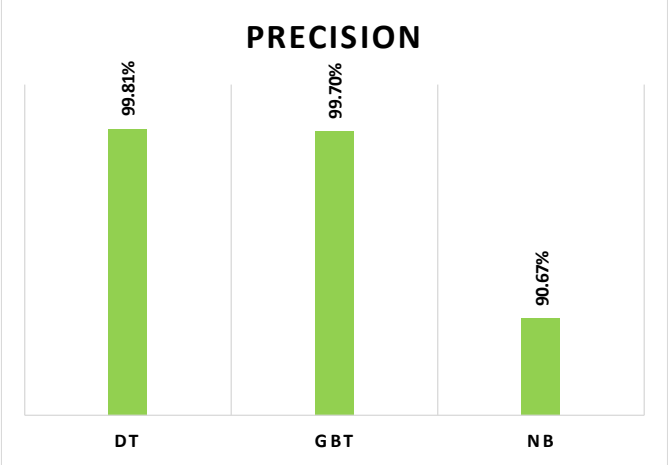

Figure 5. Precision of the FA algorithm

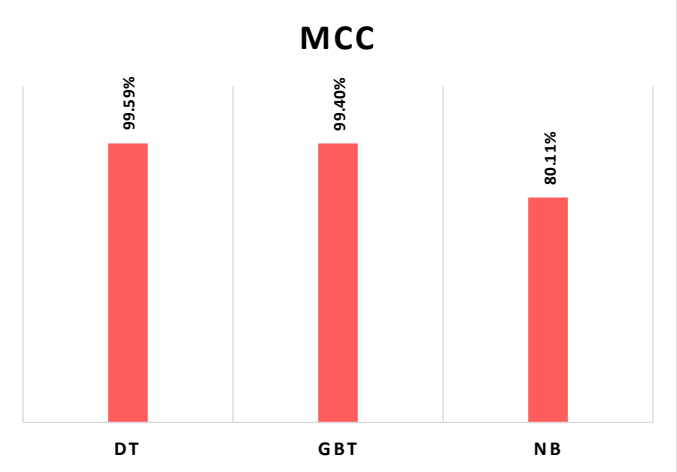

Figure 6. MCC of the FA algorithm

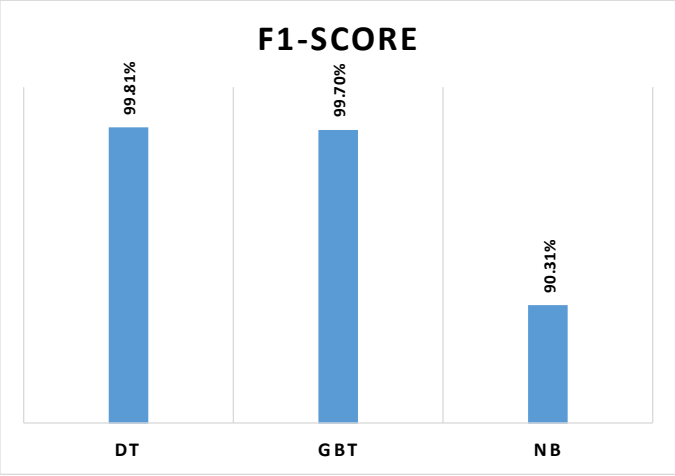

Figure 7. F1-Score of the FA algorithm

5. Conclusion

Spam spreads malicious links through emails, which causes a serious problem. The detection of these malicious links as quickly and accurately as possible is important. In this research, the FA algorithms were used to select the key features that help to detect spam. We tested the new spam sub-dataset that contains the selected features by FA with DT, GBT, and NB algorithms and optimized their hyperparameters using a random search mechanism. The Python implementation showed that the FA with DT classifier achieved the best accuracy at 99.79%, outperforming the FA with GBT and NB. The findings highlight the importance of using ML algorithms to improve spam detection. This combination of FA and DT algorithms can help make email systems reliable by effectively identifying and stopping spam. However, to ensure the reliability and generalization of the results, the combination of DT and FA algorithms should be evaluated over other datasets. In addition, other metaheuristic algorithms will be evaluated for feature selection for spam detection, and the results will be compared with those obtained by the FA algorithm.

6. Declarations

6.1. Author Contributions

Conceptualization: M.M.A., M.O.H., A.A., M.A.Z., and S.A.K.; Methodology: M.O.H. and A.A.; Software: M.M.A.; Validation: M.M.A., M.O.H., A.A., M.A.Z., and S.A.K.; Formal Analysis: M.M.A., M.O.H., A.A., M.A.Z., and S.A.K.; Investigation: M.M.A.; Resources: S.A.K. and M.A.Z.; Data Curation: S.A.K.; Writing Original Draft Preparation: M.M.A., M.O.H., A.A., M.A.Z., and S.A.K.; Writing Review and Editing: M.M.A., M.O.H., A.A., M.A.Z., and S.A.K.; Visualization: M.M.A., M.O.H., A.A., M.A.Z., and S.A.K.; All authors have read and agreed to the published version of the manuscript.

6.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

6.3. Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

6.4. Institutional Review Board Statement

Not applicable.

6.5. Informed Consent Statement

Not applicable.

6.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] J. Wei, X. Chen, J. Wang, X. Hu and J. Ma, "Enabling (End-to-End) Encrypted Cloud Emails With Practical Forward Secrecy," *in IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 4, pp. 2318-2332, 1 July-Aug. 2022, doi: 10.1109/TDSC.2021.3055495.

[2] M. M. Abualhaj, Q. Shambour, A. Alsaaidah, A. A. Abu-Shareha, S. N. Al-Khatib, and M. Hiari, "Enhancing Spam Detection Using Hybrid of Harris Hawks and Firefly Optimization Algorithms," *Journal of Applied Data Sciences*, vol. 5, no. 3, pp. 901–911, 2024, doi: 10.47738/jads.v5i3.279.

[3] M. Kolhar, F. Al-Turjman, A. Alameen and M. M. Abualhaj, "A Three Layered Decentralized IoT Biometric Architecture for City Lockdown During COVID-19 Outbreak," *in IEEE Access*, vol. 8, no. 1, pp. 163608-163617, 2020, doi: 10.1109/ACCESS.2020.3021983.

[4] G. Kambourakis, G. D. Gil and I. Sanchez, "What Email Servers Can Tell to Johnny: An Empirical Study of Provider-to-Provider Email Security," *in IEEE Access*, vol. 8, no. 1, pp. 130066-130081, 2020, doi: 10.1109/ACCESS.2020.3009122.

[5] H. Al-Mimi, N. A. Hamad, and M. M. Abualhaj, "A Model for the Disclosure of Probe Attacks Based on the Utilization of Machine Learning Algorithms," *2023 10th International Conference on Electrical and Electronics Engineering (ICEEE)*, vol. 10, no. May, pp. 241-247, May 2023, doi: 10.1109/iceee59925.2023.00051.

[6] A. AlMahmoud, E. Damiani, H. Otrok and Y. Al-Hammadi, "Spamdoop: A Privacy-Preserving Big Data Platform for Collaborative Spam Detection," *in IEEE Transactions on Big Data*, vol. 5, no. 3, pp. 293-304, 1 Sept. 2019, doi: 10.1109/TBDATA.2017.2716409.

[7] M. M. Abualhaj, Ahmad Adel Abu-Shareha, Qusai Shambour, Adeeb Alsaaidah, S. N. Al-Khatib, and M. Anbar, "Customized K-nearest neighbors' algorithm for malware detection," *International journal of data and network science*, vol. 8, no. 1, pp. 431–438, Jan. 2024, doi: 10.5267/j.ijdns.2023.9.012.

[8] A. Alsaaidah, M. M. Abualhaj, Q. Y. Shambour, et al., "Enhancing malware detection performance: leveraging K-Nearest Neighbors with Firefly Optimization Algorithm," *Multimedia Tools and Applications*, vol. 1, no. 1, pp. 1-21, 2024, doi: 10.1007/s11042-024-18914-5.

[9] K. Dev, P. K. R. Maddikunta, T. R. Gadekallu, S. Bhattacharya, P. Hegde and S. Singh, "Energy Optimization for Green Communication in IoT Using Harris Hawks Optimization," *in IEEE Transactions on Green Communications and Networking*, vol. 6, no. 2, pp. 685-694, June 2022, doi: 10.1109/TGCN.2022.3143991.

[10] M. M. Abualhaj, Ahmad Adel Abu-Shareha, M. O. Hiari, Yousef Alrabanah, Mahran Al-Zyoud, and M. A. Alsharaiah, "A Paradigm for DoS Attack Disclosure using Machine Learning Techniques," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 3, pp. 192-200, Jan. 2022, doi: 10.14569/ijacsa.2022.0130325.

[11] H. Xu and B. Yu, "Automatic thesaurus construction for spam filtering using revised back propagation neural network," *Expert Syst. Appl.*, vol. 37, no. 1, pp. 18–23, Jan. 2010.

[12] I. Idris, A. Selamat, and S. Omatu, "Hybrid email spam detection model with negative selection algorithm and differential evolution," *Eng. Appl. Artif. Intell.*, vol. 28, no. 1, pp. 97–110, Feb. 2014.

[13] I. Idris and A. Selamat, "Improved email spam detection model with negative selection algorithm and particle swarm optimization," *Appl. Soft Comput.*, vol. 22, no. 13, pp. 11–27, 2014.

[14] K. Debnath and N. Kar, "Email Spam Detection using Deep Learning Approach," *2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON), Faridabad, India*, vol. 1, no. aug., pp. 37-41, 2022, doi: 10.1109/COM-IT-CON54601.2022.9850588.

[15] A. Wijaya and A. Bisri, "Hybrid decision tree and logistic regression classifier for email spam detection," *2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE), Yogyakarta, Indonesia*, vol. 8, no. oct., pp. 1-4, 2016, doi: 10.1109/ICITEED.2016.7863267.

[16] R. Aloufi and A. R. Alharbi, "K-means and Principal Components Analysis Approach For Clustering Malicious URLs," *2023 3rd International Conference on Computing and Information Technology (ICCIT), Tabuk, Saudi Arabia*, vol. 3, no. sep., pp. 359-364, 2023, doi: 10.1109/ICCIT58132.2023.10273923.

[17] M. S. I. Mamun, M. A. Rathore, A. H. Lashkari, N. Stakhanova, and A. A. Ghorbani, "Detecting malicious urls using lexical analysis." *In Network and System Security: 10th International Conference, NSS 2016, Taipei, Taiwan, September 28-30, 2016, Proceedings 10*, vol. 10, no. sep., pp. 467-482, 2016. DOI: 10.1007/978-3-319-46298-1_30.

[18] A. Hussein and Qusai Shambour, "A Trust-enhanced Recommender System for Patient-Doctor Matchmaking in Online Healthcare Communities," *International journal of intelligent engineering and systems*, vol. 16, no. 6, pp. 684–694, Dec. 2023, doi: 10.22266/ijies2023.1231.57.

[19] F. Liu, "Nonconvex Compressed Sensing by Nature-Inspired Optimization Algorithms," *in IEEE Transactions on Cybernetics*, vol. 45, no. 5, pp. 1042-1053, May 2015, doi: 10.1109/TCYB.2014.2343618.

[20] Q. Y. Shambour, M. M. A. Alhaj, and M. M. A. Tahrawi, "A hybrid collaborative filtering recommendation algorithm for requirements elicitation," *International Journal of Computer Applications in Technology*, vol. 63, no. 1/2, pp. 135-142, 2020, doi: 10.1504/ijcat.2020.107908.

[21] R. Turner, D. Eriksson, M. McCourt, J. Kiili, E. Laaksonen, Z. Xu, and I. Guyon, "Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020," *in NeurIPS 2020 Competition and Demonstration Track*, vol. 1, no. aug., pp. 3-26, Aug. 2021.

ISSN 2723-6471

[22] H. M. Al-Mimi, N. A. Hamad, M. M. Abualhaj, S. N. Al-Khatib, and M. O. Hiari, "Improved Intrusion Detection System to Alleviate Attacks on DNS Service," *Journal of Computer Science*, vol. 19, no. 12, pp. 1549-1560, Nov. 2023.

[23] M. M. Abualhaj and S. N. Al-Khatib, "Using decision tree classifier to detect Trojan Horse based on memory data," *Telecom*, vol. 22, no. 2, pp. 393–393, Apr. 2024, doi: 10.12928/telkomnika.v22i2.25753.

[24] Monica and P. Agrawal, "A Survey on Hyperparameter Optimization of Machine Learning Models," *2024 2nd International Conference on Disruptive Technologies (ICDT), Greater Noida, India*, vol. 2, no. mar., pp. 11-15, 2024, doi: 10.1109/ICDT61202.2024.10489732.

[25] H. Yi and K. -H. N. Bui, "An Automated Hyperparameter Search-Based Deep Learning Model for Highway Traffic Prediction," *in IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 9, pp. 5486-5495, Sept. 2021, doi: 10.1109/TITS.2020.2987614.

[26] Q. Y. Shambour, M. M. Al-Zyoud, A. H. Hussein, and Q. M. Kharma, "A doctor recommender system based on collaborative and content filtering," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 1, pp. 884-893, Feb. 2023, doi: 10.11591/ijece.v13i1.pp884-893.