Enhancing spam detection using Harris Hawks optimization algorithm

Mosleh M. Abualhaj[1], Sumaya Nabil Alkhatib[1], Ahmad Adel Abu-Shareha[1], Adeeb M. Alsaaidah[1], Mohammed Anbar[2]
[1]Department of Nettworks and Cybersecurity, Faculty of Information Technology, Al-Ahliyya Amman University, Amman, Jordan
[2]National Advanced IPv6 Center (NAv6), Universiti Sains Malaysia (USM), Penang, Malaysia

Article Info

Article history:

Received Aug 25, 2024
Revised Jan 14, 2025
Accepted Jan 23, 2025

Keywords:

Feature selection
Harris Hawks algorithm
ISCX-URL2016 dataset
Machine learning
Spam

ABSTRACT

This paper employs machine learning (ML) algorithms to identify and classify spam emails. The Harris Hawks optimization (HHO) algorithm can detect the crucial features that distinguish spam from ham emails. The HHO algorithm decreased the number of features in the ISCX-URL2016 spam dataset from 72 to 10. Implementing this will enhance the efficiency and cognitive acquisition of the ML algorithms. The decision tree (DT), Naive Bayes (NB), and AdaBoost algorithms are evaluated and contrasted to identify spam emails. The random search algorithm is used to optimize the significant hyperparameters of each algorithm for the specific task of spam identification. All three ML algorithms showed exceptional accuracy in detecting spam emails during the conducted testing. The DT algorithm attained a remarkable accuracy rate of 99.75%. The AdaBoost algorithm ranks second with an incredible accuracy of 99.67%. Finally, the NB algorithm attained an accuracy of 96.30%. The results demonstrate that the HHO algorithm shows promise in recognizing the crucial features of spam emails.

Corresponding Author:

Mosleh M. Abualhaj
Department of Nettworks and Cybersecurity, Faculty of Information Technology
Al-Ahliyya Amman University
Amman 19111, Jordan
Email: m.abualhaj@ammanu.edu.jo

1. INTRODUCTION

Email is a widely used and effective means of online communication and sharing data or messages [1]. Due to email's importance and widespread use, spam emails have grown dramatically. Spam emails are unsolicited emails that contain a range of contents, such as advertisements, unsafe hyperlinks, Trojans, and malware [2], [3]. Email spam wastes time, consumes server storage, and obstructs communication channels until the recipient takes the necessary action. Eliminating spam emails may accidentally delete a vital one [2], [3]. Spammers send unsolicited emails to perpetrate email fraud; therefore, it is crucial to separate spam emails from legitimate ones using email filters. The current spam filters use machine learning (ML) techniques to counter spam emails [4], [5].

ML techniques can learn from the available spam training data to predict future spam emails. Supervised learning is a type of ML used with labelled data such as spam emails [6]. Several supervised learning techniques can be used to classify spam from ham emails, including decision tree (DT), Naive Bayes (NB), and AdaBoost [6], [7]. The supervised learning techniques use spam and ham email data to build a supervised learning model. The model contains several rules that distinguish spam and ham based on the email features. The more accurate the email features are, the better the performance of the supervised learning model will be [4], [6], [8]. The email data contains a significant number of features. Some features are crucial to distinguishing spam and ham, while others are less important or irrelevant. ML can use so-called feature selection techniques to find the key features that can be used to classify incoming emails (spam or ham) [9], [10].

The feature selection process removes redundant and irrelevant features from the email data to improve the predictive accuracy of spam filter software. Several feature selection algorithms have been used with email data to find its key features. In recent years, metaheuristic algorithms have been widely used to solve optimization problems, including finding the optimal subset of features from extensive email data [9], [11]. In this paper, the Harris Hawks optimization (HHO) algorithm will be used to find the key features that can be used to distinguish spam from ham emails [9], [12]. The widely known ISCX-URL2016 spam dataset will be used as a benchmark for email data [13]. The efficiency of the resulting subset will be tested using three common supervised learning techniques, namely DT, NB, and AdaBoost. In addition, the hyperparameters of these three techniques will be tailored to achieve the best performance and suit the problem at hand.

Several works have been proposed for spam detection. Rathi and Pareek [14] investigate a variety of ML algorithms for the spam dataset, keeping in mind the ultimate objective of determining the most effective ML algorithm for spam email categorization. Researchers evaluated the effectiveness of several different algorithms, both with and without the use of feature selection algorithms. The initial phase of the process was testing all algorithms on the entire dataset without selecting any features. After that, the Best-First feature selection technique was utilized to identify the desired features, and afterwards, several different classifiers were used. The findings demonstrated that applying the Best-First feature selection method increases accuracy. The random forest algorithm attained the highest accuracy of 99.72% among all the algorithms applied. In comparison, the NB algorithm has attained the lowest accuracy of 78.93%.

Ravi *et al.* [15] studied the detection of malicious uniform resource locators (URLs) by conducting a comparative examination of several different deep learning-based character-level embedding (DL-CLE) models. Concerning accuracy, every DL architecture possesses some degree of differentiation. On the other hand, the models that were put through their paces performed well and attained a detection rate of between 93 and 98% for malicious URLs, with a false positive rate of 0.001. This suggests that even if the DL-CLE models can identify 970 malicious URLs, they will only classify one non-malicious URL as malicious. The DL-CLE models performed better than the other models across all test cases. All DL-CLE models can deal with malicious URL drifting and provide a reliable solution in an unreliable environment.

Le *et al.* [16] suggest an end-to-end DL framework that they term URLNet. This framework aims to train nonlinear URL embedding and detect fraudulent URLs at the URL level. According to this method, the model can capture multiple sorts of semantic information, which was not achievable with the previously available models. In addition, advanced word embeddings are advocated as a potential solution to the issue of an excessive number of uncommon words being noted. Extensive experiments are carried out on a massive dataset, demonstrating a significant performance improvement over existing approaches. In addition, a component analysis study is carried out to assess the overall performance of the URLNet components.

2. METHOD

This section discusses the proposed method for detecting spam emails. First, the ISCX-URL2016 dataset will be presented. Then, the HHO Algorithm used for feature selection will be addressed. Finally, the algorithms used for the classification process will be elaborated on.

2.1. ISCX-URL2016 spam dataset

The data inside the ISCX-URL2016 dataset ought to be in a format suitable for ML algorithms. Data preprocessing is a crucial step involving transforming raw data into a refined dataset before inputting it into ML algorithms [17], [18]. The data format must be appropriate to achieve optimal outcomes from the ML algorithms utilized. For instance, the majority of ML algorithms cannot handle null values. Therefore, it is necessary to preprocess the ISCX-URL2016 dataset. The ISCX-URL2016 dataset contains numerous features with null values. These features have been excluded from the dataset, reducing the total number of features from 79 to 72. Second, as seen in Table 1, many of the dataset's features include values dispersed across a large range of values [17], [18]. These values have been condensed into relatively tight ranges by utilizing the Min-Max scaling normalizing approach [11], [19]. Table 2 presents samples of the ISCX-URL2016 spam dataset before and after the normalization process. Last but not least, the most significant features are selected (those providing the highest level of performance), and the remaining features are removed, as detailed in the following subsection.

Table 1. Sample of the ISCX-URL2016 spam dataset

#	Feature	Min value	Max value
1	Query_LetterCount	-1	1173
2	URL_Letter_Count	15	1202
3	LongestPathTokenLength	0	1393
4	pathLength	1	1402
5	Query_LetterCount	-1	1173
6	Querylength	0	1385

Table 2. Instances of the ISCX-URL2016 spam dataset before and after normalization

#	Before normalization	After normalization
1	5.5, 0, 2, 7, 2	0.318182, 0, 0, 0.049296, 0
2	5,0, 3, 8, 3	0.272727, 0, 0.333333, 0.056338, 0.333333
3	4, 2, 2, 11, 2	0.181818, 0.001444, 0, 0.077465, 0
4	4.5, 0, 2, 10, 2	0.227273, 0, 0, 0.070423, 0
5	6,19, 2, 5, 2	0.363636, 0.013718, 0, 0.035211, 0

2.2. Feature selection using Harris Hawks optimization algorithm

HHO algorithm is a population-based method that was recently introduced. This technique is derived from the cooperative behaviour exhibited by Harris Hawks birds. Figure 1 shows HHO hunting behavior. The HHO method thoroughly searches the design variable space during the exploration phase. This method is predominantly reliant on the interaction among several applicants. In addition, the HHO method involves conducting a localized search using the knowledge gained from the previous phase to enhance the quality of existing solutions. The HHO system has a phase conversion controller that utilizes an energy parameter to effectively and automatically optimize the balance between exploration and exploitation. HHO exhibits superior local search skills compared to existing algorithms due to its diverse improvement methodologies. Moreover, the HHO method outperforms other algorithms in numerical benchmark testing because it effectively balances exploiting and exploring, resulting in a high-quality solution and accelerated convergence rate [9], [12].

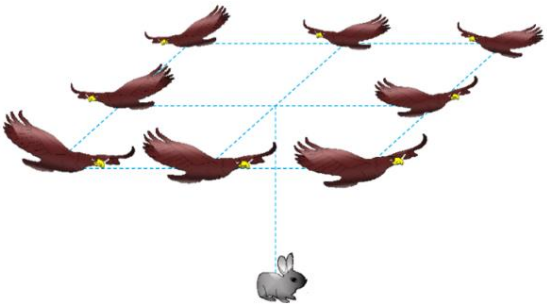

Figure 1. HHO hunting behavior [9]

The HHO technique selects only those features that significantly impact the detection of spam instances from the ISCX-URL2016 dataset. Out of 72, the HHO algorithm selected only 10 features. These features are tld, dld_getArg, pathurlRatio, ArgUrlRatio, pathDomainRatio, NumberofDotsinURL, CharacterContinuityRate, Filename_LetterCount, NumberRate_Domain, and NumberRate_DirectoryName.

2.3. Classification algorithms

2.3.1. Decision tree classification algorithm

DT is a tree-like model that uses attribute values to classify the provided datasets. A DT’s node displays an attribute along with any relevant edges. Every edge joins a node with a leaf or two nodes together. A decision value explains a leaf to identify the input data class. The primary problem using DT in spam detection is figuring out which attributes to use to divide the datasets efficiently. During the test phase, newly incoming data can be classified according to decision values using a DT built from pre-classified data from the training phase [20]. Several hyperparameters impact the DT algorithm’s performance. These hyperparameters have been tuned using the random search algorithm [21]. Table 3 shows the values of these hyperparameters.

Table 3. Hyperparameters of the DT algorithm

#	Hyperparameter	Description	Value
1	criterion	The quality of a split	gini
2	max_features	Chooses the split at each node	sqrt
3	max_depth	Maximum depth of the tree	8
4	min_samples_split	Minimum number of samples to split	6
5	min_samples_leaf	Minimum number of samples at a leaf node	4

2.3.2. Naive Bayes classification algorithm

An NB is a straightforward probabilistic algorithm that relies on the Bayes theorem's intense (naive) independence assumptions from Bayesian statistics. The NB algorithm has the benefit of requiring less training data to estimate the parameters needed for classification. In many real-world applications, it works surprisingly well, even though it makes the strong assumption that all features are conditionally independent given the class. With a known structure, this algorithm uses training data to construct class and conditional probabilities as part of its learning process. New observations are then classified based on the values of these probabilities. NB is appealing because it has a solid theoretical foundation that ensures optimal induction with clear assumptions [22]. Several hyperparameters impact the NB algorithm's performance. These hyperparameters have been tuned using the random search algorithm. Table 4 shows the values of these hyperparameters.

Table 4. Hyperparameters of the NB algorithm

#	Hyperparameter	Description	Value
1	alpha	Additive smoothing parameter	1.0
2	class_prior	Prior probabilities of the classes	None
3	fit_prior	Whether to learn class prior probabilities	True

2.3.3. AdaBoost classification algorithm

The adaptive AdaBoost algorithm gradually transforms a weak classifier into a reliable and effective one. Each cycle uses the weak classifier to classify the values in the training dataset. All the data values are assigned equal weights at the start of the training process. Nevertheless, with each subsequent cycle, the weight of the incorrectly categorized data points increases, making the classifier in that cycle rely more on them. As a result, this indicates a drop in the classifier's global error, creating a more robust and effective classifier [23]. Several hyperparameters impact the AdaBoost algorithm's performance. These hyperparameters have been tuned using the random search algorithm. Table 5 shows the values of these hyperparameters.

Table 5. Hyperparameters of the AdaBoost algorithm

#	Hyperparameter	Description	Value
1	n_estimators	Number of weak learners to use	200
2	learning_rate	Shrinks the contribution of each weak learner	0.1
3	algorithm	Algorithm to use	'SAMME.R'
4	base_estimator	The base estimator from which the boosted ensemble is built	DT(max_depth=3)
5	random_state	Controls the randomness of the estimator	123

3. RESULTS AND DISCUSSION

This section presents and discusses the results. Three ML algorithms with tuned parameters (see subsection 2.3) are used: DT, AdaBoost, and NB. The results were attained using a PC with the following software and hardware specifications: Acer Aspire E5-575G model, Windows 10 Pro, 64-bit O.S system type, Intel Core i5-7200 CPU (2.50 GHz speed, 2 Cores, and 4 Threads), 16 GB RAM, and Python programming language.

The performance metrics are based on the confusion matrix (Figure 2). A confusion matrix summarizes the number of right and wrong predictions made in a classification issue, broken down by class and summarized with count values. The performance metrics used are accuracy, recall, precision, and F1-score. Accuracy (1) is the dataset's number of correct spam and ham classifications. Precision (2) is the ratio of true positive to the sum of false positive and true positive. Recall (3) is the ratio of true positive to the sum of false negative and true positive. F1-score (4) is a measure of classification accuracy on a dataset that provides a balance between precision and recall [6], [24], [25].

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

Figure 2. Confusion matrix

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

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

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

$$F1 - score = 2 \times \frac{Pre \times Rec}{Pre+Rec} \tag{4}$$

Figures 3 to 6 show the accuracy, recall, precision, and F1-score, respectively. The results achieved by the DT algorithm for all metrics are 99.75%. The results achieved by the NB algorithm are as follows: accuracy is 96.30%%, recall is 96.30%%, precision is 96.19%, and F1-score is 96.10%, respectively. The results achieved by the AdaBoost algorithm for all metrics are 99.67%. The DT achieved the highest results of the three algorithms with all metrics, outperforming the NB by 3.45% with accuracy and recall, 3.65% with F1-score, and 3.56% with precision, and outperformed AdaBoost by 0.08% with all metrics.

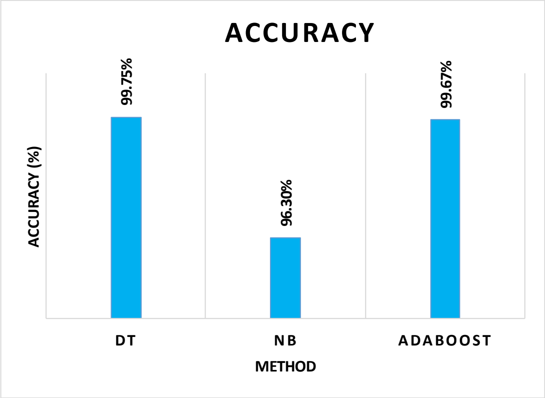

Figure 3. Accuracy of the HHO algorithm

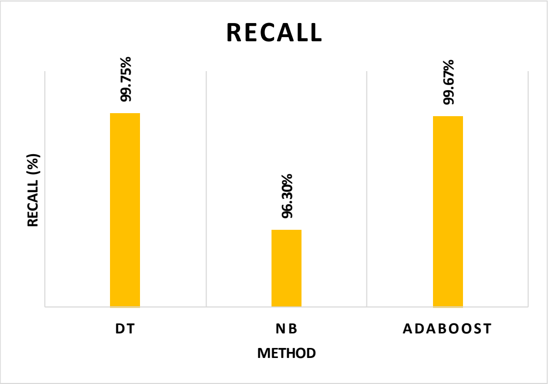

Figure 4. Recall of the HHO algorithm

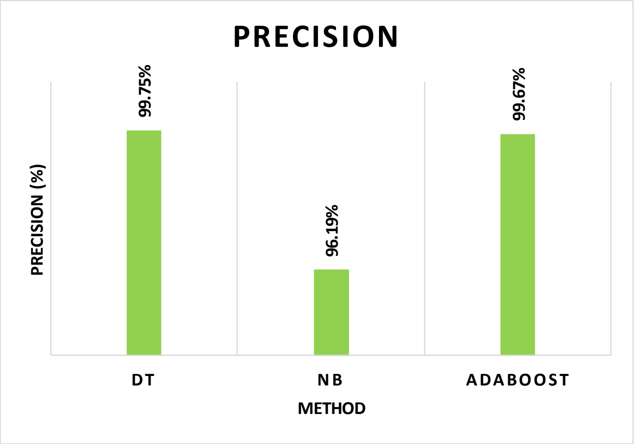

Figure 5. Precision of the HHO algorithm

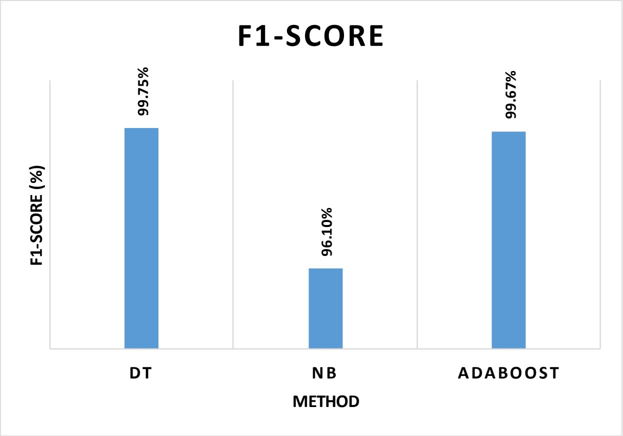

Figure 6. F1-score of the HHO algorithm

4. CONCLUSION

Spam emails are one of the threats that companies must face. Attackers exploit spam emails to spread various types of attacks. In this paper, the ML algorithms are employed to mitigate the spread of the spam emails. First, the HHO algorithm is utilized to identify the key features that help to distinguish spam from ham emails. The HHO algorithm has reduced the 72 features of the ISCX-URL2016 spam dataset to only ten features. Three well-known ML algorithms have been used to evaluate the performance of the subset of features HHO has selected: DT, AdaBoost, and NB algorithms. These three ML algorithms have been adjusted to suit the problem at hand: the spam detection problem. The DT, NB, and AdaBoost algorithms have attained a high accuracy of 99.75%, 96.30%, and 99.67%, respectively, in detecting spam emails. The DT algorithm has outperformed NB and AdaBoost algorithms by 3.45% and 0.08%, respectively. The result indicates that the HHO algorithm obtains the best result with the DT algorithm, demonstrating that the HHO and DT algorithms can potentially enhance the problems of detecting spam emails. Future works will compare the proposed model with other optimization feature selection methods, such as genetic algorithms or particle swarm optimization.

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, Jul.-Aug. 2022, doi: 10.1109/TDSC.2021.3055495.

[2] G. Kambourakis, G. D. Gil, and I. Sanchez, "What Email Servers Can Tell to Johnny: An Empirical Study of Provider-to-Provider Email Security," *IEEE Access*, vol. 8, pp. 130066-130081, 2020, doi: 10.1109/ACCESS.2020.3009122.

[3] M. M. Abualhaj, Q. S. Shambour, A. M. Alsaaidah, A. A. Abu-Shareha, S. N. Al-Khatib, and M. O. Hiari, "Enhancing Spam

Detection Using Hybrid of Harris Hawks and Firefly Optimization Algorithms," *Journal of Applied Data Sciences*, vol. 5, no. 3, 901-911, 2024, doi: 10.47738/jads.v5i3.279.

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

[5] A. Hussein and Q. Y. 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.

[6] M. M. Abualhaj, A. A. Abu-Shareha, M. O. Hiari, Y. Alrabanah, M. M. 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.

[7] X. Jiang, Y. Xu, W. Ke, Y. Zhang, Q. -X. Zhu, and Y. -L. He, "An Imbalanced Multifault Diagnosis Method Based on Bias Weights AdaBoost," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1-8, 2022, doi: 10.1109/TIM.2022.3149097.

[8] A. H. Mohammad, T. Alwada'n, O. Almomani, S. Smadi, and N. ElOmari, "Bio-inspired Hybrid Feature Selection Model for Intrusion Detection," *Computers, Materials & Continua*, vol. 73, no. 1, pp. 133–150, 2022, doi: 10.32604/cmc.2022.027475.

[9] Z. Yu, X. Shi, J. Zhou, X. Chen, and X. Qiu," Effective Assessment of Blast-Induced Ground Vibration Using an Optimized Random Forest Model Based on a Harris Hawks Optimization Algorithm," *Applied Sciences*, vol. 10, no. 4, p. 1403, 2020, doi: 10.3390/app10041403.

[10] 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.

[11] A. M. Al Saaidah *et al*., "Enhancing malware detection performance: leveraging K-Nearest Neighbors with Firefly Optimization Algorithm," *Multimedia Tools and Applications*, pp. 1-24, 2024, doi: 10.1007/s11042-024-18914-5.

[12] O. Almomani, "A Hybrid Model Using Bio-Inspired Metaheuristic Algorithms for Network Intrusion Detection System," *Computers, Materials & Continua*, vol. 68, no. 1, pp. 409–429, 2021, doi: 10.32604/cmc.2021.016113.

[13] M. M. Abualhaj, M. O. Hiari, A. Alsaaidah, M. Al-Zyoud, and S. Al-Khatib, "Spam Feature Selection Using Firefly Metaheuristic Algorithm," *Journal of Applied Data Sciences*, vol. 5, no. 4, pp. 1692–1700, 2024, doi: 10.47738/jads.v5i4.336.

[14] M. Rathi and V. Pareek, "Spam Mail Detection through Data Mining–A Comparative Performance Analysis," *International Journal of Modern Education and Computer Science*, vol. 5, no. 12, pp. 31–39, Dec. 2013, doi: 10.5815/ijmecs.2013.12.05.

[15] V. Ravi, S. Srinivasan, S. Kp, and M. Alazab, "Malicious url detection using deep learning," *TechRxiv*, pp. 1-9, 2020, doi: 10.36227/techrxiv.11492622.

[16] H. Le, Q. Pham, D. Sahoo, and S. Hoi, "Urlnet: Learning a url representation with deep learning for malicious url detection," *arXiv*, Mar. 2018, doi: 10.48550/arXiv.1802.03162.

[17] 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, 2023, pp. 359-364, doi: 10.1109/ICCIT58132.2023.10273923.

[18] 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, Proceedings, Sep. 2016, pp. 467-482. Springer International Publishing, 2016, doi: 10.1007/978-3-319-46298-1_30.

[19] L. Al-Dabbas and A. Abu-Shareha, "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, 2024, doi: 10.47738/jads.v5i3.298.

[20] D. Streeb *et al*., "Task-Based Visual Interactive Modeling: Decision Trees and Rule-Based Classifiers," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 9, pp. 3307-3323, 1 Sept. 2022, doi: 10.1109/TVCG.2020.3045560.

[21] R. Turner *et al*., "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*, Aug. 2021, pp. 3-26.

[22] F. Ramadhani, Al-Khowarizmi, and I. P. Sari, "Improving the Performance of Naïve Bayes Algorithm by Reducing the Attributes of Dataset Using Gain Ratio and Adaboost," *2021 International Conference on Computer Science and Engineering (IC2SE)*, Padang, Indonesia, 2021, pp. 1-5, doi: 10.1109/IC2SE52832.2021.9792027.

[23] S. Wu and H. Nagahashi, "Parameterized AdaBoost: Introducing a Parameter to Speed Up the Training of Real AdaBoost," *IEEE Signal Processing Letters*, vol. 21, no. 6, pp. 687-691, Jun. 2014, doi: 10.1109/LSP.2014.2313570.

[24] A. Almomani *et al*., "Ensemble-Based Approach for Efficient Intrusion Detection in Network Traffic," *Intelligent Automation & Soft Computing*, vol. 37, no. 2, pp. 2500-2517, 2023, doi: 10.32604/iasc.2023.039687.

[25] Q. Y. Shambour, N. M. Turab, and O. Y. Adwan, "An Effective e-Commerce Recommender System Based on Trust and Semantic Information," *Cybernetics and Information Technologies*, vol. 21, no. 1, pp. 103–118, Mar. 2021, doi: 10.2478/cait-2021-0008.

BIOGRAPHIES OF AUTHORS

Mosleh M. Abualhaj is a senior lecturer in Al-Ahliyya Amman University. He received his first degree in Computer Science from Philadelphia University, Jordan, in 2004, Master degree in Computer Information system from the Arab Academy for Banking and Financial Sciences, Jordan in 2007, and Ph.D. in multimedia networks protocols from Universiti Sains Malaysia in 2011. His research area of interest includes VoIP, congestion control, cybersecurity data mining, and optimization. He can be contacted at email: m.abualhaj@ammanu.edu.jo.

Sumaya Nabil Alkhatib is a senior lecturer in Al-Ahliyya Amman University. She received his first degree in Computer Science from Baghdad University, Iraq, in June 1994 and Master degree in Computer Information system from the Arab Academy for Banking and Financial Sciences, Jordan in February. Her research area of interest includes VoIP, multimedia networking, and congestion control. She can be contacted at email: sumayakh@ammanu.edu.jo.

Ahmad Adel Abu-Shareha received his first degree in Computer Science from Al Al-Bayt University, Jordan, 2004, Master degree from Universiti Sains Malaysia (USM), Malaysia, 2006, and Ph.D. degree from USM, Malaysia, 2012. His research focuses on data mining, artificial intelligent, and multimedia security. He investigated many machine learning algorithms and employed artificial intelligent in variety of fields, such as network, medical information process, knowledge construction, and extraction. He can be contacted at email: a.abushareha@ammanu.edu.jo.

Adeeb M. Alsaaidah received the Bachelor’s degree in Computer Engineering from the Faculty of Engineering, ALBalqa Applied University, the Master’s degree in Networking and Computer Security from NYIT University, and the Ph.D. degree in Computer Network from USIM, Malaysia. He is currently an Assistant Professor in Department of Network and Cybersecurity at Al-Ahliyya Amman University (AAU). His research interests include network performance, multimedia networks, network quality of service (QoS), the IoT, network modeling and simulation, network security, and cloud security. He can be contacted at email: a.alsaaidah@ammanu.edu.jo.

Mohammed Anbar (Member, IEEE) received the B.Sc. degree in Software Engineering from Al-Azhar University, Palestine, in 2008, the M.Sc. degree in Information Technology from Universiti Utara Malaysia, in 2009, and the Ph.D. degree in Advanced Internet Security and Monitoring from Universiti Sains Malaysia (USM), in 2013. He is currently a Senior Lecturer with the National Advanced IPv6 Centre (NAv6), USM. His current research interests include malware detection, intrusion detection systems (IDSs), intrusion prevention systems (IPSs), network monitoring, the internet of things (IoT), software-defined networking (SDN) security, cloud computing security, and IPv6 security. He can be contacted at email: anbar@usm.my.