

## Enhancing malware detection through self-union feature selection using gray wolf optimizer

Mosleh M. Abualhaj<sup>1</sup>, Qusai Y. Shambour<sup>1</sup>, Ahmad Adel Abu-Shareha<sup>2</sup>, Sumaya N. Al-Khatib<sup>2</sup>,  
Amal Amer<sup>2</sup>

<sup>1</sup>Department of Networks and Cybersecurity, Al-Ahliyya Amman University, Amman, Jordan

<sup>2</sup>Department of Data Science and Artificial Intelligence, Faculty of Information Technology, Al-Ahliyya Amman University, Amman, Jordan

### Article Info

#### Article history:

Received May 17, 2024

Revised Aug 23, 2024

Accepted Sep 7, 2024

#### Keywords:

Feature selection

Gray wolf optimizer

Machine learning

Malware

Random forest

### ABSTRACT

This research explores the impact of malware on the digital world and presents an innovative system to detect and classify malware instances. The suggested system combines a random forest (RF) classifier and gray wolf optimizer (GWO) to identify and detect malware effectively. Therefore, the suggested system is called RFGWO-Mal. The RFGWO-Mal system employs the GWO for feature selection in binary and multiclass classification scenarios. Then, the RFGWO-Mal system uses a novel self-union feature selection approach, combining features from different subsets of binary and multiclass classification extracted using the GWO optimizer. The RF classifier is then applied for classifying malware and benign data. The comprehensive Obfuscated-MalMem2022 dataset was utilized to evaluate the suggested RFGWO-Mal system, which has been implanted using Python. The suggested RFGWO-Mal system achieves significantly improved results using the novel self-union feature selection approach. Specifically, the RFGWO-Mal system achieves an outstanding accuracy of 99.95% in binary classification and maintains a high accuracy of 86.57% with multiclass classification. The findings underscore the achievement of a self-union feature selection approach in enhancing the performance of malware detection systems, providing a valuable contribution to cybersecurity.

*This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.*



### Corresponding Author:

Mosleh M. Abualhaj

Department of Networks and Cybersecurity, Al-Ahliyya Amman University

Amman, Jordan

Email: m.abualhaj@ammanu.edu.jo

## 1. INTRODUCTION

In the early part of the cyber era, cyberattacks were not a critical issue, and the number of attacks was very low. However, the rapid rise of the internet increases the number of threats that companies and individuals face. One of the most common threats is malware, software specifically designed to cause harm. Malware might be used for espionage, financial gain, or stealing sensitive information. Malware is exponentially increasing, causing very high damage and loss to the market [1], [2]. Statistics show that hundreds of thousands of malware are produced daily, contributing to an estimated cybercrime cost of 6 trillion dollars in 2021. The market predicted that malware analysis will increase more and more, causing growth from 3 billion USD in 2019 to 11.7 billion USD in 2024 [3].

Early malware was created using simple code; manually created rules were often enough to detect them. However, with the growing amount of malware and the pervasiveness of cyberspace, manually

constructed detection rules were no longer practical. However, malware designers constantly work on evading detections and releasing advanced malware [3]. The advanced malware can even evade the typical security means, including firewalls and antimalware [4]. Consequently, security providers must come up with new and advanced protection technologies. Security providers have adopted machine learning (ML) to boost their malware detection and classification [5]-[7].

ML refers to a collection of techniques enabling computers to acquire knowledge and improve performance without explicit programming. In other words, an ML algorithm identifies and codifies the fundamental rules that govern the data it observes. With this knowledge, the ML algorithm can deduce the characteristics of samples that have not been encountered before. In malware detection, an unencountered sample may refer to a new file. The concealed attribute could either be malware or benign [8]. Today, ML improves malware detection by utilizing diverse data types on the host, network, and cloud-based anti-malware components. The large numbers of data of various kinds present a challenge for ML algorithms to detect malware accurately [8], [9]. ML uses feature selection algorithms to address the challenge of large data of various kinds. Feature selection algorithms aim to enhance performance, mitigate overfitting, and reduce computational requirements. Careful selection of features will significantly improve the performance of ML-based malware detection systems [9], [10].

ML has encountered substantial difficulty in feature selection. Feature selection is regarded as a challenging topic due to the growing amount of time needed to identify the optimal features in a dataset with a high number of dimensions. Several scholars have recently suggested several metaheuristic algorithms for feature selection. Metaheuristic algorithms are a class of advanced techniques designed to generate solutions for complex optimization problems that are difficult to solve optimally [11], [12]. This study will employ the gray wolf optimizer (GWO), a widely recognized metaheuristic algorithm, for feature selection [13], [14]. Applying the GWO algorithm in feature selection involves customizing the algorithm to identify the most advantageous subset of features, thereby augmenting the performance of the ML system.

Several methods have been proposed to enhance malware detection. Gavrilut *et al.* [15] presented a malware detection framework that minimizes false positives. The methodology involves utilizing two combinations, simple and simple multi-stage, for variant versions of the perceptron algorithm. The core objective is to differentiate between clean files and malware effectively while minimizing false positives. The authors implemented their approach by computing various features within each binary file through malware analysis. The ML system was designed using feature-mapped and kernelized one-sided perceptrons. Key results from the study include noteworthy performance metrics for the cascade one-sided perceptron (COS-P) concerning F1 and F2 scores, achieving accuracy levels of 96.08% and 95.71%, respectively, in 3-fold cross-validation with the training dataset. However, with 20-fold cross-validation, COS-P-Map-F1 decreases slightly to 95.91%, while COS-P-Map-F2 attains higher accuracy at 95.79%. During measurements on the test dataset, the proposed algorithm demonstrates an accuracy of 85.54% for COS-P-Map-F1 and 85.17% for COS-P-Map-F2.

Li *et al.* [16] presented a malware detection method named SIGPID, focusing on permission usage analysis for Android malware. The research aimed to create three levels of pruning by incorporating data permissions to identify crucial permissions that distinguish between malicious and benign apps. The authors employed ML classification methods to classify various malware families. The proposed multi-level approach includes data pruning and permission ranking with a negative rate. The classification was utilized to categorize variants of both benign apps and malware. The evaluation was based on 22 significant permissions, and the findings indicated that the support vector machine (SVM) classifier achieved over 90% accuracy, closely matching the results generated by the baseline approach with acceptable analysis times (ranging from 4 to 32 times). The accuracy achieved in this study was 91.36% when using SVM in Android-dangerous permissions and multi-level-data pruning. SIGPID was also compared with different state-of-the-art approaches, demonstrating its effectiveness in detecting 93.62% of known malware and 91.4% of unknown/new malware.

Roseline *et al.* [17] proposed a malware detection and classification system using a deep random forest (RF) paradigm based on intelligent vision. The system employs a multilayer ensemble method inspired by the main properties of deep learning. The key idea is to analyze binary executable files (malware and cleanware) using a vision-based analysis technique. This technique leverages the observation that malware designers often reemploy the same code segments to create different malware variants, making it easier to visualize the binary code of the malware. The suggested model achieved a 97.2% detection rate for the BIG 2015 dataset, 98.65% for the Malimg dataset, and 97.43% for the MaleVis dataset. SVM, logistic regression (LR), and Naïve Bias (NB) models showed slower performance classifying malware for the entire dataset. The authors observed that test accuracy improved in Layer 2 and 5, with no further improvement in subsequent layers. The BIG2015 dataset successfully built 11 layers, but test accuracy showed no improvement after Layer 9. Similarly, no improvement was observed for the MaleVis dataset.

Akhtar and Feng [18] study focuses on detecting harmful traffic associated with computer systems to enhance network security. The research explores general variances in correlation-symmetry integrals using malware detection and analysis findings. The study presents various steps and components of the typical workflow in ML, aiming to understand the main limitations and challenges in malware detection and classification. All ML classifiers (convolutional neural networks (CNN), k-nearest neighbours (KNN), NB, SVM, RF, or decision tree (DT)) performed better in all annotated datasets. The results highlight that DT achieved the highest accuracy (99%) compared to other learning algorithms, including the proposed method. The statistical analysis computed the accuracy of classifiers as follows: CNN=98.76%, KNN=95.02%, NB=89.71%, SVM=96.41%, RF=92.01%, and DT=99%. In conclusion, the study suggests that DT, SVM, CNN, and KNN classifiers performed accurately in all circumstances.

Urooj *et al.* [19] have developed a framework to detect malicious Android applications, leveraging various elements of ML to identify such applications. Determining and choosing functions to record and examine Android app behavior is part of the suggested framework. This is achieved using AndroGuard to extract features from binary vectors and reverse application engineering. During testing, the proposed framework achieves a 96% accuracy in the given context with a low false positive rate of 0.3, especially when larger and improved feature and sample sets are used. The study shows ensemble and strong learner algorithms perform better than other approaches when handling classifications and high-dimensional data. The suggested framework has drawbacks, including its limited static analysis, absence of sustainability considerations, and neglect of a crucial multicollinearity barrier.

**2. METHOD**

The suggested system combines an RF classifier and GWO optimizer to identify and detect malware effectively. Therefore, the suggested system is called RFGWO-Mal. This section discusses the operations performed by the RFGWO-Mal to enhance the malware detection process. The section first provides an overview of the Obfuscated-MalMem2022 (MalMem) dataset, explaining its purpose in assessing the suggested system. Subsequently, it provides a comprehensive discussion of the procedures to preprocess the dataset for classification purposes. The feature selection process, which involves using the GWO optimizer, is further explained. Finally, the section explores the classification process of malware, providing a clear explanation of the role of the RF classifier.

**2.1. Data preparation**

In malware detection, the MalMem dataset is a broadly utilized benchmark. The MalMem dataset is created to simplify the assessment of malware systems by providing a more realistic malware environment for evaluation. The dataset contains 58,596 samples of network traffic gathered in a simulated environment, with samples divided evenly into malware and benign data kinds. The malware kinds are further subdivided into three primary kinds: 10,020 samples of Spyware, 9,791 samples of Ransomware, and 9,487 samples of Trojan Horse. In addition, a total of 56 features are included in the MalMem dataset, including the output column, as shown in Table 1 [2], [20].

Table 1. MalMem dataset features

#	Feature name	#	Feature name	#	Feature name	#	Feature name
1	pslist.nproc	15	handles.nthread	29	malfind.protection	43	psxview.not_in_session_fals e_avg
2	pslist.nppid	16	handles.ndirectory	30	malfind.uniqueInjections	44	psxview.not_in_deskthrd_fal se_avg
3	pslist.avg_threads	17	handles.nsemaphore	31	psxview.not_in_pslist	45	modules.nmodules
4	pslist.nprocs64bit	18	handles.ntimer	32	psxview.not_in_eprocess _pool	46	svcsan.nservices
5	pslist.avg_handlers	19	handles.nsection	33	psxview.not_in_ethread_ pool	47	svcsan.kernel_drivers
6	dlllist.ndlls	20	handles.nmutant	34	psxview.not_in_pspcid_li st	48	svcsan.fs_drivers
7	dlllist.avg_dlls_per _proc	21	ldrmodules.not_in_lo ad	35	psxview.not_in_csrrs_ha ndles	49	svcsan.process_services
8	handles.nhandles	22	ldrmodules.not_in_in it	36	psxview.not_in_session	50	svcsan.shared_process_serv ices
9	handles.avg_handle s_per_proc	23	ldrmodules.not_in_m em	37	psxview.not_in_deskthrd	51	svcsan.interactive_process_ services

The data within the MalMem dataset should be prepared for the ML model. Data preparation involves two main processes, namely data transformation and data normalization. Data transformation is necessary because ML systems cannot handle non-numeric data. It is central to represent the data numerically to avoid any issues at later stages. Converting all values into numerical form is the solution to this issue. Label encoding technique can be used for data transformation [21]. In the case of binary classification, Label-encoding assigns the values 0 and 1 to the 'normal' and 'malware' labels, respectively. In the case of multiclass classification, the MalMem dataset contains four different values in the output column: 'normal,' 'Trojan Horse,' 'Spyware,' and 'Ransomware'. The Label-encoding assigns 0 to 'normal', 1 to 'Trojan Horse', 2 to 'Spyware', and 3 to 'Ransomware'. As for data normalization, the numerical values are scaled to fit within a specified range, preventing any bias towards variables with larger scales. The min-max technique can be used for data scaling, which scales the data to range between 0 and 1 [21], [22].

## 2.2. Feature selection

Feature selection have become a major tool in reducing the dimension of huge datasets available in cybersecurity, yet maintaining performance accuracy is crucial. Typically, the number of samples representing the distribution of the high-dimensional feature spaces is insufficient. Therefore, reducing dimensionality is critical in numerous cybersecurity fields, such as malware detection. Effective feature selection method leads to improving the performance of an ML system. This study aims to provide a unique self-union method for selecting features by utilizing the GWO optimizer.

### 2.2.1. GWO optimizer

The GWO algorithm was built based on the chasing behavior of a pack of gray wolves in their natural environment. The algorithm mimics gray wolves' chasing strategy and hierarchical leadership in the wild. Lately, the GWO has been used for feature selection issues in data mining. The GWO algorithm provide several benefits to detect of malware. The GWO is efficient algorithm in exploring the feature space and thus select the most relevant feature for malware and increase the ML-based malware detection systems. The exploration process in GWO adapts dynamically to the evolving attributes of malware, ensuring resilience against shifting threat environments. In addition, the rapid convergence speed of GWO is important for detecting new malware patterns in real-time. Furthermore, the GWO algorithm impose computation operations, which make it useful for environments with limited resources, as it allows for efficient malware identification without the need for costly hardware. The GWO optimizer performs several operations to find the relevant features for malware detection [13], [14], [23], [24].

### 2.2.2. Self-union feature selection using GWO

The suggested RFGWO-Mal system employs the GWO optimizer for feature selection. Specifically, GWO has identified 4 out of 55 features for binary classification and 8 out of 55 for multiclass classification, as detailed in Table 2. Conventional feature selection (C-FS) involves selecting a subset of relevant features from the initial feature set [25], [26]. In addition to the C-FS approach, the suggested RFGWO-Mal system introduces a novel union feature selection (U-FS) approach. This novel approach combines features from different subsets to determine the most relevant ones. The union feature selection method is anticipated to outperform the traditional single feature selection approach. This is attributed to its potential to enhance accuracy and efficiency in the feature selection process, ultimately improving the overall performance of the ML system and mitigating overfitting [25], [26].

Typically, the U-FS approach combines features from different subsets of different optimizers. The RFGWO-Mal system introduces a novel self-union U-FS method that combines features from different subsets of a single optimizer. The suggested self-union U-FS method works as follows. Initially, the GWO optimizer conducts feature selection for binary classification. Then, the GWO optimizer conducts feature selection for multiclass classification. Subsequently, the selected features from binary and multiclass classification are combined into a single subset. This innovative self-union feature selection method aims to enhance the overall performance of the RFGWO-Mal system while mitigating overfitting. Figure 1 depicts the suggested self-union U-FS method processes. Table 2 provides a detailed list of the union of features derived from binary and multiclass classification.

Table 2. Selected feature by different methods

Method	Selected features (feature number)
Binry	2, 20, 23, 46
Multiclass	13, 14, 16, 18, 20, 24, 27, 54
Self-union U-FS	2, 13, 14, 16, 18, 20, 23, 24, 27, 46, 54

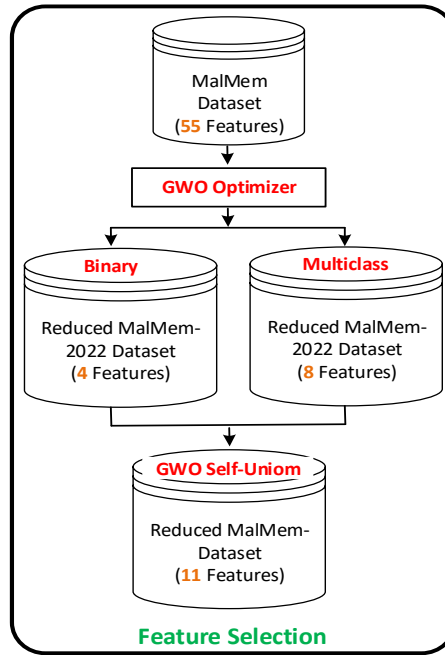


Figure 1. The suggested self-union method

### 2.3. Classification

In the previous steps (section 3.1 and 3.2), the data was processed and prepared for classification, distinguishing malware from benign data. In the RFGWO-Mal system, the classification task will utilize the RF classifier. RF is an ML classifier specifically developed to address classification tasks. RF utilizes ensemble learning, which involves the integration of several classifiers to tackle intricate problems [27], [28]. At this level, the suggested RFGWO-Mal system is completed to detect malware. Section 4 presents the performance of the RFGWO-Mal system.

## 3. RESULTS AND DISCUSSION

### 3.1. Implementation environment

The suggested RFGWO-Mal system was conducted on a desktop with Intel Core i9-14900KF (3.2 GHz 24 Core CPU, 32 Thread, 6.0 GHz Turbo), 32 GB DDR5 RAM, 1 TB SSD, Intel UHD Graphics 770, and Ubuntu 23.10 O.S. Python was utilized to test and evaluate the suggested system. Several libraries from Python were utilized to implement the RFGWO-Mal system for detecting malware. Some of these libraries are ‘pandas’, ‘numpy’, ‘sklearn.preprocessing’, ‘mealpy.swarm\_based.GWO’, and ‘RandomForestClassifier’.

### 3.2. Performance evaluation criteria

The confusion matrix is widely utilized for measuring the performance of ML systems, such as the suggested RFGWO-Mal malware detection system. The elements of the confusion matrix are true positives (TPo), true negatives (TNe), false positives (FPo), and false negatives (FNe). The metrics that have been utilized in this work are derived from the four elements of the confusion matrix. These metrics are the accuracy (Acc) which calculated using (1), precision (Pr) which calculated using (2), and recall (Re) which calculated using (3). Besides, K-fold cross-validation, with k equal to 5, is utilized to ensure consistent performance of the system across different subsets of the data, reducing the risk of overfitting to a particular train-test split [8], [25], [28].

$$Acc = \frac{(TPo+TNe)}{(TPo+TNe+FPo+FNe)} \tag{1}$$

$$Pr = \frac{TPo}{(TPo+FPo)} \tag{2}$$

$$Re = \frac{TPo}{(TPo+FNe)} \tag{3}$$

### 3.3. Results

The suggested RFGWO-Mal system was tested with binary and multiclass classification types. For each type, the subset of features selected by the GWO optimizer was tested using the C-FS and suggested self-union U-FS approaches (refer to section 3.2). The RF classifier will perform the classification task in the suggested RFGWO-Mal system (refer to section 3.3).

#### 3.3.1. Binary classification

Figures 2 to 4 show the Acc, Re, and Pr of the suggested RFGWO-Mal system with binary classification, respectively. Across all three measures, the typical C-FS method using GWO demonstrated an impressive value of 99.95%. Also, the suggested union U-FS method has achieved the same value of 99.95% with all three metrics. Based on the tested measures, the results show that the suggested U-FS method works just as well in binary classification as the C-FS method with the RFGWO-Mal system.

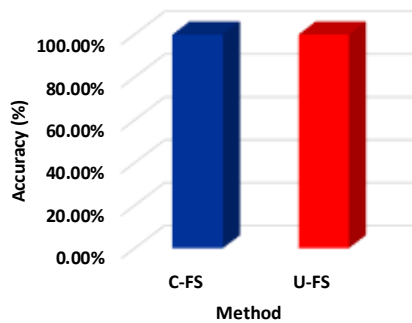


Figure 2. Acc of the RFGWO-Mal system with binary classification

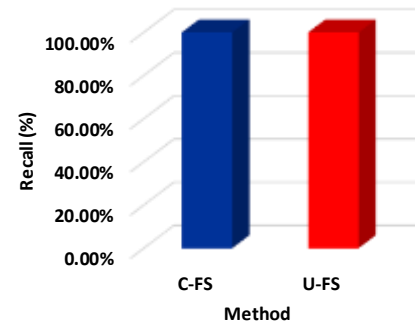


Figure 3. Re of the RFGWO-Mal system with binary classification

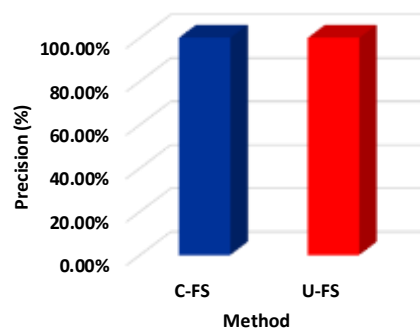


Figure 4. Pr of the RFGWO-Mal system with binary classification

#### 3.3.2. Multiclass classification

Figures 5 to 7 show the Acc, Re, and Pr of the suggested RFGWO-Mal system with multiclass classification, respectively. Across all three measures, the typical C-FS method using GWO demonstrated high results: Acc=86.33%, Re=86.33%, and Pr=86.32%. In contrast, the suggested union U-FS method using GWO surpassed this, achieving higher results: Acc=86.57%, Re=86.57%, and Pr=86.58%. Consequently, the results obtained through the suggested U-FS method surpass those achieved by the typical C-FS method, indicating its superior performance in multiclass classification based on the evaluated measures.

In summary, the self-union feature selection approach demonstrated equivalent performance to conventional feature selection in binary classification tasks. However, it significantly outperformed conventional methods in multiclass classification scenarios. Specifically, the self-union feature selection approach showed improvements in accuracy, recall, and precision by 0.24%, 0.24%, and 0.26%, respectively, compared to traditional feature selection techniques. These enhancements highlight the effectiveness of the self-union feature selection method, particularly in complex classification tasks. Consequently, the proposed RFGWO-Mal system proves to be a highly efficient tool in the field of malware detection, offering superior performance and reliability. This advancement not only strengthens cybersecurity measures but also provides a robust framework for future research and development in malware detection technologies.

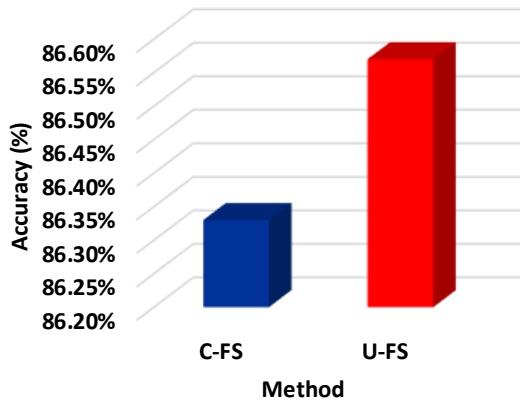


Figure 5. Acc of the RFGWO-Mal system with multiclass classification

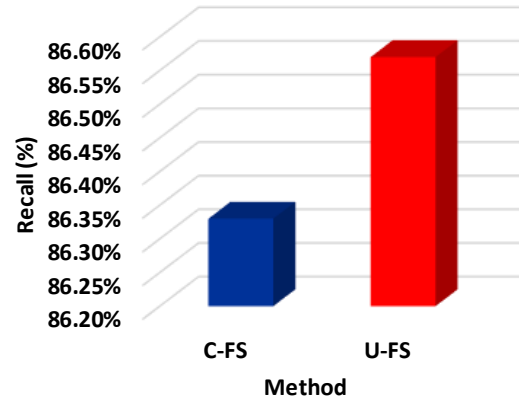


Figure 6. Re of the RFGWO-Mal system with multiclass classification

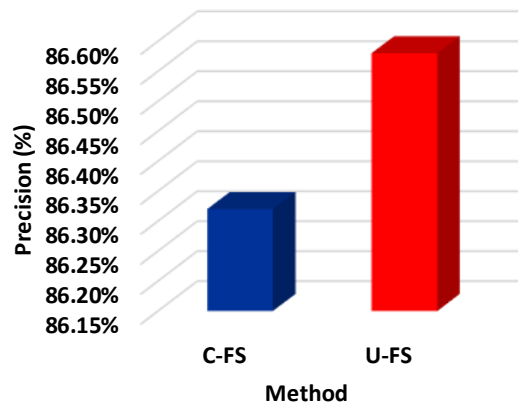


Figure 7. Pr of the RFGWO-Mal system with multiclass classification

#### 4. CONCLUSION

This research suggested the RFGWO-Mal system, a novel approach to detecting and classifying malware instances in the digital landscape. The RFGWO-Mal system employs two well-known ML algorithms, namely GWO and RF algorithms, to enhance the malware detection process. Besides, the RFGWO-Mal system utilizes a novel self-union feature selection approach that combines the features from the same optimizer. The combined features were extracted using a GWO optimizer with binary and multiclass classification. Then, the RFGWO-Mal system employs the RF classifier to perform the classification task, predicating the malware files. The evaluation of the RFGWO-Mal system on the Obfuscated-MalMem2022 dataset demonstrated the system's outstanding performance. The RFGWO-Mal system achieves an accuracy of 99.95% and 86.57% with binary and multiclass classification, respectively. These results highlight the effectiveness of the self-union feature selection approach in enhancing malware detection systems, demonstrating that this approach has made a significant contribution to cybersecurity. In future research, the self-union method will be evaluated with other ML classifiers such as KNN, DT, and SVM. In addition, the RFGWO-Mal system will be evaluated with other malware datasets.

#### REFERENCES

- [1] J. H. Park, "Symmetry-adapted machine learning for information security," *Symmetry*, vol. 12, no. 6, pp. 1-4, Jun. 2020, doi: 10.3390/sym12061044.
- [2] M. M. Abualhaj, A. A. Abu-Shareha, Q. Shambour, A. 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.
- [3] M. Gopinath and S. C. Sethuraman, "A comprehensive survey on deep learning based malware detection techniques," *Computer Science Review*, vol. 47, pp. 100529, Feb. 2023, doi: 10.1016/j.cosrev.2022.100529.
- [4] M. Belaoued, A. Derhab, S. Mazouzi, and F. A. Khan, "MACOMAL: a multi-agent based collaborative mechanism for Anti-Malware assistance," *IEEE Access*, vol. 8, pp. 14329-14343, Jan. 2020, doi: 10.1109/access.2020.2966321.

- [5] K. Shaukat, S. Luo, and V. Varadharajan, "A novel deep learning-based approach for malware detection," *Engineering Applications of Artificial Intelligence*, vol. 122, p. 106030, Jun. 2023, doi: 10.1016/j.engappai.2023.106030.
- [6] V. Vasani, A. K. Bairwa, S. Joshi, A. Pljonkin, M. Kaur, and M. Amoon, "Comprehensive analysis of advanced techniques and vital tools for detecting malware intrusion," *Electronics*, vol. 12, no. 20, p. 4299, Oct. 2023, doi: 10.3390/electronics12204299.
- [7] 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," *IEEE Access*, vol. 8, pp. 163608–163617, Jan. 2020, doi: 10.1109/access.2020.3021983.
- [8] M. M. Abualhaj, A. A. Abu-Shareha, M. O. Hiari, Y. Alrabanah, M. Al-Zyouid, 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, Jan. 2022, doi: 10.14569/ijacsa.2022.0130325.
- [9] H. Zhao, Q. Hu, P. Zhu, Y. Wang, and P. Wang, "A recursive regularization-based feature selection framework for hierarchical classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 7, pp. 2833–2846, Jul. 2021, doi: 10.1109/tkde.2019.2960251.
- [10] T. Zhang, T. Zhu, P. Xiong, H. Huo, Z. Tari, and W. Zhou, "Correlated differential privacy: feature selection in machine learning," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 2115–2124, Mar. 2020, doi: 10.1109/tii.2019.2936825.
- [11] L. Jovanovic *et al.*, "Improving phishing website detection using a hybrid two-level framework for feature selection and XGBoost tuning," *Journal of Web Engineering*, Jul. 2023, doi: 10.13052/jwe1540-9589.2237.
- [12] O. M. Alyasiri, Y.-N. Cheah, A. K. Abasi, and O. M. Al-Janabi, "Wrapper and hybrid feature selection methods using metaheuristic algorithms for english text classification: a systematic review," *IEEE Access*, vol. 10, pp. 39833–39852, Jan. 2022, doi: 10.1109/access.2022.3165814.
- [13] L. Xu *et al.*, "Accurate and efficient performance prediction for mobile IOV networks using GWO-GR neural network," *IEEE Internet of Things Journal*, vol. 9, no. 17, pp. 16463–16471, Sep. 2022, doi: 10.1109/jiot.2022.3152739.
- [14] X. Sun, Y. Zhang, X. Tian, J. Cao, and J. Zhu, "Speed sensorless control for IPMSMs using a modified MRAS with gray wolf optimization algorithm," *IEEE Transactions on Transportation Electrification*, vol. 8, no. 1, pp. 1326–1337, Mar. 2022, doi: 10.1109/tte.2021.3093580.
- [15] D. Gavriluț, M. Cimpoeșu, D. Anton, and L. Ciortuz, "Malware detection using machine learning," *2009 International Multiconference on Computer Science and Information Technology*, Mragowo, Poland, 2009, pp. 735-741, doi: 10.1109/IMCSIT.2009.5352759.
- [16] J. Li, L. Sun, Q. Yan, Z. Li, W. Srisa-An, and H. Ye, "Significant permission identification for machine-learning-based android malware detection," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 3216–3225, Jul. 2018, doi: 10.1109/tii.2017.2789219.
- [17] S. A. Roseline, S. Geetha, S. Kadry, and Y. Nam, "Intelligent vision-based malware detection and classification using deep random forest paradigm," *IEEE Access*, vol. 8, pp. 206303–206324, Jan. 2020, doi: 10.1109/access.2020.3036491.
- [18] M. S. Akhtar and T. Feng, "Malware analysis and detection using machine learning algorithms," *Symmetry*, vol. 14, no. 11, p. 2304, Nov. 2022, doi: 10.3390/sym14112304.
- [19] B. Urooj, M. Shah, C. Maple, M. Abbasi, and S. Riasat, "Malware detection: a framework for reverse engineered android applications through machine learning algorithms," *IEEE Access*, vol. 10, pp. 89031–89050, 2022.
- [20] T. L. Carrier, P. Victor, A. Tekeoglu, and A. H. Lashkari, "Detecting obfuscated malware using memory feature engineering," in *The 8th International Conference on Information Systems Security and Privacy (ICISSP)*, Jan. 2022, doi: 10.5220/0010908200003120.
- [21] H. 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, Dec. 2023, doi: 10.3844/jcssp.2023.1549.1560.
- [22] H. Al-Mimi, N. A. Hamad, M. M. Abualhaj, M. S. Daoud, M. Rasmi, "Enhanced intrusion detection system for protecting HTTP services from attacks," *International Journal of Advances in Soft Computing and Its Applications*, vol. 15, no. 3, pp. 67-84, 2023, doi: 10.3844/jcssp.2023.1549.1560.
- [23] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46-61, 2014.
- [24] H. Faris, I. Aljarah, M. A. Al-Betar, and S. Mirjalili, "Grey wolf optimizer: a review of recent variants and applications," *Neural Computing and Applications*, vol. 30, pp. 413-435, 2018.
- [25] Ü. Çavuşoğlu, "A new hybrid approach for intrusion detection using machine learning methods," *Applied Intelligence*, vol. 49, no. 7, pp. 2735–2761, Feb. 2019, doi: 10.1007/s10489-018-01408-x.
- [26] Y. Hou, H. Gao, Z. Wang, and C. Du, "Improved grey wolf optimization algorithm and application," *Sensors*, vol. 22, no. 10, p. 3810, May 2022, doi: 10.3390/s22103810.
- [27] Y. Ren, X. Zhu, K. Bai, and R. Zhang, "A new random forest ensemble of intuitionistic fuzzy decision trees," *IEEE Transactions on Fuzzy Systems*, vol. 31, no. 5, pp. 1729–1741, May 2023, doi: 10.1109/tfuzz.2022.3215725.
- [28] P. N and S. Sugave, "Ensemble approach with hyperparameter tuning for credit worthiness prediction," *2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT)*, Bangalore, India, 2022, pp. 1-5, doi: 10.1109/GCAT55367.2022.9971879.





## BIOGRAPHIES OF AUTHORS







**Prof. 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, and cybersecurity data mining and optimization. He can be contacted at email: m.abualhaj@ammanu.edu.jo.









**Dr. Qusai Y. Shambour**     received the B.Sc. degree in Computer Science from Yarmouk University, Jordan, in 2001, the M.S. degree in computer networks from University of Western Sydney, Australia, in 2003, and the Ph.D. degree in software engineering from the University of Technology Sydney, Australia, in 2012. Currently, he is a Professor at the Department of Software Engineering, Al-Ahliyya Amman University, Jordan. His research interests include information filtering, recommender systems, VoIP, machine learning, and data science. He can be contacted at email: q.shambour@ammanu.edu.jo.







**Dr. 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.



**Ms. Sumaya N. Al-Khatib**     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.



**Ms. Amal Amer**     is a lecturer in Al-Ahliyya Amman University. She received her first degree in Computer Information Science from Al Balqa Applied University, Jordan, in August 2012 and Master degree in Information Systems from University of Jordan, Jordan in August 2016. Her research area of interest includes VoIP, blockchain, and cybersecurity data mining and optimization. She can be contacted at email: a.amer@ammanu.edu.jo.