Bulletin of Electrical Engineering and Informatics
Vol. 15, No. 2, April 2026, pp. 1265~1282
ISSN: 2302-9285, DOI: 10.11591/eei.v15i2.9887

A firewall model for attack detection using machine learning and metaheuristic feature selection algorithms

Mosleh M. Abualhaj[1], Sumaya Nabil Al-Khatib[2], Nida Al-Shafi[1], Mohammad O. Hiari[1], Mohammad Sh. Daoud[3], Mohammed Anbar[4], Mahran M. Al-Zyoud[1]
[1]Department of Networks and Cybersecurity, Faculty of Information Technology, Al-Ahliyya Amman University, Amman, Jordan
[2]Department of Computer Science, Faculty of Information Technology, Al-Ahliyya Amman University, Amman, Jordan
[3]College of Engineering, Al Ain University, Abu Dhabi, United Arab Emirates
[4]Cybersecurity Research Center (CYRES), Universiti Sains Malaysia (USM), Penang, Malaysia

Article Info

Article history:

Received Jan 14, 2025
Revised Dec 15, 2025
Accepted Feb 22, 2026

Keywords:

Bat Algorithm
Dragonfly Algorithm
Feature selection
Machine learning
UNSW-NB15 dataset

ABSTRACT

This research presents a firewall model designed to enhance network attack detection by integrating machine learning (ML) and advanced feature selection techniques. The study introduces a union-based (DAUBA) feature selection method that combines the exploratory capability of the Dragonfly Algorithm (DA) with the exploitation efficiency of the Bat Algorithm (BA). By combining these two bio-inspired optimizers, the method generates complementary feature subsets that enhance both accuracy and efficiency. The proposed DAUBA feature selection method is incorporated into a ML–based firewall and evaluated on the UNSW-NB15 dataset using three classifiers: adaptive boosting (AdaBoost), K-nearest neighbor (KNN), and Naïve Bayes (NB). Experimental results demonstrate that the approach achieves near-perfect accuracy (100% with AdaBoost), along with strong precision, recall, and F1-scores, while maintaining computational costs compatible with real-time deployment. These findings highlight the novelty and practical value of combining DA and BA in feature selection for next-generation firewall systems.

Corresponding Author:

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

1. INTRODUCTION

Currently, humankind heavily relies on the digital world in all aspects of life, including jobs, study, communication, and even entertainment [1]. However, the digital world contains a considerable number of vulnerabilities that are exploited by malicious actors, commonly referred to as hackers. Hackers are using specialized tools and techniques to launch attacks against the digital world, exploiting its vulnerabilities. These attacks are targeting the infrastructure and services of the digital world [2], [3]. In 2022, the cost of launched attacks exceeded 8 trillion USD [4]. This huge cost requires the attention of security engineers to protect against severe attacks. Security engineers have developed numerous tools over the years to defend against digital world attacks. These tools include web vulnerability scanning, antivirus software, and firewalls [5], [6].

Firewalls are security tools that use various techniques to defend against attacks. Traditional firewalls use simple rules to protect against attacks. The firewall searches for specific information in the network traffic and matches it to the corresponding firewall rules. Then, network traffic is permitted or denied based on these rules [7]–[9]. However, hackers are using advanced tools and techniques that can easily penetrate traditional

firewalls. Therefore, the firewall must employ smart techniques that enable advanced inspection of the entire network traffic to detect attacks [9], [10]. Recently, AI has become an essential component of the algorithms used by firewalls to detect attacks. Specifically, the current firewalls are using a branch of AI called machine learning (ML) [9], [10].

ML algorithms analyze previous benign and malicious network traffic to prevent future attacks [11]. However, network traffic is huge and hard to study and analyze using ML algorithms. Additionally, several attributes of the network traffic are unrelated to the attacks. Therefore, the accuracy of detecting attacks by the firewall based on ML could be reduced [12], [13]. For that, the volume of the traffic must be reduced by keeping only the key attributes that can be used effectively to detect the attacks. ML uses so-called feature selection algorithms to achieve this goal. Feature selection algorithms filter out the characteristics of network traffic that have less impact on detecting attacks and retain only the key attributes. Several types of algorithms are used for feature selection, including metaheuristic algorithms [12], [14], [15].

Over the last decade, metaheuristic algorithms have been applied in various domains to tackle complex problems. Metaheuristic algorithms have been proven to be efficient and reliable, achieving high results in the cybersecurity domain. One of the main groups of metaheuristic algorithms is derived from the behavior of animals in getting their food [16]–[18]. Dragonfly Algorithm (DA) and Bat Algorithm (BA) are widely used metaheuristic algorithms in various domains to solve different problems [19], [20]. In this work, the DA and BA metaheuristic algorithms will be used to improve the performance of the AI-based firewall. Specifically, the DA and BA will be utilized as feature selection algorithms to identify the key attributes that facilitate the identification of attacks in network traffic. Additionally, adaptive boosting (AdaBoost), K-nearest neighbor (KNN), and Naïve Bayes (NB) algorithms will be employed with the proposed ML–based firewall [21]–[23].

Several researchers have employed ML to detect cyber intrusions. Several researchers have employed ML to detect cyber intrusions. Chuang and Chen [12] have proposed an intrusion detection system (IDS) for protecting industrial IoT (IIoT) devices against attacks. The proposed IDS system utilizes the PCC technique to identify and evaluate the relationships between packet features. Then, a filtering rule is employed to refine the features, reducing the TON_IoT dataset from 45 to only ten key features for intrusion detection. The experiments that were conducted demonstrate that the proposed IDS can achieve an accuracy exceeding 96%.

Habeeb and Babu [24] have proposed a network intrusion detection system (NIDS) system that has been designed specifically for IoT networks. The proposed NIDS system uses a novel two-stage feature selection method. In the first stage, a coarse feature selection is implemented to analyze the relations among the features of the IoT traffic. In the second stage, a fine feature selection is implemented using the whale optimization algorithm (WOA) combined with the genetic algorithm (GA). The GA is combined with the WOA to enhance the search space and avoid local optima through crossover and mutation operations. The suggested feature selection method has reduced the number of features to only 32. The performance of the suggested NIDS system is assessed using the BoT-IoT 2020 dataset. The results indicate that the proposed NIDS system utilizing the GA-WOA feature selection method achieved an accuracy of 99.5%.

Wisanwanichthan and Thammawichai [25] proposed an IDS system that employs a double-layered hybrid approach (DLHA) to address the significant disparity in attack patterns. The first layer of DLHA uses an NB classifier to detect DoS and probe attacks. The second layer of DLHA uses a support vector machine (SVM) classifier to detect R2L and U2R attacks. The DLHA approach detects anomalies in network traffic by finding and comparing the probability of each feature in a new network packet to the normal behavior. The proposed IDS system was evaluated on the NSL-KDD dataset. The experiments conducted demonstrate that the DLHA approach outperforms NB and SVM classifiers when working independently. Whereas the DLHA approach achieves an accuracy of 88.97% and a false-positive rate of 0.12%.

Robinson *et al.* [26] proposed an IDS system that utilized ML and data mining to distinguish between normal traffic and various types of attacks. The proposed IDS system employs two distinct feature selection techniques: correlation-based and information gain-based. Correlation-based analysis is used to find and prioritize features that have a strong relationship with attacks. On the other hand, the information gain technique is used to find significant information about the attacks. Additionally, the proposed IDS system utilizes an oversampling method to enhance the number of samples for minority attacks, thereby ensuring the IDS system is adequately trained to detect these attacks. The CICIDS 2017 dataset is used in the valuation process. The experiments conducted show an achieved accuracy of 99.8%.

Ali [27] proposes a PSO-ML hybrid model for IoT-based DDoS detection, integrating particle swarm optimization (PSO) with ML classifiers to improve detection accuracy and resource efficiency. PSO is utilized for feature selection and parameter optimization, enabling the model to learn more effectively and make precise decisions. The system employs random forest (RF), SVM, and multilayer perceptron (MLP) classifiers to analyze traffic patterns and classify network traffic as normal or malicious in real time. The approach was extensively evaluated using four benchmark datasets (UNSW-NB15, CICIDS2017, KDDCUP99, and SDN),

achieving an accuracy of up to 99.64% using MLP on UNSW-NB15. While simulation results demonstrate near-real-time detection and high performance, further validation is needed under real-world IoT conditions.

Almomani [28] proposed a hybrid IDS model using bio-inspired metaheuristic algorithms that combines several approaches. The model focuses on wrapper-based feature selection with k-NN fitness evaluation, aiming to reduce dimensionality while preserving classification accuracy. It was evaluated on the UNSW-NB15 dataset, which contains nine attack types, with a particular emphasis on detecting Generic attacks, the most frequent in the dataset. For classification, J48, SVM, and RF were used. Results show that feature reduction was achieved from 49 features to as few as 8, though the best-performing hybrids relied on 24 features. Specifically, MVO-BAT retained 24 features with accuracy equivalent to the full set. Overall, the proposed hybridization strategy achieved an accuracy of ≈approximately 92.8%, demonstrating the practical value of combining metaheuristic algorithms for intrusion detection.

Kumar and Kumar [29] proposed a hybrid IDS model using the GWO–LOA–RF scheme, which combines Grey Wolf and Lion Optimizers with RF. The method employs a hybrid feature selection strategy that combines RF importance with a metaheuristic search for dimensionality reduction. An SVM classifier is then used as the final detection model. Preprocessing steps include categorical encoding and min–max normalization to prepare the data for analysis. The model was evaluated on the UNSW-NB15 and CICDDoS2019 benchmark datasets, with a focus on both feature selection efficiency and classification accuracy. It achieved 99.23% accuracy on UNSW-NB15 and 99.13% on CICDDoS2019, outperforming other tested models.

Shuaibu and Alabi [30] proposed a hybrid IDS feature selection method that combines BGSA and BGWO, utilizing an intersection strategy to integrate GS-decision tree (DT) and GW-DT. In this design, wrapper-based feature selection is employed with a DT as the evaluator, while DT, AdaBoost, and RF are later used as the final classifiers. The resulting ensemble, GSGW-DT, selects only four optimal features, significantly reducing dimensionality. Preprocessing includes categorical encoding and min–max normalization to prepare the dataset. The model was evaluated on the UNSW-NB15 dataset, which contains nine attack types, and a Pearson correlation analysis confirmed low redundancy among the selected features. In terms of performance, GSGW-DT-RF achieved 99.41% accuracy with the lowest FPR of 0.03%, GSGW-DT-AB achieved 99.36% accuracy with the highest precision of 99.94%, and GSGW-DT-DT achieved 99.02% accuracy with an FPR reduced to 0.24%.

More *et al.* [31] built IDS models using the UNSW-NB15 dataset, applying exploratory data analysis with correlation filtering and XGBoost feature importance for effective feature selection. To enhance the dataset, they introduced a derived feature, network_bytes=sbytes+dbytes, and performed preprocessing through categorical encoding and standard scaling. The study evaluated logistic regression, linear SVM, DT, RF, and gradient boosted (XGBoost), with optimization performed using grid search and tuned hyperparameters. Results showed that feature selection significantly enhanced detection performance and reduced false alarms. In particular, RF with feature selection achieved the highest test accuracy of 99.45% along with an F1-score of 0.9965 and the lowest false alarm rate (FAR) of 1.94%. XGBoost followed closely, reaching 99.41% accuracy with an F1-score of 0.9966 and FAR of 2.33%. DT achieved around 99.04% accuracy but suffered from a higher FAR of 5.75%. The linear SVM achieved 98.79% accuracy, although it attained the highest test AUC of 0.9916, while logistic regression reached 98.93% accuracy with an AUC of 0.9829.

Although studies using filter/wrapper selection, hybrid metaheuristics, and ensembles on public benchmarks (TON_IoT, BoT-IoT, NSL-KDD, CICIDS2017, and UNSW-NB15) report strong accuracy [12], [24]–[31], three key gaps remain. First, the DA and BA are rarely examined as stand-alone feature selectors in AI-based firewall settings. Second, the simple union of DA- and BA-selected features—expected to capture complementary discriminative signals—has not been systematically evaluated under a single, controlled pipeline. Third, prior work often prioritizes headline accuracy while under-reporting efficiency (selected-feature count and training/inference time) on lightweight, deployable classifiers (AdaBoost, KNN, and NB). Motivated by these gaps, we study DA and BA as dedicated selectors and assess their union-based feature set within an ML-based firewall, jointly evaluating detection performance and efficiency. The novelty and main contributions of this article can be summarized as follows:

- Introduction of a new feature selection strategy, the DA∪BA feature selection method, which unifies the DA and BA to capture complementary feature subsets.
- Integration of the DA∪BA feature selection method into a ML–based firewall and evaluation using the UNSW-NB15 dataset.
- Execution of a comprehensive experimental assessment (accuracy, precision, recall, F1-score, and computational cost), demonstrating that the DA∪BA feature selection method achieves perfect detection performance (100% with AdaBoost) while remaining efficient for real-time deployment.
- Demonstration that the union strategy enhances both detection accuracy and operational efficiency, providing a practical pathway for strengthening next-generation firewall systems.

2. METHOD

This section presents the dataset and preprocessing pipeline, including label encoding for categorical fields and normalization to standardize feature scales. We then apply the DA and BA as wrapper-based feature selectors under a unified setup. Finally, we introduce our union strategy, which merges the DA-and BA-selected attributes to yield a compact and discriminative subset for subsequent modeling. Figure 1 shows the ML-based firewall pipeline with DA/BA feature selection.

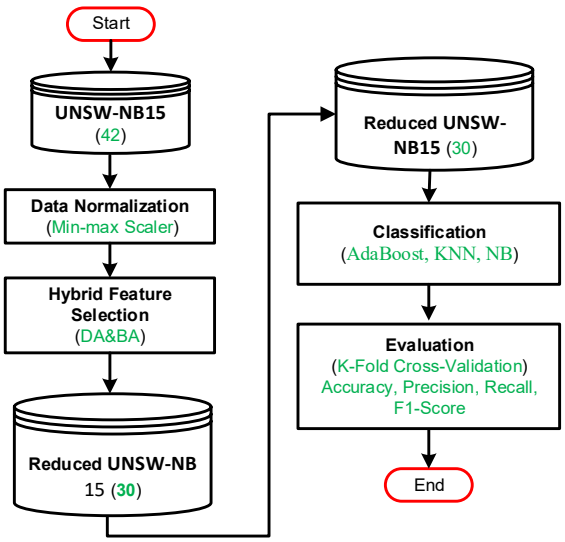


Figure 1. Workflow the key components of the proposed ML-based firewall pipeline

2.1. Dataset preparation

The proposed ML-based firewall aims to prevent attacks on the network traffic. Therefore, the model should be assessed using a credible dataset such as the UNSW_NB15 dataset. Several research studies have used the UNSW_NB15 dataset for evaluation purposes. This is because the UNSW_NB15 dataset contains large amounts of data distributed over normal and attack network traffic. In addition, the attack traffic varies and is distrusted over several types of common attacks, including Analysis, Backdoor, DoS, Exploits, Fuzzers, Generic, Reconnaissance, Shellcode, and Worms. Moreover, the kind of traffic is distributed over many of the network's main services, including DHCP, DNS, FTP, and HTTP. Finally, the UNSW_NB15 dataset comprises 68,661 records of normal (32734 records) and attack (35927 records) network traffic, along with 42 features. The number of records is large enough to evaluate the proposed ML-based firewall reliably. In addition, the 42 features are diverse and well represent the key characteristics that can be used to distinguish between normal and attack network traffic [32]–[34]. Table 1 describes the features of the UNSW-NB15 dataset.

As shown in Table 1, several features within the UNSW-NB15 dataset contain non-numeric data. This data is not acceptable in any ML-based system, as most ML algorithms work only with numeric data. Therefore, all non-numeric data within the UNSW-NB15 dataset must be converted to numeric values. ML provides several algorithms that can be used to numerize non-numeric data. Label encoding is one of the most reliable and most used algorithms. The Label-encoding algorithm encodes the non-numeric data by replacing each different value with a unique integer [33], [35]. For example, the service feature contains the following values: DHCP, DNS, FTP, FTP-data, HTTP, IRC, POP3, radius, SMTP, SNMP, SSH, SSL, and others (-). The Label-encoding algorithm replaces the service feature values with the following: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, and 13, respectively.

Moreover, several features in the UNSW-NB15 dataset contain widely varying values, as shown in Table 1. Again, these values will bias the ML algorithms toward the larger values, resulting in inaccurate results. Therefore, the wide range of values must be narrowed. One of the algorithms widely used by ML systems to normalize a wide range of values is the min-max algorithm. The min-max algorithm normalizes the wide range of values to be between 0 and 1 [36], [37]. The min-max algorithm can be calculated using (1):

$$X_{new} = \frac{(X_{.}+X_{min})}{(X_{max}+X_{min})} \quad (1)$$

Table 1. Overview of the features in the UNSW-NB15 dataset

#	Feature	Type	Description	Min. value	Max. value
1	dur	Numeric	Duration of the network connection.	0	59.996017
2	proto	Nominal	Protocol used in the network flow (e.g., TCP, UDP, ICMP).	N/A	N/A
3	service	Nominal	Network service associated with the connection (e.g., HTTP, FTP).	N/A	N/A
4	state	Nominal	State and status of the network connection.	N/A	N/A
5	spkts	Numeric	Number of packets sent by the source.	1	512
6	dpkts	Numeric	Number of packets sent by the destination.	0	800
7	sbytes	Numeric	Number of source bytes transferred.	65	44196
8	dbytes	Numeric	Number of destination bytes transferred.	0	60800
9	rate	Numeric	Ethernet data rates transmitted and received.	0	1000000
10	sttl	Numeric	Time-to-live value of packets from the source.	31	255
11	dttl	Numeric	Time-to-live value of packets from the destination.	0	254
12	sload	Numeric	Average source bits per second.	0	2.224E+09
13	dload	Numeric	Average destination bits per second.	0	818390.81
14	sloss	Numeric	Number of lost packets from the source.	0	2
15	dloss	Numeric	Number of lost packets from the destination.	0	6
16	sinpkt	Numeric	Average time between packets sent by the source.	0	13992.212
17	Dinpkt	Numeric	Average time between packets sent by the destination.	0	13992.285
18	sjit	Numeric	Jitter between source packets.	0	19787.972
19	djit	Numeric	Jitter between destination packets.	0	19788
20	swin	Numeric	Source TCP window size.	0	255
21	stcpb	Numeric	Source TCP base sequence number.	0	4277446941
22	dtcpb	Numeric	Destination TCP base sequence number.	0	4088422545
23	dwin	Numeric	Destination TCP window size.	0	255
24	tcprtt	Numeric	Round trip time for TCP packets.	0	0.265426
25	synack	Numeric	Time to establish a TCP connection (SYN-ACK time).	0	0.176175
26	ackdat	Numeric	Time for acknowledgment packets (ACK-DATA time).	0	0.136995
27	smean	Numeric	Mean packet size from the source.	50	834
28	dmean	Numeric	Mean packet size from the destination.	0	865
29	trans_depth	Numeric	Transaction depth, indicating the HTTP request/response chain level.	0	131
30	response_body_len	Numeric	Response body length in bytes.	0	5242880
31	ct_srv_src	Numeric	Number of connections to the same service by the source.	1	59
32	ct_state_ttl	Numeric	Number of connections with the same state and time-to-live.	0	6
33	ct_dst_ltm	Numeric	Number of connections to the same destination over time.	1	59
34	ct_src_dport_ltm	Numeric	Number of connections from the source to the same destination port over time.	1	59
35	ct_dst_sport_ltm	Numeric	Number of connections to the destination from the same source port over time.	1	38
36	ct_dst_src_ltm	Numeric	Number of connections to the destination from the same source over time.	1	59
37	is_ftp_login	Numeric	Binary indicator for successful FTP login.	0	1
38	ct_ftp_cmd	Numeric	Number of FTP commands in the flow.	0	2
39	ct_flw_http_mthd	Numeric	Number of HTTP request methods in the flow (e.g., GET, POST).	0	16
40	ct_src_ltm	Numeric	Number of connections from the same source over time.	1	60
41	ct_srv_dst	Numeric	Number of connections to the same service by the destination.	1	59
42	is_sm_ips_ports	Numeric	Binary indicator for same source/destination IP and port.	0	1

Finally, the UNSW-NB15 dataset contains 42 features. Several features are highly important in distinguishing between normal and attack network traffic. However, some of the 42 features are less important or not important in distinguishing between normal and attack network traffic. These features degraded the performance of an ML system and increased processing time due to the need to handle large amounts of unimportant data. Therefore, only the most important features should be retained in the dataset, while the other features should be excluded. In this work, the DA and BA will be combined to select features that are highly important in distinguishing between normal and attack network traffic [19], [20].

The DA is a swarm-based optimization algorithm inspired by the static and dynamic behaviors of dragonflies during foraging and migration. In feature selection, DA mimics the movement of dragonflies by simulating social interactions, such as attraction, repulsion, and alignment. These behaviors help the algorithm explore the feature space (diversity) and exploit promising regions (intensity) to identify the most relevant subset of features. Algorithm 1 shows the pseudocode of the DA. The BA is a metaheuristic optimization algorithm inspired by the echolocation behavior of bats. In feature selection, BA utilizes this echolocation mechanism to explore the feature space by emitting virtual "sound waves" and adjusting the frequency, loudness, and pulse rates to identify the optimal feature subset. Bats initially explore globally and gradually exploit local optima by fine-tuning their movements. Algorithm 2 shows the pseudocode of the BA algorithm. Table 2 compares and contrasts the DA and BA in feature selection [19], [20], [38], [39].

Algorithm 1. DA

```
Input: Dataset D with n features, population size N, max iterations T
Output: Best feature subset S_best
1.  Initialize N dragonflies with random binary vectors Xi (length n).
2.  Initialize velocities Vi for each Xi.
3.  Evaluate fitness of each Xi using: Fitness = (α × classification error) + (β ×
    (#selected features / n)).
4.  Identify food source (best solution) and enemy (worst solution).
5.  For t = 1 to T:
6.     For each dragonfly Xi:
7.         a. Compute swarming factors: Separation (Si),
    Alignment (Ai), Cohesion (Ci), Attraction to food (Fi),
    Distraction from enemy (Ei).
8.         b. Update velocity Vi: Vi = w·Vi + s·Si + a·Ai + c·Ci + f·Fi + e·Ei.
9.         c. Update position: Xi = Xi + Vi.
10.        d. Convert Xi to binary using a transfer function (e.g., sigmoid).
11.        e. Evaluate new fitness and update food/enemy if needed.
12. Return S_best = food source.
```

Algorithm 2. BA

```
Input: Dataset D with n features, population size N, max iterations T
Output: Best feature subset S_best
1.  Initialize N bats with random binary vectors Xi (length n).
2.  Initialize velocities Vi, pulse rates ri, loudness Ai, and frequencies fi.
3.  Evaluate fitness of each Xi.
4.  Identify global best solution G.
5.  For t = 1 to T:
6.     For each bat i:
7.         a. Update frequency fi and velocity Vi.
8.         b. Update position Xi = Xi + Vi.
9.         c. Convert Xi to binary using a transfer function.
10.        d. If rand > ri, generate local solution around G.
11.        e. Evaluate new fitness.
12.        f. If rand < Ai and fitness improved:
13.            - Accept new solution, update Ai and ri.
14.            - Update global best G if better solution is found.
15. Return S_best = G.
```

Table 2. Comparison between the DA and BA

Aspect	DA	BA
Inspiration	Based on the behavior of dragonflies, focusing on their swarming dynamics for foraging and migration.	Inspired by the echolocation ability of bats to detect prey and navigate in the dark.
Exploration vs. exploitation	Balances exploration and exploitation through swarming behaviors (attraction, repulsion, and alignment).	Balances exploration and exploitation by adjusting frequency and loudness of echolocation pulses.
Search strategy	Uses group dynamics: individuals move towards the best solutions based on social and cognitive factors.	Combines global search (based on frequency) and local search (fine-tuning via loudness and pulse rates).
Convergence speed	May exhibit slower convergence due to a strong focus on exploration in the early stages.	Typically, faster convergence, especially when local optima are close to the global optimum.
Tuning complexity	Requires careful adjustment of parameters like step size, inertia weights, and neighborhood radius.	Parameters such as pulse rate and loudness require fine-tuning but are relatively simpler than DA.
Scalability	Scales well for high-dimensional datasets but may require more computational resources due to complex swarm dynamics.	Scales efficiently but may struggle with maintaining diversity in very high-dimensional feature spaces.

DA and BA often outperform other swarm-based optimizers for IDS feature selection because of their complementary search strategies. DA maintains population diversity through swarming dynamics (separation, alignment, cohesion, attraction, and distraction), which improves exploration and avoids local optima in high-dimensional spaces [19], [38]. BA, by contrast, employs echolocation-based frequency tuning and adaptive pulse rates to balance global and local search, resulting in faster convergence and more compact feature subsets [20], [39]. Recent IDS studies support these advantages, showing that DA is effective for firewall log analysis and BA achieves dimensionality reduction with high detection accuracy [38], [40]. Consistent with these reports, our union DA∪BA method leverages both exploration and exploitation to deliver robust intrusion detection performance.

2.2. Proposed machine learning-based firewall model

The proposed ML-based firewall experiment will be conducted using the UNSW-NB15 dataset to assess its performance. The UNSW-NB15 dataset has undergone several operations before it can be used for

training and testing. First, all non-numeric data in the UNSW-NB15 dataset have been converted to a numeric form using the Label-encoding algorithm mentioned in section 2.1. Then, the min-max algorithm is applied to scale down the large differences in values in the UNSW-NB15 dataset, as described in section 2.1. The final step of the UNSW-NB15 dataset preparation process involves identifying features that effectively differentiate between normal and attack traffic using the proposed feature selection method, which integrates the DA and BA. The proposed feature selection method is highlighted in section 2.2.1. At this stage, the UNSW-NB15 dataset is ready for use in training and testing the proposed ML-based firewall. Three of the most efficient and widely used ML classifiers will be used at this stage of the training and testing. The three classifiers are AdaBoost, KNN, and NB.

2.2.1. The proposed feature selection method

Feature selection is a crucial step in establishing an effective IDS that differentiates normal and attack traffic. Therefore, applying an efficient feature selection method to the proposed ML-based firewall will confer the following advantages. One of the main benefits is increasing the firewall's performance by enhancing its ability to differentiate between normal and attack traffic, focusing on key features rather than irrelevant ones. The second and equally important benefit is that using this approach simplifies the training of the firewall, decreasing its training time and incremental complexity. Therefore, this work will employ an improved feature selection method that incorporates the DA and BA algorithms. Tables 3 and 4 present the DA and BA parameter settings, respectively.

Table 3. Parameter settings of the DA algorithm

Parameter	Value	Brief description
Population size	24	Number of candidate feature masks evaluated per iteration.
Max iterations	100	Optimization steps before termination.
Separation weight	0.2	Repels agents to prevent crowding.
Alignment weight	0.2	Aligns agent movement directions.
Cohesion weight	0.2	Pulls agents toward the group center.
Food attraction	2	Bias toward the current best solution.
Enemy distraction	2	Bias away from poor regions of the search space.
Initial neighborhood radius	0.4	Starting neighborhood size (decays during search).
Initial step size	0.5	Initial movement amplitude (decays during search).
Binary threshold	0.55	Converts continuous positions to a 0/1 feature mask (encourages parsimony).

Table 4. Parameter settings of the BA

Parameter	Value	Brief description
Population size	24	Number of bats (candidate masks) per iteration.
Max iterations	100	Optimization steps before termination.
Frequency range	[0, 2]	Controls exploration step size.
Initial loudness	0.9	Acceptance likelihood for new solutions (decays over time).
Loudness decay	0.9	Exponential decay factor for loudness.
Initial pulse rate	0.1	Initial probability of local search (increases over time).
Pulse growth	0.9	Growth factor governing pulse-rate increase.
Local search scale	0.01	Step size for random walk near the best solution.
Binary threshold	0.55	Maps velocities/positions to a 0/1 feature mask (promotes compact subsets).

The proposed DA∪BA feature selection method is grounded in the mathematical concept of the union operation from set theory. The union combines elements from two or more sets while eliminating duplicates. In our approach, the DA and BA algorithms are applied in parallel to the preprocessed UNSW-NB15 dataset, producing two independent feature subsets. Let D and B represent the subsets returned by DA and BA, respectively. The final subset is then defined as the union: $U=D \cup B=\{j| j\in D \text{ or } j \in B\}$. This means that any feature selected by either algorithm is retained, with duplicates removed. Classifiers are then trained and evaluated directly on U.

Figure 2 illustrates the proposed DA∪BA feature selection method. First, the DA identifies a subset of key features from the UNSW-NB15 dataset, represented as: DA={1, 3, 5, 6, 8, 9, 10, 13, 14, 15, 17, 18, 19, 20, 22, 24, 25, 26, 27, 29, 30, 31, 34, 35, 37, 38, 39, 40}. Next, the BA algorithm produces another subset of key features: BA={4, 10, 13, 14, 28, 29, 38, 39}. Finally, the two subsets are combined using the union function, yielding: DA∪BA={1, 3, 4, 5, 6, 8, 9, 10, 13, 14, 15, 17, 18, 19, 20, 22, 24, 25, 26, 27, 28, 29, 30, 31, 34, 35, 37, 38, 39, 40}.

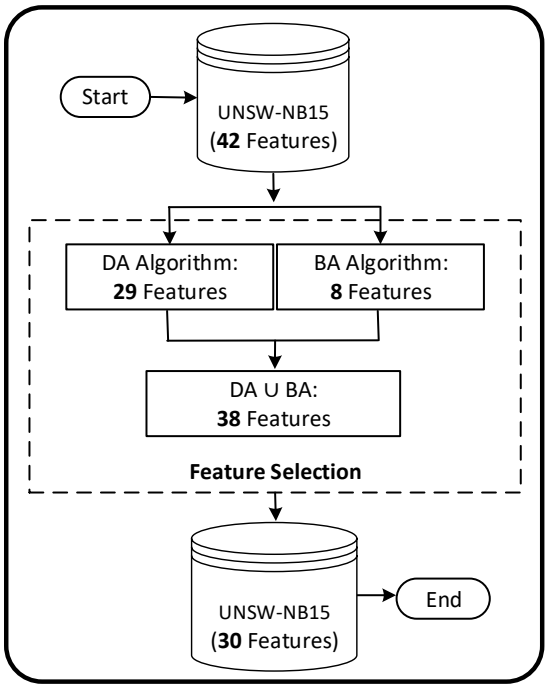


Figure 2. Workflow of the proposed DA∪BA feature selection method

Using the union function to combine the features selected by the DA and BA algorithms has several advantages. First, the union reduces the possibility of missing key features by producing a more diverse feature set. Second, the union of features provides a more comprehensive feature set, which improves accuracy and leads to better generalization. In addition, the union allows both global and local feature interactions to be considered, which helps to handle complex datasets [28], [37]. Finally, the DA algorithm identifies broad patterns in the feature space, while the BA algorithm ensures precision by refining solutions in focused areas [12], [13]. Therefore, the union of DA and BA offers a more comprehensive approach that leverages the two algorithms.

2.2.2. Classification

The final stage of the proposed ML-based firewall is to classify network traffic as normal or malicious. We trained and evaluated three widely used classifiers—AdaBoost, KNN, and NB—on the preprocessed data. AdaBoost, KNN, and NB were chosen to represent three complementary learning paradigms. AdaBoost demonstrates the power of ensemble boosting, KNN captures local decision boundaries with instance-based learning, and NB provides a fast probabilistic Baseline. This mix ensures a balanced evaluation of the DA∪BA FS method in terms of accuracy, simplicity, and efficiency. Table 5 shows the properties of the AdaBoost, KNN, and NB algorithms [20], [41]–[43]. The fixed hyperparameters of the three classifers were: AdaBoost (n_estimators=50, learning_rate=1.0, base DecisionTreeClassifier (max_depth=1), algorithm="SAMME.R"), KNN (n_neighbors=5, weights="uniform", metric="minkowski"), and NB (alpha=1.0, fit_prior=True, class_prior=None).

Table 5. Key properties of the AdaBoost, KNN, and NB algorithms

Aspect	AdaBoost	KNN	NB
Algorithm type	Ensemble learning, boosting algorithm	Instance-based learning (lazy learner)	Probabilistic classifier, based on Bayes' theorem
Learning approach	Combines weak classifiers (e.g., decision stumps)	Non-parametric, no explicit learning phase	Parametric, relies on prior probabilities and likelihood
Working mechanism	Iteratively improves weak learners by focusing on misclassified samples	Classifies based on the majority class of KNN	Assumes independence among features and calculates probabilities
Strengths	- Focuses on hard-to-classify samples - High accuracy with weak learners	- Simple and intuitive - No training time required	- Fast and efficient for high-dimensional data - Works well for small datasets
Computational complexity	Moderate to high (depends on the No. of iterations and weak learners)	High during inference, as distances to all training samples are calculated	Low, involves basic probabilistic computations

To ensure robust evaluation, the UNSW-NB15 dataset was split using stratified 5-fold cross-validation. In each fold, 80% of the data was used for training and 20% for testing, preserving the ratio of normal to attack records. All preprocessing steps, including label encoding and normalization, were fitted only on the training portion of each fold and then applied to the test portion to avoid data leakage. Model performance was averaged across folds to reduce variance due to random partitioning. These procedures minimize the risk of overfitting and ensure that reported results reflect the generalization ability of the classifiers under the DA∪BA method [6], [10].

3. RESULTS AND DISCUSSION

Experiments were conducted using Python 3.12, scikit-learn 1.4 or later, and NumPy 1.26 or later on Ubuntu 22.04 LTS. The hardware environment consisted of an Intel Core i9-14900KS CPU with 32 GB of DDR5 RAM. The proposed ML-based firewall will be evaluated using four different metrics. These metrics are the firewall accuracy, firewall precision, firewall recall, and firewall F1-score. Table 6 clarifies these five metrics. These metrics are computed based on the confusion matrix CN. The four elements of the CN, in the case of the proposed ML-based firewall, are true positive (F_{tp}), true negative (F_{tn}), false positive (F_{fp}), and false negative (F_{fn}). Accuracy, recall, precision, and F1-score are calculated based on these elements using (2)-(5), respectively [44]–[47].

$$\text{Accuracy} = \frac{(F_{tp}+F_{tn})}{(F_{tp}+F_{tn}+F_{fp}+F_{fn})} \quad (2)$$

$$\text{Recall} = \frac{F_{tp}}{(F_{tp}+F_{fn})} \quad (3)$$

$$\text{Precision} = \frac{F_{tp}}{(F_{tp}+F_{fp})} \quad (4)$$

$$\text{F1} - \text{score} = 2 \times \frac{F_{pr} \times F_{re}}{F_{pr}+F_{re}} \quad (5)$$

Table 6. Description of the evaluation metrics

Metric	Definition	Purpose
Accuracy	The proportion of total instances that were correctly classified by the model.	Measure overall performance on balanced datasets.
Precision	The proportion of correctly predicted positive instances among all instances predicted as positive.	Assess model's ability to minimize false positives.
Recall	The proportion of correctly predicted positive instances out of all actual positive instances.	Assess model's ability to minimize false negatives.
F1-score	The harmonic mean of precision and recall.	Balance precision and recall, especially for imbalanced datasets.

3.1. Classification results

Figure 3 presents the classification accuracy for AdaBoost, KNN, and NB under four feature–selection settings: DA∪BA (DA∪BA), DA, BA, and Baseline (NoFs). The models achieve uniformly high accuracy. For AdaBoost, the DA∪BA method achieves the best result at 100.00%, surpassing DA and BA at 99.99% and the Baseline at 99.98%, i.e., gains of 0.01–0.02 percentage points. For KNN, the union, DA, and BA tie for the highest accuracy at 99.99%, each edging the Baseline at 99.98% by 0.01 points. For NB, the union and DA are co-best at 99.84%, outperforming BA at 99.78% by 0.06 points and the Baseline at 99.74% by 0.10 points. Overall, the union strategy is best for AdaBoost and co-best for KNN and NB; DA matches the union on KNN and NB; BA remains competitive but trails slightly on NB, while the Baseline is consistently lowest. The consistently high values indicate strong separability between malicious and benign traffic with minimal misclassification.

Figure 4 presents the classification recall for AdaBoost, KNN, and NB under four feature–selection settings: DA∪BA, DA, BA, and Baseline (NoFs). The models achieve uniformly high recall. For AdaBoost, the union and DA are co-best at 100.00%, each exceeding BA and Baseline at 99.99% by 0.01 percentage points. For KNN, the union achieves the highest recall at 100.00%, surpassing DA, BA, and Baseline (all 99.99%) by 0.01 points. For NB, the Baseline achieves the best recall at 100.00%, surpassing the union at 99.91% by 0.09 points, DA at 99.90% by 0.10 points, and BA at 99.83% by 0.17 points. Overall, the union strategy is best for AdaBoost and KNN, while the Baseline leads for NB; BA remains competitive but trails

slightly behind NB. The consistently high recall indicates strong sensitivity with very low false-negative rates (few missed attacks).

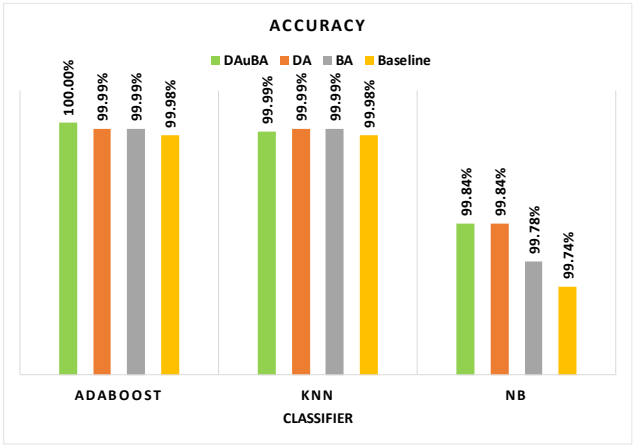


Figure 3. Accuracy of the proposed ML-based firewall

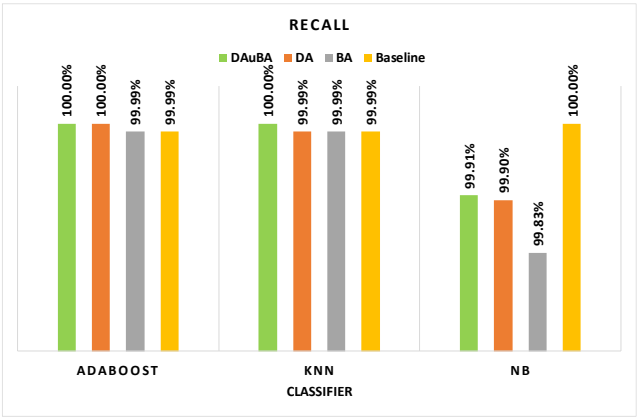


Figure 4. Recall of the proposed ML-based firewall

Figure 5 presents the classification precision for AdaBoost, KNN, and NB under four feature–selection settings: DAUBA, DA, BA, and Baseline (NoFs). The models achieve uniformly high precision. For AdaBoost, the DAUBA method achieves the best result at 100.00%, surpassing DA and BA at 99.99% and the Baseline at 99.97%, i.e., gains of 0.01–0.03 percentage points. For KNN, DA, and BA, the highest precision is achieved at 99.99%, surpassing the union at 99.98% by 0.01 points and the Baseline at 99.97% by 0.02 points. For NB, the union attains the best precision at 99.91%, topping DA and BA at 99.90% by 0.01 points and the Baseline at 99.53% by 0.38 points. Overall, the union strategy is best for AdaBoost and NB, and remains competitive for KNN. DA and BA are co-best for KNN, while the Baseline is consistently the lowest. The consistently high precision indicates strong positive predictive value with very low false-positive rates (few benign flows flagged as attacks).

Figure 6 presents the classification F1-score for AdaBoost, KNN, and NB under four feature–selection settings: DAUBA, DA, BA, and Baseline (NoFs). The models achieve uniformly high F1-scores. For AdaBoost, the DAUBA method achieves the best result at 100.00%, surpassing DA and BA at 99.99% and the Baseline at 99.98%, i.e., gains of 0.01–0.02 percentage points. For KNN, the union, DA, and BA tie for the highest F1-score at 99.99%, each edging the Baseline at 99.98% by 0.01 points. For NB, the union attains the best F1-score at 99.91%, topping DA at 99.90% by 0.01 points, BA at 99.87% by 0.04 points, and the Baseline at 99.76% by 0.15 points. Overall, the union strategy is best for AdaBoost and NB, and is co-best for KNN. DA and BA remain competitive for KNN, while the Baseline is consistently the lowest. The consistently high F1-scores indicate a strong balance between precision and recall with minimal misclassification.

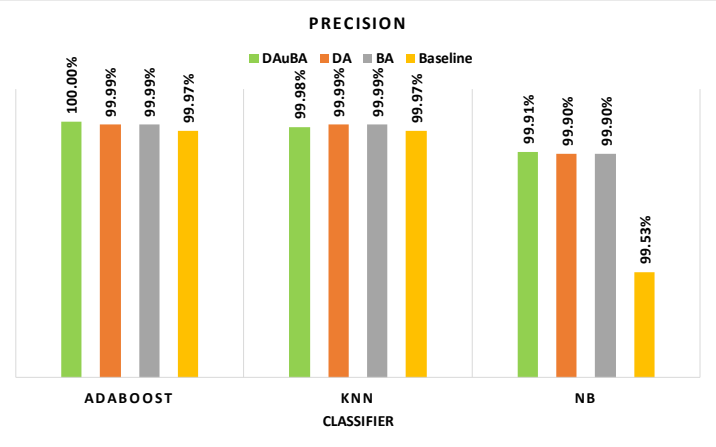


Figure 5. Precision of the proposed ML-based firewall

Figure 6. F1-score of the proposed ML-based firewall

Table 7 summarizes accuracy, precision, recall, and F1-score for all methods across the three classifiers. To make the differences explicit, Table 8 reports union-versus-comparator gaps (in percentage points) in accuracy, precision, recall, and F1-score for each classifier (values are Union − Other; positive values indicate that the union is better). Overall, the union strategy consistently improves or matches the strongest alternative on accuracy and F1—notably for AdaBoost (e.g., +0.02 pp accuracy and +0.02 pp F1 over the Baseline) and NB (e.g., +0.10 pp accuracy and +0.15 pp F1 over the Baseline), while it ties the best methods for KNN on accuracy/F1 and yields a +0.01 pp recall gain over all comparators. In terms of precision, the union is best for AdaBoost and NB (e.g., +0.03 pp and +0.38 pp over their Baselines, respectively), but trails KNN by 0.01 pp relative to DA/BA. For recall, the union is the best for AdaBoost and the second-best for KNN, whereas the Baseline achieves the highest NB recall (by 0.09 pp over the union) at the cost of markedly lower precision and F1.

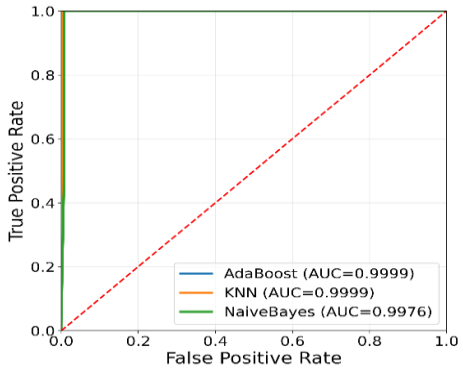
Table 7. Summary of classification performance across classifiers and feature-selection methods

	Accuracy (%)			Precision (%)			Recall (%)			F1-Score (%)		
Method	AdaBoost	KNN	NB	Ada Boost	KNN	NB	AdaBoost	KNN	NB	AdaBoost	KNN	NB
DAUBA	100	99.99	99.84	100	99.98	99.91	100	100	99.91	100	99.99	99.91
DA	99.99	99.99	99.84	99.99	99.99	99.9	100	99.99	99.9	99.99	99.99	99.9
BA	99.99	99.99	99.78	99.99	99.99	99.9	99.99	99.99	99.83	99.99	99.99	99.87
Baseline	99.98	99.98	99.74	99.97	99.97	99.53	99.99	99.99	100	99.98	99.98	99.76

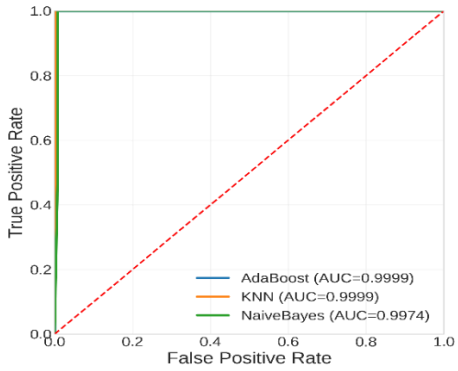
Table 8. Pairwise percentage-point differences between the DAUBA method and comparators (DA, BA, and Baseline)

Metric	Classifier	Union (%)	Δ vs DA (pp)	Δ vs BA (pp)	Δ vs Baseline (pp)
Accuracy	AdaBoost	100	0.01	0.01	0.02
	KNN	99.99	0	0	0.01
	NB	99.84	0	0.06	0.1
Precision	AdaBoost	100	0.01	0.01	0.03
	KNN	99.98	-0.01	-0.01	0.01
	NB	99.91	0.01	0.01	0.38
Recall	AdaBoost	100	0	0.01	0.01
	KNN	100	0.01	0.01	0.01
	NB	99.91	0.01	0.08	-0.09
F1-score	AdaBoost	100	0.01	0.01	0.02
	KNN	99.99	0	0	0.01
	NB	99.91	0.01	0.04	0.15

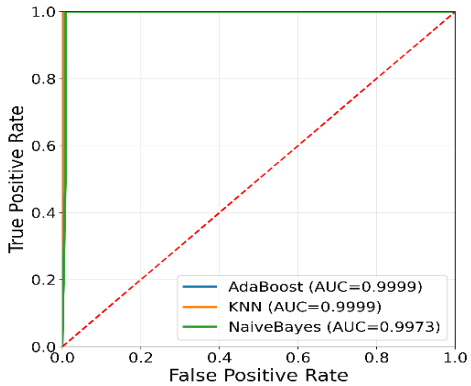
Figures 7-9 show the ROC–AUC curves for AdaBoost, KNN, and NB, using, respectively, DA, BA, and the union DAUBA feature selection methods. In all cases, the curves cluster near the top-left corner—well above the no-skill diagonal—indicating very high TPRs at negligible FPRs and thus strong attack–benign separability for the proposed ML-based firewall. In Figure 7 (DA), AUCs are ≈0.9999 (AdaBoost), ≈0.9998 (KNN), and ≈0.9976 (NB), with AdaBoost marginally leading and NB slightly lower but still excellent. Figure 8 (BA) shows a similarly tight clustering, with AdaBoost ≈0.9999, KNN ≈0.9999 (essentially tied), and NB ≈0.9974. Figure 9 (DAUBA) maintains comparable AUCs—AdaBoost ≈0.9999, KNN ≈0.9999, NB ≈0.9973—and equally steep rises at very low FPRs, suggesting saturated ranking performance across classifiers. Taken together, these ROC–AUC results reflect the discriminative strength of all three feature sets (DA, BA, and DAUBA) and indicate that the firewall can be thresholder to achieve near-perfect detection with minimal false alarms; the union and BA selections provide the most uniform classifier behavior, while DA remains virtually indistinguishable in AUC.



Figures 7. ROC curves of DA feature selection method



Figures 8. ROC curves of BA feature selection method



Figures 9. ROC curves of DAUBA feature selection method

3.2. Comparison with existing approaches

Figure 10 presents the accuracy of the proposed variants and prior Baselines, all evaluated on the same dataset. Among our methods, AdaBoost-DAUBA achieves the highest accuracy at 100.00%, followed by AdaBoost-DA at 99.99% and AdaBoost-BA at 99.99% (tie). All proposed variants exceed the strongest published reference. Using AdaBoost-DAUBA as a reference point, the margins over prior work are +0.36% versus. [19] 99.64%, +7.20% versus [28] 92.80%, +0.77% versus [29] 99.23%, +0.59% versus [30] 99.41%, and +0.55% versus [31] 99.45%. Even the lower-performing proposed variants (AdaBoost-DA/BA 99.99%) surpass the strongest Baseline ([27] 99.64%) by 0.35 percentage points.

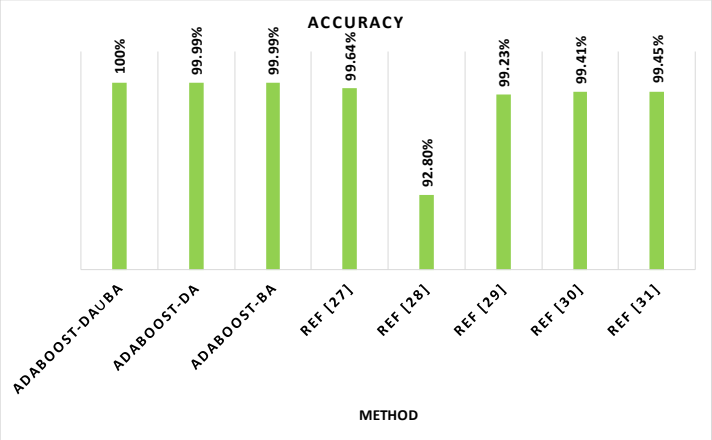


Figure 10. Accuracy of the proposed ML-based firewall compared to the previous works on UNSW-NB15

Figure 11 presents an accuracy comparison between our variants and prior studies evaluated on different datasets. Among the proposed methods, AdaBoost-DAUBA achieves 100.00%, followed by AdaBoost-DA at 99.99% and AdaBoost-BA at 99.99% (tie). All proposed variants exceed external references: taking AdaBoost-DAUBA as the anchor, the margins are +4.00% over [9] (96.00%), +0.50% over [24] (99.50%), +11.03% over [25] (88.97%), +0.20% over [26] (99.80%), and +0.87% over [29] (99.13%). Even the lower-performing proposed variants (AdaBoost-DA/BA 99.99%) surpass the strongest reference ([26] 99.80%) by 0.19%. While cross-study comparisons span different datasets, the consistent margins reinforce the general effectiveness of the proposed bio-inspired feature selection within the proposed method.

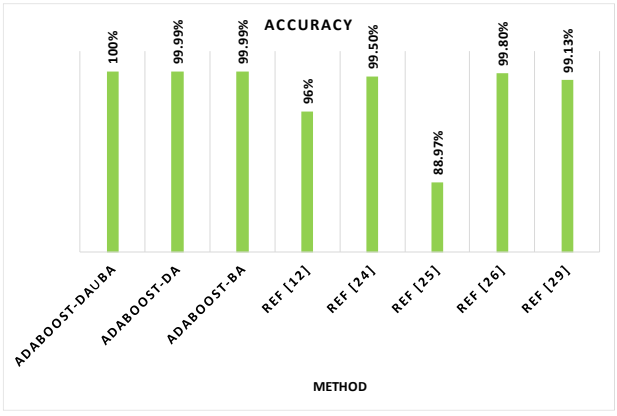


Figure 11. Accuracy of the proposed ML-based firewall compared to the previous works across various datasets

3.3. Computational cost analysis

Tables 9 and 10 report training time (s) and per-sample inference time (µs) for AdaBoost, KNN, and NB under DAUBA, DA, BA, and Baseline. For training, Baseline is fastest for AdaBoost (3.805 s vs 9.885 s

DAUBA, 9.070 s DA, 5.039 s BA) and KNN (0.00639 s vs 0.01168 s DAUBA, 0.01145 s DA, 0.08014 s BA), while BA is fastest for NB (0.01229 s vs 0.04025 s DAUBA, 0.02606 s DA, 0.01691 s Baseline). For per-sample inference, DAUBA gives the lowest AdaBoost latency (31,830.9 µs, ≈31.4 samples/s) versus DA 38,223.7 µs (≈26.2 s/s), BA 38,241.8 µs (≈26.2 s/s), and Baseline 38,816.4 µs (≈25.8 s/s); BA is best for KNN (1,418.5 µs, ≈705 s/s) compared with DAUBA 6,556.1 µs (≈153 s/s), DA 6,671.2 µs (≈150 s/s), and Baseline 4,974.0 µs (≈201 s/s); and BA is also best for NB (896.8 µs, ≈1,115 s/s) versus DAUBA 1,227.0 µs (≈815 s/s), DA 1,047.3 µs (≈955 s/s), and Baseline 1,008.4 µs (≈992 s/s). In a firewall deployment context, these latencies translate directly into real-time processing capacity—e.g., KNN-BA offers ~3.5× the per-core throughput of the Baseline, while AdaBoost-DAUBA lowers latency by ~18% relative to Baseline—quantifying the trade-off between faster retraining (Baseline/BA, depending on classifier) and higher inline throughput (BA for KNN/NB, DAUBA for AdaBoost).

Table 9. Comparison of model training times

Method	DAUBA	DA	BA	Baseline
AdaBoost	9.884596	9.070031	5.038871	3.805394
KNN	0.011681	0.011449	0.080135	0.006386
NB	0.040249	0.026061	0.012288	0.016907

Table 10. Comparison of model inference time

Method	DAUBA	DA	BA	Baseline
AdaBoost	31830.92604	38223.74854	38241.7602	38816.42692
KNN	6556.112815	6671.19787	1418.53607	4974.02036
NB	1226.974065	1047.28321	896.810805	1008.404815

3.4. Key findings and implications

The DAUBA method delivered the strongest results across all classifiers by combining the exploratory search of DA with the fast convergence of BA. It achieved 100.00% accuracy with AdaBoost, 99.99% with KNN, and 99.84% with NB. These values were higher than the Baseline (99.98%, 99.98%, and 99.74%) and improved on DA and BA individually by margins as large as 0.10. Precision increased by up to 0.38, and F1-score improved by as much as 0.15, showing that the union not only enhanced accuracy but also reduced both false alarms and missed intrusions. The improvement stems from the union's ability to preserve complementary features from both algorithms, producing subsets that generalized well across classifiers. Efficiency tests also confirmed practicality: DAUBA required 9.88 seconds for AdaBoost training and reduced AdaBoost inference time to 31,830 microseconds per sample, which is about 18% faster than the Baseline.

These findings indicate that DAUBA enables intrusion detection with near-perfect accuracy while remaining efficient enough for real-time deployment. Even small improvements in accuracy translate into thousands of additional flows correctly identified in high-volume traffic. By combining high detection accuracy with acceptable training and inference times, the method provides a reliable and scalable solution for enhancing firewalls against network attacks.

4. CONCLUSION

This study presents a robust ML-based firewall that leverages advanced feature selection and ML techniques to effectively detect network attacks. By integrating the DA and BA algorithms for feature selection, the model identifies the most relevant features from the UNSW-NB15 dataset, enhancing classification performance. Among the classifiers evaluated, AdaBoost demonstrated superior accuracy at 100%, followed closely by KNN at 99.99% and NB at 99.84%. These results highlight the effectiveness of the proposed approach in improving IDS. The combination of metaheuristic algorithms and ML classifiers offers a promising direction for developing reliable and scalable solutions to address the growing challenges in network security.

This study was limited to the UNSW-NB15 dataset, which may restrict generalizability to other network settings and attack behaviors. There remains a risk of overfitting due to dataset-specific feature patterns. As part of our future work, we plan to evaluate the model on additional benchmarks, such as CICIDS2017 and BoT-IoT, to confirm its robustness and reduce the risk of overfitting. We also intend to integrate deep learning classifiers with the DAUBA feature subsets to further improve adaptability and detection accuracy. Finally, live-deployment trials will be pursued to validate real-time performance and scalability in operational environments.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Mosleh M. Abualhaj	✓	✓			✓				✓	✓		✓		
Sumaya Nabil Al-Khatib		✓		✓		✓				✓				
Nida Al Shafi		✓	✓			✓	✓	✓		✓	✓			
Mohammad O. Hiari	✓			✓	✓			✓	✓		✓			
Mohammad Sh. Daoud		✓		✓		✓		✓		✓	✓			
Mohammed Anbar	✓				✓	✓			✓		✓			
Mahran M. Al-Zyoud	✓		✓		✓	✓			✓		✓			

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are openly available in [Canadian Institute for Cybersecurity] at https://www.unb.ca/cic/datasets/cic-unsw-nb15.html [doi: 10.1109/MilCIS.2015.7348942], reference number [34].

REFERENCES

[1] S. T. Alrawashdeh *et al.*, "Individual and technological factors affecting the adoption of AI-powered remote auditing in the Jordanian banking sector," *Data and Metadata*, vol. 3, Jan. 2024, doi: 10.56294/dm2024.408.

[2] Y. Gao, J. Chen, H. Miao, B. Song, Y. Lu, and W. Pan, "Self-learning spatial distribution-based intrusion detection for industrial cyber-physical systems," *IEEE Transactions on Computational Social Systems*, vol. 9, no. 6, pp. 1693–1702, Dec. 2022, doi: 10.1109/TCSS.2021.3135586.

[3] N. K. Al-Okbi, S. A. Alomari, W. J. Hadi, A. S. Ajrash, A. Smerat, and L. Abualigah, "Defending digital integrity: Advances in media forgery analysis research and cybersecurity development," *SN Computer Science*, vol. 6, no. 8, Oct. 2025, doi: 10.1007/s42979-025-04462-8.

[4] L. Petrosyan, "Estimated cost of cybercrime worldwide 2018-2029," *Statista*, 2024. https://www.statista.com/forecasts/1280009/cost-cybercrime-worldwide. (Accessed Jan. 05, 2025).

[5] J. Hajny, S. Ricci, E. Piesarskas, O. Levillain, L. Galletta, and R. De Nicola, "Framework, tools and good practices for cybersecurity curricula," *IEEE Access*, vol. 9, pp. 94723–94747, 2021, doi: 10.1109/ACCESS.2021.3093952.

[6] M. Ali, M. F. Mushtaq, U. Akram, S. Ramzan, S. Tahir, and M. Ahsan, "An ensemble approach for firewall log classification using stacked machine learning models," *Journal of Computing & Biomedical Informatics*, vol. 8, no. 2, pp. 1–14, 2025, doi: 10.56979/802/20245.

[7] A. H. Abdi *et al.*, "Security control and data planes of SDN: A comprehensive review of traditional, AI, and MTD approaches to security solutions," *IEEE Access*, vol. 12, pp. 69941–69980, 2024, doi: 10.1109/ACCESS.2024.3393548.

[8] T. Chomsiri, X. He, P. Nanda, and Z. Tan, "Hybrid tree-rule firewall for high speed data transmission," *IEEE Transactions on Cloud Computing*, vol. 8, no. 4, pp. 1237–1249, Oct. 2020, doi: 10.1109/TCC.2016.2554548.

[9] S. Bagheri and A. Shameli-Sendi, "Dynamic firewall decomposition and composition in the cloud," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3526–3539, 2020, doi: 10.1109/TIFS.2020.2990786.

[10] A. Saman and S. Rasool, "A feature-level hybrid model approach for automated phishing email detection," *Journal of Computing & Biomedical Informatics*, vol. 9, no. 1, 2025.

[11] A. Q. Saeed *et al.*, "Integrating three machine learning algorithms in ensemble learning model for improving content-based spam email recognition," *Journal of Soft Computing and Data Mining*, vol. 5, no. 2, pp. 188–196, Dec. 2024, doi: 10.30880/jscdm.2024.05.02.014.

[12] H.-Y. Chuang and R.-M. Chen, "Feature selection for malicious detection on industrial IoT using machine learning," *Sensors and Materials*, vol. 36, no. 3, p. 1035, Mar. 2024, doi: 10.18494/SAM4666.

[13] S. Azeem, S. Javed, I. Naseer, O. Ali, and T. M. Ghazal, "A new hybrid PSO-HHO wrapper based optimization for feature selection," *IEEE Access*, vol. 13, pp. 87090–87099, 2025, doi: 10.1109/ACCESS.2025.3570901.

[14] L. Abualigah *et al.*, "Optimizing intrusion detection in wireless sensor networks via the improved chameleon swarm algorithm for

feature selection," *IET Communications*, vol. 19, no. 1, Jan. 2025, doi: 10.1049/cmu2.70029.
[15] R. Masadeh *et al.*, "Narwhal optimizer: A nature-inspired optimization algorithm for solving complex optimization problems," *Computers, Materials & Continua*, vol. 85, no. 2, pp. 3709–3737, 2025, doi: 10.32604/cmc.2025.066797.
[16] M. Ghasemi *et al.*, "Birds of prey-based optimization (BPBO): a metaheuristic algorithm for optimization," *Evolutionary Intelligence*, vol. 18, no. 4, Aug. 2025, doi: 10.1007/s12065-025-01052-8.
[17] A. Zraiqat *et al.*, "Library and readers algorithm (LRA): A novel human-inspired parameter-free metaheuristic for efficient global optimization," *International Journal of Intelligent Engineering and Systems*, vol. 18, no. 9, pp. 526–540, Oct. 2025, doi: 10.22266/ijies2025.1031.34.
[18] A. Al-Ou'n *et al*., "Numbat Optimization Algorithm (NOA): A bio-inspired metaheuristic for solving optimization problems," *International Journal of Intelligent Engineering and Systems*, vol. 19, no. 2, pp. 996–1015, 2026, doi: 10.22266/ijies2026.0228.60.
[19] S. Chen, A. Slowik, Y. Chen, Y.-H. She, and X.-S. He, "Quantum-computing-driven bat algorithm: Balancing exploration–exploitation with predictive mutation and frequency adjustment for optimization," *Expert Systems with Applications*, vol. 283, Jul. 2025, doi: 10.1016/j.eswa.2025.127714.
[20] S. Rani S, R. AbuRukba, and K. El-Fakih, "Optimizing predictive maintenance in industrial IoT cloud using dragonfly algorithm," *IEEE Internet of Things Journal*, vol. 12, no. 17, pp. 36001–36018, Sep. 2025, doi: 10.1109/JIOT.2025.3582671.
[21] N. Tashtoush, A. Al-Ghraibah, S. A. S. Sulaiman, F. A. Amran, S. K. Rahim, and S. N. Harun, "Cancer pain detection based on physiological parameters and machine learning," *Cogent Engineering*, vol. 11, no. 1, Dec. 2024, doi: 10.1080/23311916.2024.2431581.
[22] A. Arabiat and M. Altayeb, "Driving behavior analytics: an intelligent system based on machine learning and data mining techniques," *Bulletin of Electrical Engineering and Informatics*, vol. 14, no. 3, pp. 2055–2065, Jun. 2025, doi: 10.11591/eei.v14i3.9095.
[23] B. Xu *et al.*, "Machine learning-assisted computation of water activity for ionic liquid-based aqueous ternary elements," *Desalination and Water Treatment*, vol. 324, Oct. 2025, doi: 10.1016/j.dwt.2025.101484.
[24] M. S. Habeeb and T. R. Babu, "Coarse and fine feature selection for network intrusion detection systems (IDS) in IoT networks," *Transactions on Emerging Telecommunications Technologies*, vol. 35, no. 4, Apr. 2024, doi: 10.1002/ett.4961.
[25] T. Wisanwanichthan and M. Thammawichai, "A double-layered hybrid approach for network intrusion detection system using combined Naive Bayes and SVM," *IEEE Access*, vol. 9, pp. 138432–138450, 2021, doi: 10.1109/ACCESS.2021.3118573.
[26] R. R. R. Robinson, K. P. A. Madhav, and C. Thomas, "Improved minority attack detection in intrusion detection system using efficient feature selection algorithms," *Expert Systems*, vol. 41, no. 7, Jul. 2024, doi: 10.1111/exsy.13546.
[27] A. S. H. M. Ali, "Detection and prevention of distributed denial of service (DDoS) attacks using metaheuristic and machine learning techniques," *International Journal of Science and Research (IJSR)*, vol. 13, no. 11, pp. 633–642, Nov. 2024, doi: 10.21275/SR241107015558.
[28] 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.
[29] A. Kumar and S. Kumar, "Metaheuristic-enhanced feature selection for high-accuracy intrusion detection in cloud computing," *Procedia Computer Science*, vol. 259, pp. 640–649, 2025, doi: 10.1016/j.procs.2025.04.014.
[30] Y. B. Shuaibu and I. O. Alabi, "Utilizing metaheuristic ensemble feature selection to enhance intrusion detection systems," in *Proceedings of the 38th iSTEAMS Bespoke Conference*, 2024, pp. 243–264.
[31] S. More, M. Idrissi, H. Mahmoud, and A. T. Asyhari, "Enhanced intrusion detection systems performance with UNSW-NB15 data analysis," *Algorithms*, vol. 17, no. 2, Feb. 2024, doi: 10.3390/a17020064.
[32] M. Luqman *et al.*, "Intelligent parameter-based in-network IDS for IoT using UNSW-NB15 and BoT-IoT datasets," *Journal of the Franklin Institute*, vol. 362, no. 1, Jan. 2025, doi: 10.1016/j.jfranklin.2024.107440.
[33] P. Verma, J. G. Breslin, D. O'Shea, N. Mehta, N. Bharot, and A. Vidyarthi, "Leveraging gametic heredity in oversampling techniques to handle class imbalance for efficient cyberthreat detection in IIoT," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 1940–1951, Feb. 2024, doi: 10.1109/TCE.2023.3319439.
[34] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, Nov. 2015, pp. 1–6, doi: 10.1109/MilCIS.2015.7348942.
[35] M. Sarfraz, I. A. Sumra, B. Khalid, and E. Fatima, "AI-driven predictive threat detection and cyber risk mitigation: A survey," *Journal of Computing & Biomedical Informatics*, vol. 8, no. 2, pp. 1–18, 2025, doi: 10.56979/802/2025.
[36] M. U. Javeed, S. M. Aslam, H. A. Sadiqa, A. Raza, M. M. Iqbal, and M. Akram, "Phishing website URL detection using a hybrid machine learning approach," *Journal of Computing & Biomedical Informatics*, vol. 9, no. 1, 2025.
[37] O. Almomani, A. Alsaaidah, A. A. Abu-Shareha, A. Alzaqebah, M. A. Almaiah, and Q. Shambour, "Enhance URL defacement attack detection using particle swarm optimization and machine learning," *Journal of Computational and Cognitive Engineering*, vol. 4, no. 3, pp. 296–308, Feb. 2025, doi: 10.47852/bonviewJCCE52024668.
[38] W. A. H. M. Ghanem *et al.*, "Cyber intrusion detection system based on a multiobjective binary bat algorithm for feature selection and enhanced bat algorithm for parameter optimization in neural networks," *IEEE Access*, vol. 10, pp. 76318–76339, 2022, doi: 10.1109/ACCESS.2022.3192472.
[39] R. A. Gulhane, S. R. Gupta, and M. A. Pund, "Feature optimization using hybrid metaheuristic red deer and dragonfly algorithms for multi-disease prediction," *Multimedia Tools and Applications*, vol. 84, no. 13, pp. 11533–11549, May 2024, doi: 10.1007/s11042-024-19369-4.
[40] S. Khanam, M. Sharif, M. Raza, W. Ishaq, M. Fayyaz, and S. Kadry, "Anomaly recognition in surveillance based on feature optimizer using deep learning," *PLOS One*, vol. 20, no. 5, May 2025, doi: 10.1371/journal.pone.0313692.
[41] Y. Wang *et al.*, "Modeling of concrete-filled PVC tube columns confined with CFRP strips under uniaxial eccentric compression: machine learning and finite element approaches," *Journal of Big Data*, vol. 12, no. 1, Feb. 2025, doi: 10.1186/s40537-024-01058-6.
[42] M. M. Abualhaj, S. N. Al-Khatib, M. Al Zyoud, I. Qaddara, and M. Anbar, "Enhancing intrusion detection system performance using a hybrid of Harris Hawks and whale optimization algorithms," *Engineering, Technology & Applied Science Research*, vol. 15, no. 4, pp. 24354–24361, Aug. 2025, doi: 10.48084/etasr.10919.
[43] A. A. Abu-Shareha, "A framework for diabetes detection using machine learning and data preprocessing," *Journal of Applied Data Sciences*, vol. 5, no. 4, pp. 1654–1667, Dec. 2024, doi: 10.47738/jads.v5i4.363.
[44] M. Hussain, W. Sharif, M. R. Faheem, Y. Alsarhan, and H. A. Elsalamony, "Cross-platform hate speech detection using an attention-enhanced BiLSTM model," *Engineering, Technology & Applied Science Research*, vol. 15, no. 6, pp. 29779–29786, 2025, doi: 10.48084/etasr.13249.

[45] N. M. Alaskar, M. Hussain, S. J. Almheiri, A. Khan, and K. M. Adnan, "Big Data-Driven Federated Learning Model for Scalable and Privacy-Preserving Cyber Threat Detection in IoT-Enabled Healthcare Systems," *Computers, Materials & Continua*, vol. 87, no. 1, pp. 1–10, 2026, doi: 10.32604/cmc.2025.074041.
[46] H. Hussain, C. Chen, M. Hussain, M. Anwar, M. Abaker, A. Abdelmaboud, and Q. Yamid, "Optimised knowledge distillation for efficient social media emotion recognition using DistilBERT and ALBERT," *Scientific Reports,* vol. 15, p. 30104, 2025, doi: 10.1038/s41598-025-16001-9.
[47] M. Zubair *et al*., "An interpretable framework for gastric cancer classification using multi-channel attention mechanisms and transfer learning approach on histopathology images," *Scientific Reports*, vol. 15, no. 1, p. 13087, 2025, doi: 10.1038/s41598-025-97256-0.

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.

Ms. Sumaya Nabil 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 2007. Her research area of interest includes VoIP, congestion control, and cybersecurity data mining and optimization. She can be contacted at email: sumayakh@ammanu.edu.jo.

Ms. Nida Al-Shafi is a lecturer in Al-Ahliyya Amman University. She received her first degree in Telecommunication and Electronic Engineering from Philadelphia University, Jordan, in January 2013 and master degree in Cybersecurity from Al-Ahliyya Amman University, Jordan in January 2024. Her research area of interest includes cybersecurity and telecommunication. She can be contacted at email: n.shafi@ammanu.edu.jo.

Mr. Mohammad O. Hiari is a lecturer in Al-Ahliyya Amman University. He received his first degree in Software Engineering from Philadelphia University, Jordan, in August 2004 and master degree in Computer Science from Al Balqa Applied University, Jordan in February 2016. His research area of interest includes VoIP and cybersecurity data mining and optimization. He can be contacted at email: m.hyari@ammanu.edu.jo.

Mohammad Sh. Daoud received the Ph.D. degree in computer science from De Montfort University, U.K. He is currently an Associate Professor with the College of Engineering, Al Ain University, United Arab Emirates. His research interests include artificial intelligence, swarm systems, secured systems and networks, and smart applications. He can be contacted at email: mohammad.daoud@aau.ac.ae.

Dr. Mohammed Anbar 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.

Dr. Mahran M. Al-Zyoud is an assistant professor of Networks and Cybersecurity at Al-Ahliyya Amman University. He received a B.Sc. in Computer Science and a M.Sc. in Computer Information Systems from the University of Jordan in 2004 and 2012. He received a Ph.D. in Computer Science from the University of Alabama, USA, in 2019. His research interests include data privacy and IoT security. He can be contacted at email: m.zyoud@ammanu.edu.jo.