Journal of Computing & Biomedical Informatics
ISSN: 2710 - 1606

Volume 10 Issue 01
2025

Research Article
https://doi.org/10.56979/1001/2025/1156

Intelligent Firewall for Attack Detection: Integrating Dragonfly and Bat Algorithms with Machine Learning

Ali Al-Allawee[1*], Sultan Aldossary[2], Radhwan M. Abdullah[3], and Lway Faisal Abdulrazak[4]

[1]Department of Computer Science, College of Education for Pure Science, University of Mosul, Mosul, Iraq.
[2]Department of Computer Engineering and Information, College of Engineering in Wadi Alddawasir, Prince Sattam bin Abdulaziz University, Saudi Arabia.
[3]Department of Agricultural Machines and Equipment, College of Agriculture and Forestry, University of Mosul, Mosu 41002, Iraq.
[4]Department of space and UAV Engineering Technologies, Electrical Engineering Technical College, Middle Technical University, Baghdad, Iraq.
[*]Corresponding Author: Ali Al-Allawee. Email: aliabd@uomosul.edu.iq

Received: September 12, 2025 **Accepted:** November 29, 2025

Abstract: The increasing sophistication of cyber threats necessitates the development of advanced attack detection methods capable of handling high-dimensional network traffic data efficiently. This paper introduces an AI-driven firewall model that leverages the Dragonfly Algorithm (DA) and Bat Algorithm (BA) for optimal feature selection, enhancing attack detection accuracy. The proposed approach utilizes the UNSW-NB15 dataset and employs a union-based feature selection strategy, combining the best-selected features from DA and BA to maximize classification performance. Three classifiers— utilize Decision Tree (DT), Support Vector Machine (SVM), and Logistic Regression (LR)—are implemented for attack detection. Experimental results demonstrate that DT achieved 100% accuracy, SVM achieved 99.99% accuracy, while LR achieved 99.94%, confirming the effectiveness of the proposed model. The AI-embedded firewall significantly reduces false positives and enhances detection robustness.

Keywords: Attack Detection; Machine Learning; Feature Selection; Dragonfly Algorithm; Bat Algorithm

1. Introduction

Humankind extensively depends on the digital environment in all facets of life, including work, education, interaction, and even entertainment [1-2]. However, the digital environment has significant weaknesses that intruders exploit. The intruders are utilizing specific tools and methods to initiate intrusion acts in the digital environment. These intrusion acts are causing serious damage to the digital environment [3-55]. A report from 2023 shows that the cost of intrusions into the digital environment has reached 11 trillion USD [6]. Therefore, computer security experts have made great efforts to protect the digital environment and stop or at least reduce the cost of intrusion acts. In recent years, several security tools, including packet sniffers, anti-malware, and firewalls, have been developed to shield the digital environment [7-9].

Firewalls are security systems that employ various methods to protect digital environment networks and endpoints from attacks. Typical firewalls, such as stateless firewalls, are built from simple rules similar to if-else stamens in programming languages to mitigate intrusion. The data that matches the rules is blocked or allowed based on the matching rule [6] [10] [11]. However, the intruders are using advanced instruments and methods that are beyond the capability of typical firewalls. Hence, the typical firewall should be updated to include more advanced techniques that can cope with new intrusion types. The most recent firewalls use AI techniques to prevent intrusion. Particularly, modern firewalls utilize customized Machine Learning (ML) algorithms to analyze traffic and avoid intrusion [12-14].

ML systems learn and investigate the preceding malicious and benign network traffic to stop upcoming attacks [15-16]. Nevertheless, network traffic is massive and requires more effort to learn and investigate utilizing ML systems. Moreover, numerous features of the network traffic are irrelevant to the attacks. Hence, the precision of identifying the attacks by ML systems would be condensed [17-18]. Accordingly, massive network traffic should be lessened using feature selection methods in ML systems. The main goal of feature selection methods is to eliminate irrelevant traffic features and keep only the relevant features. Numerous kinds of algorithms are utilized for feature selection, including metaheuristic methods [19] [20].

In previous years, metaheuristic methods have been utilized in many fields to handle difficult issues [21-22]. One of the fields that have extensively utilized the metaheuristic methods is cybersecurity. Particularly, animal-based metaheuristic methods are widely utilized by researchers to secure the digital environment from intruders [23], [24]. Dragonfly Algorithm (DA) and Bat Algorithm (BA) are metaheuristic methods that are frequently utilized in many fields to handle difficult issues [25]. In this study, the DA and BA methods are utilized to improve the effectiveness of the ML-based firewalls. Specifically, the DA and BA methods are utilized as feature selection methods to determine the essential features of the data that assist in finding intrusions in the digital environment. In addition, the suggested ML-based firewall will utilize Decision Tree (DT), Support Vector Machine (SVM), and Logistic Regression (LR) methods [26-27].

2. Related works

A. M. Aleesa et al. [28] developed a deep learning–based intrusion detection system (IDS) using three models: Artificial Neural Network (ANN), Deep Neural Network (DNN), and Recurrent Neural Network with Long Short-Term Memory (RNN-LSTM). The UNSW-NB15 dataset was preprocessed by replacing missing values with zeros, encoding categorical data numerically, and applying min–max normalization. The data was then split into 70% training, 15% testing, and 15% validation sets. Each model was trained for both binary and multi-class classification, with accuracy as the evaluation metric. For binary classification, the ANN, DNN, and RNN-LSTM achieved accuracies of 99.26%, 99.22%, and 85.42%. For multi-class classification, their accuracies were 97.89%, 99.59%, and 85.38%. The DNN model demonstrated the best overall performance on the UNSW-NB15 dataset.

S. Bagui et al. [29] proposed a hybrid feature selection approach to improve intrusion detection accuracy. The method combines k-means clustering with correlation-based feature selection to identify the most informative attributes, including dur, service, sttl, dttl, and ct_srv_src. To evaluate the approach, two classifiers, Naïve Bayes (NB) and J48, were applied to 8,000 samples from the UNSW-NB15 dataset. Experimental results showed that NB achieved significant performance gains with feature selection; for example, its accuracy in detecting worm attacks rose from 84% to 99%. In contrast, J48 showed only a minor improvement, increasing from 99.59% to 99.94% under the same conditions. These results show that the hybrid feature selection method effectively improves classification performance, especially for probabilistic models such as Naïve Bayes.

Y. B. Shuaibu and I. O. Alabi [30] introduced a hybrid feature selection framework for intrusion detection that combines Binary Gravitational Search Algorithm (BGSA) and Binary Grey Wolf Optimizer (BGWO) using an intersection strategy. Specifically, the approach integrates GS-DT and GW-DT models within a wrapper-based selection process, where a Decision Tree (DT) serves as the evaluator. DT, AdaBoost, and Random Forest (RF) are then used as the final classifiers. The resulting ensemble, GSGW-DT, identified only four optimal features, substantially reducing dimensionality. For data preprocessing, categorical encoding and min–max normalization were employed. Evaluation on the UNSW-NB15 dataset, which covers nine attack types, and Pearson correlation analysis verified minimal redundancy among features. In terms of performance, GSGW-DT-RF achieved 99.41% accuracy with a 0.03% FPR. Similarly, GSGW-DT-AB reached 99.36% accuracy with 99.94% precision, while GSGW-DT-DT attained 99.02% accuracy with a reduced 0.24% FPR.

S. More et al. [31] developed intrusion detection models using the UNSW-NB15 dataset, incorporating exploratory data analysis, correlation filtering, and XGBoost feature importance for efficient feature selection. A new derived feature, network_bytes = sbytes + dbytes, was introduced to enrich the dataset, followed by categorical encoding and standard scaling during preprocessing. The study evaluated multiple classifiers—Logistic Regression, Linear SVM, Decision Tree (DT), Random Forest (RF), and XGBoost—with hyperparameters optimized via grid search. Results demonstrated that feature selection notably improved detection accuracy and reduced false alarms. The RF model achieved the best performance, with 99.45% accuracy, an F1-score of 0.9965, and a FAR of 1.94%. This was followed by XGBoost, which achieved 99.41% accuracy and a FAR of 2.33%. DT, SVM, and Logistic Regression also performed competitively, confirming the effectiveness of the optimized feature selection process.

3. Method

3.1. UNSW-NB 15 Dataset

The UNSW-NB 15 dataset's raw network packets were generated using the IXIA PerfectStorm tool within the Cyber Range Lab at UNSW Canberra, producing a combination of authentic modern normal activities and synthetic contemporary attack behaviors. The tcpdump tool was employed to capture 100 GB of raw traffic, specifically in Pcap file format. The dataset comprises nine categories of attacks: DoS, Fuzzers, Backdoors, Generic, Analysis, Exploits, Shellcode, Reconnaissance, and Worms. The Argus and Bro-IDS tools are utilized, and twelve algorithms are developed to generate features along with the class label. A partition of this dataset was designated as a training set and a testing set, specifically, UNSW_NB15_training-set.csv and UNSW_NB15_testing-set.csv, respectively. The training set comprises 175,341 records, while the testing set contains 82,332 records, categorized into attack and normal types. The UNSW-NB 15 dataset shall consist of 42 features utilized for differentiating between normal and attack network traffic. The 42 features of the UNSW-NB15 dataset are: dur (f1), proto (f2), service (f3), state (f4), spkts (f5), dpkts (f6), sbytes (f7), dbytes (f8), rate (f9), sttl (f10), dttl (f11), sload (f12), dload (f13), sloss (f14), dloss (f15), sinpkt (f16), dinpkt (f17), sjit (f18), djit (f19), swin (f20), stcpb (f21), dtcpb (f22), dwin (f23), tcprtt (f24), synack (f25), ackdat (f26), smean (f27), dmean (f28), trans_depth (f29), response_body_len (f30), ct_srv_src (f31), ct_state_ttl (f32), ct_dst_ltm (f33), ct_src_dport_ltm (f34), ct_dst_sport_ltm (f35), ct_dst_src_ltm (f36), is_ftp_login (f37), ct_ftp_cmd (f38), ct_flw_http_mthd (f39), ct_src_ltm (f40), ct_srv_dst (f41), is_sm_ips_ports (f42) [32],[33],[34].

In this work, the training set and testing set have been combined in one dataset that contains 257,673 records and 42 features. The data type of three of the 42 features is text [32] [34]. The values of these features are converted to numbers as most of the machine learning classifiers readily work with the numerical values. The label encoder mechanism was implemented to convert the text values to numbers. After that, the min-max scaler is used to map the values of the 42 features to the same scale. This is because some classifiers are sensitive to feature magnitude [26] [35], [36]. Finally, the key features will be identified using a union of DA and BA metaheuristic methods. Feature selection helps to reduce the overfitting, improve ML model performance, and reduce training time [7], [25].

The DA is a bio-inspired optimization algorithm that simulates the social and dynamic behaviors of dragonflies during hunting and migration. For feature selection, DA aims to find the optimal subset of features from a dataset by balancing exploration (searching broadly) and exploitation (focusing on the best solutions). The algorithm is based on three key behaviors of dragonflies: attraction to food sources, repulsion from enemies, and alignment with neighbors. The BA is a metaheuristic optimization algorithm inspired by the echolocation behavior of bats. In feature selection, BA is used to identify the most relevant subset of features by mimicking the way bats navigate their environment and locate prey using sound waves. The algorithm is based on loudness and pulse emission rate, balancing global exploration and local exploitation. Echolocation is used to refine the search, allowing the algorithm to focus on promising subsets of features. Table 1 compare and contrast the DA and BA algorithms in feature selection [25] [37-39].

Table 1. Comparison of the DA and BA algorithms.

Aspect	Dragonfly Algorithm (DA)	Bat Algorithm (BA)
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).
Feature Selection Strengths	Captures global patterns and relationships among features due to swarm-based behavior.	Excels at refining feature subsets and finding locally optimal solutions due to adaptive parameters.
Performance on Complex Data	Effective in datasets with highly interdependent features because of social interaction modeling.	Performs well in datasets requiring precise optimization, especially for subtle or less prominent feature sets.
Diversity of Solutions	Maintains high diversity by simulating attraction, alignment, and repulsion behaviors, reducing premature convergence.	May reduce diversity during later stages as it converges toward the global optima, risking premature convergence.
Sensitivity to Local Optima	Less prone to local optima because of swarm dynamics and diverse movement patterns.	Can be sensitive to local optima if the exploration phase is not robustly parameterized.
Flexibility	Flexible in adapting to various optimization problems but may need additional mechanisms for fine-tuning.	Highly adaptable with simple mechanisms for switching between global and local search.

3.2. The Suggested Feature Selection Technique

Choosing the right features is the first step in making an ML-based firewall work well since it has a direct effect on how well the system can separate normal and malicious traffic. A good feature selection method improves the ML-based firewall's performance by focusing learning on relevant features. This makes the training process easier and more accurate. It reduces computational complexity and training time, hence making the ML-based firewall practical and effective [25] [40].

This study proposes an enhanced feature selection methodology that combines the DA and BA algorithms within the mathematical framework of Union set theory. The Union function combines the key features identified by DA and BA without duplication, which produces a comprehensive feature set. The DA algorithm identifies broad patterns in the feature space, selecting features of {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}, while the BA algorithm focuses on precision and refinement, producing a subset of {4, 10, 13, 14, 28, 29, 38, 39}. The Union of these subsets yields a combined feature set: {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}.

This approach of union DA and BA brings several advantages. First, it lessens the possibility of missing pivotal features by creating a very diversified and inclusive set of features. Also, it makes it possible to include both global and local feature relationships, which are very crucial in dealing with intricate data sets. When you combine DA and BA, the whole feature selection process works well because DA focuses on breeding patterns, and BA makes sure the results are accurate. These properties significantly enhance the accuracy and generality of ML-based firewalls across various datasets [25] [37-39].

3.3. Classification

The key purpose of the proposed ML-based firewall is to be able to differentiate between normal and attacked traffic. This process will occur after the completion of comprehensive data preprocessing. Accordingly, foolproof measures were taken to safeguard the data quality being processed, like scaling, transforming, and feature selection. Three key classification algorithms, namely DT, SVM, and LR, are trained and tested to identify the most effective algorithm for the firewall. The data was divided into training and testing sets with a ratio of 80% for training and 20% for testing to ensure the reliability of the model is preserved. Furthermore, cross-validation is used to decrease the chances of bias and variability that come from the usage of a single train-test split. This rigorous assessment backbone not only makes the results reliable but also gives a guarantee that the selected algorithm functions properly in many different cases. The details and performance of DT, SVM, and LR algorithms are presented in Table 2, which demonstrates their possible deployment in the firewall model [41] [42-43].

Hyperparameters play a crucial role in controlling model complexity, regulating learning behavior, and ultimately determining the classifier's generalization performance. The main Hyperparameters used in the three classifiers are listed in Table 3, which highlights their default values and relative impact on performance. Figure 1 demonstrates the proposed ML-based firewall model.

Table 2. DT, SVM, and LR algorithms.

Aspect	DT	SVM	LR
Learning Approach	Supervised learning; uses a tree-like structure to partition the feature space based on information gain or Gini index.	Supervised learning; finds an optimal hyperplane (or decision boundary) to maximize the margin between classes.	Supervised learning; models the relationship between input features and the probability of target classes.
Working Mechanism	Recursively splits data into subsets based on feature thresholds, creating branches and leaves for decisions.	Constructs a hyperplane (or multiple for multi-class) by maximizing margin and using kernel tricks for non-linearity.	Estimates probabilities using a linear equation and maps them to classes using a sigmoid or softmax function.
Strengths	- Simple and interpretable. - Handles categorical and continuous data. - No feature scaling needed.	- Effective for high-dimensional and non-linear data. - Robust to small changes in the data. - Uses kernels.	- Fast and efficient for linear problems. - Provides probabilistic outputs. - Suitable for binary classification.
Weaknesses	- Prone to overfitting without pruning. - Less effective for continuous large feature spaces.	- Computationally expensive for large datasets. - Sensitive to outliers.	- Assumes linearity in relationships. - Struggles with non-linear data. - Sensitive to correlated features.
Computational Complexity	Low; training complexity depends on the number of splits and depth of the tree (O (n log n)).	High; training complexity depends on the kernel used (O (n^2) to O (n^3) for large datasets).	Moderate; scales well with large datasets (O(nk) for k features and n samples).

Table 3. Key Hyperparameters values for DT, SVM, and LR

Algorithm	Hyperparameter	Valu e	Purpose	Impact on Performance
DT	criterion	"gini"	Measures split quality.	Minor impact; rarely changes performance significantly.
	max_depth	None	Controls maximum tree depth.	Major impact—prevents overfitting; shallower depth improves generalization.

	min_samples_split	2	Minimum samples to split a node.	High impact—larger values reduce overfitting and improve stability.
	min_samples_leaf	1	Minimum samples per leaf.	High impact—smooths the model and prevents noisy leaves.
	max_features	None	Number of features considered per split.	Medium impact—reduces variance and speeds up training.
	random_state	None	Controls randomness.	No performance effect, only reproducibility.
SVM	C	1	Regularization strength.	Critical impact—balances margin vs errors; biggest influence on accuracy.
	kernel	"rbf"	Type of decision surface.	Critical impact—defines model behaviour (linear vs nonlinear).
	gamma	"scale"	Kernel influence.	Critical impact—controls overfitting/underfitting; very sensitive.
	degree	3	Only used for polynomial kernel.	Low impact unless polynomial kernel is selected.
	max_iter	-1	Unlimited iterations.	No direct performance effect; only training time.
LR	penalty	"l2"	Regularization type.	Medium impact—L1 can perform feature selection.
	C	1	Inverse regularization strength.	High impact—controls overfitting; smaller C improves generalization.
	solver	"lbfgs"	Optimization algorithm.	Medium impact—affects speed and compatibility with penalties.
	max_iter	100	Max optimization steps.	No major performance effect; only affects convergence.
	fit_intercept	TRUE	Adds bias term.	Low impact—rarely changes performance.

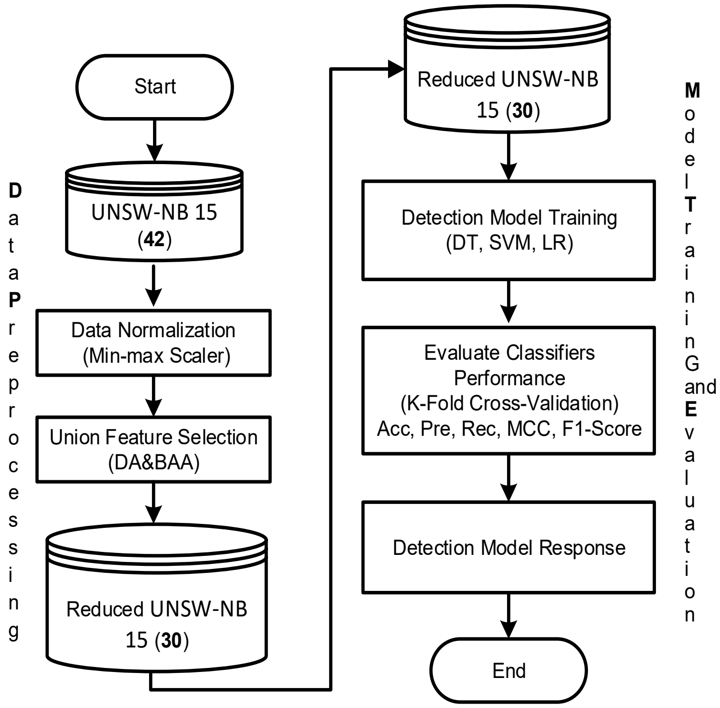
Start
UNSW-NB 15 (42)
Data Normalization (Min-max Scaler)
Union Feature Selection (DA&BAA)
Reduced UNSW-NB 15 (30)
Data Preprocessing
Reduced UNSW-NB 15 (30)
Detection Model Training (DT, SVM, LR)
Evaluate Classifiers Performance (K-Fold Cross-Validation) Acc, Pre, Rec, MCC, F1-Score
Detection Model Response
End
Model TraininG and Evaluation

Figure 1. The proposed ML-based firewall.

4. Results and Discussion

The efficiency of the suggested firewall framework will be assessed on Dell Alienware m18 R2 Gaming Laptop with the following specification: 14th Gen Intel Corei9 14900HX CPU (5.80 GHz speed, 8 Performance-cores, 16 Efficient-cores, 32 Threads, and 36 MB Cache), 32 GB DDR5-4800 RAM, 2 TB SSD, NVIDIA GeForce RTX 4090 (24 GB memory), Ubuntu 24.4.1. Moreover, several libraries and tools have been used from Python 3.13 to develop the proposed firewall model. Some of these libraries and tools are pandas, numpy, MinMaxScaler, LabelEncoder, mealpy.music_based.BA, mealpy.bio_based.DA, DecisionTreeClassifier, SVC, LogisticRegression, and confusion_matrix.

The suggested firewall will be assessed utilizing four different metrics. These metrics are the firewall accuracy (F_{Acc}), firewall precision (F_{Pre}), firewall recall (F_{Rec}), and firewall (F_{f1}). These metrics are calculated based on the confusion matrix (CN). The four elements of the CN, in the case of the suggested firewall, are true positive (F_{TPo}), true negative (F_{TNe}), false positive (F_{FPo}), and false negative (F_{FNe}). F_{Acc}, F_{Pre}, F_{Rec}, and F_{f1} are calculated based on these elements using Equations 1, 2, 3, and 4, respectively. Table 4 illustrates the differences between the four evaluation metrics [44-52].

Table 4. Characteristics of common classification metrics.

Criteria	Accuracy	Precision	Recall	F1-Score
Definition	Percentage of total predictions that are correct.	Out of predicted positives, how many are truly positive.	Out of actual positives, how many the model correctly identifies.	A balanced measure combining precision and recall.
What It Measures	Overall correctness across all classes.	Exactness and reliability of positive predictions.	Completeness and ability to capture all true positives.	Trade-off between precision and recall.

Best Used When	Classes are balanced and error costs are equal.	False positives are costly (e.g., blocking legitimate emails).	False negatives are costly (e.g., missing attacks or diseases).	Dataset is imbalanced and you want a single balanced score.
Strengths	Simple and easy to interpret.	Reduces false alarms; makes positive predictions trustworthy.	Ensures fewer missed positives; ideal for safety-critical tasks.	Good for comparing models; handles imbalance better than accuracy.
Weaknesses	Misleading with imbalanced datasets.	Ignores missed positives; may miss many true cases.	May produce many false alarms.	Does not include true negatives; different precision/recall combinations can produce same score.

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

$$F_{Rec} = \frac{F_{TPo}}{(F_{TPo}+F_{FNe})} \quad (2)$$

$$F_{Pre} = \frac{F_{TPo}}{(F_{TPo}+F_{FPo})} \quad (3)$$

$$F_{f1} = 2 \times \frac{F_{Pre} \times F_{Rec}}{F_{Pre}+F_{Rec}} \quad (4)$$

Figure 2 displays the F_{Acc} of the suggested firewall system. The DT, SVM, and LR algorithms will be utilized to show the F_{Acc} of the suggested firewall system. The DT attained the optimal F_{Acc} of 100%, the SVM attained the almost optimal F_{Acc} of 99.99%, and the LR attained a very high F_{Acc} of 99.94%. Even though the DT attained the uppermost F_{Acc}, the SVM and LR algorithms have attained an extraordinary F_{Acc}. These results indicate that the proposed firewall system delivers highly accurate traffic classification across all models. The minimal variation among the classifiers further confirms that the system's detection performance is consistent, robust, and not dependent on a particular algorithm.

Figure 3 displays the F_{Rec} of the suggested firewall system. The DT, SVM, and LR algorithms will be utilized to show the F_{Rec} of the suggested firewall system. Obviously, the three algorithms have attained extraordinary F_{Rec}, whereas the FRec attained by DT is 100%, and by SVM and LR is 99.98%. Hence, the suggested firewall system has successfully reduced the FPs. These high recall values indicate that the system is highly effective in identifying nearly all malicious traffic with minimal missed detections. The negligible difference among the classifiers also demonstrates that the system maintains strong and consistent detection capabilities regardless of the chosen model.

Figure 4 displays the F_{Pre} of the suggested firewall system. The DT, SVM, and LR algorithms will be utilized to show the FPre of the suggested firewall system. The DT and SVM attained and optimal FPre of 100%, while the LR attained very high F_{Pre} of 99.95%. Though the DT and SVM algorithms attained optimal F_{Pre}, the LR algorithm have attained an outstanding F_{Pre}. Hence, the suggested firewall system has successfully reduced the FNs. These precision results indicate that the system is highly reliable in correctly identifying benign traffic and minimizing false alarms. The close performance among the classifiers further confirms the system's stability and strong predictive accuracy across different models.

Figure 5 displays the F_{f1} of the suggested firewall system. The DT, SVM, and LR algorithms will be utilized to show the F_{f1} of the suggested firewall system. The DT attained an optimal F_{f1} of 100%, the LR attained a very high F_{f1} of 99.97%, and the SVM attained also a very high F_{f1} of 99.77%. Though the DT attained optimal F_{f1}, the SVM and LR algorithms have attained an excellent F_{f1}. These results demonstrate that the proposed firewall system achieves a strong balance between precision and recall across all classifiers. The consistently high Ff1 scores further indicate that the system maintains robust and dependable detection performance regardless of the algorithm used.

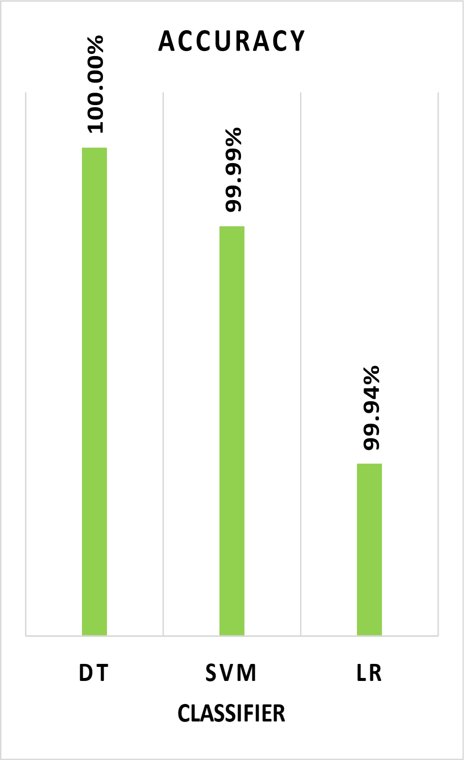

Figure 2. The F_{Acc} of the suggested firewall system

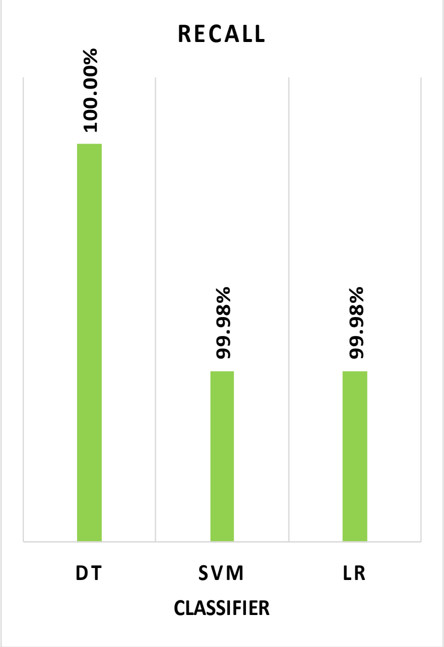

Figure 3. The F_{Rec} of the suggested firewall system.

Figure 6 presents the accuracy comparison of the proposed models (DT, SVM, and LR) and previously published baselines, all on the same dataset. The DT classifier achieved the highest accuracy of 100.00%. SVM achieved 99.99%, and LR achieved 99.94%.

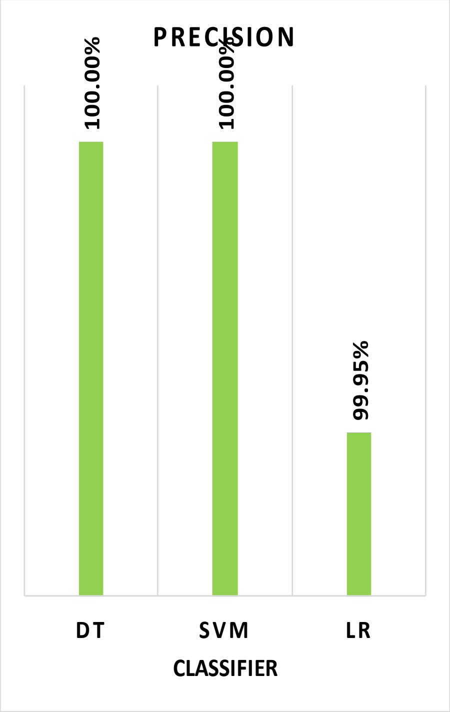

Figure 4. The F_{Pre} of the suggested firewall system

All three outperform the strongest prior works. Taking DT (100.00%) as the reference, the accuracy margins are +0.74% over Ref [20] (99.26%), +0.06% over Ref [21] (99.94%), +0.64% over Ref [22] (99.36%), and +0.55% over Ref [23] (99.45%). Even the lower-performing models—SVM (99.99%) and LR (99.94%)—surpass all existing baselines. This confirms the robustness and effectiveness of the proposed framework.

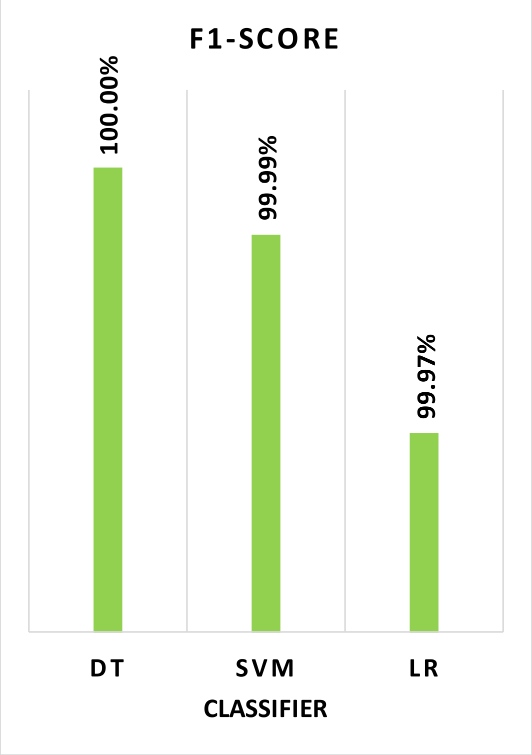

Figure 5. The F_{f1} of the suggested firewall system.

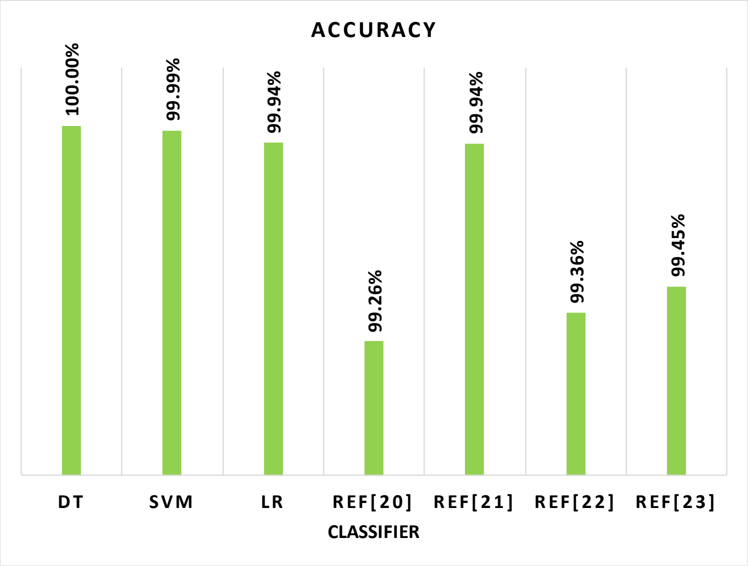

Figure 6. The F_{Acc} of the suggested firewall system versus existing woks

In summary, The F_{Acc} result of the suggested firewall system verifies that the DA and BA optimizers have effectively selected the significant features to identify the attacks. In addition, the result of all metrics also establishes that combining the features chosen by DA and BA optimizers has found the optimal subset of features to find the attack, particularly with a 100% result of DT. Moreover, the F_{Pre} results demonstrate that the subset of features makes the system do well across all classes, avoids bias toward the dominant class, and helps the system balance the predictions by avoiding FPs and FNs.

5. Conclusion

This paper proposed an AI-embedded firewall that integrates DA and BA algorithms for feature selection and employs DT, SVM, and LR classifiers for attack detection. By leveraging the strengths of DA and BA through the Union Set Theory function, the system effectively preserves critical features, enhancing classification performance. Experimental evaluation on the UNSW-NB15 dataset demonstrated that DT achieved 100% accuracy, SVM achieved 99.99% accuracy, while LR achieved 99.94%, confirming the effectiveness of the proposed approach. The results validate the reliability of the AI-based firewall, which minimizes false positives and negatives, ensuring robust intrusion detection. This research highlights the potential of metaheuristic-based feature selection in improving cybersecurity defenses. Future work will focus on expanding the dataset, testing additional classifiers, and implementing real-time attack mitigation strategies.

Reference

1. E. B. Amin, R. Al-Dmour, H. Al-Dmour, and A. Al-Dmour, "Technostress Impact on Educator Productivity: Gender Differences in Jordan's Higher Education," Electronic Journal of e-Learning, vol. 22, no. 8, pp. 60–75, 2024, doi: 10.34190/ejel.22.8.3608.
2. M. M. Abualhaj, A. A. Abu-Shareha, and S. N. Al-Khatib, "An Innovative Approach for Enhancing Capacity Utilization in Point-to-Point Voice over Internet Protocol Calls," International Journal of Electrical and Computer Engineering, vol. 14, no. 1, 2024.
3. M. N. Al-Sharabati, A. A. Abu-Shareha, and M. A. AlSharaiah, "An Adaptive Framework for Classification and Detection of Android Malware," International Journal of Interactive Mobile Technologies, vol. 18, no. 5, pp. 1–17, 2024.
4. H. Othman, M. M. Abu Al-Hija, and M. A. AlSharaiah, "Innovative Malware Detection: Practical Swarm Optimization and fuzzyKNN Model in Honeypot Environment," International Journal of Intelligent Engineering and Systems, vol. 17, no. 1, pp. 299–308, Feb. 2024.
5. H. Othman, M. M. Abu Al-Hija, and M. A. AlSharaiah, "Toward Enhancing Malware Detection Using Practical Swarm Optimization in Honeypot," International Journal of Intelligent Engineering and Systems, vol. 17, no. 1, pp. 288–298, Feb. 2024.
6. J. R. Hernandez, "What Is the Actual Cost of Cybercrime?," Evolve Security, Jan. 15, 2025. [Online]. Available: https://www.evolvesecurity.com/blog-posts/actual-cost-of-cybercrime.
7. R. Alabdallat, M. Abualhaj, and A. Abu-Shareha, "Android Malware Detection Using a Modified Dwarf Mongoose Algorithm," International Journal of Intelligent Engineering and Systems, vol. 18, no. 8, 2025, doi: 10.22266/ijies2025.0930.21.
8. A. A. Abu-Shareha, M. Al-Zyoud, and Q. Y. Shambour, "Cascaded Spin Shuffle: A Transposition Cipher Using Spin Motion and Grid Cascading," International Journal of Intelligent Engineering and Systems, vol. 17, no. 6, pp. 824–838, 2024.
9. M. M. Abualhaj, S. N. Al-Khatib, A. A. Abu-Shareha, A. Hyassat, and M. Sh. Daoud, "Smart Firewall for Phishing Detection Powered by Bio-Inspired Algorithms," Journal of Advances in Information Technology, vol. 16, no. 11, pp. 1529–1539, 2025.
10. 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, 2020, doi: 10.1109/TCC.2016.2554548.
11. M. M. Abualhaj et al., "Improving Firewall Performance Using Hybrid of Optimization Algorithms and Decision Trees Classifier," IAES International Journal of Artificial Intelligence, vol. 14, no. 4, pp. 2839–2848, Aug. 2025, doi: 10.11591/ijai.v14.i4.pp2839-2848.
12. Z. Eman, A. Sherin and L. Bayan, "Exploring Healthcare Students' Perspectives on Artificial Intelligence in Healthcare," 2024 25th International Arab Conference on Information Technology (ACIT), Zarqa, Jordan, 2024, pp. 1-6, doi: 10.1109/ACIT62805.2024.10876903.
13. N. Tashtoush et al., "Cancer Pain Detection Based on Physiological Parameters and Machine Learning," Cogent Engineering, 2024.
14. Z. S. Mufti et al., "Spectral Analysis of CuO and GO via Machine Learning," Egyptian Informatics Journal, vol. 29, art. 100632, 2025.
15. A. Arabiat, M. Hassan, and O. Almomani, "Traffic Congestion Prediction Using Machine Learning: Amman City Case Study," Proc. SPIE, vol. 13188, 1318806, 2024.
16. A. M. Ali, S. Nashwan, A. Al-Qerem, A. Almomani, M. A. Sakhnini and A. Aldweesh, "Machine Learning Models for Brain Signal Classification: A Focus on EEG Analysis in Epilepsy Cases," 2024 2nd International Conference on Cyber Resilience (ICCR), Dubai, United Arab Emirates, 2024, pp. 1-8, doi: 10.1109/ICCR61006.2024.10532919.
17. M. M. Abualhaj et al., "Spam Detection Boosted by Firefly-Based Feature Selection and Optimized Classifiers," IJASCA, vol. 17, no. 3, pp. 1–19, 2025.
18. R. Alabdallat, M. Abualhaj, and A. Abu-Shareha, "Enhanced Multiclass Android Malware Detection Using a Modified Dwarf Mongoose Algorithm," International Journal of Analysis and Applications, vol. 23, pp. 1–23, Jan. 2025.

19. V. Mittal, “Synthesis of Circular Antenna Arrays for Achieving Lower Side Lobe Level and Higher Directivity Using Hybrid Optimization Algorithm,” Algorithms, vol. 17, no. 6, art. 256, 2024.
20. U. Mohammed et al., “ICSOMPA: A Novel Improved Hybrid Algorithm for Global Optimisation,” Evolutionary Intelligence, vol. 17, no. 5, pp. 3337–3440, 2024.
21. A. Abu-Khadrah et al., “The Impact of an Inverse-Buffalo Variant Optimization Algorithm on Search Space Expansion,” IEEE Access, vol. 12, pp. 119775–119788, 2024.
22. A. Ishtaiwi et al., “A Hybrid JADE–Sine Cosine Approach for Advanced Metaheuristic Optimization,” Applied Sciences, vol. 14, no. 22, art. 10248, 2024.
23. A. A. Abu-Shareha, M. M. Abualhaj, A. A. Ali, A. Munther, and M. Anbar, “Enhancing Malware Detection with Firefly and Grey Wolf Optimization Algorithms,” in 2024 11th International Conference on Electrical and Electronics Engineering (ICEEE), Amman, Jordan, Apr. 2024, pp. 394–398, doi: 10.1109/ICEEE62185.2024.10779310.
24. G. Kaur et al., “An Efficient Approach for Localizing Sensor Nodes in 2D Wireless Sensor Networks Using Whale Optimization-Based Naked Mole Rat Algorithm,” Mathematics, vol. 12, no. 15, art. 2315, 2024.
25. M. M. Abualhaj et al., “Enhanced Network Communication Security Through Hybrid Dragonfly-Bat Feature Selection for Intrusion Detection,” Journal of Communications, vol. 20, no. 5, pp. 607–618, 2025.
26. M. M. Abualhaj et al., “A Paradigm for DoS Attack Disclosure Using Machine Learning Techniques,” IJACSA, vol. 13, no. 3, pp. 192–200, 2022.
27. A. Arabiat et al., “WEKA-based Machine Learning for Traffic Congestion Prediction in Amman City,” IJAI, vol. 13, no. 4, pp. 4422–4434, 2024.
28. A. Aleesa et al., “Deep-Intrusion Detection System with Enhanced UNSW-NB15 Dataset,” Journal of Engineering Science and Technology, vol. 16, no. 1, pp. 711–727, 2021.
29. S. Bagui et al., “Using Machine Learning Techniques to Identify Rare Cyber-Attacks on the UNSW-NB15 Dataset,” Security and Privacy, vol. 2, no. 6, p. e91, 2019.
30. I. O. Alabi and Y. B. Shuaibu, “Utilizing Metaheuristic Ensemble Feature Selection to Enhance IDS,” Proc. iSTEAMS Conf., 2024.
31. S. More et al., “Enhanced Intrusion Detection Systems Performance with UNSW-NB15 Data Analysis,” Algorithms, vol. 17, no. 2, art. 64, 2024.
32. 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, art. 64, 2024, doi: 10.3390/a17020064.
33. J. Xiao et al., “Correction to: Load Balancing Strategy for SDN Multi-Controller Clusters Based on Load Prediction,” Journal of Supercomputing, vol. 80, pp. 7120–7121, 2024.
34. J. Xie et al., “Network Intrusion Detection Based on Dynamic Intuitionistic Fuzzy Sets,” IEEE Transactions on Fuzzy Systems, vol. 30, no. 9, pp. 3460–3472, 2022.
35. Y. A. Maz et al., “Transfer Learning-Based Approach with an Ensemble Classifier for Detecting Keylogging Attack,” Computers, Materials & Continua, vol. 76, no. 1, pp. 1–10, 2025.
36. A. Alsokkar, “Sentiment Analysis for Arabic Call Center Notes Using Machine Learning Techniques: A Case Study of Jordanian Dialect,” Journal of Artificial Intelligence, vol. 7, no. 3, pp. 29–39, 2023, doi: 10.32629/jai.v7i3.940.
37. S. Mirjalili, “Dragonfly Algorithm,” Neural Computing and Applications, vol. 27, no. 4, pp. 1053–1073, 2015.
38. P.-W. Tsai et al., “Bat Algorithm Inspired Algorithm for Solving Numerical Optimization Problems,” Applied Mechanics and Materials, vols. 148–149, pp. 134–137, 2011.
39. W. A. H. M. Ghanem et al., “Cyber Intrusion Detection System Based on a Multiobjective Binary Bat Algorithm,” IEEE Access, vol. 10, pp. 76318–76339, 2022.
40. A. Adel, A. Munther, M. M. Abualhaj, A. Al-Allawee, and M. Anbar, “Enhancing Malware Detection with Firefly and Grey Wolf Optimization Algorithms,” in 2024 11th International Conference on Electrical and Electronics Engineering (ICEEE), Amman, Jordan, Apr. 2024, pp. 394–398, doi: 10.1109/ICEEE62185.2024.10779310.
41. D. S. Kapoor, K. J. Singh, A. Singh, B. Mulakala, K. Singh, P. Prashant, R. Singh, and S. Mahajan, “Comparative Evaluation and Prediction of Exoplanets Using Machine Learning Methods,” in Integrating Metaheuristics in Computer Vision for Real-World Optimization Problems, A. Abraham, V. Snášel, and M. Gandhi, Eds. Hoboken, NJ, USA: Wiley, 2024, pp. 163–184, doi: 10.1002/9781394230952.ch9.

42. T. Lhamo et al., "Heart Disease Diagnosis by Machine Learning Techniques," 2024 International Conference on Science, Engineering and Business for Driving Sustainable Development Goals (SEB4SDG), Omu-Aran, Nigeria, 2024, pp. 1-7, doi: 10.1109/SEB4SDG60871.2024.10630058.
43. Q. Y. Shambour et al., “A Multi-Criteria Trust-Enhanced Collaborative Filtering Algorithm for Personalized Tourism Recommendations,” IJECE, vol. 36, no. 3, pp. 1919–1928, 2024.
44. Ibrahim, C. Gatzoulis, M. M. Eid, F. H. Rizk, L. Abualigah, and E. S. M. El-Kenawy, “Regression Models for Predicting Gamified Educational Outcomes,” in 2024 International Telecommunications Conference (ITC-Egypt), Cairo, Egypt, Jul. 2024, pp. 537–542, doi: 10.1109/ITC-Egypt61665.2024.10629538.
45. N. M. Alaskar, M. Hussain, S. J. Almheiri, Atta-ur-Rahman, 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, early access, Dec. 18, 2025, doi:10.32604/cmc.2025.074041.
46. M. Hussain, W. Sharif, M. R. Faheem, Y. Alsarhan, and H. A. Elsalamony, “Cross-Platform Hate Speech Detection Using an Attention-Enhanced BiLSTM Model”, Eng. Technol. Appl. Sci. Res., vol. 15, no. 6, pp. 29779–29786, Dec. 2025. https://doi.org/10.48084/etasr.13249
47. Z. Awais et al., "ISCC: Intelligent Semantic Caching and Control for NDN-Enabled Industrial IoT Networks," in IEEE Access, vol. 13, pp. 169881-169898, 2025, doi: 10.1109/ACCESS.2025.3614984.
48. Zubair, M.; Hussain, M.; Albashrawi, M.A.; Bendechache, M.; Owais, M. A comprehensive review of techniques, algorithms, advancements, challenges, and clinical applications of multi-modal medical image fusion for improved diagnosis. Computer Methods and Programs in Biomedicine. 2025, 272, 109014. https://doi.org/10.1016/j.cmpb.2025.109014.
49. Hussain, M., Chen, C., Hussain, M. et al. Optimised knowledge distillation for efficient social media emotion recognition using DistilBERT and ALBERT. Sci Rep 15, 30104 (2025). https://doi.org/10.1038/s41598-025-16001-9
50. Zubair, M., Owais, M., Hassan, T. et al. An interpretable framework for gastric cancer classification using multi-channel attention mechanisms and transfer learning approach on histopathology images. Sci Rep 15, 13087 (2025). https://doi.org/10.1038/s41598-025-97256-0
51. Y. Sanjalawe, S. Fraihat, S. Al-E’mari, M. M. Abualhaj, S. Makhadmeh, and E. Alzubi, “Smart load balancing in cloud computing: Integrating feature selection with advanced deep learning models,” PLOS One, vol. 20, no. 9, Sep. 2025, Art. no. e0329765. DOI: 10.1371/journal.pone.0329765.
52. Al-Ghraibah, A. Al-Abbas, M. B. Kmainasi, I. E. Mustafa, and K. Brmo, “Automated Detection of Lung Cancer Levels Based on Patient’s Lifestyle Information,” in 2024 Second Jordanian International Biomedical Engineering Conference (JIBEC), Amman, Jordan, Nov. 2024, pp. 22–27, doi: 10.1109/JIBEC62419.2024.10731194.