Improved Intrusion Detection System Using a Modified African Vultures Algorithm

Yasameen A. Al-Shadeedi [1,*], Mosleh Abualhaj [1], Ahmad Abu-Shareha [2], Mohamed Yousif [3], and Anusha Achuthan [4]

[1] Department of Networks and Cybersecurity, Al-Ahliyya Amman University, Amman, Jordan
[2] Department of Data Science and Artificial Intelligence, Al-Ahliyya Amman University, Amman, Jordan
[3] School of Technologies, Cardiff Metropolitan University, Cardiff, UK
[4] School of Computer Sciences, Universiti Sains Malaysia, Gelugor, Penang, Malaysia
Email: yasmeen_ameer1994@yahoo.com (Y.A.A-S.); m.abualhaj@ammanu.edu.jo (M.A); a.abushareha@ammanu.edu.jo (A.A-S); MYousif@cardiffmet.ac.uk (M.Y); anusha@usm.my (A.A).
*Corresponding author

***Abstract*—The evolution of cyber threats has imposed the need for robust Intrusion Detection Systems (IDSs) that are capable of detecting intrusions with high accuracy. This paper proposes an IDS based on Machine Learning (ML), which uses a modified African Vultures Optimization Algorithm (AVOA) for Feature Selection (FS). The AVOA was enhanced using a crossover function. The crossover was implemented to enhance the algorithm's ability to explore the FS solution space. The proposed model was further evaluated on the UNSW-NB15 dataset with binary classification. Binary classification showed an excellent performance using the XGBoost and Hist Gradient Boosting classifiers, with an accuracy and a precision of 99.77%. These results further indicate the efficiency of using AVOA for intrusion detection. This study depicts the potential of ML in the development of accurate and robust IDS solutions, thus setting a benchmark for future research in cybersecurity.**

***Keywords*—Machine Learning (ML), Feature Selection (FS), Intrusion Detection System (IDS), African Vultures Algorithm (AVOA)**

I. Introduction

Technology plays an important role in every sphere of life. Technology influences the lives of millions of people, from software to self-driving cars, computer networks, and many more. Technologies have also introduced important innovations in factories, hospitals, hotels, cities, and territories. Overall, technology has improved every aspect of our lives, from the way we communicate, work, and make decisions [1]. However, this increasing reliance on technology has also increased the risks that accompanied. It has led to an increase in cyber intrusions, posing significant threats to personal, corporate, and national security [2, 3]. Cyber intrusions include attacks such as data breaches, phishing, ransomware, and many more [4]. It targets individuals, digital systems, and networks, leading to substantial financial and reputational damage. With over 2328 attacks per day, an average of 800,000 attacks happen every year; there is a hacker attack every 39 seconds. It is estimated that around 33 billion account breaches occurred in Ref. [5]. Fig. 1 shows the losses made by cyber intrusions, which are on a constant rise [6]. The high number of intrusions necessitates the deployment of strong cyber defense systems and measures to protect private information and to preserve systems' integrity against these rapidly increasing intrusions. The prevention of intrusions has become a serious challenge because of the vast increase in cybersecurity threats [7].

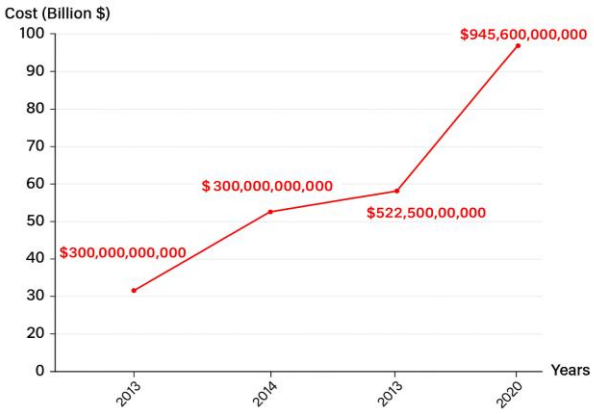

Fig. 1. Financial losses caused by cyber intrusions in billions [6].

Traditional cybersecurity measures, such as firewalls and anti-malware software, while still essential, need to be supplemented with newer techniques such as Intrusion Detection Systems (IDSs). IDS is essential for threat detection [8]. IDS can be software or hardware; it works by monitoring network flows and behavioral patterns in order to identify any suspicious activity and generate reports [9]. Modern IDS employs AI models to predict malicious activities in real time [10]. Machine Learning (ML) can be used to enhance the IDS performance in threat detection [11] and allows for the detection of previously unknown attacks through pattern recognition, enabling proactive responses to security breaches. ML-based IDS is implemented commonly using a supervised ML model that is trained using a classification algorithm and a dataset.

Manuscript received November 13, 2025; revised December 19, 2025; accepted January 26, 2026; published April 14, 2026.

The classification algorithm recognizes common patterns in the data using complex mathematical operations [12, 13].

The labeled dataset contains samples of network flows; these flows are labeled as normal or intrusion. Then, the trained model using the dataset is used with new unlabeled flows to predict whether they are normal or intrusion. This is beneficial for real-time protection as well as it predicts abnormalities and previously unrecorded attacks [14]. However, numerous challenges can impact the performance of ML-based IDS, such as the large amount of data in the network traffic that is analyzed by the IDS, as well as the irrelevant information contained in the analyzed data [15]. Here, Feature Selection (FS) emerges as a critical step in the development of IDS models [16]. FS involves identifying and selecting the most crucial features out of a larger set of features. By selecting the relevant features, the model will increase the IDS performance by reducing the amount of irrelevant data in the prediction process done [17, 18].

Metaheuristic algorithms have demonstrated their efficiency in Refs. [19−22]; among these metaheuristics, the AVOA stands out due to its effectiveness in navigating the complex search space of FS [23]. Inspired by the behavior of African vultures, this algorithm efficiently identifies the most relevant features, thus enhancing the predictive capabilities of IDS [24]. This work will introduce an IDS model using a modified version of AVOA for the model development. The AVOA will be applied to improve the performance of the IDS model. This study will evaluate the proposed AVOA performance on the UNSW-NB15 dataset. UNSW-NB15 dataset contains various attack types, including DoS, worms, Backdoors, and Fizzers. This makes it a suitable benchmark dataset for testing and development of intrusion detection techniques.

II. Literature Review

Researchers focused on IDS as it is an important tool for cybersecurity. IDS model performance is improved using FS, and for that, multiple studies were done in this area using the UNSW-NB15 dataset for intrusion detection. This confirms the importance of this dataset in making a strong foundation for IDS ML models.

Rajeyyagari *et al*. [25] proposed a framework for enhancing security in smart grid networks using both blockchain modules and Deep Learning (DL) using CNN for classification and the AVO algorithm for hyperparameter tuning. The model was trained on the UNSW-NB15 dataset. The model achieved a classification accuracy of 97.40%, a precision of 98.25%, a recall of 98.37%, and an F1−Score of 99.17%.

In the study by Alsirhani *et al*. [26], a detection model for securing a smart grid was introduced. The authors applied AVOA for FS, Deep Belief Network (DBN), and Long Short-Term Memory (LSTM) for the classification of network traffic into normal and attack. The model was trained on the NSL-KDD dataset with results of 98.99% for Accuracy, 99.39% for Specificity, 96.97% for Recall, 96.97% for Precision, 96.97% for F-Measure, 00.30 % for FNR, and 00.06 % for FPR.

Zareian *et al.* [27] proposed the Bitterling Fish Optimization (BFO) Algorithm, a meta-heuristic algorithm inspired by the mating behavior of bitterling fish. In this paper, the authors have applied BFO in the FS in IDS and multiple optimization problems. FS performance is evaluated on several COVID-19, diabetes, kidney disease, and intrusion detection datasets. In intrusion detection, it uses the dataset UNSW-NB15. For classification, the multi-layer perceptron was used, whose accuracy result was 96.72%.

Zhiqiang *et al.* [28] proposed a DL-based IDS model to help solve the challenges of identifying attacks in networks. The authors used the UNSW-NB15 dataset for training the model. In FS, the features most impactful were identified using the Gedeon method. The feed-forward artificial neural network algorithm was used to train the DL model with ten hidden layers. This model attained an accuracy score of 99.5% with a low false alarm rate of 0.47%.

Shushlevska *et al.* [29] proposed an IDS using the UNSW-NB15 dataset; they evaluated several ML algorithms for anomaly identification in network traffic, comparing the performances of Naive Bayes (NB), Logistic Regression (LR), Decision Tree (DT), and Random Forest (RF) classifiers, with a classification accuracy of 70.62%, 70.60%, 86.12%, and 87.09% respectively. In terms of F1−score and Recall measures, the RF classifier was the best performing.

Ramasubramanian and Rajaprakash [30] proposed a DL model for intrusion detection for 5G-IoT networks. The A2VO was used for FS, and RNN LSTM was used for classification. The model was trained on the NSL-KDD. With an accuracy score of 99%, precision of 98.9%, and F1−Score of 98.9%. On AWID the results were 99.3%, 99.1%, and 99% for accuracy, precision, and F1−score respectively. On the UNSW-NB15 dataset, the results were 99.2%, 99%, and 99% for accuracy, precision, and F1−Score respectively.

III. Materials and Methods

This study aims to develop an ML model for intrusion detection. The components of the proposed IDS model are shown in Fig. 2. The model will be trained and evaluated using the UNSW-NB15 dataset. The dataset underwent the necessary steps of preprocessing. The preprocessing steps used for the dataset are data transformation, normalization, and FS. The transformation was done using the label encoder to transform categorical values into numerical ones, which is necessary because some ML algorithms work better with numerical values.

Normalization was completed using the Min-Max Scaler to convert the numerical values into a common range. This is crucial for ML algorithms that operate with the expectation that all input variables are in the same range. FS is crucial because of the irrelevant data that could impact the IDS performance negatively. The FS stage is completed using the AVOA algorithm to minimize irrelevant data processed by the model, thus improving its performance. The optimal feature subset will be used with various ML algorithms for classification.

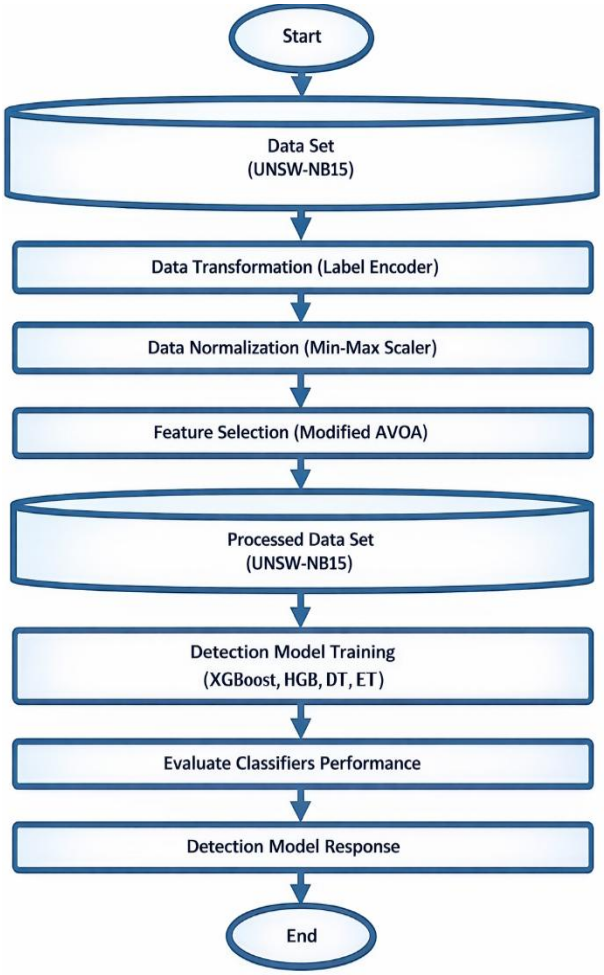

Fig. 2. The proposed IDS model.

A. Dataset

The UNSW-NB15 dataset has already been used for the IDS model development. The dataset consists of approximately 2.5 million records and 49 features. This dataset was created by the Australian Centre for Cyber Security (ACCS) [31]. UNSW-NB 15 data set is created by the IXIA PerfectStorm tool in the Cyber Range Lab. This data set contains nine families of attack types: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms. This makes it a suitable benchmark dataset for testing and development of intrusion detection models. The 49 features in the dataset can be classified into six categories. Table I describes the features of the UNSW-NB15 dataset and their categories [31].

The UNSW-NB15 major categories of the records are normal and attack. The attack records are further classified into nine families according to the nature of the attacks. Table II shows the UNSW-NB15 data set record distribution.

The features in the UNSW-NB15 dataset are designed to capture a wide range of network data that can be an indicator of normal or malicious behavior for network traffic. UNSW-NB15 supports the creation of a robust IDS ML model.

TABLE I. UNSW-NB15 FEATURE DESCRIPTION

Feature category	#	Name	Type
Flow features	1	srcip	nominal
	2	sport	integer
	3	dstip	nominal
	4	dsport	integer
	5	proto	nominal
Basic features	6	state	nominal
	7	dur	Float
	8	sbytes	Integer
	9	dbytes	Integer
	10	sttl	Integer
	11	dttl	Integer
	12	sloss	Integer
	13	dloss	Integer
	14	service	nominal
	15	Sload	Float
	16	Dload	Float
	17	Spkts	integer
	18	Dpkts	integer
Content features	19	swin	integer
	20	dwin	integer
	21	stcpb	integer
	22	dtcpb	integer
	23	smeansz	integer
	24	dmeansz	integer
	25	trans_depth	integer
	26	res_bdy_len	integer
Time features	27	Sjit	Float
	28	Djit	Float
	29	Stime	Timestamp
	30	Ltime	Timestamp
	31	Sintpkt	Float
	32	Dintpkt	Float
	33	tcprtt	Float
	34	synack	Float
	35	ackdat	Float
General-purpose features	36	is_sm_ips_ports	Binary
	37	ct_state_ttl	Integer
	38	ct_flw_http_mthd	Integer
	39	is_ftp_login	Binary
	40	ct_ftp_cmd	integer
Connection features	41	ct_srv_src	integer
	42	ct_srv_dst	integer
	43	ct_dst_ltm	integer
	44	ct_src_ltm	integer
	45	ct_src_dport_ltm	integer
	46	ct_dst_sport_ltm	integer
	47	ct_dst_src_ltm	integer
Labeling features	48	attack_cat	nominal
	49	Label	binary

TABLE II. UNSW-NB15 DATA SET RECORD DISTRIBUTION

Attack Type	Number of Records
Normal	2,218,761
Fuzzers	24,246
Analysis	2,677
Backdoors	2,329
DoS	16,353
Exploits	44,525
Generic	215,481
Reconnaissance	13,987
Shellcode	1,511
Worms	174

B. Data Cleaning

The first step of data preprocessing for UNSW-NB15 is data cleaning. This is done by handling the missing data in the datasets. UNSW-NB15 contains some missing data, this data may be missing because it was never collected, the records were lost, or for many other reasons. This missing data, in turn, creates a huge problem for the proposed ML model. The missing values negatively impact the performance of the ML algorithms and the AVOA [32]. Hence, it is necessary to handle missing values present in the dataset to ensure more accurate prediction and training for the IDS model, reducing false predictions. In the UNSW-NB15 dataset, the specific feature service has missing values marked as (-). The common way to handle missing values is to drop the column or row containing them. In this case, the service feature has missing values, and using the mean would not be appropriate [33]. This feature indicates the protocol used for communication (e.g., HTTP, SMTP, FTP, DNS), which is critical for identifying the type of attack. Assigning an incorrect value (such as the mean) could lead to false predictions. Therefore, the missing values were handled by dropping the rows that contained them. This row (instances) associated with the (-) are therefore dropped to ensure that they don't negatively impact the model's performance.

C. Numeric Conversion

The next step of data preprocessing for UNSW-NB15 is numeric conversion. The dataset features are categorized into four different types based on their nature: nominal, integer, binary, and float. The information for the features category is mentioned in the dataset files. The feature types are also outlined in Table I.

Numeric conversion is necessary to ensure that mathematical operation in the ML algorithms and AVOA can be applied without errors, as they work better with numerical values [34]. The integer, binary, and float features (e.g., 8, 9, 15, and 16 in Table I) were converted to ensure consistency across the dataset, while the nominal features (e.g., 5, 6, and 14) were converted using one-hot encoding.

One-hot encoding is used for nominal features such as 'protocol_type', 'service', and 'flag' to transform them into a format that can be effectively utilized by ML algorithms [35] since one-hot encoding converts each category into a new binary column, where the presence of a category is marked with one and its absence with a 0. This method prevents the ML model from assuming a natural ordering among categories, which could lead to biased or inaccurate results.

The conversion step is crucial for numerical operations and ensures that mathematical functions and algorithms can be applied without errors. The integer and binary data types undergo conversion to ensure all numerical features are in a format suitable for ML models, which improve the algorithms performance.

D. Label Encoding

The second stage of data preprocessing is a necessary stage in the preparation of the data for the proposed IDS ML model. The attack category and label features are transformed into numerical values using label encoding. This technique is crucial because ML algorithms work better with numerical values. Label encoding works by assigning a unique integer value to each distinct category within a feature [36]. The encoding process ensures that the categorical features, for the category of the attack and label, are converted into a numerical value for better ML algorithm performance. Fig. 3 shows an example of a label encoding for the attack category label. Converting categorical labels to numerical values through label encoding is a fundamental step in building the IDS ML model. It is done to ensure that the model can handle the categorical data efficiently.

Label	Encoded Label
Normal	0
Analysis	1
Backdoor	2
DoS	3
Exploits	4
Fuzzers	5
Generic	6
Reconnaissance	7
Shellcode	8
Worms	9

Fig. 3. Label encoding.

E. Data Normalization

The third stage in data preprocessing is data normalization, as it has a vital role in preparing data for the proposed ML IDS. Normalization ensures that all variables are adjusted to a consistent range, which is critical for optimizing the performance of classification algorithms [37]. One of the most used methods for normalization is the Min-Max Scaling, which rescales features to fall within a range of 0 to 1 [38]. This step is important because many classification algorithms, as well as optimization techniques, assume that the input variables are on the same scale. If this assumption is not met, it can negatively impact the model's performance. Therefore, applying normalization helps prevent such issues and ensures more accurate and reliable detection by the IDS model. Therefore, the MinMaxScaler is implemented to ensure all features are on the same scale. The MinMaxScaler is used with the Eq. (1) [39]:

$$X\_Scaled=(X-X\_Min) / (X\_Max-X\_Min) \quad (1)$$

where X_Scaled is the result for the MinMaxScaler, X is the original value, X_Min is the minimum value in the column, and X_Max is the maximum value in the column. The MinMaxScaler was used for the features in the dataset.

F. Feature Selection Using the Modified African Vulture Optimization Algorirhm

FS is the process of identifying features that impact the performance of an ML IDS [40]. The AVOA is a meta-

heuristic algorithm effective in FS in IDS. However, all Metaheuristics algorithm (e.g., AVOA) suffer from the local optima issue [41]. Local optima are where the algorithm reaches a solution and labels it as the best [42]. However, if the search space had been explored more efficiently, a better solution would have been found. A major effective measure against the local optima is by using a crossover [43]. The crossover increases the capacity of the algorithm to explore the search space and produces more solutions [44]. The AVOA has been modified by adding a crossover function, which increases the diversity of the solution space and helps prevent the algorithm from getting stuck in local optima. This modification enhances the algorithm's ability to thoroughly explore the solution space, leading to the discovery of more optimal feature subsets to improve the performance of the ML IDS. Fig. 4 shows the Flow Chart for the modified AVOA, which is the same as the original AVOA, however, with one additional function for the crossover.

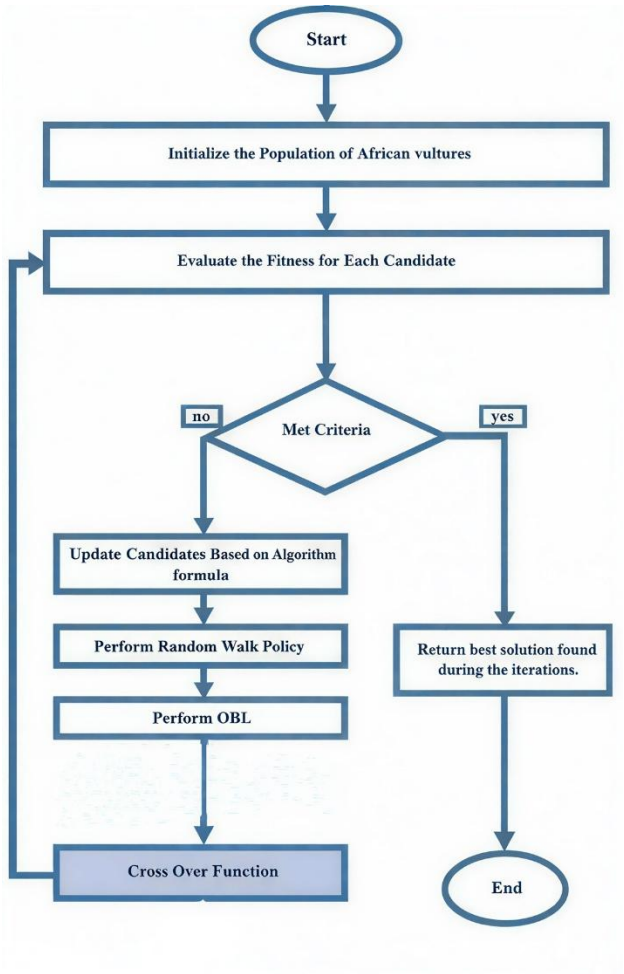

Fig. 4. Flow chart for the modified AVOA.

The crossover function in the modified AVOA is applied between the best solution (the top vulture based on fitness) and the rest of the population. This process generates new offspring solutions by combining features from the best-performing parent and other members of the population, ensuring diversity and improving the algorithm's exploration of the solution space.

A Gaussian crossover was employed to introduce controlled stochastic variability during feature selection, enhancing population diversity while maintaining stable convergence in the high-dimensional UNSW-NB15 feature space. Gaussian noise and the blending factor were empirically selected to balance exploration and exploitation, and the comparison with the original AVOA was performed to isolate the effect of the proposed crossover enhancement. The steps for this crossover process are as follows:

Step one: Selecting Parents for Crossover: The algorithm selects the best solution as a parent, and the second parent is selected from the remaining population. The crossover is done between the best solution and the rest of the population. This generates two offspring solutions from the selected parents. When it comes to the generation of new offspring solutions, a Gaussian crossover is used, leveraging a blending factor and Gaussian-distributed noise to introduce variability. By iterating over each gene (feature) of each parent, it mixes the corresponding genes from both parents, adjusting them with random noise, and generates two new genetically diverse offspring.

Step two: Combining the Population: Once the offspring are created, the algorithm combines the current population and the newly generated offspring into a single larger population. This increases the diversity of the population by introducing new potential solutions.

Step three: Fitness Evaluation: The algorithm evaluates the fitness of the entire population (both the original individuals and the new offspring). Each individual is assessed based on their ability to classify data accurately using the objective function, helping to identify the most promising solutions.

Step four: Selecting Top Individuals: The top N solutions, where N is the size of the original population, are selected based on their fitness values. The algorithm discards the less effective solutions while keeping the best-performing individuals for future iterations.

Using the crossover, enhanced AVOA algorithm increases the diversity of the solution space while also focusing on refining the best solutions. This helps the algorithm avoid local optima and improves its ability to identify the best feature subsets that enhance the performance of the IDS model.

G. Classification

Potential is most evident in classification. It makes accurate predictions possible and facilitates well-informed decision-making. It is essential to comprehend classification methods in order to get reliable findings in a variety of fields. Knowledge discovery and the extraction of important insights are facilitated by efficient data management and a thorough understanding of categorization techniques. The IDS ML model relies on algorithms that learn from patterns in data using complex mathematical equations, enabling the model to make predictions without requiring explicit programming. These algorithms are known as classifiers, where the target values (labels) are known in advance. Once trained, the IDS ML model can classify new, previously unseen data

instances to determine whether the network traffic is normal or an attack. Classification is crucial for IDS to predict whether incoming network traffic represents normal activity or an attack. The classifiers used to train the IDS ML model include XGBoost, DT, Extra Trees (ET), and Histogram-based Gradient Boosting (HGB). The reason for employing a variety of classifiers is that each algorithm has a unique approach to recognizing patterns and making predictions. Some classifiers might be more effective for network data. By using multiple classifiers, the chances of identifying a model that performs exceptionally well in detecting intrusions. Table III shows advantages and disadvantages of classifiers [45−48].

TABLE III. ADVANTAGES AND DISADVANTAGES OF CLASSIFIERS

ML algorithm	Advantages	Disadvantages
DT	Simple to understand and interpret. Requires little data preparation. Able to handle both numerical and categorical data.	Prone to overfitting. It can become unstable with small variations in data. Poor performance on unbalanced datasets.
ET	Less likely to overfit than regular DTs. Effective in high-dimensional spaces. Robust to noise and outliers.	Computationally intensive. Difficult to interpret compared to simple DTs. Can still overfit with very noisy data.
HGB	Fast training speed. Effective handling of missing values. Good performance on imbalanced data.	Prone to overfitting without proper tuning. Not as interpretable as simpler models. Requires careful setup of hyperparameters.
XGBoost	High performance on a variety of problems. Handles missing data well. GPU support speeds up computation.	Can be prone to overfitting if not tuned properly. Requires good hardware for optimal performance. Complex parameter tuning is needed.

IV. IMPLEMENTATION RESULTS AND DISCUSSION

The study was done on a computer running Windows with the Visual Studio Code (VSC) IDE because it has powerful and user-friendly development tools. A high-performance computing setup is used to put the ML system into action. The ML model was implemented using Python version 3.11. Python comes with prepared libraries for ML and data handling that were useful for the model implementation. Table IV provides a short overview of the libraries utilized in setting up the ML Model.

The effectiveness of the proposed intrusion detection system in this study is evaluated using several performance metrics. Accuracy indicates the overall ability of the model to correctly classify network traffic. Precision reflects the reliability of the generated intrusion alerts by measuring how many detected attacks are truly malicious. The F1−Score provides a balanced evaluation of the model's detection performance, thereby indicating the overall robustness and effectiveness of the proposed IDS model [49−51].

TABLE IV. IMPORTANT LIBRARIES USED IN CREATING THE MODEL

Role in Implementation	Libraries and Descriptions	Specific Uses
Data Handling and Manipulation	Pandas: Data manipulation and analysis library NumPy: Numerical computing with arrays and matrices	Data handling, reading/writing CSV files Mathematical operations, array manipulations
ML Algorithms	sklearn.Ensemble: Ensemble methods from scikit-learn sklearn.tree: DT module from scikit-learn xgboost: Optimized distributed gradient boosting library	Classification using RF, ET, AdaBoost, GBT Classification using DT Classification using XGBoost
Model Selection and Validation	sklearn.Model_selection: Model selection tools from scikit-learn	Data splitting, cross-validation, learning curves
Model Evaluation	sklearn.metrics: Metrics module from scikit-learn	Evaluation metrics like accuracy, ROC AUC
Preprocessing and Pipeline	sklearn.preprocessing: Preprocessing tools from scikit-learn sklearn.pipeline: Pipeline tools from scikit-learn	Label Encoding, MinMaxScaling Building preprocessing pipelines
Visualization	matplotlib.pyplot: 2D plotting library for Python seaborn: Data visualization library based on matplotlib	Graph plotting Heatmaps for confusion matrices
Mathematical and Scientific Computations	math: Standard Python math library scipy: Library for mathematics, science, and engineering	Mathematical functions like square root Scientific computations
Utility and Miscellaneous	time: Standard Python library for time tasks, os: Standard Python library for OS interaction	Measuring execution time File and directory operations

A. The Modified AVO Results

The used ML classifiers are XGBoost, DT, ET, and HGB. These classifiers were used to evaluate the model performance; these classifiers demonstrated remarkable efficacy, with most achieving near-perfect scores in accuracy, precision, and F1−Score. The detailed results are consolidated in Table V, which outlines the performance metrics along with convergence times, providing a comprehensive view of each classifier's effectiveness.

TABLE V. THE CLASSIFICATION RESULTS FOR ALL CLASSIFIERS

Classifier	Accuracy	Precision	F1−score	Convergence Time
XGBoost	99.77%	99.77%	99.51%	0.0042 s
HGB	99.77%	99.77%	99.52%	0.0042 s
DT	99.75%	99.75%	99.47%	0.0042 s
ET	99.71%	99.71%	99.39%	0.0042 s

In the evaluation of classifiers for binary classification, XGBoost and HGB topped the charts with the highest accuracy, precision, and F1−Score at 99.77%, showcasing their robustness in handling complex classification tasks in

the context. Meanwhile, the RF and HGB demonstrated excellent performance, with near-perfect scores across all metrics. The convergence time for these classifiers was notably quick, enhancing their suitability for real-time applications. Fig. 5 shows the remarkable performance of XGBoost and HGB, as it shows the classifier's performance in binary classification.

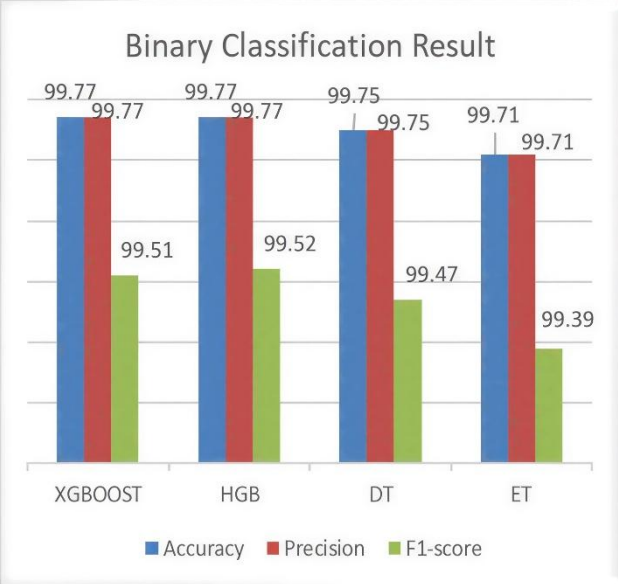

Fig. 5. Binary classification results.

The XGBoost classifier achieved an excellent performance in binary classification. The XGBoost Confusion Matrix (CM) detailed in Fig. 6 shows a small number of FP (the model identified 26 cases as intrusions when they were normal) and FN (the model marked 12 cases as normal when they were intrusions). As for the true predictions, the XGBoost correctly identified 12358 instances as normal and 3839 as intrusions.

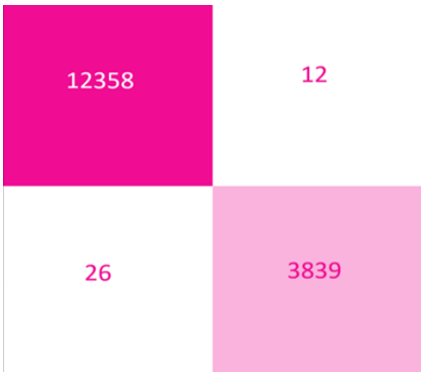

Fig. 6. XGBoost confusion matrix.

The HGB classifier achieved an excellent performance in binary classification. The HGB CM detailed in Fig. 7 shows a small number of FP (the model identified 27 cases as intrusions when they were normal) and FN (the model marked 10 cases as normal when they were intrusions). As for the true predictions, the HGB correctly identified 12,360 instances as normal and 3,838 as intrusions.

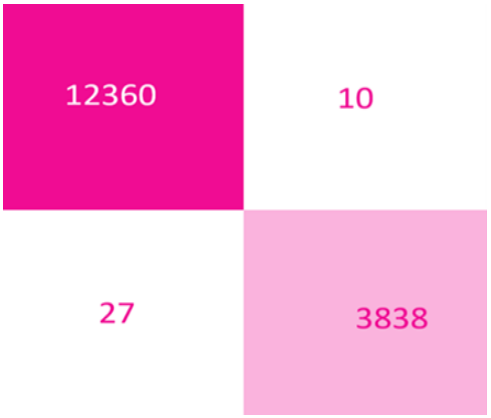

Fig. 7. Histogram-based gradient boosting confusion matrix.

The DT classifier achieved a good performance in binary classification. The DT CM detailed in Fig. 8 shows a small number of FP (the model identified 23 cases as intrusions when they were normal) and FN (the model marked 18 cases as normal when they were intrusions). As for the true predictions, the DT correctly identified 12,352 instances as normal and 3,842 as intrusions.

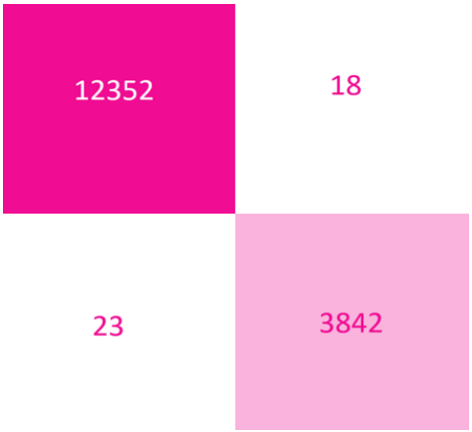

Fig. 8. Decision trees confusion matrix.

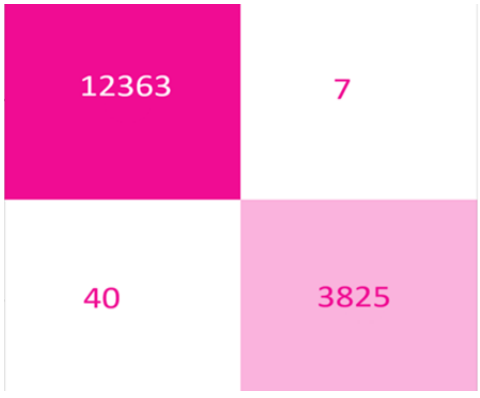

Fig. 9. Extra trees confusion matrix.

The ET classifier achieved a good performance in binary classification. The ET CM detailed in Fig. 9 shows a small number of FP (the model identified 40 cases as intrusions when they were normal) and FN (the model marked 7 cases as normal when they were intrusions). The

ET correctly identified 12,363 instances as normal and 3,825 as intrusions.

B. The Modified AVO vs. The Original AVO

The modified AVO demonstrated enhanced accuracy across various ML algorithms for binary classification of network traffic. Notable improvements include XGBoost, which saw an accuracy increase from 99.72% to 99.77%, and DT, which improved from 99.60% to 99.75%. ET also showed a slight increase from 99.6% to 99.71%. Similarly, HGB accuracy moved from 99.72% to 99.77%. The bar graph in Fig. 10 highlights these improvements in accuracy, comparing the results of the modified AVO versus the original AVO.

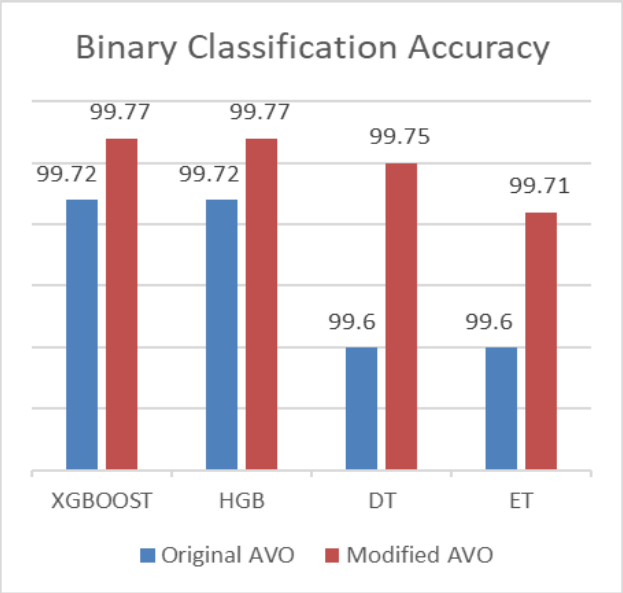

Fig. 10. Modified AVO vs. original AVO binary classification accuracy comparison.

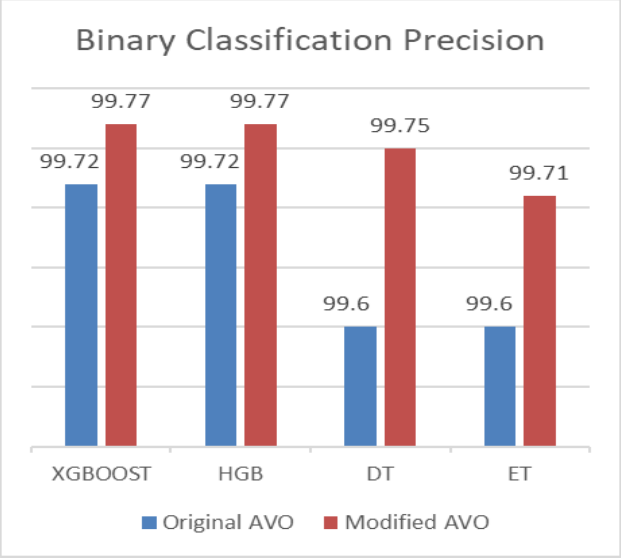

Fig. 11. Modified AVO vs. original AVO binary classification precision comparison.

The modified AVO showed significant improvements in precision for several ML algorithms used in the binary classification of network traffic, as shown in Fig. 11. Notably, DT precision saw a considerable rise from 99.60% to 99.75%. XGBoost's precision increased slightly from 99.72% to 99.77%, aligning closely with HGB, which also improved from 99.72% to 99.77%. ET also showed a slight increase from 99.6% to 99.71. The accompanying bar graph vividly illustrates these enhancements in precision, comparing the results from the original AVO to those achieved with the modified AVO.

The modified AVO exhibited advancements in F1−Score for various ML algorithms tailored for binary classification tasks. For instance, DT saw an improvement in the F1−Score from 99.16% to 99.47%. XGBoost also showed a slight enhancement, increasing from 99.4% to 99.51%. Similarly, ET made a modest gain, improving from 99.15% to 99.39%. HGB displayed a marginal increase from 99.42% to 99.52%. Fig. 12 shows the improvements in the F1−Score for the various ML algorithms using the modified AVO.

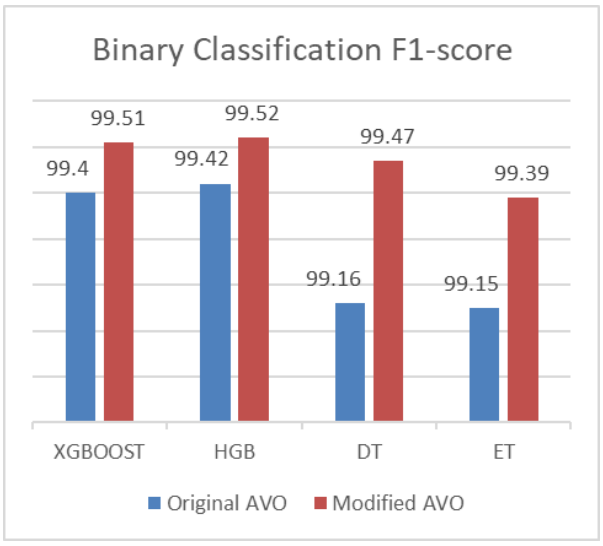

Fig. 12. Modified AVO vs. original AVO binary classification F1−Score comparison.

The modified AVO outperforms the original AVO across various ML algorithms and in binary classification of network traffic in terms of accuracy, precision, and F1−Score.

V. Conclusion

The study results show that the modified AVO had an improved performance in intrusion detection in both binary and multiclass classification tasks. The AVO model showed excellent results in binary classification tasks, reaching almost perfect scores with classifiers such as XGBoost, HGB, DT, and ET, with all scoring above 99.70% in accuracy, precision, and F1−Score. Improved performance of the modified AVO using crossover. The crossover increased the performance of ML algorithms used compared to using the original AVO. This improvement is shown in the accuracy, precision, and F1−Score for the ML algorithms. This outlines the implemented crossover effectiveness in increasing the

AVO algorithm performance. The modified AVO model outperforms other models. The results show that the proposed model achieves higher results in accuracy and precision.

Although the UNSW-NB15 dataset is a well-established benchmark for intrusion detection research, it may not fully capture all real-world network traffic characteristics or newly emerging attack behaviors, which represents a limitation of this study. Future work will therefore extend the proposed IDS to Software-Defined Networking (SDN) environments with a focus on DDoS detection using deep learning models, adapt the framework to QoS-aware software-defined IoT systems to jointly consider security and performance constraints, and evaluate the approach on additional benchmark datasets to further validate its generalization capability across diverse network environments and traffic conditions.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Yasameen A. Al-Shadeedi conceived the research idea, designed the proposed intrusion detection framework, implemented the modified African Vultures Optimization Algorithm with the crossover enhancement, conducted the experiments, analyzed the results, and wrote the original draft of the manuscript; Mosleh Abualhaj contributed to the methodological design, provided technical guidance on machine learning and feature selection, and reviewed the manuscript; Ahmad Abu-Shareha assisted in dataset preparation, experimental validation, and result interpretation; Mohamed Yousif supported the performance evaluation, comparative analysis, and contributed to refining the experimental results; Anusha Achuthan contributed to manuscript editing, formatting according to journal guidelines, and critical revision of the paper; All authors reviewed and approved the final version of the manuscript.

REFERENCES

[1] M. K. Alghamry, E. H. Abuhamdi, M. Shehab, A. Smerat, and L. Abualigah, "Intelligent robotics using optimization algorithms: A survey," *Mastering the Minds of Machines: A Journey into Deep Learning and AI*, pp. 187−193, 2025.

[2] M. A. Almaiah *et al*., "Classification of cybersecurity threats, vulnerabilities and countermeasures in database systems," *Computers, Materials and Continua*, vol. 81, no. 2, 2024.

[3] A. R. Shorman *et al*., "Adaptive intrusion detection for IoT networks using artificial immune system techniques: A comparative study," *Journal of Robotics and Control*, vol. 6, no. 2, pp. 570−582, 2025.

[4] R. Alabdallat, M. Abualhaj, and A. A. Shareha, "Android malware detection using a modified dwarf mongoose algorithm," *International Journal of Intelligent Engineering and Systems*, vol. 18, no. 8, 2025.

[5] N. James. 90 + Cyber crime statistics 2026: Cost, industries and trends. [Online]. Available: https://www.getastra.com/blog/security-audit/cyber-crime-statistics/

[6] A. Djenna, E. Barka, A. Benchikh, and K. Khadir, "Unmasking cybercrime with artificial-intelligence-driven cybersecurity analytics," *Sensors*, vol. 23, no. 14, pp. 6302–6302, Jul. 2023.

[7] A. A. Shareha and A. Amer, "Improving linear support vector machine classifier using mutual information feature weighting for intrusion detection system," in *Proc. Seventh International Conference on Image, Video Processing, and Artificial Intelligence*, 2025, vol. 13731, pp. 231−238.

[8] M. M. Abualhaj, S. N. A. Khatib, M. A. Zyoud, I. Qaddara, M. O. Hiari, and S. M. A. Aldossary, "Enhanced network communication security through hybrid dragonfly-bat feature selection for intrusion detection," *Journal of Communications*, vol. 20, no. 5, pp. 607–618, 2025.

[9] M. Hiari, Y. Alraba'nah, and I. Qaddara, "A deep learning-based intrusion detection system using refined LSTM for DoS attack detection. engineering," *Technology and Applied Science Research*, vol. 15, no. 4, pp. 25627−25633, 2025.

[10] A. A. Mosuli, M. Abualhaj, A. A. Shareha, M. Yous, and M. Daoud, "Enhancing intrusion detection system performance using a modified grey wolf optimizer," *Annals of Emerging Technologies in Computing (AETiC),* 2026, pp. 21-44, vol. 10, no. 1.

[11] M. Otair *et al*., "Enhanced arithmetic optimization algorithm for intrusion detection in wireless sensor networks," *International Journal of Communication System*s, vol. 38, no. 10, 2025.

[12] E. Hanandeh *et al*., "Optimizing deep learning scalability: Harnessing distributed systems and cloud computing for next-generation AI," *Mastering the Minds of Machines*, pp. 142−148, 2025.

[13] M. H. Almomani *et al*., "Reinforcement learning: Machines that learn by doing," *Mastering the Minds of Machines: A Journey into Deep Learning and AI*, vol. 7, p. 42, 2025.

[14] 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, p. 34, 2025.

[15] I. Qaddara and Y. Alraba'nah, "Enhancing requirements classification using machine learning techniques," *SN Computer Science*, vol. 6, no. 6, p. 649, 2025.

[16] M. Amin *et al*., "A hybrid brain stroke prediction framework: Integrating feature selection, classification, and hyperparameter optimization," *Engineering Reports*, vol. 7, no. 7, 2025.

[17] M. M. Abualhaj, M. O. Hiari, A. Alsaaidah, M. A. Zyoud, and S. A. Khatib, "Spam feature selection using firefly metaheuristic algorithm," *Journal of Applied Data Sciences*, vol. 5, no. 4, pp. 1692–1700, 2024.

[18] M. M. Abualhaj, M. O. Hiari, A. Alsaaidah, and M. M. A. Zyoud, "Comparative analysis of whale and Harris Hawks optimization for feature selection in intrusion detection," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 37, no. 1, pp. 179–185, 2025.

[19] M. M. Abualhaj, Q. Y. Shambour, A. A. A. Shareha, S. N. A. Khatib, and A. Amer, "Enhancing malware detection through self-union feature selection using gray wolf optimizer," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 37, no. 1, pp. 197–205, 2025.

[20] M. M. Abualhaj, S. Al-Khatib, M. O. Hiari, and Q. Y. Shambour, "Enhancing spam detection using hybrid of Harris hawks and firefly optimization algorithms," *Journal of Soft Computing and Data Mining*, vol. 35, no. 2, pp. 161–174, 2024

[21] M. A. A. Betar *et al*., "Ameliorated elk herd optimizer for global optimization and engineering problems," *Artificial Intelligence Review*, vol. 58, no. 11, pp. 1−80, 2025.

[22] M. M. Abualhaj *et al*., "Spam detection boosted by firefly-based feature selection and optimized," *Int. J. Adv. Soft Comput. Appl.*, vol. 17, no. 3, pp. 1–19, 2025.

[23] O. Almomani *et al*., "Enhance URL defacement attack detection using particle swarm optimization and machine learning," *Journal of Computational and Cognitive Engineering*, vol. 4, no. 3, 2025.

[24] S. Rajeyyagari *et al.*, "Convolutional neural network-based African vulture optimization algorithm for the enhancement of cybersecurity in the blockchain-based smart grid," *Multimedia Tools and Applications*, vol. 83, no. 20, 2023.

[25] M. M. Abualhaj and S. N. A. Khatib, "Using decision tree classifier to detect Trojan Horse based on memory data," *Telecommunication Computing Electronics and Control*, vol. 22, no. 2, pp. 393–400, 2024.

[26] Alsirhani, M. M. Alshahrani, A. M. Hassan, A. I. Taloba, R. M. A. El-Aziz, and A. H. Samak, "Implementation of African vulture optimization algorithm based on deep learning for cybersecurity

intrusion detection," *Alexandria Engineering Journal*, vol. 79, pp. 105–115, 2023.
[27] L. Zareian, J. Rahebi, and M. J. Shayegan, "Bitterling Fish Optimization (BFO) algorithm," *Multimedia Tools and Applications*, pp. 1–34, 2024.
[28] L. Zhiqiang, G. M. U. Din, B. Bing, J. Jianchao, Y. Ye, and L. Zhijun, "Modeling network intrusion detection system using feed-forward neural network using UNSW-NB15 dataset," in *Proc. IEEE 7th Int. Conf. Smart Energy Grid Engineering (SEGE)*, 2019, pp. 299–303.
[29] M. Shushlevska *et al*., "Anomaly detection with various machine learning classification techniques over UNSW-NB15 dataset," 2022.
[30] G. Ramasubramanian and S. Rajaprakash, "An Avant-Garde African Vulture Optimization (A2VO) based deep RNN-LSTM model for 5G-IoT security," *Journal of Advanced Research in Applied Sciences and Engineering Technology*, vol. 32, no. 1, pp. 1−17, 2022.
[31] S. B. Baghbadorani, S. A. Johari, Z. Fakhri, E. K. Shahmirzadi, S. N. Shavkatovich, and S. Lee, "A new version of African vulture optimizer for apparel supply chain management based on reorder decision-making," *Sustainability*, vol. 15, no. 1, p. 400, 2022.
[32] J. Yoon, J. Jordon, and M. Schaar, "Gain: Missing data imputation using generative adversarial nets," in *Proc. International Conference on Machine Learning*, 2018. pp. 5689−5698.
[33] M. Ahmad *et al*., "Intrusion detection in internet of things using supervised machine learning based on application and transport layer features using UNSW-NB15 data-set," *EURASIP Journal on Wireless Communications and Networking*, pp. 1−23, 2021.
[34] O. I. Ramadan *et al*., "Enhancing breast cancer classification based on BPSO feature selection and machine learning techniques," *Engineering, Technology and Applied Science Research*, vol. 15, no. 3, pp. 23907−23916, 2025.
[35] Z. Wang, "Deep learning-based intrusion detection with adversaries," *IEEE Access*, vol. 6, pp. 38367−38384, 2018.
[36] M. K Dahouda and I. Joe, "A deep-learned embedding technique for categorical features encoding," *IEEE Access*, vol. 9, pp. 114381−114391, 2021.
[37] Y. A. Maz, M. Anbar, S. Manickam, M. M. Abualhaj, S. A. Almalki, and B. A. Alabsi, "Transfer learning-based approach with an ensemble classifier for detecting keylogging attack on the internet of things," *Computers, Materials and Continua*, pp. 1–10, Jan. 2025.
[38] R. Alabdallat, M. Abualhaj, and A. A. 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.
[39] N. Islam, "DTization: A new method for supervised feature scaling," arXiv preprint arXiv:2404.17937, 2024.
[40] M. M. Abualhaj, S. N. A. Khatib, M. A. Alsharaiah, and M. O. Hiari, "Performance comparison of whale and Harris Hawks optimizers with network intrusion prevention systems," *Journal of Applied Data Sciences*, vol. 5, no. 4, pp. 1530–1538, 2024.
[41] P. Agrawal, H. F. Abutarboush, T. Ganesh, and A. W. Mohamed, "Metaheuristic algorithms on feature selection: A survey of one decade of research (2009–2019)," *IEEE Access*, vol. 9, pp. 26766–26791, 2021.
[42] J. D. Knowles, R. A. Watson, and D. W. Corne, "Reducing local optima in single-objective problems by multi-objectivization," in *Proc. Int. Conf. Evolutionary Multi-Criterion Optimization*, 2001, pp. 269–283.
[43] D. C. Dang *et al.*, "Escaping local optima with diversity mechanisms and crossover," in *Proc. Genetic and Evolutionary Computation Conf. (GECCO)*, 2016, pp. 645–652.
[44] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Systems*, vol. 9, no. 2, pp. 115–148, 1995.
[45] 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, 2025.
[46] R. S. Ghoneim, "Machine learning for adaptive facade design: Enhancing thermal performance in urban architecture," *Civil Engineering and Architecture*, vol. 13, no. 2, pp. 1273–1288, 2025.
[47] B. Xu *et al*., "Machine learning-assisted computation of water activity for ionic liquid-based aqueous ternary elements," *Desalination and Water Treatment*, 101484, 2025.
[48] I. S. Samanta *et al*., "Artificial intelligence and machine learning techniques for power quality event classification: A focused review and future insights," *Results in Engineering*, 103873, 2024.
[49] M. M. Abualhaj *et al*., "A paradigm for DoS attack disclosure using machine learning techniques," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 3, 2022.
[50] H. A. Mimi *et al*., "Improved intrusion detection system to alleviate attacks on DNS service," *Journal of Computer Science*, vol. 19, no. 12, pp. 1549–1560, 2023.
[51] A. A. A. Shareha *et al*., "Diabetes prediction using hybrid supervised and unsupervised techniques based on PIMA dataset," *Journal of Artificial Intelligence and Technology*, Nov. 2025.