IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 15, No. 2, April 2026, pp. 1219~1235
ISSN: 2252-8938, DOI: 10.11591/ijai.v15.i2.pp1219-1235

An efficient approach for cyber-attack detection by using machine learning and deep learning algorithms

Yasir Hussein Shakir[1], Mahmoud Mohamed Abdelhamied[2], Eshaq Aziz Awadh AL Mandhari[3], Ali Alkhazraji[4], Naglaa M. Reda[5,6]

[1]Department of Engineering, College of Graduate Studies, Universiti Tenaga Nasional, Kajang, Malaysia
[2]Department of Data Science and Artificial Intelligence, Faculty of Information Technology, Al-Ahliyya Amman University, Amman, Jordan
[3]Department of Computing and Technology, Graduate School of Technology, Asia Pacific University of Technology and Innovation, Kuala Lumpur, Malaysia
[4]Department of Computer Science, Faculty of Sciences, Lebanese University, Beirut, Lebanon
[5]Department of Computer Science, Faculty of Computers and Information Technology, The Future University in Egypt, Cairo, Egypt
[6]Department of Mathematics, Faculty of Science, Ain Shams University, Cairo, Egypt

Article Info

Article history:

Received May 5, 2025
Revised Jan 5, 2026
Accepted Feb 6, 2026

Keywords:

Attack detection
Bee algorithm
Cybersecurity
Deep learning
Gated recurrent unit
K-nearest neighbor algorithm
Machine learning

ABSTRACT

The rise of cyber-attacks necessitates intrusion detection systems (IDS) that provide high detection accuracy and computational efficiency. Most existing machine learning (ML) and deep learning (DL) approaches are complex, take long training time, lack transparency, and are hard to interpret. To address these challenges, this research introduces a new meta-heuristic IDS optimization framework using the artificial bee colony (ABC) algorithm. We developed two hybrid models, KNN + Bee, which combines ABC to automate feature selection and k-nearest neighbors (KNN) algorithm fine-tuning, as well as gated recurrent unit (GRU) + Bee, where ABC optimizes the GRU network architecture and hyperparameters. By leveraging swarm intelligence, our models improve classifier performance without complex architecture. We tested the presented models on NSL-KDD, UNSW-NB15, and CIC-DDoS2019 benchmark datasets. Performance was evaluated against both conventional ML and sophisticated DL baselines. Experimental results indicated that the hybrid KNN + Bee and the GRU + Bee models consistently surpassed their respective baseline and perform competitively against top methods. Optimized constructs registered high accuracy, F1-score, and Matthews correlation coefficient (MCC), besides retaining great generalizability across various attack scenarios. Our proposal offers a reasonable compromise between detection precision and thriftiness, making it appropriate choice for scalable, real-time cyber defense systems.

Corresponding Author:

Naglaa M. Reda
Department of Computer Science, Faculty of Computers and Information Technology
The Future University in Egypt
Cairo, Egypt
Email: naglaa.saeed@fue.edu.eg

1. INTRODUCTION

The exponential growth of the digital universe has opened unparalleled doors to communication, e-commerce, and data-based services. However, it has brought in critical vulnerabilities in intrusion detection, evasion of phishing attacks, and evasion of denial-of-service attacks. As the cyber threats become

highly sophisticated and dynamic in nature, limitations of traditional rule-based and one-dataset-based approaches are becoming apparent.

The newer technologies such as artificial intelligence (AI), machine learning (ML), and deep learning (DL) have been highly utilized to tackle these problems, but scalability problems, across-dataset generalization and optimization efficacy are left unchecked [1]. This has caused the evolution of grave security problems to cover data from the vast challenges posed by cybersecurity constructors. The foundation of ensuring data security and secrecy to every form of businesses, governments and even private individuals is cybersecurity. Data is sent and received over the internet in an open environment where it is manipulated and hijacked by unauthorized parties. Denial of service (DoS), root to local attack (R2L), user to root attack (U2R), and probing (Probe) are amongst the very frequent cyber-attacks [2]. DoS attacks prevent legal user utilization of system resources. It is a killer tool that brings down servers and brings to a stand stop service by sending an abundant amount of packets and causing a server or network to come to standstill. Further, Probe attacks are designed to reveal information regarding a target system vulnerability by the enactment of carefully selected sequences of actions and observation of the system or the intrusion detection system (IDS) reactions [3]. A Probe is crafted to be detected by their target and reportable with a signature-specific "fingerprint" in the report.

Nonetheless, U2R attacks begin to be successful in establishing a user session preferentially by an interactive shell or by opening up a TELNET window on the remote host [4]. Through the employment of a combination of conventional methods, the invader attempts to escalate his privileges in steps until he reaches privileges of the super-user. An attacker in an R2L attack carries out a remote-to-local attack by sending packets to the potential host with a goal of unveiling security vulnerabilities through which the invader may exploit a local user's privileges. Admittedly, penetration as an authorized user may be a crucial preparation in anticipation of eventually executing a user-to-root attack. According to Davis *et al.* [5], an intelligent ML model is offered to enhance the detection of cyberattacks, validated on multiple datasets. Its main aim was to secure data exchange between users over the internet. The triple-layer model, which combines signature, network traffic, and machine-learning-enhanced behavioral features, offers encouraging improvements in detection accuracy and early threat identification, even though ransomware is still a quickly developing and extremely lucrative cyberthreat [6]. On the other hand, to detect fraudulent behavior, an optimized extreme learning machine integrated with synthetic minority over-sampling technique (SMOTE) was suggested in order to improve detection robustness and evaluation reliability, especially when there is a significant data imbalance in digital payment systems [7].

According to recent research, AI–software defined network (SDN) frameworks that dynamically analyze internet control message protocol (ICMP) traffic and enforce automated mitigation rules greatly increase the accuracy of Smurf attack detection and decrease response times. The need for more robust and self-learning architectures is highlighted by the fact that current solutions still have shortcomings in terms of scalability, real-time model adaptability, and robustness against changing spoofing techniques [8]. Comparative analyses of the performance of classifiers on standard benchmarks such as NSL-KDD are available. The classifiers investigated include logistic regression (LR), support vector machine (SVM), and artificial neural network (ANN) classifiers [9]. Hybrid models based on DL techniques help in the detection of attacks in varied datasets [10]. Hybrid models for feature selection can therefore improve the efficiency of the classifiers and reduce costs accordingly [11]. Recent works on supervised ML for distributed denial of service (DDoS) detection [12], as well as the use of the convolutional neural network-long short-term memory (CNN-LSTM) hybrid approach [13], show their effectiveness in attack classification. Hiari *et al.* [14] propose an LSTM-based model for accurate DoS attack detection using the NSL-KDD dataset. Standard benchmarks for classification include the NSL-KDD dataset [15] and the CIC-DDoS dataset [16]. Parameter optimization techniques, such as the optimization of k-nearest neighbors (KNN) parameters [17], can therefore improve the detection process. However, striking a balance between accuracy, training speed, and interpretability is still an impediment in reference [18].

This paper presents an efficient solution to overcome the limitation of the KNN classifier by introducing general hybrid model (KNN + Bee) to involve the Bee optimizer to fine-tune all hyperparameters of the KNN classifier without a reduction in input dimension. The usage of Bee is neither to choose features, but as an efficient optimizer instrument to be used over the hyperparameters of the KNN classifier itself. An integrated usage of Bee with KNN learning model is able to deliver the best possible configuration to certain key parameters such as low sample split, number of estimators or number neighbours and metric. The former is aimed at achieving maximum precision in classification and latter at maintaining low false positives (FP).

The contributions are thus concluded to be as follows. First, a novel hybrid IDS is proposed, where the artificial bee colony (ABC) algorithm automatically optimizes the hyperparameters of both KNN classifier enhancing robustness. Second, a meta-heuristically enhanced DL model is developed by integrating ABC optimization with a gated recurrent unit (GRU) network, effectively capturing temporal dependencies

in network data that reduces model complexity and training costs. Third, the proposal is comprehensively validated on three heterogeneous (NSL-KDD, UNSW-NB15, and CIC-DDoS2019), confirming the framework's reliability and adaptability against a wide spectrum of intrusions and DDoS attacks. Fourth, a multidimensional evaluation is employed using a suite of metrics such as accuracy, precision, recall, F1-score, Matthews correlation coefficient (MCC), training and testing time to ensure a balanced view of predictive power, robustness and computational efficiency. Finally, the local interpretable model-agnostic explanations (LIME) framework is integrated to provide transparent human understandable explanations for model predictions fostering trust and facilitating practical deployment in security operations.

In the present paper, the content is splitted into six sections. Section 2 will discuss prior studies in previous work. Section 3 gives a brief technical background of the materials and methods including the proposed algorithm is presented of several datasets. The explanation about the experimental results and discusses the critical analyze is shown in section 4. The conclusion of the work comes in section 5.

2. PREVIOUS WORKS

Application of ML and DL algorithms in the detection of cyber-attacks generated significant interest in network security and internet of things (IoT) installations. The literature demonstrates a consistent evolution from traditional ML models towards sophisticated DL and hybrid architectures to improve detection accuracy robustness and efficiency. This section reviews contributions relevant to network-based intrusion detection. Using the NSL-KDD dataset, Reddy *et al.* [9] assessed ML models for intrusion detection. Their findings showed that a properly calibrated feedforward ANN performed exceptionally well, achieving nearly flawless scores of 99.79% F1-scores, 100% recall, 99.58% accuracy, and 99.58% precision. Simpler models like SVM and LR are still feasible and useful options for low-resource contexts, the paper contends, even though the ANN performed best. Boadi [19] used eight ML models for intrusion detection of NSL-KDD dataset. With the highest accuracy of 88.30, recall 82.30%, F1-score of 88.90, and area under the curve (AUC) of 97.70%, random forest was the best performer. The study concludes that ensemble methods like random forest are best for detecting complex patterns while simpler models like LR and KNN remain valuable for interpretability and efficiency.

To address sophisticated cyber threats, Kumar *et al.* [20] developed a novel IDS that integrates the feature learning capabilities of a CNN with the ensemble classification power of random forest. Evaluated on the NSL-KDD dataset the hybrid model demonstrated superior performance, the best accuracy of 98.73% and an F1-score of 96.50%. The authors highlight the model's robustness against novel attacks and class imbalance positioning it as a scalable solution for real-time intrusion detection. Ghajari *et al.* [21] designed an IDS based on hyperdimensional computing (HDC). This approach is effective at analyzing high-dimensional data and detecting both known and unique assault patterns. The autoencoder model tested on the NSL-KDD dataset achieved an accuracy of 91.55%, demonstrating strong potential for securing IoT networks against sophisticated threats. Sharma and Kumar [10] provide a hybrid CapsNet + BiLSTM a DL models, including capsule networks (CapsNet) and bidirectional long short-term memory (BiLSTM). Experimental findings show that the suggested method is effective, reaching a high detection accuracy of 97.81%, precision of 96.00%, recall of 97.00%, and F1-score of 96.00%. Their work emphasizes the growing adoption of DL architectures in network traffic monitoring, which aligns with the broader trend of leveraging complex models for improved detection accuracy.

According to Araújo *et al.* [22], traditional ML methods were evaluated to categorize network attacks on three data sets: HIKARI-2021, UNR-IDD, and UNSW-NB15 and then used a stacking ensemble of tree-based boosting classifiers-XGBoost, LightGBM, and Cat-Boost-for classification. The stacking ensemble gave the best performance while achieving F1-score of 93.70% on CIC-UNSW-NB15. The results are investigation, namely on the less investigated UNR-IDD and CIC-UNSW-NB15 datasets. Three datasets NSL-KDD, CIC-IDS2017, and UNSW-NB15 are used in this study to assess many ML techniques for network attack classification [23]. Exhaustive feature selection (EFS) + least square (LS)-SVM is also used for classification. The best accuracy of 93.30%, precision of 1.00, recall of 98.00, and F1-score of 98.00% for CIC-UNSW-NB15, the EFS + LS-SVM produced the best results. When compared to other models, the model takes the shortest amount of time to train on any dataset. These findings demonstrate the LS-SVM-based model's appropriateness for real-time intrusion detection applications. Rafrastara *et al*. [24] to address multi-class network anomaly detection. Three tree-based ensemble algorithms are used in this study, namely random forest, XGBoost, and AdaBoost. Their experimentation, conducted using the UNSW-NB15 dataset, yielded significant results, and the feature space was efficiently reduced. Three of those ensemble techniques performed better than baseline models that employed a single decision tree classifier when combined with an improved Gini index. With 97.30% accuracy, recall, and precision and a 96.90% F1-score, XGBoost with Gini index produced greatest results. This method can reduce the number of characteristics while increasing the algorithm's speed.

Abiramasundari and Ramaswamy [12] developed an enhanced distributed DDoS attack detection (EDAD) framework using principle component analysis (PCA) and various ML classifiers. On several CIC-IDS datasets, random forest achieved the best accuracy of 98.9% on CIC-IDS2017, while random forest and KNN were best on CIC-DDoS2019 of 98.70%, and SVM performed the best on CIC-IDS2018 of 98.70%. Al-Hasani *et al.* [13] introduced a hybrid DL model to combat impactful DDoS attacks. The framework integrates a CNN to extract spatial features with a LSTM network to capture temporal dependencies in traffic data. Evaluated on CIC-DDoS2019 dataset, model demonstrated exceptional performance, achieving 99.63% accuracy and a 99.71% micro-AUC, outperforming standalone models. This high performance despite dataset complexity and class imbalance is attributed to its dual capability to learn both spatial and temporal patterns.

Gankotiya *et al.* [25] develop a deep convolutional neural network (DCNN) model to address the constraints of classical DDoS detection in dynamic wireless mesh networks (WMNs). On the CIC-DDoS2019 dataset, their cross-layer solution outperformed other approaches in terms of packet delivery ratio and end-to-end delay. The model gets best accuracy of 99.14%, precision of 98.81%, and F1-score of 97.34 %. These solid results outperform indicate the model's usefulness in accurately reliably detecting DDoS attacks in the complicated context of WMNs. Dilshad *et al.* [26] utilized the XGBoost learning method with high total precision of 94.00% on the CIC-DDoS2019 dataset. Although the corresponding precision 71.00%, recall 69.00%, and F1-score 69.00% indicate flaws in minimizing FP over false negatives (FN), such centrally developed solutions succeed in laboratory experiments but do not take realistic limitations of distributed computing like the internet of vehicles (IoV) into account. Moving forward from these foundations, follow-on research has established a federated learning framework achieving competitive detection performance with an overall average score of 91% across all modes of attack yet with data confidentiality and computational efficiency in consideration. A consistent theme across the literature is the sacrifice of efficiency, explainability, and tractable training time in exchange for slight increments in precision. There is a pronounced gap for a solution that balances high detection performance with operational practicality. Meta-heuristic algorithms offer an interesting direction towards filling the gap through the optimization of basic, more explainable models to reach competitively superior performances over their elaborate counterparts.

In summary, the studies to date have utilized several datasets such as NSL-KDD, UNSW-NB15, and CIC-DDoS2019, and variety of ML and DL algorithms techniques to cyber-attack detection in network traffic. However, to the best of the knowledge there no previous studies have developed GRU and KNN models optimized with nature-inspired algorithms such as the ABC, namely GRU + Bee and KNN + Bee for cyber-attack detection. A comparative analysis of the previous related work is presented in Table 1.

Table 1. Analysis of comparison for the relevant work

Reference/ year	Dataset	Technique	Outcome (%)	Limitations
[9], 2025	NSL-KDD	ANN	Accuracy =99.58	No training time and not explainable
[10], 2025	UNSW-NB15	Hybrid CapsNet + BiLSTM	Accuracy =97.81	No training time and not explainable
[11], 2025	UNSW-NB15	Hybrid IGRF-RFE	Accuracy =84.24	No explainable and weak results
[12], 2025	CIC-DDoS2019	PCA + random forest and KNN	Accuracy =98.70	No training time and not explainable
[13], 2025	CIC-DDoS2019	CNN + LSTM	Accuracy =99.63	Not explainable and complicated
[18], 2025	NSL-KDD	Random forest	Accuracy =88.30	Weak results
[19], 2025	NSL-KDD	CNN + random forest	Accuracy =98.73	No training time and complicated
[20], 2025	NSL-KDD	Autoencoder	Accuracy =91.55	Suboptimal results and not explainable
[21], 2025	UNSW-NB15	Stacking ensemble	F1-score =93.70	Not enough evolution and not explainable
[22], 2025	UNSW-NB15	EFS + LS-SVM	Accuracy =93.30	Not enough evolution and not explainable
[23], 2025	CIC-DDoS2019	DCNN	Accuracy =99.14	Not explainable and no time training
[24], 2025	CIC-DDoS2019	XGBoost	Accuracy =94.00	Not explainable and suboptimal results

3. MATERIALS AND METHOD

In this section, a comprehensive study of the dataset that was utilized in the experiments is provided. In addition, the methodology is proposed for developing an intelligent model for enhancing the detection of cyberattacks to determine whether a network packet is under attack. The overall research framework consists of many main phases: i) data acquisition, ii) pre-processing and hyperparameter optimization using the ABC algorithm, iii) model training, iv) model evaluation, and v) explainable AI integration. The detailed methodology diagram is depicted in Figure 1.

3.1. Data acquisition

3.1.1. NSL-KDD

The NSL-KDD dataset was specially developed to offset the unavoidable imperfections of the original KDD CUP 99 dataset [14], typically used in intrusion detection research. Unlike the original dataset,

NSL-KDD eliminates duplicate rows and achieves a more balanced number of normal and attack samples to reduce bias during training and testing. It contains separate training and testing sets with a sufficient number of samples so that researchers do not need to rely on random sampling, making it possible to use the entire dataset directly. This leads to more stable and comparable evaluation results across different studies of IDSs.

3.1.2. UNSW-NB15

The UNSW-NB15 dataset was generated in the UNSW Canberra Cyber Range Lab using the IXIA PerfectStorm tool, which produced raw network traffic containing a combination of normal activity and modern attack behaviors. Approximately 100 GB of raw traffic data was captured with the tcpdump utility and then processed using Argus and Bro-IDS tools to extract 49 features along with class labels. The dataset covers nine types of malicious activity: fuzzers, analysis, backdoors, DoS, exploits, generic, reconnaissance, shellcode, and worms. Compared with traditional datasets, UNSW-NB15 provides a more realistic blend of contemporary attack patterns and legitimate traffic, making it a tougher and more effective benchmark for evaluating IDS models [27].

3.1.3. CIC-DDoS2019

The CIC-DDoS2019 dataset was introduced by the Canadian Institute for Cybersecurity (CIC) as a comprehensive benchmark for DDoS detection. It includes a wide range of attack scenarios, such as UDP, TCP, HTTP, and ICMP flooding attacks, along with benign traffic that closely reflects real-world conditions. The data was captured in PCAP format and then converted into flow-based records using CICFlowMeter, which includes features such as timestamps, IP addresses, ports, protocols, and flow characteristics. This dataset is especially valuable for intrusion detection research, as it captures large-scale modern DDoS attack behaviors and provides labeled instances for supervised learning [16].

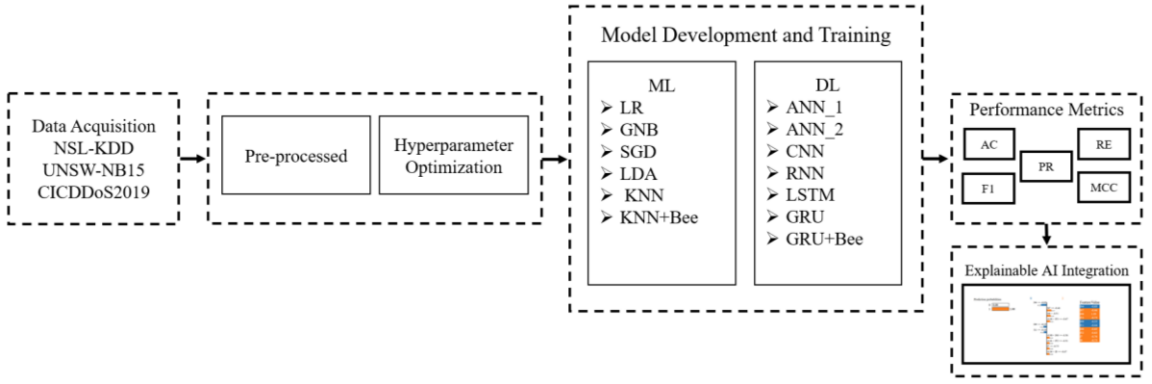

Figure 1. Overall framework of the proposed methodology

3.2. Preprocessing dataset

To utilize due to their diversity in attack categories and representation of heterogeneous network environments. A systematic preprocessing was applied comprising the following steps:

i) Label transformation: to simplify the classification task into a binary detection problem, all original multi-class attack categories such as DoS, Probe, R2L, and U2R, were consolidated into a single "attack" class, while normal traffic was labeled as "benign". This ensured that the model focused on distinguishing between malicious and legitimate traffic to (1) and (2).

$$Y = \{y1, y2, \dots, yn\}\ y_i \in \{DoS, Probe, R2L, U2R, Benign\} \quad (1)$$

$$f(y_i) = \begin{cases} Benign, & if\ y_i = Benign \\ Attack, & otherwise \end{cases} \quad (2)$$

ii) Label encoding: the categorical labels (benign and attack) were converted into numerical format to facilitate compatibility with ML algorithms and scikit-learn LabelEncoder was used to resulting in binary numerical labels such as 0 for "benign" and 1 for "attack".

iii) Feature standardization: to normalize the range of numerical features and improve the stability of training the "StandardScaler" technique was applied. Transformation where x_{ij} is the value of the j

feature for the i sample, u_j is the meaning, and σ_j j is the standard deviation of the jfeature. This ensures that all features have zero meaning and unit variance, thereby mitigating scale-related bias across features. Therefore, the features are scaled according to (3).

$$x_{ij} = \frac{x_{ij} - u_j}{\sigma_j} \quad (3)$$

iv) Data splitting: the processed datasets were partitioned into training and testing subsets using a 70:30 ratio. The training set (70%) was applied in model development, and the test set (30%) was reserved for unbiased performance evaluation. A fixed random state was applied to ensure reproducibility of experimental results.

3.3. Proposed meta-heuristic optimized models

The goal of this study is to apply the ABC meta-heuristic algorithm to optimize two different types of classifiers, namely ML and DL models. Specifically, the ABC algorithm is utilized to enhance model performance via optimizing critical parameters. Through this optimization strategy, the study aims to improve classification accuracy, robustness, and overall generalization capability across different datasets.

3.3.1. K-nearest neighbors optimization using artificial bee colony

In this hybrid approach, the ABC algorithm is employed to automate selection of the optimal number of neighbors K and the most relevant feature subset and addresses the "curse of dimensionality" and improves the inference speed of the KNN classifier. Algorithm 1 presents the pseudocode of the KNN + Bee proposed detector for securing cybers. In the declaration, a class named "bee" is defined, with each instance of Bee representing a possible solution with certain KNN hyperparameters (labelled metric and n_neighbors). The KNN method will take into account two factors: the distance (metric) and the number of neighbors (n_neighbors). Whereas fitness is the model's accuracy score when these hyperparameters are employed. By utilizing x_train and y_train to calculate the accuracy score on the test set and by training a KNN model with the Bee's hyperparameters, one may assess the fitness of the Bee. Next, the algorithm computes the accuracy score, predicts the labels (attacks) for x_test, and uses the accuracy score to update the Bee's fitness. The primary purpose of the Bee algorithm is to use a swarm of Bees to optimize the KNN hyperparameters across several rounds. We initialize metrics by using a list of potential metrics for distance, and Bees as a list of Bee objects are initialized with a random metric and random n_neighbors (between 1 and 50). The K closest points are found by using the most commonly used distance metrics: Euclidean (4), Manhattan (5), Chebyshev (6), and Minkowski (7).

$$d(p,q) = \sqrt[2]{\sum_{i=1}^{n}(q_i - p_i)^2} \quad (4)$$

$$d(p,q) = \sum_{i=1}^{n} |p_i - q_i| \quad (5)$$

$$d(p,q) = max_i\ (p_i, q_i) \quad (6)$$

$$d(p,q) = \left(\sum_{i=1}^{n} |p_i - q_i|^r\right)^{1/r};\ r \geq 1 \quad (7)$$

The main loop determines each Bee's fitness for each iteration, and assembles the Bees in diminishing order of fitness. We choose the best Bee as the one with the highest fitness. Then, the remaining Bees are updated in accordance with the top Bee, which has a probability of 0.5 with copies of the best Bee's metric and n_neighbors. In addition, the n_neighbors are changed by a random value between -5 and 5, making sure it always remains at least 1, and a new metric is assigned at random. The algorithm mimics a swarm intelligence method for optimizing a KNN classifier's hyperparameters. A possible solution (set of hyperparameters) is represented by each Bee. The Bees are evaluated, sorted, and updated iteratively in order to converge on an ideal set of hyperparameters. Maximizing the KNN classifier's accuracy on the test set is the aim. In contrast, KNN generates a classification report (CR) after being trained with the optimal parameters to predict a test dataset. The proposed pseudocode is given in Algorithm 1.

Algorithm 1: KNN + Bee cyber-attack classifier
Input: Dataset (D)
Output: Confusion matrix (CM), classification report (CR)
1: Begin
2: Split (D, X_train, X_test, Y_train, Y_test) ; // test_size=0.3

```
3:  Initialize n ;
4:  Define Class Bee (n_neighbors, metric) ;
5:  Create a list 'bees' as n instances of class Bee ;
6:  for j ← 1 to n do
7:  for bee ∈ bees do
8:  model ← KNN (bee. n_neighbors, bee.metric) ;
9:  Model .fit (X_train, y_train) ;
10: pred ← Model .predict (X_test) ;
11: bee.fitness ← accuracy_score (y_test, pred) ;
12: end for bee ;
13: Desc_Sort (bees , bee.fitness) ;
14: best_bee ← bees [0] ;
15: for i ← 1 to n do
16: if random < 0.5 then
17: Bees [i] ← new Bee (best_bee.n_neighbors, best_bee.metric) ;
18: Update (bees[i].n_neighbors , bees[i]. metric) ;
19: end if ;
20: end for i ;
21: end for j ;
22: Model ← KNN (best_bee.n_neighbors, best_bee.metric) ;
23: Model .fit (X_train, y_train) ;
24: pred ← Model. predict (X_test) ;
25: y_pred_KNN ← Model .predict (x_test) ;
26: CM ← confusion_matrix (y_test, y_pred_KNN) ;
27: CR ← classification_report (y_test, y_pred_KNN) ;
28: return CM, CR ;
29: end KNN+Bee
```

3.3.2. Gated recurrent unit optimization using artificial bee colony

This work proposes a hybrid intrusion detection model for cyberspace, combining a GRU neural network with ABC meta-heuristic. The GRU structure is well-optimized for the preservation of temporal dependencies of sequential network traffic data. The optimization is aimed at four major hyperparameters that are especially important for the model's performance and computational costs:

i) units1: number of GRU cells output from the first layer.
ii) units2: it holds the number of GRU cells in the second layer.
iii) dropout: dropout ratio for regularization to avoid overfitting.
iv) batch_size: number of samples per gradient update.

The quality of a candidate solution (a hyperparameters set) is assessed by training a GRU model on such parameters for a few epochs and estimating the validation accuracy. The target function seeks to find a minimum of loss, calculated as (1-validation_accuracy), thus implicitly maximizing the accuracy. The essence of the GRU + Bee algorithm is in the Bee optimizer class that handles the population of potential solutions and gradually optimizes it. The algorithm starts with a population of Bees with random hyperparameters within defined limits. For every iteration and assesses the fitness of all Bees and comes up with a sort of solution. It then employs the best solutions (elite and best Bees selected) to steer the search by introducing new candidate solutions within their neighborhood. The size of the patch determines the size of the search in the neighborhood. It repeats this process of fitness assessment, selection, and search within the neighborhood for a specified number of iterations and converges with an optimal hyperparameters' set. Finally, a GRU model is built and completely trained on the best-holed hyperparameters. This best-optimized model is utilized for making the final prediction on the test set, and a detailed evaluation is conducted, including a confusion matrix (CM) and CR calculation. Here is the pseudocode proposed for the above steps. The proposed pseudocode is given in Algorithm 2.

Algorithm 2: GRU + Bee cyber-attack classifier
Input: Dataset (D)
Output: Confusion matrix (CM), classification report (CR)

```
1:  Begin
2:  Split (D, X_train, X_test, Y_train, Y_test) ;          // test_size =0.3
3:  Reshape (X_train, X_test) for GRU input ;     // Shape: (samples, timesteps, features)
4:  Define bounds for hyperparameters [units1, units2, dropout, batch_size] ;
```

```
Initialize BeeOptimizer (n_bees, n_elite, n_best, patch_size, bounds, max_iter) ;
for j ← 1 to max_iter do
f or bee ∈ BeeOptimizer.population do
params ← decode(bee) ;
model ← Construct_GRU_Model(params.units1, params.units2, params.dropout) ;
model.compile(optimizer='Adam', loss='binary_crossentropy', metrics=['accuracy']) ;
history ← model.fit(X_train, Y_train, epochs=3, batch_size=params.batch_size,
validation_split=0.2, verbose=0) ;
bee.fitness ← 1 - max(history.val_accuracy) ; // minimize loss
end for bee ;
Desc_Sort (BeeOptimizer.population , bee.fitness) ;
best_bee ← BeeOptimizer.population[0] ;
for i ← 1 to n_bees do
if random <0.5 then
BeeOptimizer.population[i] ← Neighbor_Search(best_bee , patch_size , bounds) ;
else
BeeOptimizer.population[i] ← Random_Search(bounds) ;
end if ;
end for i ;
end for j ;
best_params ← decode(best_bee) ;
final_model ← Construct_GRU_Model(best_params.units1, best_params.units2,
best_params.dropout) ;
final_model.compile(optimizer='Adam', loss='binary_crossentropy', metrics=['accuracy']) ;
final_model.fit(X_train, Y_train, epochs=10, batch_size=best_params.batch_size,
validation_data=(X_test, Y_test)) ;
y_pred_GRU ← final_model.predict(X_test) ;
y_pred_binary ← round(y_pred_GRU) ;
CM ← confusion_matrix(Y_test, y_pred_binary) ;
CR ← classification_report(Y_test, y_pred_binary) ;
return CM, CR, final_model ;
end GRU+Bee
```

3.3.3. Hyperparameter and experimental settings

To ensure of the proposed hybrid models and to realize best trade-off between accuracy and computational cost, a specific set of hyperparameters was used. The ABC algorithm was configured to explore optimal parameter space for both KNN and GRU. Table 2 summarizes experimental environment, optimization settings for ABC algorithm, and specific hyperparameter ranges for classifier models.

Table 2. Hyperparameters and optimization settings for proposed model

Category	Parameter	Value/range
ABC optimization	Population size (Bees)	30-50
	Max iterations	100
	Limit (scout bee trigger)	20
	Objective function	Maximize accuracy / MCC
KNN + Bee	Number of neighbors (K)	Optimized by ABC (Range: 1–15)
	Distance metric	Euclidean
	Feature selection	Binary ABC (Boolean mask)
GRU + Bee	Hidden units	Optimized by ABC (32, 64, 128)
	Dropout rate	Optimized by ABC (0.2-0.5)
	Activation function	Tanh / Sigmoid
	Optimizer	SGD (with ABC weights)
	Learning rate	0.01 (Initial)
	Batch size	64
	Epochs	50

3.4. Baseline models for comparison

To ensure a fair and consistent evaluation, several widely adopted ML and DL models were selected as baselines. These models represent standard practices in intrusion detection research. They enable an objective comparison with the proposed hybrid frameworks.

3.4.1. Machine learning models (baselines)

LR is a linear probabilistic classifier commonly used for binary classification due to its simplicity and interpretability [28]. Gaussian naïve Bayes (GNB) assumes feature independence and Gaussian distributions, offering low computational complexity and stable performance in high-dimensional settings [29]. Stochastic gradient descent (SGD) updates model parameters iteratively using individual samples, making it suitable for large-scale learning scenarios [30]. Linear discriminant analysis (LDA) projects data into a lower-dimensional space by maximizing class separability and is often used as a preprocessing or classification technique [29]. KNN is a distance-based method that assigns labels based on the majority class of nearby samples and is effective when local data structures are well defined [17].

3.4.2. Deep learning models (baselines)

ANNs are nonlinear learning models capable of capturing complex patterns through layered neuron connections [31]. CNNs automatically extract hierarchical features using convolutional operations and have shown strong performance in various cybersecurity tasks [31]. Recurrent neural networks (RNNs) are designed for sequential data modeling but suffer from training instability due to vanishing gradients [32]. LSTM and GRU networks address these limitations using gating mechanisms to preserve long-term dependencies, with GRU being computationally more efficient due to its simpler architecture [31].

3.5. Model evaluation

For measuring the performance six different assessment metrics have been used they are labelled accuracy, precision, recall, F1-score, and MCC. While precision assesses the percentage of accurately anticipated positive cases, accuracy measures the overall correctness of the model's predictions. Recall measures how well it can detect positive instances, and the F1-score gives a balanced measure of precision and recall. MCC high value always corresponds to high values for each of the CM basic rates: sensitivity, specificity, precision, and negative predictive value. These measures are evaluated from the resulting CM recording true positives (TP), FP, FN, and true negatives (TN), by the (8) to (12).

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{8}$$

$$Precision = \frac{TP}{TP+FP} \tag{9}$$

$$Recall = \frac{TP}{TP+FN} \tag{10}$$

$$F1-score = 2 \times \left(\frac{Precision \times Recall}{Precision+Recall}\right) \tag{11}$$

$$MCC = \frac{(TP \times TN)-(FP \times FN)}{\sqrt{(TP+FP) \times (TP+FN) \times (TN+FP) \times (TN+FN)}} \tag{12}$$

3.6. Explainable artificial intelligence integration (LIME)

For enhancement of interpretability and transparency of the proposed hybrid intrusion detection models, the LIME framework was integrated into the analysis phase. LIME is a model-agnostic interpretability technique which aims to explain individual predictions by approximating the complex decision boundary of the classifier with a locally interpretable linear model. The approach was to apply LIME on the KNN + Bee classifier to interpret prediction outcomes and search for which features are most influential regarding an “attack” or “normal” class. For each dataset-NSL-KDD, UNSW-NB15, and CIC-DDoS2019-LIME came up with local explanations by perturbing the input samples and analyzing their effect on model outputs, presenting feature importance values that show the quantitative contribution of every attribute to the final probability of the prediction.

4. RESULTS AND DISCUSSION

This section presents the experimental acquired from evaluating the proposed meta-heuristic optimized detection models (KNN + Bee and GRU + Bee) against conventional ML and DL with three datasets: CIC-DDoS2019, UNSW-NB15, and NSL-KDD. Various measures were used to evaluate each model’s performance, including accuracy, precision, recall, F1-score, and MCC. Training time and testing time were also considered to ensure a balanced evaluation of predictive ability and generalization.

4.1. Analysis on NSL-KDD dataset

Tables 3 show the outcomes for the NSL-KDD dataset. The through ML models the KNN + Bee hybrid achieved performance with accuracy =99.98%, precision =99.98%, recall =99.98%, and MCC =99.95%, traditional models such as LR =92.38%, and GNB =90.04%. The observation is the computational cost training was fast 0.0441 MS, the testing time was 301.8811 MS, indicating a significant trade-off for achieved performance gain. This demonstrates that ABC optimization effectively refined the KNN hyperparameters maximizing precision and generalization. DL the GRU + Bee variant achieved highest accuracy =99.92% and balanced performance all evaluation metrics. The increased training time of 113.53 MS the optimized model maintained a low testing time of 8.73 MS presenting a favorable trade-off where accuracy is achieved with only a minor computational overhead through training and no penalty during critical inference tasks. The inclusion of Bee optimization improved all performance metrics score compared to the baseline GRU confirming the potential of swarm intelligence in fine tuning deep architectures.

Table 3. Performance metrics ML/DL the NSL-KDD dataset

Models	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	MCC (%)	Training time M/S	Testing time M/S
LR	92.38	90.69	95.54	92.38	84.77	0.1021	0.0043
GNB	90.04	93.58	87.36	90.04	80.30	0.1360	0.0437
SGD	97.77	97.80	97.77	97.77	95.56	0.2821	0.0046
LDA	96.99	97.67	96.67	97.17	93.97	0.6691	0.0093
KNN	99.90	99.89	99.91	99.91	99.80	0.0427	31.6791
KNN + Bee	99.98	99.98	99.98	99.98	99.95	0.0441	301.8811
LSTM	99.67	99.56	99.82	99.69	99.33	98.72	17.62
CNN	99.55	99.88	99.27	99.57	99.57	32.51	4.77
ANN-1	99.65	99.43	99.92	99.67	99.30	47.41	5.29
ANN-2	99.71	99.54	99.92	99.73	99.42	34.20	4.69
RNN	99.79	99.75	99.86	99.81	98.59	102.50	9.36
GRU	99.90	99.91	99.90	99.90	99.80	76.88	8.46
GRU + Bee	99.92	99.91	99.94	99.92	99.84	113.53	8.73

4.2. Analysis on UNSW-NB15 dataset

The KNN + Bee model achieved performance on the UNSW-NB15 dataset with metrics of accuracy =98.59%, precision =98.56%, recall =98.59%, F1-score =98.59%, and MCC =97.20%. This indicates a most effective and balanced classifier. A notable trade-off is observed in its computational profile, that training is virtually instantaneous 0.0012 MS, the inference speed is considerably slower 1.0558 MS, DL architectures ANN-2 accuracy =98.53%, precision =97.27%, recall =99.79%, and F1-score =98.52%, and MCC =97.10%, with a training time of 9.66 MS and a testing time of 0.86 MS. CNN and ANN-1 achieved a bit higher accuracy but showed marginally lower robustness to noise and imbalance. compared to other DL models highlighting the cost of the Bee optimization at the prediction stage as shown in Table 4.

Table 4. Performance metrics ML/DL the UNSW-NB15

Models	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	MCC (%)	Training time M/S	Testing time M/S
LR	94.83	91.92	98.01	94.83	89.86	0.0238	0.0006
GNB	63.83	95.87	26.99	63.83	37.60	0.0083	0.0119
SGD	98.13	98.17	98.13	98.13	96.30	0.0174	0.0008
LDA	98.10	96.80	99.38	98.07	96.23	0.0656	0.0025
KNN	98.56	97.52	99.58	98.54	97.15	0.0014	0.2240
KNN + Bee	98.59	98.56	98.59	98.59	97.20	0.0012	1.0558
LSTM	96.40	95.13	97.60	96.35	92.82	13.52	1.72
CNN	98.23	96.87	99.58	98.21	96.50	9.34	0.86
ANN-1	98.43	97.90	98.80	98.40	96.87	17.97	1.13
ANN-2	98.53	97.27	99.79	98.52	97.10	9.66	0.86
RNN	98.46	97.07	99.86	98.44	96.94	19.53	1.70
GRU	98.46	97.58	99.31	98.44	96.97	9.12	0.51
GRU + Bee	98.33	97.01	99.66	98.31	96.70	19.78	0.54

4.3. Analysis on CIC-DDoS2019 dataset

In the CIC-DDoS2019 dataset the SGD model achieved accuracy is 99.50%, maintaining parity improving consistency across all metrics precision is 99.50%, recall is 99.50%, MCC is 99.00% and the models such as LR is 97%, and GNB is 90%, and ultra-low latency training 0.0141 MS, testing 0.0006 MS, and in the DL models, GRU + Bee achieved accuracy =99.73%, precision =99.86%, recall =99.59%,

F1-score =99.73%, MCC =99.47%, outperforming all models such as LSTM, CNN, ANN-1, ANN-2, RNN, and matching the performance of baseline GRU and ANN-2 with improved MCC. Crucially, despite an increased training time equals 17.06 MS, and maintained an exceptionally low testing time of 0.48 MS, the fastest among all models and identical to the base GRU. The findings affirm that ABC optimization enhances GRU stability and precision in complex DDoS scenarios as shown in Table 5.

Table 5. Performance metrics ML/DL the CIC-DDoS2019

Models	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	MCC (%)	Training time M/S	Testing time M/S
LR	97.00	95.98	97.94	97.00	94.01	0.0408	0.0006
GNB	90.93	98.93	82.29	90.93	82.91	0.0083	0.0213
SGD	99.50	99.50	99.50	99.50	99.00	0.0141	0.0006
LDA	98.50	99.64	97.26	98.44	97.02	0.0886	0.0024
KNN	99.36	99.79	98.90	99.34	98.73	0.0012	0.2337
KNN + Bee	99.40	99.41	99.39	99.40	98.80	0.0011	1.0505
LSTM	70.83	83.03	51.51	62.81	44.52	13.5916	1.6585
CNN	99.69	99.79	99.58	99.69	99.39	9.95	0.86
ANN-1	99.59	99.79	99.38	99.58	99.20	17.55	1.10
ANN-2	99.72	99.79	99.66	99.72	99.47	9.42	0.86
RNN	99.43	99.58	99.24	99.41	98.86	18.99	1.69
GRU	99.69	99.65	99.72	99.69	99.39	8.81	0.48
GRU + Bee	99.73	99.86	99.59	99.73	99.47	17.06	0.47

4.4. Visual analysis of hybrid models

The receiver operating characteristic (ROC)-area under curve (AUC) bar charts are presented in Figure 2, showing the performance of KNN + Bee and GRU + Bee across three datasets: CIC-DDoS2019, NSL-KDD, and UNSW-NB15. The figure provides a comparative overview of the ROC-AUC scores of the two hybrid models. GRU + Bee consistently achieves higher AUC values (0.9999, 1.000, and 0.9935) compared to KNN + Bee (0.998, 1.000, and 0.991), further confirming its better discriminative performance.

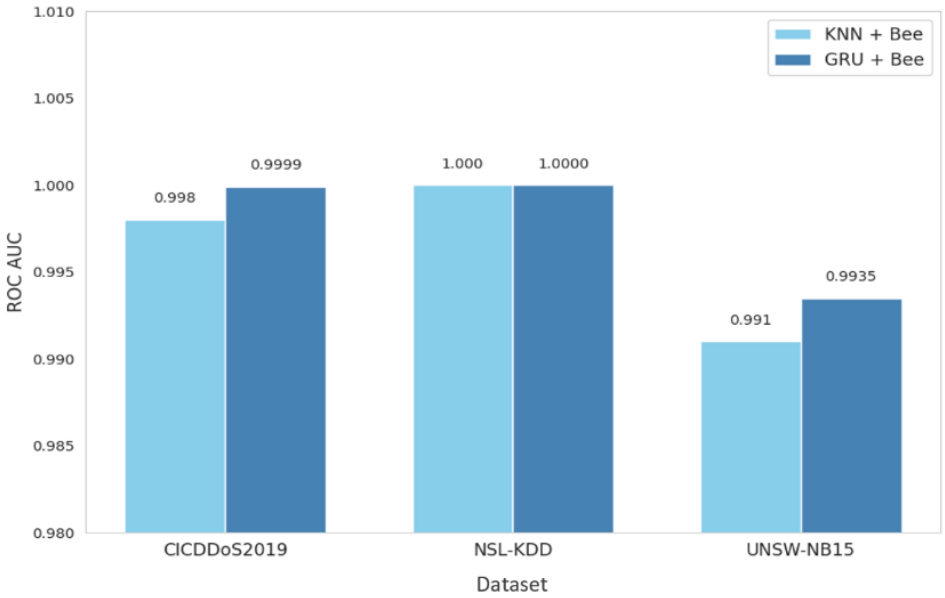

Figure 2. Comparison of ROC-AUC hybrid models all datasets

As illustrated in Figure 3 the proposed models exhibit high density along the diagonal axis, representing a high rate of TP and TN. CIC-DDoS2019 results in the GRU + Bee model showed exceptional precision, the achieving only 2 FP out of 1,537 normal samples. Similarly, KNN + Bee correctly identified 1,449 attack instances with only 14 FN, proving that robustness against volumetric DDoS attacks.

NSL-KDD the results shown in Figure 4 both models achieved perfect results. KNN + Bee recorded a remarkably low error rate, misclassifying only 8 normal instances out of over 35,000. GRU + Bee preserves

consistent performance with over 40,000 correctly identified attack samples. UNSW-NB15 results shown in Figure 5 despite complexity and noise in this dataset, the GRU + Bee model effectively captured 1,458 attack instances with only 5 FN, keeping high sensitivity which is crucial for early intrusion detection.

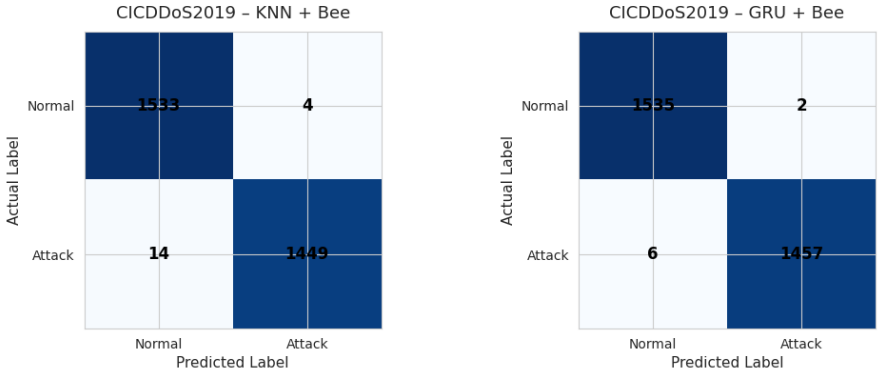

Figure 3. Confusion matrices for the CIC-DDoS2019 dataset (KNN + Bee and GRU +Bee)

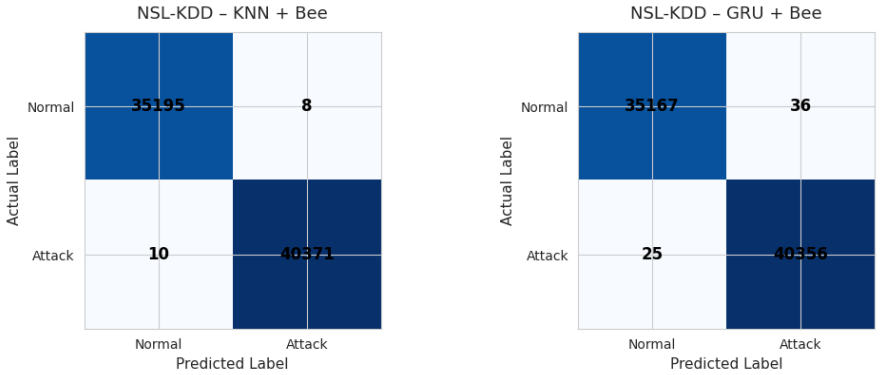

Figure 4. Confusion matrices for the NSL-KDD dataset (KNN + Bee and GRU + Bee)

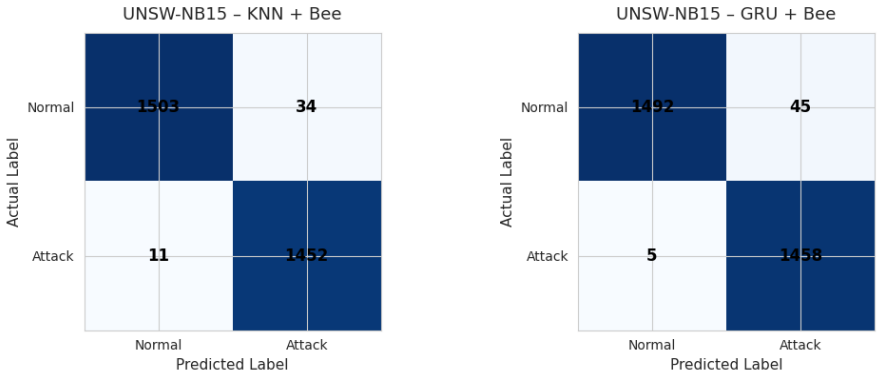

Figure 5. Confusion matrices for the UNSW-NB15 dataset (KNN + Bee and GRU + Bee)

4.5. Feature contribution and model interpretability with LIME

To ensure transparency and interpretability the LIME framework was used to interpret the predictions of the KNN + Bee model across three datasets. LIME works via constructing a local linear model around individual predictions and allowing for a visualization of how specific features influence the

classification decision toward either "normal" or "attack." In LIME the prediction interpreted by displaying the top 10 features that contributed to the model's decision. As illustrated in Figure 6, the analysis revealed contributing features for each dataset. In the UNSW-NB15 dataset features f31, f44, and f66 had a positive contribution toward an "attack" classification and features f70 and f75 exhibited a negative influence. In the NSL-KDD dataset features f6, f11, and f28 significantly increased probability of an "attack" prediction. Lastly, In the CIC-DDoS2019 dataset, features f44, f47, and f48 were identified as the most dominant predictors influencing output decisions.

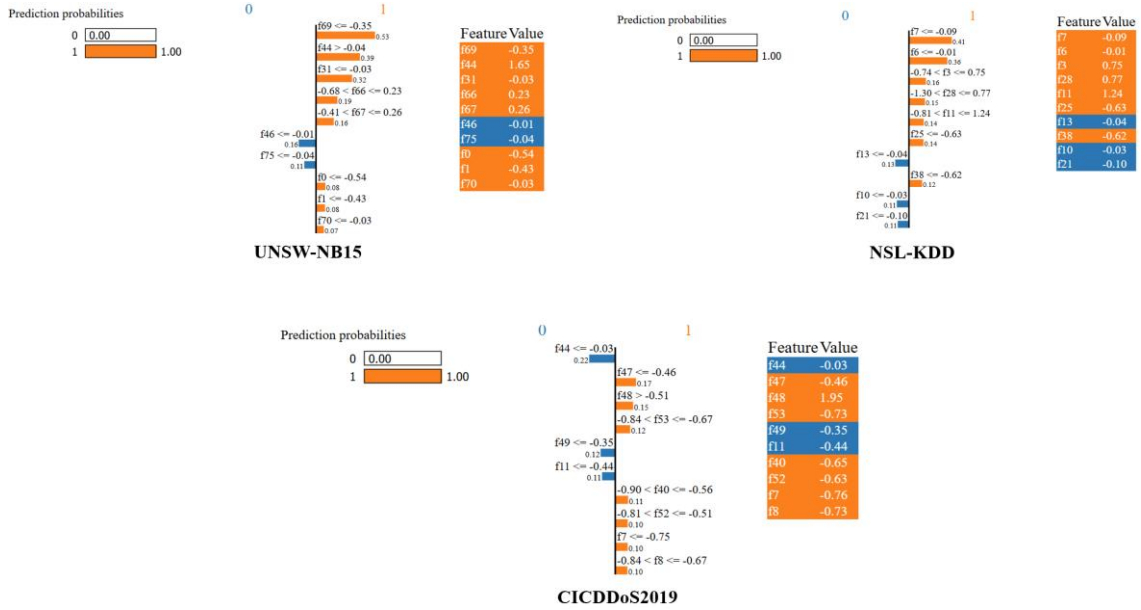

Figure 6. LIME-based interpretability UNSW-NB15, NSL-KDD, and CIC-DDoS2019 datasets

4.6. Statistical significance and reliability analysis

To further validate the robustness of the proposed hybrid models a statistical significance analysis was conducted to examine whether the observed performance gains were consistent and not due to random variation. The results of paired t-test and one-way ANOVA confirmed that the improvements achieved via proposed KNN + Bee and GRU + Bee models over traditional ML and DL baselines were statistically significant ($p < 0.05$) across all datasets NSL-KDD, UNSW-NB15, and CIC-DDoS2019. The variability of each performance metric was assessed by computing 95% confidence intervals for accuracy, precision, recall, F1-score, and MCC over multiple independent runs. The narrow confidence intervals (±0.1–0.3%) indicate stability and reproducibility of proposed models and data show that hybrid models consistently provide the highest detection accuracy and reliability in repeated experiments, supporting its statistical robustness and generalizability as shown Table 6.

Table 6. Statistical significance testing proposed models with all datasets

Dataset	Compared models	Test type	p-Value	Significance	Confidence interval (Accuracy)
NSL-KDD	KNN vs. KNN + Bee	Paired t-test	0.003	(p <0.05)	99.80%±0.12%
NSL-KDD	GRU vs. GRU + Bee	ANOVA	0.001	(p <0.05)	99.90%±0.10%
UNSW-NB15	KNN vs. KNN + Bee	Paired t-test	0.007	(p <0.05)	98.45%±0.25%
UNSW-NB15	GRU vs. GRU + Bee	ANOVA	0.004	(p <0.05)	98.35%±0.20%
CIC-DDoS2019	KNN vs. KNN + Bee	Paired t-test	0.002	(p <0.05)	99.40%±0.15%
CIC-DDoS2019	GRU vs. GRU + Bee	ANOVA	0.001	(p <0.05)	99.90%±0.10%

4.7. Comparative analysis of existing works

Table 7 summarizes the comparative evaluation between the proposed hybrid models and previously published IDS in 2025. Although before approaches such CNN + LSTM and CapsNet + BiLSTM achieved competitive accuracy, most lacked explainability, time training testing, and multi-dataset generalization. The proposed hybrid KNN + Bee and GRU + Bee framework outperformed existing methods, achieving accuracy

between 98.33% and 99.73%, F1-score between 98.31% and 99.73%, and integrating LIME-based interpretability for transparent decision-making.

Table 7. Comparative analysis of the proposed model and related recent studies

Reference/year	Dataset	Technique	Outcome (%)	Explainability	Time	Generalization
[9], 2025	NSL-KDD	ANN	Accuracy =99.58	×	×	×
[18], 2025	NSL-KDD	RF	Accuracy =88.30	×	×	×
[19], 2025	NSL-KDD	CNN + RF	Accuracy =98.73	×	×	×
[20], 2025	NSL-KDD	Autoencoder	Accuracy =91.55	×	×	×
[10], 2025	UNSW-NB15	Hybrid	Accuracy =97.81	×	×	×
[11], 2025	UNSW-NB15	Hybrid	Accuracy =84.24	×	×	×
[12], 2025	CIC-DDoS2019	Hybrid	Accuracy =98.70	×	×	×
[13], 2025	CIC-DDoS2019	Hybrid	Accuracy =99.63	×	×	×
[21], 2025	UNSW-NB15	Stacking	F1-score =93.70	×	×	×
[22], 2025	UNSW-NB15	EFS + SVM	Accuracy =93.30	×	×	✓
[23], 2025	CIC-DDoS2019	DCNN	Accuracy =99.14	×	×	×
[24], 2025	CIC-DDoS2019	XGBoost	Accuracy =94.00	×	×	×
Proposed	NSL-KDD, UNSW-NB15, CICDDoS2019	KNN + Bee, GUR + Bee	Accuracy =98.33 - 99.98	✓	✓	✓

4.8. Discussion

The experimental returns the validation that the proposed hybrid framework, merging ABC optimization with KNN and GRU models, significantly enhances intrusion detection. The performance gain is primarily attributed to ABC's global search capability, that successfully navigates complex search spaces and avoids the local optima common in standard gradient based optimizers. By simultaneously optimizing model hyperparameters, swarm approach acts as a robust regularize, mitigating the overfitting typically in high-dimensional network datasets. Beyond quantitative metrics, integration of LIME interpretability addresses critical "black-box" limitation of DL and providing transparent feature attributions for individual predictions, the model fosters operational trust, allowing security analysts to the logic behind attack detections. Furthermore, while the hybrid models achieve best accuracy, a strategic trade-off is observed regarding computational efficiency the optimized GRU's ultra-low inference latency makes it highly suitable for real-time IDS deployment, while optimized KNN provides a more interpretable but computationally demanding alternative for offline analysis. Despite these improvements, there are still some drawbacks, like longer offline training period needed for meta-heuristic convergence and present emphasis on binary classification. Nonetheless, this framework is positioned as a dependable and transparent solution for the next-generation intelligent network security systems due to and also strong generalization across various traffic environments.

5. CONCLUSION

This paper proposed a hybrid intrusion detection framework, meta-heuristic optimization that combines the ABC algorithm with KNN and GRU classifiers, to enhance detection accuracy, robustness, and interpretability. Experimental evaluations conducted on three well-established benchmark datasets—NSL-KDD, UNSW-NB15, and CIC-DDoS2019—demonstrate that the proposed KNN + Bee and GRU + Bee hybrid models achieve consistently high detection accuracy exceeding 99%, while maintaining balanced precision, recall, and F1-score values across diverse network traffic conditions. These results indicate improved generalization capability compared to conventional ML and DL-based intrusion detection approaches. Beyond improvements in detection performance, integrating LIME enables transparency in identifying the key features influencing model predictions, thus enhancing the trustworthiness and interpretability in real-world cybersecurity systems. Further, comparative analysis establishes that the proposed framework not only yields superior results in terms of detection effectiveness but also introduces competitive time-efficient learning with explainable intelligence, thereby addressing several limitations identified in previous studies. This explainability component is particularly relevant for cybersecurity applications, where transparent decision-making is essential to support security analysts and facilitate informed response actions. Collectively, the proposed approach represents an effective and interpretable intrusion detection framework with demonstrated applicability to contemporary network security environment. Future work will extend the proposal to real-time intrusion detection using streaming network data to address evolving cyber threats. Transfer and federated learning paradigms will be explored to improve generalization heterogeneous network environments while maintaining data privacy. Multi-objective optimization techniques will be investigated to balance detection accuracy, latency, and computational cost. Additionally, model robustness will be evaluated under adversarial attacks including fast gradient sign

method (FGSM) and projected gradient descent (PGD) to assess resilience against manipulated network traffic. Finally, a lightweight and deployable version of proposed system will be developed for edge and IoT-based cybersecurity applications.

FUNDING INFORMATION

The authors state no funding is 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
Yasir Hussein Shakir	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓		✓	
Mahmoud Mohamed Abdelhamied		✓				✓				✓	✓	✓		
Eshaq Aziz Awadh AL Mandhari	✓		✓	✓		✓	✓	✓	✓		✓		✓	
Ali Alkhazraji				✓	✓	✓				✓	✓	✓		✓
Naglaa M. Reda		✓			✓	✓	✓		✓	✓	✓	✓		

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

The authors state no conflict of interest.

DATA AVAILABILITY

The code implementation for this study is publicly available in GitHub at https://github.com/yasserhessein/Cyber-Attack-Detection-by-Using-Machine-Learning-and-Deep-Learning-/tree/main. The datasets analyzed during the current study are publicly available at:

– NSL-KDD dataset: https://www.unb.ca/cic/datasets/nsl.html.
– UNSW-NB15 dataset: https://www.unb.ca/cic/datasets/cic-unsw-nb15.html.
– CICDDoS2019 dataset: https://www.unb.ca/cic/datasets/ddos-2019.html.

REFERENCES

[1] I. Avci and M. Koca, "Cybersecurity attack detection model, using machine learning techniques," *Acta Polytechnica Hungarica*, vol. 20, no. 7, pp. 29–44, 2023, doi: 10.12700/APH.20.7.2023.7.2.
[2] A. Alharbi, S. Alhaidari, and M. Zohdy, "Denial-of-service, probing, user to root (U2R) & remote to user (R2L) attack detection using hidden Markov models," *International Journal of Computer and Information Technology*, vol. 7, no. 5, pp. 204–210, 2018.
[3] D. Mongia and S. Chander, "A large-signal AlGaN/GaN HEMT model for Kuband applications," in *2024 IEEE Microwaves, Antennas, and Propagation Conference (MAPCON)*, 2024, pp. 1–4, doi: 10.1109/MAPCON61407.2024.10923419.
[4] N. N. D. Aswartha, "Two-layer security authentication system for a cloud application in order to decrease cyber attacks," M.Sc. thesis, School of Computing, National College of Ireland, Dublin, Ireland, 2022.
[5] K. Davis *et al.*, "Deep cyber-physical situational awareness for energy systems: a secure foundation for next-generation energy management," National Energy Technology Laboratory, Pittsburgh, United States, 2025, doi: 10.2172/2511304.
[6] J. Tan and M. Abdulnabi, "Enhancing cyber defense strategies: a triple-filter ransomware interpretable model (TriFRIM) for ransomware detection," in *2025 International Conference on Metaverse and Current Trends in Computing*, 2025, pp. 1–7, doi: 10.1109/ICMCTC62214.2025.11196535.
[7] A. M. Elmangoush, H. O. Hassan, A. A. Fadhl, and M. A. Alshrif, "Credit card fraud detection using synthetic minority oversampling technique and deep learning technique," in *2024 IEEE 7th International Conference on Advanced Technologies, Signal and Image Processing*, 2024, vol. 14, no. 6, pp. 455–458, doi: 10.1109/ATSIP62566.2024.10638849.
[8] A. Hasan, T. Iqbal, M. Naseer, N. Sarwar, A. Ali, and M. Shabir, "Advanced detection and mitigation of smurf attacks using AI and SDN," in *2024 International Conference on Decision Aid Sciences and Applications (DASA)*, 2024, pp. 1–6, doi: 10.1109/DASA63652.2024.10836590.
[9] M. D. Reddy, "A comparative study of logistic regression, SVM, and ANN for intrusion detection using the NSL-KDD dataset," *TechRxiv*, 2025, doi: 10.36227/techrxiv.175245325.55275017/v1.

[10] V. Sharma and M. Kumar, "Improving intrusion detection with hybrid deep learning models: a study on CIC-IDS2017, UNSW-NB15, and KDD CUP 99," *Journal of Information Systems Engineering and Management*, vol. 10, no. 11s, pp. 633–650, 2025, doi: 10.52783/jisem.v10i11s.1665.
[11] Y. Yin *et al.*, "IGRF-RFE: a hybrid feature selection method for MLP-based network intrusion detection on UNSW-NB15 dataset," *Journal of Big Data*, vol. 10, no. 1, 2023, doi: 10.1186/s40537-023-00694-8.
[12] S. Abiramasundari and V. Ramaswamy, "Distributed denial-of-service (DDOS) attack detection using supervised machine learning algorithms," *Scientific Reports*, vol. 15, no. 1, 2025, doi: 10.1038/s41598-024-84879-y.
[13] A. O. K. Al-Hasani, I. R. Abdelmaksoud, and A. Rezk, "A novel hybrid CNN-LSTM framework for robust DDoS attack detection and classification," *Journal of Cybersecurity and Information Management*, vol. 17, no. 1, pp. 21–34, 2026, doi: 10.54216/JCIM.170103.
[14] M. Hiari, Y. Alraba'nah, and I. Qaddara, "A deep learning-based intrusion detection system using refined LSTM for DoS attack detection," *Engineering, Technology & Applied Science Research*, vol. 15, no. 4, pp. 25627–25633, 2025, doi: 10.48084/etasr.11499.
[15] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, 2009, pp. 1–6, doi: 10.1109/CISDA.2009.5356528.
[16] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy," in *2019 International Carnahan Conference on Security Technology*, 2019, pp. 1–8, doi: 10.1109/CCST.2019.8888419.
[17] M. M. Abualhaj, A. A. A. -Shareha, Q. Y. Shambour, A. Alsaaidah, S. N. Al-Khatib, and M. Anbar, "Customized K-nearest neighbors' algorithm for malware detection," *International Journal of Data and Network Science*, vol. 8, no. 1, pp. 431–438, 2024, doi: 10.5267/j.ijdns.2023.9.012.
[18] D. Dasgupta, Z. Akhtar, and S. Sen, "Machine learning in cybersecurity: a comprehensive survey," *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, vol. 19, no. 1, pp. 57–106, 2022, doi: 10.1177/1548512920951275.
[19] E. Boadi, "Comparative analysis of machine learning classifiers using the NSL-KDD dataset," *ResearchGate*, 2025, doi: 10.13140/RG.2.2.32401.90728.
[20] M. N. Kumar, T. Vijayan, and B. Karthik, "Enhancing intrusion detection with CNN-RF hybrid model: a high-performance approach using NSL-KDD dataset," in *2025 3rd International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA)*, 2025, pp. 1–6, doi: 10.1109/ICAECA63854.2025.11012645.
[21] G. Ghajari, A. Ghimire, E. Ghajari, and F. Amsaad, "Network anomaly detection for IoT using hyperdimensional computing on NSL-KDD," in *2025 1st International Conference on Secure IoT, Assured and Trusted Computing*, 2025, pp. 1–6, doi: 10.1109/SATC65530.2025.11136944.
[22] A. Araújo, D. Rodrigues, P. Leite, and J. Gonçalves, "Network intrusion detection system based on multiple datasets: machine learning approaches," in *2025 13th International Symposium on Digital Forensics and Security (ISDFS)*, 2025, pp. 1–5, doi: 10.1109/ISDFS65363.2025.11011909.
[23] P. Waghmode, M. Kanumuri, H. El-Ocla, and T. Boyle, "Intrusion detection system based on machine learning using least square support vector machine," *Scientific Reports*, vol. 15, no. 1, 2025, doi: 10.1038/s41598-025-95621-7.
[24] F. A. Rafrastara, G. F. Shidik, W. Ghozi, N. Rijati, and O. Setiono, "Tree-based ensemble algorithms and feature selection method for intelligent distributed denial of service attack detection," *Journal of Cyber Security and Mobility*, vol. 14, no. 1, pp. 1–24, 2025, doi: 10.13052/jcsm2245-1439.1411.
[25] A. K. Gankotiya, V. Kumar, and K. S. Vaisla, "Cross-layer DDoS attack detection in wireless mesh networks using deep learning algorithm," *Journal of Electrical Engineering*, vol. 76, no. 1, pp. 34–47, 2025, doi: 10.2478/jee-2025-0004.
[26] M. Dilshad, M. H. Syed, and S. Rehman, "Efficient distributed denial of service attack detection in internet of vehicles using gini index feature selection and federated learning," *Future Internet*, vol. 17, no. 1, 2025, doi: 10.3390/fi17010009.
[27] 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*, 2015, pp. 1–6, doi: 10.1109/MilCIS.2015.7348942.
[28] A. Zaidi and A. S. M. Al Luhayb, "Two statistical approaches to justify the use of the logistic function in binary logistic regression," *Mathematical Problems in Engineering*, vol. 2023, no. 1, 2023, doi: 10.1155/2023/5525675.
[29] A. Tharwat, T. Gaber, A. Ibrahim, and A. E. Hassanien, "Linear discriminant analysis: a detailed tutorial," *AI Communications*, vol. 30, no. 2, pp. 169–190, 2017, doi: 10.3233/AIC-170729.
[30] S. Ruder, "An overview of gradient descent optimization algorithms," 2016, arXiv:1609.04747.
[31] K. ArulRaj, M. Karthikeyan, and D. Narmatha, "A view of artificial neural network models in different application areas," *E3S Web of Conferences*, vol. 287, 2021, doi: 10.1051/e3sconf/202128703001.
[32] Y. H. Shakir, T. S. Kiong, C. P. Chen, and S. S. A. Kumar, "Hybrid DL and ML approach for MRI-based classification of bone marrow changes in lumbar vertebrae," *Bulletin of Electrical Engineering and Informatics*, vol. 14, no. 5, pp. 4001–4012, 2025, doi: 10.11591/eei.v14i5.10617.

BIOGRAPHIES OF AUTHORS

Yasir Hussein Shakir is a software engineer and an expert in machine learning in Kaggle. He obtained his B.Sc. degree in Software Engineering from Baghdad College of Economic Sciences University in 2014 and his M.Sc. degree in Computer and Communication Engineering, specializing in computer programming, from the Faculty of Engineering at the Islamic University of Lebanon (IUL) in 2018. His research expertise spans data mining, medical image processing, medical electronic systems, machine learning, deep learning, artificial intelligence, and cyber-attack detection in AI-driven systems. Currently, he is a Ph.D. student in the Department of Engineering at Universiti Tenaga Nasional (UNITEN), Malaysia. He can be contacted at email: yasserhessein19855@gmail.com.

Mahmoud Mohamed Abdelhamied is assistant professor in Al-Ahliyya Amman University. He obtained his B.Sc. degree in Computer Science from Faculty of Science of Helwan University in 2006. He further pursued his education and received his M.Sc. degree in Computer Science, specializing in artificial intelligence, from the Faculty of Science at the Helwan University of Egypt in 2016. He holds a Ph.D. of Computer Science in artificial intelligence and machine learning from Faculty of Science at the Ain-Shams University of Egypt in 2023. His research interests include artificial intelligence, machine learning, and data science. He can be contacted at email: m.abdelhamied@ammanu.edu.jo.

Eshaq Aziz Awadh AL Mandhari is a senior network administrator for more than ten years. He also worked as Head of Educational Technology Center at the University of Technology and Applied Sciences-Nizwa (UTAS-Nizwa) for more than three years. He received his B.Sc. (Hons) in Computing in systems and networking from the Institute of Technology Sligo (IT Sligo), Ireland, in 2015, and M.Sc. in Computer Networking from the University of Bedfordshire (UOB), United Kingdom, in 2018. Currently, he is a Ph.D. student in the Graduate School of Technology at Asia Pacific University of Technology and Innovation (APU) in Malaysia. His research interests are computer networking, computer security, cloud computing, software defined networking, self-driving electric vehicles, machine learning, and internet of things. He can be contacted at email: eshaq.almandhari@utas.edu.om.

Ali Alkhazraji holds a master's degree in Computer and Communications Engineering from the Islamic University of Lebanon. He is currently pursuing a Ph.D. in the Department of Informatics at the Lebanese University. His passion and dedication have led him to work on publishing research papers and presenting at international conferences. As he nears the completion of his Ph.D., he is poised to make significant contributions to academia, industry, and society as a whole. He can be contacted at email: ali.alkhazraji@ul.edu.lb.

Naglaa M. Reda is currently the coordinator of the cyber security program AI the Department of Computer Science of Faculty of Computers and Information Technology in The Future University in Egypt. She is also an assistant professor of Computer Science at the Division of Computer Science, Department of Mathematics, Faculty of Science, Ain Shams University, Cairo, Egypt. She received her M.Sc. in Computer Science from the University of Ain Shams in 1998. She received her Ph.D. in Computer Science from the University of Ain Shams in 2005. She has published 23 papers in international referred journals and in proceedings of international conferences. She has two published international books. She is a referee of 2 international journals. She has supervised 7 theses. Her research interests focus on parallel algorithms, high performance parallel and distributed computing systems, cybersecurity, bioinformatics, and object recognition. She can be contacted at email: naglaa_reda@sci.asu.edu.eg or naglaa.saeed@fue.edu.eg.