Expert Systems With Applications 326 (2026) 132661

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Hybrid ABC–HBA feature optimization with self-training using simulated unlabelled data for robust intrusion detection

Sasangi Harischandra [a], U.U. Samantha Rajapaksha [a], Bhagya Nathali Silva [a], Chandimal Jayawardena [b,*]

[a] Department of Information Technology, Faculty of Computing, Sri Lanka Institute of Information Technology, Malabe, 10115, Sri Lanka
[b] School of Computer Science and Information Technology, College of Engineering and Information Technology, Adelaide University, Australia

ARTICLE INFO

Keywords:
Intrusion detection systems
Feature optimisation
Machine learning
Pseudo-labelling
Swarm intelligence algorithms
Anomaly detection

ABSTRACT

The increasing scale and heterogeneity of network traffic pose significant challenges for intrusion detection systems (IDS), particularly in detecting extremely rare attack classes and generalising to previously unseen threats under severe class imbalance. This study proposes a hybrid intrusion detection framework that integrates swarm intelligence–based feature optimisation with self-training using unlabelled data simulation to address these limitations. A novel ABC–HBA feature selection strategy is introduced, combining the efficient exploration capability of the Artificial Bee Colony (ABC) algorithm with the strong global exploitation and fast convergence of the Honey Badger Algorithm (HBA), resulting in a highly discriminative and compact feature subset. A Random Forest(RF) classifier augmented with a pseudo-labelling mechanism is then employed to enhance learning from unlabelled and unseen attack samples, enabling effective detection of novel attack patterns absent from the training set. To further mitigate extreme class imbalance, a hybrid resampling strategy is applied. Experimental evaluation on the KDD Cup 1999 dataset demonstrates that the proposed framework achieves an overall accuracy of 99.95% and a detection rate of 98.16%, while significantly improving the recognition of extremely rare attack classes, including a 92.86% detection rate for U2R attacks, which constitute less than 0.01% of the dataset. The proposed method consistently outperforms baseline RF, ABC-based, and several other state-of-the-art meta-heuristic and deep learning approaches, confirming its effectiveness in enhancing rare attack detection and generalisation to unseen threats in realistic intrusion detection scenarios.

1. Introduction

With the rapid advancement of cyberspace, innovative networking and computing technologies have continually evolved, significantly enhancing global connectivity, operational efficiency, and the digital transformation of various sectors (Li, 2018). These developments have revolutionised how data are communicated, stored, and processed across critical infrastructures such as financial institutions, energy grids, transportation systems, and healthcare services. However, this unprecedented expansion of cyberspace has simultaneously intensified cybersecurity challenges, as cyber threats continue to grow in complexity, frequency, and scale, posing severe risks to the reliability and security of essential digital systems (Guan et al., 2017).

Traditionally, network security has relied on static defence mechanisms, including specialised hardware and software tools such as firewalls, intrusion detection systems (IDSs), and intrusion prevention systems (IPSs), deployed at fixed points within the network such as nodes and gateways (Abraham & Bindu, 2021). These systems operate using predefined rule sets to monitor traffic, detect anomalous behaviour, and block potential intrusions. While effective against known threats, such rule-based mechanisms are inherently reactive and lack the adaptability required to counter rapidly evolving attack strategies, rendering them increasingly inadequate against advanced persistent threats (APTs) and zero-day vulnerabilities (Li, 2018). Furthermore, the availability of automated and low-cost attack tools has enabled adversaries to launch large-scale attacks that can overwhelm static defence infrastructures.

Within this evolving threat landscape, IDSs remain a fundamental component of defence-in-depth architectures; however, their effectiveness is often limited in dynamic and high-volume network environments. As illustrated in Fig. 1, existing IDS models–whether rule-based or based on traditional machine learning (ML), frequently suffer from low detection accuracy, high false positive rates, and limited capability

* Corresponding author.
E-mail address: chandimal.jayawardena@adelaide.edu.au (C. Jayawardena).

https://doi.org/10.1016/j.eswa.2026.132661
Received 14 December 2025; Received in revised form 18 April 2026; Accepted 26 April 2026
Available online 28 April 2026

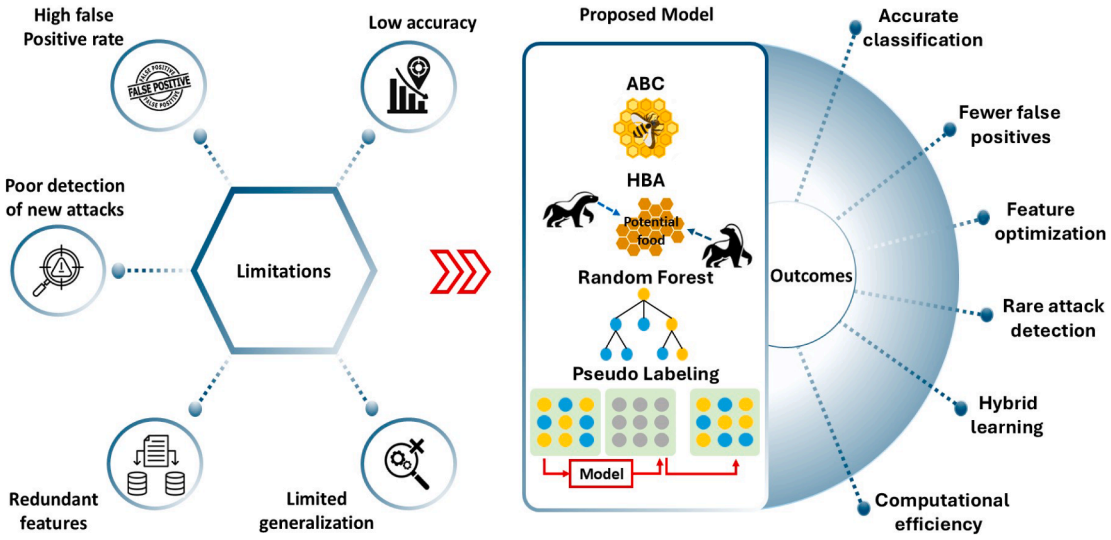

Fig. 1. Overview of the proposed hybrid ABC–HBA model and its outcomes.

to recognise unseen or rare attack patterns. These shortcomings are primarily attributed to redundant or suboptimal feature representations, insufficient feature optimisation, and poor generalisation across diverse attack scenarios, particularly under highly imbalanced network traffic conditions.

To mitigate these limitations, numerous studies have explored ML-based IDS approaches that leverage intelligent traffic analysis and adaptive classification techniques. Classical algorithms such as k-nearest neighbours (KNN) (Liao & Vemuri, 2002; Pathak & Pathak, 2020), Naïve Bayes (NB) (Mukherjee & Sharma, 2012), Decision Trees (DT) (Resende & Drummond, 2018), Support Vector Machines (SVM) (Gao et al., 2009; Vapnik, 2000; Yang & Wang, 2008), and Random Forest (RF) (Resende & Drummond, 2018) have been widely adopted due to their simplicity and computational efficiency. However, despite achieving high overall accuracy in some cases, these models often struggle to maintain consistent performance when faced with complex, evolving, or highly imbalanced traffic. In particular, detection rates for minority attack categories such as User-to-Root (U2R), Remote-to-Local (R2L), and probing attacks remain significantly low (Iftikhar et al., 2025), highlighting their limited robustness and adaptability to unseen intrusion patterns.

Recent research has therefore shifted toward deep learning(DL)-based IDS architectures and meta-heuristic optimisation techniques to enhance detection robustness and generalisation. DL models, including Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM) networks, have demonstrated improved feature representation and classification performance in complex network environments (Ullah et al., 2024). In parallel, meta-heuristic algorithms such as Genetic Algorithms (GA), Particle Swarm Optimisation (PSO), Ant Colony Optimisation (ACO), and Artificial Bee Colony (ABC) have been applied to feature selection and dimensionality reduction to reduce redundancy and improve classifier efficiency. Despite these advances, many existing frameworks still suffer from high computational complexity, overfitting, and reduced sensitivity to rare or emerging attack types, underscoring the need for an adaptive and data-efficient hybrid detection approach.

The ABC algorithm is widely recognized for its conceptual simplicity, ease of implementation, and limited number of control parameters compared to other meta-heuristic algorithms such as ACO and GA (Karaboga & Basturk, 2008). Prior studies have demonstrated that ABC achieves an effective balance between exploration and exploitation through the coordinated behaviours of employed, onlooker, and scout bees, leading to competitive performance on benchmark optimization and feature selection problems (Alatas, 2010; Karaboga, 2005; Karaboga & Akay, 2011). Nevertheless, ABC has been reported to suffer from relatively slow convergence and insufficient local exploitation when applied to complex or high-dimensional search spaces, particularly in scenarios requiring rapid escape from local optima. In contrast, the Honey Badger Algorithm (HBA) is a population-based meta-heuristic inspired by the dynamic foraging and hunting strategies of honey badgers, incorporating digging and honey-seeking mechanisms to enhance exploitation and accelerate convergence (Hashim et al., 2022). Since its introduction in 2020, HBA has attracted significant research interest due to its straightforward structure, ease of use, efficient computational time, and accelerated convergence speed. Comprehensive survey studies report that HBA and its variants have been successfully applied to a wide range of optimization problems, with numerous improvements and hybridisations proposed to further enhance its exploitation efficiency and convergence behaviour. These studies highlight HBA's strong ability to refine solutions in complex and high-dimensional search spaces while reducing the risk of premature convergence through adaptive search behaviours and diversified population movement mechanisms (Hassan et al., 2024). An RF classifier is then employed for reliable intrusion detection with enhanced accuracy and reduced false positive rates. To identify a suitable classification model, multiple ML algorithms, including RF, LR, DT, SVM, Gradient Boosting, and XGBoost, were evaluated. While some boosting-based models achieved competitive performance for rare attack categories, RF demonstrated the most consistent overall performance, stability, and generalisation across all classes; therefore, it was selected as the base classifier for the proposed framework. Detailed comparative results are provided in Supplementary Table 1.

Despite advances in ML and DL-based IDS, the detection of extremely rare attack classes (approximately 0.01%) remains a critical unresolved challenge. Existing feature selection and optimization frameworks primarily optimise global accuracy, often eliminating discriminative features associated with minority attack patterns. This leads to extremely low recall for rare attacks such as U2R and R2L. Therefore, a feature selection mechanism specifically designed to preserve minority-discriminative features while maintaining global search efficiency is required. In this context, the key contribution of this study is the development of a feature-selection–driven framework specifically designed to enable the detection of extremely rare attack types under severe class imbalance. The proposed approach integrates the ABC algorithm with the HBA to achieve a balanced optimisation process that combines effective global exploration with adaptive local exploitation. This hybrid optimisation strategy facilitates the preservation of minority-discriminative features while maintaining efficient

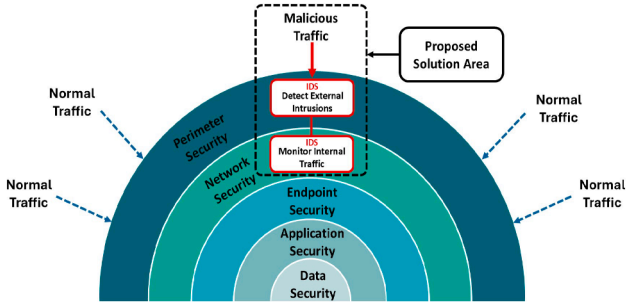

Fig. 2. Proposed solution area within the Defence-in-Depth architecture for detecting and monitoring malicious traffic.

convergence behaviour. Consequently, the framework demonstrates enhanced capability in detecting extremely rare attack classes while preserving overall classification stability, providing a robust and practically applicable solution for highly imbalanced intrusion detection scenarios.

The proposed framework is designed to detect both external and internal intrusions across the Perimeter and Network Security layers of a defence-in-depth architecture, as illustrated in Fig. 2. By operating across both layers, the framework strengthens perimeter defences against external threats such as Denial-of-Service (DoS) and probing attacks while simultaneously monitoring internal traffic to detect lateral movements, insider threats, and complex intrusion types including U2R and R2L. Through advanced feature selection and hybrid optimisation, the framework minimises redundancy, prioritises discriminative attributes, and achieves improved detection accuracy with reduced false positive rates, while maintaining computational efficiency and scalability for large-scale, high-volume network environments.

The organisation of this paper is as follows: Section 2 presents a comprehensive review of intrusion detection methods based on metaheuristic and ML algorithms. Section 3 describes the proposed approach. Section 4 presents the experimental results and analysis, and Section 5 outlines future research directions and concludes the paper.

2. Background and related work

IDSs are designed to enhance network security by identifying and responding to malicious activities that may bypass traditional firewall defences. Despite extensive research and the development of numerous IDS models, a major challenge is designing systems that can effectively generalise across multiple attack categories, adapt to continuously evolving threats, and maintain low false-positive rates, particularly in multi-class and imbalanced data environments (Ahmad et al., 2018). As attack strategies continue to evolve and network infrastructures grow in complexity, the limitations of conventional intrusion detection approaches have become increasingly evident, underscoring the necessity for more adaptive and intelligent detection frameworks.

2.1. ML Techniques for intrusion detection

ML has proven effective in a wide range of cybersecurity applications, including network intrusion detection, anomaly detection, and threat classification.

NB is a straightforward yet efficient probabilistic classifier based on Bayes' theorem (Mukherjee & Sharma, 2012). NB performs well in practical scenarios, such as spam filtering, text classification, and intrusion detection, due to its scalability and efficiency in high-dimensional data environments (Abraham & Bindu, 2021). Zeng (2024) proposed a hybrid model for network anomaly detection that integrates DL models such as multilayer perceptrons and deep neural networks with classical ML classifiers, including DT, RF, and SVM. An anomaly-based intrusion detection system using an ensemble classification approach was proposed to detect unknown attacks on web servers (Kamarudin et al., 2017). SVMs have been a popular choice for binary and multi-class classification tasks on the dataset (Gao et al., 2009; Kausar et al., 2011; Vapnik, 2000; Yang & Wang, 2008). Mukkamala, Janoski, and Sung (2002) demonstrated the effectiveness of SVMs in detecting both frequent and rare attack types. However, they noted computational inefficiency when applied to the full dataset, prompting interest in feature selection and dimensionality reduction. Kausar et al. (2011) presented a comprehensive review of SVM and its kernel-based approaches in IDS, highlighting their effectiveness in improving detection rates and minimising false alarms for enhanced network security.

KNN is a widely used, instance-based learning classification algorithm based on the principle of similarity. It classifies a given unknown point using the majority class of its k nearest neighbours in the feature space. Unlike most other algorithms, KNN is often referred to as a lazy learner because it does not involve an explicit training phase; instead, it stores all the training data and performs computation only when classification is required (Pathak & Pathak, 2020). Liao and Vemuri (2002) proposed an algorithm based on KNN for modelling program behaviour in intrusion detection, demonstrating its effectiveness in reducing false positives and computational overhead compared to traditional sequence-based methods, while highlighting its potential scalability and relevance to text categorisation techniques.

DT-based models, particularly RF, have shown high accuracy and interpretability (Resende & Drummond, 2018). The fundamental idea behind RF is to reduce variance and increase model generalisation by combining the results from a diverse set of trees, each trained on a different subset of the data (Zhang & Zulkernine, 2006). This ensemble approach is particularly advantageous in the domain of IDS, where the ability to generalise over complex, high-dimensional network data is essential (Kathiresan et al., 2022). Studies (Farsi et al., 2024; Tavallaee et al., 2009) have highlighted the ability of RF models to handle the dataset's mix of categorical and numerical features effectively. The study by Farnaaz and Jabbar (2016) developed an intrusion detection model using an RF classifier, demonstrating a high detection accuracy and a low false alarm rate on the NSL-KDD dataset, highlighting its robustness compared to the other traditional classifiers. Zhang et al. (2008) proposed systematic frameworks for misuse, anomaly, and hybrid IDS using the RF algorithm, enabling both pattern-based intrusion detection and outlier-based anomaly detection. Their experimental results on the KDD'99 dataset demonstrated superior detection accuracy and lower false-positive rates than existing methods, particularly highlighting the effectiveness of the hybrid approach.

Chandrashekhar and Raghuveer (2012) proposed clustering algorithms for classification. It is impractical to always have a labelled dataset for intrusion detection systems. There should be a proper system to classify these unlabelled attacks to maintain a safe network environment. The proposed method used the K-means algorithm, the fuzzy c-means algorithm, and the Mountain Clustering algorithm for classification. Songma et al. (2012) introduced an unsupervised clustering-based model to detect intrusions. The proposed solution is a two-phase classification method. In the first phase, a set of data is clustered by the k-means algorithm. In the second phase, outliers are identified by a distance-based technique, and a class label is assigned to each pattern. Existing statistical anomaly detection methods, such as the nearest neighbour approach, K-means clustering, or probabilistic analysis, involve costly point-to-point computation in organising data into clusters (Sarasamma et al., 2005). The proposed method is called the Multilevel Hierarchical Kohonen Net (K-Map), which uses a Single-layer winner-takes-all K-Map algorithm and a Multilayer hierarchical winner-takes-all K-Map algorithm. The study by Aung and Min (2017) proposed a hybrid intrusion detection model combining K-means clustering and RF classification, demonstrating improved detection accuracy and reduced computational cost compared to using RF alone. Recent studies have explored the integration of DL and blockchain technologies for intrusion detection in Industrial Control Systems (ICS), aiming to enhance detection accuracy while ensuring data integrity and trust.

Recent advances in IDS have increasingly focused on DL architectures due to their ability to model complex and high-dimensional network traffic patterns. Jose and Jose (2023) conducted a comparative study examining the effectiveness of DL models in IoT-based IDS environments. Their analysis evaluated DNN, LSTM, and CNN. Experimental results demonstrated that DNN achieved an accuracy of 94.61%, while LSTM and CNN achieved 97.67% and 98.61%, respectively, indicating the superior capability of DL models in capturing intricate traffic behaviours and outperforming conventional ML approaches. In parallel, federated learning has emerged as a significant paradigm for IDS, particularly in distributed and IoT environments. Federated learning enables decentralised model training across multiple nodes without sharing raw network data, thereby addressing privacy, scalability, and data sovereignty concerns. Recent surveys (Hernandez-Ramos et al., 2025) highlight the growing adoption of federated IDS frameworks, outlining both their advantages and associated challenges, including communication overhead, model heterogeneity, and robustness against adversarial attacks. While DL and federated learning approaches demonstrate strong performance in large-scale environments, they often require substantial computational resources and may not explicitly address extreme class imbalance or feature redundancy in intrusion datasets. In particular, blockchain-assisted DL–based IDS frameworks have demonstrated strong performance on both legacy (NSL-KDD) and modern (CICIDS) datasets, while highlighting open challenges related to computational overhead and deployment on resource-constrained ICS environments (Devi Priya et al., 2025). Pawana et al. (2025) introduced the first roaming-specific intrusion detection dataset for cloud-native 5G networks and showed that sequence-aware DL models, particularly LSTM and GRU, significantly outperform feed-forward and convolutional architectures in detecting rare, high-impact roaming attacks under severe class imbalance.

2.2. Feature selection techniques for intrusion detection

ML models often operate on large datasets that contain numerous features, many of which may be redundant or irrelevant. Feature selection is a crucial preprocessing step that helps improve model accuracy and efficiency by reducing the dimensionality of data while preserving essential information (Amiri et al., 2011; Battiti, 1994). Many researchers have identified that the presence of irrelevant features can harm the performance of learning systems (Balasaraswathi et al., 2017). Nimbalkar and Kshirsagar (2021) highlighted the significance of feature selection for intrusion detection in IoT, aiming to develop models that require minimal time for training while ensuring high performance. High-dimensional data presents several challenges that can significantly impact the performance of ML models (Hasan et al., 2016). Hasan et al. (2016) addressed these challenges, emphasising the importance of feature selection in mitigating issues such as the curse of dimensionality, overfitting, feature redundancy, and irrelevance, ultimately improving model accuracy and efficiency. Dimensional reduction techniques can be broadly categorised into feature selection and feature extraction methods (Saini & Sharma, 2018). Feature selection is the process of selecting a non-redundant subset of features from the original set and removing redundant attributes. Traditional feature selection techniques have been grouped into four main categories: filter, wrapper, embedded, and hybrid (Padmaja & Vishnuvardhan, 2016; Zebari et al., 2020). Much later, another technique, called ensemble feature selection, was also adopted (Lazar et al., 2012). Feature extraction is the method that extracts new features from the existing dataset; it is a very effective way to decrease the number of features for preprocessing without losing relevant features in the dataset (Zebari et al., 2020).

Ambusaidi et al. (2016) proposed a novel approach to intrusion detection with a filter-based feature selection algorithm known as Flexible Mutual Information Feature Selection (FMIFS). FMIFS is an enhancement over the existing Mutual Information Feature Selection (MIFS) and Modified Mutual Information Feature Selection (MMIFS) algorithms (Hoque et al., 2014; Song et al., 2014). In their work, the authors integrated FMIFS algorithm with the Least Squares Support Vector Machine (LSSVM) classifier to develop an enhanced IDS. LSSVM is a variation of the conventional SVM that replaces inequality constraints with equality constraints in its optimisation formulation. This modification reduces the computational complexity by converting the quadratic programming problem into a set of linear equations. The effectiveness of the proposed LSSVM-IDS integrated with FMIFS was evaluated using three benchmark intrusion detection datasets: KDD Cup 99, NSL-KDD, and Kyoto 2006+. The experimental results demonstrated that the integrated model achieved superior classification accuracy, detection rate, false positive rate, and F-measure compared to several existing detection systems.

In their study, Louvieris et al. (2013) proposed an effects-based feature identification method that combines k-means clustering, NB feature selection, and C4.5 DT classification. This method facilitates the classification of relevant and statistically significant feature sets and provides a statistical standard for the validity of the approach. The inclusion of the NB feature selection and the Kruskal–Wallis test enables the identification of statistically significant features, reducing noise and increasing analysis efficiency by filtering out irrelevant data. This proposed method demonstrates that a statistically relevant and reduced feature set can effectively identify cyber attacks as anomalies in a complex cyber network environment. The significance of noisy data, the optimal number of principal components (PCs), and the effectiveness of Principal Component Analysis (PCA) for intrusion detection were investigated by Keerthi Vasan and Surendiran (2016). The study investigates the impact of dimensionality reduction using PCA on network traffic analysis for intrusion detection. The objective is to determine the ideal number of PCs that maintain classification accuracy while reducing data dimensionality. Experimental evaluation using the KDD CUP and UNB ISCX datasets shows that the first 10 PCs achieve classification accuracies of 99.7% and 98.8%, respectively, which is nearly identical to the accuracies obtained using the original 41 and 28 features. This demonstrates that PCA effectively reduces dimensionality while retaining essential data properties for intrusion detection. Additionally, the study highlights the impact of noisy data on PCA performance, emphasising the importance of noise reduction in network traffic data preprocessing. Tan et al. (2010) have proposed an LDA-based feature selection method to reduce the computational cost of attack detection using payload-based anomaly IDS.

2.3. Meta-Heuristic algorithms for feature selection

With the rapid advancement in technology, the volume of datasets has increased. ML techniques such as classification, clustering, data analysis, and feature selection can be used to address these challenges. Nature-inspired algorithms, which mimic biological and physical processes, have also become significant in addressing such problems. Feature selection, in particular, benefits from these algorithms, which help identify the most relevant features and remove redundant features (Kauser & Senthil, 2018).

Balasaraswathi et al. (2017) provide a survey of feature selection techniques for IDS, including bioinspired algorithms and non-bioinspired algorithms. The authors conclude that integrating bio-inspired algorithms with other techniques enhances performance and offers effective solutions to complex problems. Ghanem et al. (2022) proposed a new method known as MOB-EBATMLP. In the first step, the multi-objective BAT algorithm (MOBBAT) is used to develop a feature selection algorithm based on a proficient wrapper approach. To improve IDS performance, the next step uses the features of the first stage to classify traffic using the recently improved BAT algorithm (EBAT) to train multilayer perceptrons (EBATMLP).

A modified NB algorithm based on an ABC algorithm is proposed by Yang et al. (2018), and the method is tested on two public datasets. Experimental results indicate that, compared with NB classifiers using genetic algorithms, Grey Wolf Optimiser, Water Wave Optimisation, and

the basic NB classifier, the proposed approach substantially increases intrusion detection accuracy, effectively identifying a wide range of network attacks and improving overall network security performance.

Aghdam and Kabiri (2016) proposed an ACO-based method for intrusion detection. This method reduced the number of features by approximately 88% and lowered the detection error by around 24% on the KDD Cup 99 dataset. Lee et al. (2006) introduced a bio-inspired feature-subset selection approach using genetic algorithms and ant colony algorithms, with the resulting subset evaluated by a neural network. The findings suggest that ant colony algorithms provide a promising strategy addressing the feature-subset selection in inductive learning for pattern classifiers.

Pandithurai et al. (2024) proposed hybrid models for Distributed Denial of Service (DDoS) attack detection in cloud environments. One such approach integrates HBO for feature selection with a Bi-LSTM classifier, achieving 97% accuracy and outperforming traditional models such as LSTM, DNN, DBN, and ANN. This method enhances DDoS attack prediction by optimising key features and improving classification performance.

The ABC algorithm continues to receive considerable attention in optimisation research due to its strong exploration–exploitation balance and adaptability to dynamic search environments. Recent comprehensive surveys (Ibrahim et al., 2025) document numerous variants, hybridisations, and enhancements of the ABC algorithm, including multi-objective formulations and applications across engineering, finance, healthcare, and social sciences. These studies emphasise improvements in convergence behaviour, diversity preservation, and global search capability. In the context of intrusion detection and feature selection, several hybrid ABC-based approaches have been proposed. Alsaleem (2025) introduced a two-stage hybrid feature selection framework combining ABC and GA. In their method, ABC is applied in the first stage, followed by GA as a wrapper-based refinement mechanism. The approach was evaluated on NSL-KDD, UNSW-NB15, and CIC-IDS2017 datasets using RF and XGBoost classifiers. Despite these advances, existing ABC-based and hybrid optimisation approaches primarily focus on improving global accuracy and feature reduction, often without explicitly addressing extreme class imbalance or rare attack preservation.

3. Methodology

In this study, a novel intrusion detection model is developed by integrating ML techniques with meta-heuristic optimisation methods to effectively detect four major types of attacks. The proposed hybrid approach plays a vital role in identifying abnormal traffic patterns and improving detection accuracy, particularly for rare attack types. The implementation of this model uses two levels: feature selection using the hybrid ABC-HBA algorithm, and a self-training classification stage using simulated unlabelled data for intrusion detection. Fig. 3 presents the detailed architecture and functional components of the proposed hybrid IDS framework.

3.1. Dataset

Researchers have long recommended using the KDD Cup 1999 dataset for several decades to create anomaly-based intrusion detection systems and other tools for protecting computer networks (UCI Machine Learning Repository, 1999). Stolfo developed this dataset, which was constructed using the information gathered during the DARPA'98 IDS evaluation program (Al-Mamory & Jassim, 2013; Protić, 2018). About 5 million connection records, each containing roughly 100 bytes, can be created from the approximately 4 gigabytes of compressed raw (binary) tcpdump data of seven weeks' worth of network traffic that makes up DARPA'98. There are about 2 million connection records in the two weeks of test data. The 4,900,000 single connection vectors that make up the KDD training dataset each have 41 features and are classified as either normal or attacked, with only one specific attack type (Tavallaee et al., 2009).

This study is conducted as a feasibility analysis to evaluate the effectiveness of the proposed framework in detecting network intrusions, with particular emphasis on extremely rare attack categories. The primary objective is not to model contemporary network traffic patterns, but to evaluate the effectiveness of a feature-selection–driven optimisation framework under conditions of extreme class imbalance, where minority attack classes constitute approximately 0.01% of the dataset. Although the KDD Cup 1999 dataset does not fully reflect modern traffic, it provides a controlled benchmark to rigorously assess the framework's robustness in detecting extremely rare attack classes. The severe imbalance and clearly defined minority categories, such as U2R, create a challenging optimisation landscape in which feature selection methods may easily bias toward majority classes. Indeed, while many prior studies report high overall accuracy on this dataset, detection performance for rare attack classes remains consistently low (Bhati et al., 2020). By explicitly addressing optimisation bias under skewed data distributions, the proposed hybrid framework enhances minority-class sensitivity while preserving overall classification stability. The observed improvement in detecting low-frequency attack patterns therefore supports the feasibility and imbalance-robustness of the proposed approach.

Since the high volume of data to be processed requires significant computational resources, a subset containing only 10% of the training data, taken randomly from the original dataset, was used in this study as part of the training process. Furthermore, the Corrected KDD dataset was also utilised during training to enhance model robustness and evaluate performance against previously unseen attack types. As illustrated in Fig. 3(a), both labelled and unlabelled data from these datasets are incorporated into the training phase for model development. The Corrected KDD dataset differs from the 10% KDD and Whole KDD datasets as it includes 14 new types of attacks, designed to test the IDS performance on unknown attack forms. In the complete and 10% KDD datasets, there are 24 attack types in total, while the Corrected KDD dataset contains 38. There are mainly five classes in the dataset: Normal, DoS, Probe, U2R, and R2L. It is also important to highlight that the KDD's training dataset and Corrected KDD contain a large number of attacks for the categories Normal, Probe, and DoS, representing approximately 99.76% of the entire dataset (Araujo et al., 2010).

3.2. Data preprocessing

As shown in Fig. 3(b), the data preprocessing stage standardises and cleans the raw dataset by performing normalisation, encoding, and the removal of irrelevant or redundant features. The processed data is then split into training and testing sets, ensuring effective learning and unbiased model evaluation before feature optimisation.

3.2.1. Hybrid approach to address data imbalance

In real-world datasets, class imbalance can significantly impact the performance of ML models. In this study, the KDD Cup 1999 dataset exhibits a highly imbalanced distribution of attack classes, where certain threats are severely under-represented. To mitigate this issue, a hybrid resampling approach is employed, combining the Synthetic Minority Over-sampling Technique (SMOTE) and Random Under-Sampling (RUS) to achieve a more balanced class distribution. This strategy increases the representation of minority classes while reducing the dominance of the majority class, thereby preventing the model from being biased toward prevalent attack categories. Furthermore, by applying this resampling prior to the feature optimisation stage, the evaluation of candidate feature subsets is conducted on a more balanced dataset, which supports minority-sensitive feature selection and reduces the likelihood of favouring features that primarily enhance majority-class performance.

First, the majority class was identified, and any class with a sample size less than 50% of the majority class was considered a minority class. To ensure unbiased evaluation and prevent data leakage, the

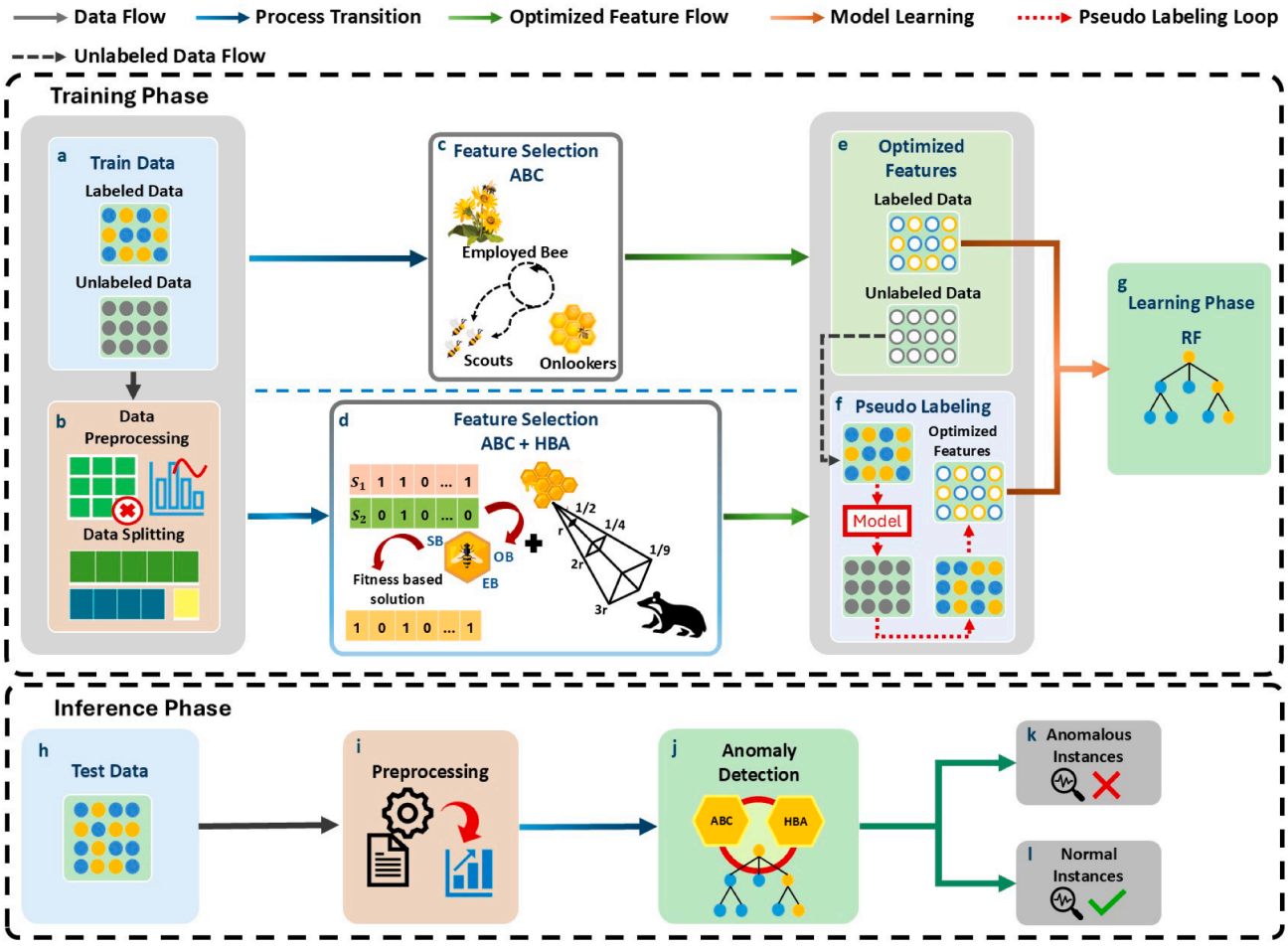

Fig. 3. Architecture of the proposed intrusion detection framework.

dataset was first split into training and test sets. SMOTE was then applied exclusively to the training set to generate synthetic samples for the under-represented classes, thereby increasing their representation within the training data. Subsequently, RUS was applied to the majority class within the training set, reducing it to 50% of its original size. This hybrid resampling strategy ensures a more balanced class distribution while retaining meaningful data characteristics. The effectiveness of the approach was verified by analysing the resulting class distribution after resampling. Overall, this combined strategy enables the model to learn from a more representative dataset, improving generalisation performance, reducing bias toward majority classes, and mitigating the risk of overfitting.

3.3. Proposed hybrid feature optimisation

The ABC algorithm is a swarm-based meta-heuristic optimisation technique developed by Karaboga (2005) to solve numerical optimisation problems. The algorithm draws inspiration from the intelligent foraging behaviour observed in honey bees. In the ABC algorithm, three types of bees play a role in optimisation (Karaboga, 2010). Employed bees search for previously known food sources, which are candidate solutions, and communicate their results to onlooker bees. Onlooker bees analyse these communicated solutions and choose the most valuable ones, with higher-quality solutions having a higher probability of being selected. Meanwhile, scout bees abandon poor solutions and randomly search for new reasonable solutions to prevent stagnation and encourage exploration (Karaboga & Basturk, 2008). This cooperative search strategy enables the ABC algorithm to effectively balance exploration and exploitation in the search space.

ABC techniques are an effective and robust means of selecting features based on honey bee foraging patterns. ABC algorithms improve the accuracy of ML models by reducing feature-set complexity. In this study, the ABC algorithm is employed as a feature selection mechanism, thereby improving the subset of top features that contribute to intrusion detection performance. The search algorithm progresses through three major phases:

1. Employed bees phase, where each bee explores a neighbouring solution by flipping a randomly selected feature bit and updates its position if an improvement is found;
2. Onlooker bees phase, where solutions are selected probabilistically based on their fitness values, and further refinements are made through local modifications; and
3. Scout bees phase, where the worst-performing solutions are replaced with randomly generated feature subsets to maintain diversity and avoid premature convergence.

ABC algorithm was employed to optimise the subset of features that maximises the classification performance. Each i-th bee represents a binary vector $x_i = [x_{i1}, x_{i2}, \ldots, x_{iD}]$, where $x_{ij} \in \{0, 1\}$ indicates whether the j^{th} feature is selected (1) or not (0), and D denotes the total number of features in the dataset. This binary representation enables the algorithm to efficiently explore different combinations of features and assess their contribution to intrusion-detection performance.

1) Fitness evaluation

To ensure robustness under extreme class imbalance, the macro-averaged F1-score is adopted as the optimisation objective. In contrast to the overall accuracy or weighted F1-score, which are dominated by majority-class performance, macro-F1 assigns equal importance to each class irrespective of its frequency:

$$F_{1,\text{macro}} = \frac{1}{C}\sum_{i=1}^{C} F_{1,i} \tag{1}$$

where C denotes the total number of classes and $F_{1,i}$ represents the F1-score of class i.

Under severe imbalance conditions, where minority attack categories may constitute approximately 0.01% of the dataset, optimisation driven by accuracy can lead to feature subsets that ignore minority-discriminative attributes. In contrast, macro-F1 penalises poor recall or precision in rare classes equally to majority classes, thereby enforcing minority-sensitive search pressure during feature selection. This formulation ensures that the optimisation process preserves features that contribute to rare attack detection rather than favouring majority-dominant patterns.

The fitness of a feature subset x_i is defined based on the macro F1-score of an RF classifier, adjusted by a feature penalty term that encourages smaller subsets. The fitness value is calculated as:

$$\text{Fitness}(x_i) = \alpha \cdot \text{F1}(x_i) - \beta \cdot \frac{|S_i|}{|F|} \tag{2}$$

where $\text{F1}(x_i)$ denotes the macro F1-score obtained using the selected features S_i, $|S_i|$ is the number of selected features, $|F|$ is the total number of available features, and α and β are weight coefficients controlling the balance between accuracy and feature reduction.

In this study, these parameters are empirically set as $\alpha = 1.0$ and $\beta = 0.1$.

2) Employed bees phase

During this phase, each employed bee generates a new candidate solution v_i by exploring the neighbourhood of its current solution x_i. In the standard ABC algorithm, this is achieved using:

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \tag{3}$$

where x_k is a randomly selected neighbouring solution and x_{kj} is the j^{th} dimension of x_k. The parameter $\phi_{ij} \in [-1, 1]$ controls the step size.

Since the feature selection problem is defined in a binary search space, a modified update strategy is adopted. Each solution x_i is encoded as a binary vector, where each bit represents the inclusion (1) or exclusion (0) of a feature.

To ensure controlled and reproducible exploration, the stochastic update in Eq. (3) is replaced with a deterministic single-bit flipping mechanism. At each iteration, exactly one feature index j is selected, and the corresponding bit is flipped as:

$$v_{ij} = \begin{cases} 1 - x_{ij}, & \text{if } j = j^* \\ x_{ij}, & \text{otherwise} \end{cases} \tag{4}$$

where j^* denotes the selected feature index for the current iteration.

This strategy ensures that only one feature is modified at a time, enabling a fine-grained local search while improving stability and reproducibility compared to fully stochastic updates.

3) Onlooker bees phase

In the onlooker phase, each food source is selected with a probability that is proportional to its fitness value. This probability is defines as:

$$p_i = \frac{\text{Fitness}_i}{\sum_{n=1}^{N} \text{Fitness}_n} \tag{5}$$

This probabilistic selection mechanism ensures that higher-quality feature subsets are more likely to be chosen for further refinement, allowing the algorithm to focus its search on more promising regions of the feature space while still maintaining diversity.

4) Scout bees phase

If a food source fails to improve after a predefined number of iterations, it is replaced by a new randomly generated feature subset. This is achieved using the update rule:

$$x_{ij} = x_{\min,j} + \text{rand}(0,1) \times (x_{\max,j} - x_{\min,j}) \tag{6}$$

In the deterministic variant used in this study, the worst-performing bee is replaced with a fixed alternating binary pattern to maintain diversity and avoid stagnation. Formally, the new feature vector is defined as:

$$x_d^{\text{new}} = d \bmod 2, \quad d = 0, 1, \ldots, |F| - 1 \tag{7}$$

where d denotes the feature index. This generates a sequence such as 0,1,0,1,… which is deterministic and reproducible across iterations. By using this pattern, the algorithm preserves diversity in the population while avoiding random initialization that could slow convergence.

Although the ABC algorithm demonstrates strong exploration capabilities and robustness in feature selection, it suffers from several inherent limitations. One of the key drawbacks is its tendency to converge prematurely due to insufficient exploitation during later iterations (Djellali et al., 2018). The random perturbation mechanism in the employed and onlooker phases may lead to the discarding of potentially useful solutions, especially in high-dimensional feature spaces. Moreover, the stochastic nature of scout bee replacement can slow convergence and reduce stability in approaching the global optimum (Djellali et al., 2018). To overcome these limitations, the ABC algorithm is integrated with the HBA, a recently developed swarm intelligence method that provides a stronger balance between exploration and exploitation through its dynamic density and foraging modes (Hashim et al., 2022). The hybridisation of ABC and HBA combines the exploration strength of ABC with the exploitation efficiency of HBA, resulting in faster convergence, improved feature subset quality, and enhanced classification accuracy for intrusion detection.

3.4. Hybrid ABC–HBA feature optimization

HBA operates using two primary modes: digging and honey searching, to explore and refine the search space effectively. In the context of feature selection, each honey badger represents a candidate feature subset encoded as a binary vector, where each bit indicates whether a specific feature is selected. The position of each honey badger corresponds to a feature subset, and the objective is to minimise a fitness function based on classification performance while reducing the number of selected features. The algorithm updates positions adaptively using a density factor that controls the transition from exploration to exploitation over time, allowing it to converge efficiently toward optimal subsets of features.

In this study, the optimisation process employs a tightly integrated ABC–HBA meta-heuristic, where exploration and exploitation operate concurrently within each iteration. Fig. 3(d). The ABC algorithm is first employed to generate diverse feature subsets through local search and probabilistic neighbourhood updates. This phase ensures effective exploration of the high-dimensional search space and avoids premature convergence. Subsequently, the HBA is integrated into the exploitation phase to refine candidate solutions by adjusting feature subsets based on the adaptive intensity and distance from the global best solution. The adaptive flipping probability in HBA is modulated based on the proximity of each candidate to the global best solution, allowing the search to fine-tune promising feature subsets while maintaining diversity. This integration ensures that features relevant to both majority and minority classes are preserved, improving the detection of extremely rare attack types.

1) Fitness function

In the proposed ABC–HBA framework, the same fitness function defined in Eq. (2) is used to ensure consistency with the ABC algorithm.

The HBA is inherently a minimization-based optimization method; therefore, the objective function is reformulated into a minimization form while preserving equivalence with the original fitness definition. Specifically, the optimization problem is defined as:

$$\min\left(1 - \text{Fitness}(x_i)\right) \tag{8}$$

This transformation ensures that solutions with higher fitness values in the original formulation correspond to lower objective values in the HBA search space. The parameters $\alpha = 1.0$ and $\beta = 0.1$ are kept identical across both ABC and HBA to maintain fairness. Thus, both algorithms operate on the same evaluation function, differing only in their optimization direction (maximization vs. minimization) and search strategy.

2) Distance and intensity calculation

To guide the exploitation phase, the HBA computes the Euclidean distance between a candidate solution x_i and the global best solution x_{best}, defined as:

$$d_i = \|x_i - x_{\text{best}}\|_2 \tag{9}$$

The smell intensity, modelled after prey-detection behaviour, is then calculated as follows:

$$I_i = \frac{1}{4\pi d_i^2} \tag{10}$$

It should be noted that this formulation does not represent physical diffusion in Euclidean space; rather, it serves as a non-linear distance-based weighting mechanism. In the binary feature space, the Euclidean distance between two solutions is proportional to their Hamming distance, thereby providing a meaningful measure of similarity between feature subsets. The inverse-square relationship ensures that solutions closer to the global best receive significantly higher intensity values, thereby increasing the probability of exploitation, while distant solutions experience rapidly diminishing influence. This adaptive decay mechanism enhances fine-grained local search without imposing a literal physical interpretation.

3) Adaptive probability update

The adaptive flipping probability for each feature dimension is computed using the intensity value I_i and a random scaling factor $\beta \in [0, 1]$.

$$P_i = \text{clip}(I_i \times \beta,\ 0,\ 1) \tag{11}$$

This probabilistic mechanism controls the extent of feature bit-flipping during the exploitation stage, balancing random exploration and adaptive convergence.

4) Position update rule

Based on the computed adaptive probability, each bit in the candidate solution is updated as follows:

$$x_{ij}^{(t+1)} = \begin{cases} 1 - x_{ij}^{(t)}, & \text{if rand}(0,1) < P_i \\ x_{ij}^{(t)}, & \text{otherwise} \end{cases} \tag{12}$$

This operation introduces an adaptive local search behaviour where feature bits are selectively inverted according to their likelihood of improvement, promoting better feature combinations across iterations. The workflow of Hybrid ABC-HBA feature Optimisation is presented in Fig. 4 as Algorithm 1.

After obtaining the optimised feature subset through the hybrid ABC-HBA algorithm, the proposed framework transitions into a self-training phase using simulated unlabelled data, where pseudo-labels are iteratively generated for previously unseen samples. This is followed by supervised model training, iterative refinement, and final inference for multi-class intrusion classification. This multi-stage process enables the model to leverage both labelled and pseudo-labelled data while progressively improving classification performance.

Algorithm 1: Hybrid ABC - HBA Feature Optimization

Initialize random binary population $P = \{S_1, S_2, \ldots, S_N\}$
for each solution $S_i \in P$ **do**
  evaluate fitness F_i
End for
Identify global best solution S_{best} and F_{best}

for each iteration t $\in T_{\max}$ **do**
  for each bee $S_i \in P$ **do**

    ABC Exploration - Employed Bee Phase
      Choose random feature index $j \in [1, D]$
      Randomly select another bee S_k
        while selected bee is same as current bee **do**
          Randomly select another bee S_k
        End while
      If exploration condition ($r < |\phi|$) is satisfied then
        Randomly flip the selected feature $S_{i,j}$ to explore a new possibility $S'_{i,j}$
      End if

    HBA-Driven Exploitation - Adaptive Intensity Phase
      Compute Euclidean d distance to global best
      Compute intensity I
      Compute adaptive probability p
      for each feature $f = 1,2,3, \ldots\ldots D$ **do**
        if $threshold <$ adaptive probability **then**
          Invert the selected features $S_{i,f}$ bit to move closer to the best solution $S'_{i,f}$
        End if
      End for

    Fitness Evaluation and Greedy Selection
      Evaluate the fitness of the candidate solution S'_i using the fitness function
      if $F'_i < F_i$ **then**
        Replace the current solution S_i and its fitness F_i with the new ones S'_i and F'_i
      End if
  End for
  Global Best Solution Update
    Identify current best bee S_b with $F_b = \min_i F_i$
    if $F_b < F_{\text{best}}$ **then**
      $S_{\text{best}} = S_b,\ F_{\text{best}} = F_b$
    End if
End for

Fig. 4. Pseudo-code of the hybrid ABC–HBA algorithm.

Definition of Notations

Notation	Definition
P	Population matrix containing all candidate feature subsets
N	Population size, the number of bees (candidate solutions)
D	Number of features in the solution vector
j	Randomly selected feature index for ABC exploration (single feature)
f	Feature index used in HBA exploitation loop (all features)
k	Index of another randomly selected bee used in ABC exploration
ϕ	Random coefficient $\in [-1, 1]$ used in ABC exploration phase
r	Random number $\in [0, 1]$ used for exploration condition
β	Random factor $\in [0, 1]$ used in HBA adaptive intensity phase
p	Adaptive probability computed from intensity and β in HBA exploitation -------- Eqa (11)
I	Intensity of smell, computed based on Euclidean distance to global best in HBA -------- Eqa (10)
d	Euclidean distance between candidate solution S'_i and global best solution S_{best} -------- Eqa (9)
S_i	Original solution vector for bee i (before exploration/exploitation)
S'_i	Candidate solution vector for bee i after exploration/exploitation updates
$S_{i,j}$	Feature j of the original solution S_i during ABC exploration
$S'_{i,j}$	Feature j of the candidate solution S'_i after ABC exploration flip
$S_{i,f}$	Feature f of the original solution S_i during HBA exploitation
$S'_{i,f}$	Feature f of the candidate solution S'_i after HBA exploitation inversion
S_{best}	Global best solution found so far
S_b	Current best solution in the current iteration before comparing with S_{best}
F_i	Fitness of the original solution S_i -------- Eqa (2)
F'_i	Fitness of the candidate solution S'_i after exploration/exploitation updates
F_{best}	Fitness of the global best solution S_{best}
F_b	Fitness of the current best solution S_b in the current iteration
T_{max}	Maximum number of iterations
$threshold$	User-defined threshold for deciding whether to invert a feature during HBA exploitation

Fig. 5. Notation definitions used in the proposed framework.

3.5. Optimized feature representation

The optimised feature matrix produced by the hybrid meta-heuristic process contains the most relevant and discriminative attributes from both the labelled and simulated unlabelled subsets of the training data. These features serve as the unified input for the self-training phase based on simulated unlabelled data, ensuring that only high-quality and non-redundant characteristics contribute to model learning. Both labelled and simulated unlabelled data share the same optimised feature representation, thereby maintaining consistency during pseudo-label generation and iterative retraining.

3.6. Pseudo-labeling mechanism

The pseudo-labelling strategy enables the model to effectively utilise additional data through an iterative self-training mechanism (Amini et al., 2025; Yang et al., 2021; Zhang & Li, 2020; Zhu, 2005). However, pseudo-labelling can introduce noisy or incorrect supervision, as generated labels depend on the model's current predictions. Such sub-optimal supervision may degrade performance, highlighting the need for robust learning strategies to handle noisy labels (Sharma & Silva, 2026). The corrected KDD Cup 1999 test dataset was designed to provide a realistic evaluation scenario for intrusion detection systems by introducing 14 novel attack types that are absent from the training set, while excluding two attack types present during training, as documented in the official KDD Cup specification (UCI Machine Learning Repository, 1999). These additional attacks introduce a distribution shift, reflecting real-world conditions where previously unseen threats emerge in network traffic.

Although the corrected dataset is fully labelled, the labels are intentionally withheld during training and treated as unlabelled to simulate realistic deployment scenarios where ground-truth annotations for emerging threats are not readily available. Accordingly, the proposed approach follows a self-training strategy with unlabelled data simulation, rather than relying on inherently unlabelled data.

To facilitate the categorization of unseen attack instances into broader attack categories, a pseudo-labelling mechanism is employed. As illustrated in Fig. 6, initially, the RF classifier is trained on the labelled portion of the optimized dataset to establish a base model capable of identifying fundamental traffic behaviour patterns. The trained model is then used to generate pseudo-labels for the corrected dataset, assigning predicted labels along with associated confidence scores. These pseudo-labelled instances are subsequently incorporated into the training process, enabling the model to adapt to novel attack patterns and improving robustness under distributional differences between training and test data.

The confidence threshold τ was empirically set to 0.90, such that only predictions with at least 90% posterior probability are retained as pseudo-labels. This choice reflects a balance between reliability and data utilisation: a higher threshold reduces the likelihood of incorporating incorrectly labelled instances, while still preserving a sufficient number of samples for effective self-training with simulated unlabeled data.

To validate this selection, a sensitivity analysis was conducted using $\tau \in \{0.80, 0.85, 0.90, 0.95\}$. The results indicated that $\tau = 0.90$ provides the most favourable trade-off between pseudo-label precision and the number of retained samples, leading to stable improvements in downstream classification performance. Formally, the pseudo-labelled dataset D_p is represented as:

$$D_p = \Big\{ (x_i, \hat{y}_i) \;\Big|\; x_i \in D_u, \quad \hat{y}_i \in \{\text{Normal, DoS, Probe, R2L, U2R}\}, \quad P(\hat{y}_i \mid x_i) \geq \tau \Big\} \tag{13}$$

where D_u denotes the unlabelled dataset, x_i represents an unlabelled instance, $\hat{y}_i$ is the predicted intrusion class assigned by the model, and $P(\hat{y}_i \mid x_i)$ is the confidence associated with that prediction. Only predictions with confidence greater than or equal to the threshold τ are retained to ensure reliable pseudo-label generation.

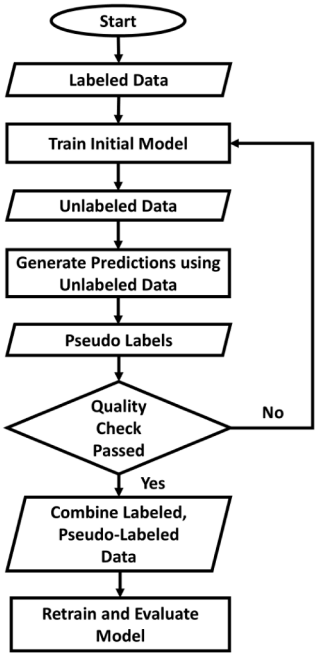

Fig. 6. Pseudo-labeling process flowchart.

3.6.1. Iterative self-training and model refinement

The high-confidence pseudo-labelled instances are merged with the original labelled dataset to form an expanded training set:

$$D' = D_l \cup D_p \tag{14}$$

where D_l is the original labelled subset. The RF classifier is then retrained on this expanded dataset to incorporate additional knowledge extracted from the unlabelled samples. After retraining, the updated model predicts new pseudo-labels for the remaining unlabelled data, and the process repeats iteratively. Through this iterative process, the classifier's decision boundaries become more generalised, enabling it to effectively identify both frequent attack types and rare, infrequent attack types.

3.7. Learning phase with RF

The RF model functions as the core learning component due to its robustness, ensemble averaging mechanism, and adaptability to mixed data types. It constructs multiple decision trees, each trained on bootstrapped samples of the expanded dataset, and aggregates their outputs via majority voting (Breiman, 2001). This ensemble approach ensures reduced variance, improved generalisation, and balanced detection across multiple intrusion categories (Farsi et al., 2024). The RF model's feature importance scores also offer interpretability, providing insights into which optimised attributes most influence the classification of specific attack types.

3.8. Inference and multi-class intrusion classification

Once the pseudo-labelling iterations converge, the finalised RF model is deployed for inference. The unseen *test data* are preprocessed and transformed using the same optimised feature set obtained during training. The trained model classifies each instance into one of the five major network traffic categories: *Normal*, *DoS*, *Probe*, *R2L*, and *U2R*.

The classification output for each test instance x_j is defined as:

$$\hat{y}_j = \arg\max_{c \in C} P(y = c \mid x_j) \tag{15}$$

where $C = \{\text{Normal}, \text{DoS}, \text{Probe}, \text{R2L}, \text{U2R}\}$ represents the set of possible intrusion classes. The model assigns the class with the highest posterior probability to each instance, ensuring accurate differentiation among multiple intrusion behaviours. This probabilistic framework improves detection accuracy and enhances interpretability, as lower confidence values may indicate ambiguous or previously unseen traffic patterns.

3.9. Evaluation metrics

There are a number of evaluation metrics that can be used to measure the performance of the model. The most commonly used metrics in intrusion detection are f1-score, precision, and recall. These metrics provide complementary insights into the classifier's effectiveness, particularly in scenarios involving class imbalance and multi-class intrusion behaviour.

4. Results and discussion

In this section, we present a comprehensive evaluation and discussion of the performance of the proposed framework. For comparative analysis, four models were developed: the baseline Random Forest (RF) model, the RF with pseudo-labelling, the RF-ABC hybrid model, which integrates the ABC optimization algorithm, and the RF-HBA model, which incorporates the HBA algorithm.

All experiments were conducted on a machine equipped with an AMD Ryzen 3 5300U CPU (4 cores, 2.6 GHz), Radeon Graphics, and 16 GB RAM, running Windows 11 (64-bit). All models were implemented in Python 3.10 with Scikit-learn, and no GPU acceleration was used. The same computational environment was used for all models to ensure a fair comparison of training and testing times.

4.1. Convergence behaviour of ABC and ABC–HBA

Figs. 7 and 8 illustrate the convergence behaviour of the ABC and ABC–HBA algorithms using both the fitness value and the F1-score across optimisation iterations. To provide a comprehensive understanding, the fitness evolution reflects the optimisation objective, while the F1-score is used as the primary performance indicator for classification effectiveness. As shown in the fitness convergence plot, the ABC algorithm demonstrates a gradual increase in fitness values due to its maximisation-based formulation, whereas the ABC–HBA model exhibits a decreasing trend consistent with its minimisation objective of $1 - F_{\text{fitness}}(x)$. Despite this difference in optimisation direction, both methods converge steadily, indicating stable search behaviour.

In terms of classification performance, both algorithms show consistent improvements in F1-score over iterations, as illustrated in the second plot. The ABC model improves from an initial F1-score of 95.21% to 97.14% at convergence, while the ABC-HBA model starts at 95.38% and achieves a higher final F1-score of 97.78%. This demonstrates the superior convergence capability of the hybrid ABC-HBA approach. The improved performance of ABC-HBA can be attributed to the enhanced balance between exploration and exploitation introduced by the HBA component, which allows the algorithm to escape suboptimal regions more effectively.

4.2. Computational efficiency analysis

Fig. 9 compares the computational efficiency of the base RF model, the ABC-optimised model, and the proposed hybrid ABC-HBA model. The base model required 25.36 seconds for training and 0.3827 seconds for testing. The integration of the ABC algorithm reduced the training time to 18.40 seconds (a 27.4% reduction) and the testing time to 0.3546 seconds (a 7.3% reduction). The proposed hybrid ABC-HBA model achieved the best performance, with a training time of 16.47 seconds and a testing time of 0.1761 seconds, representing reductions of

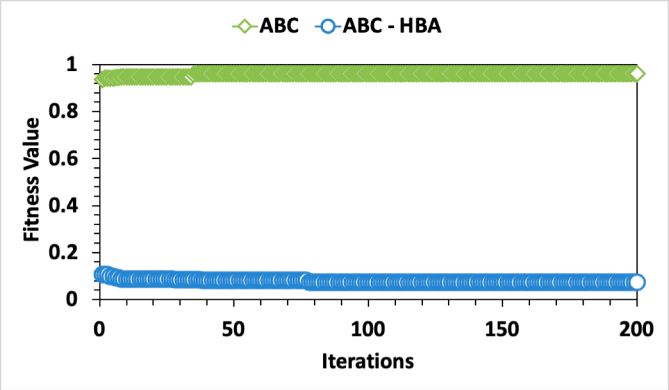

Fig. 7. Fitness values for the ABC and ABC-HBA models over iterations.

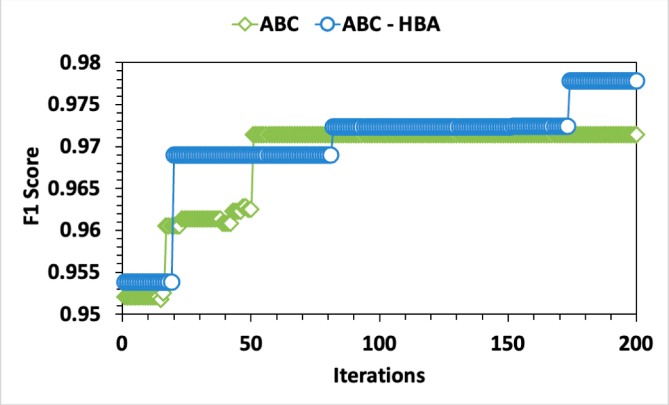

Fig. 8. F1 Score for the ABC and ABC-HBA models over iterations.

35.0% and 54.0%, respectively, compared to the base model. The computational cost of feature selection for both methods shows that ABC-HBA has selected 20 features, whereas ABC selected slightly fewer, 18 features. Both methods use a population size of 20 and 200 iteration, resulting 4000 fitness evaluation per method. Despite select more features, ABC-HBA required less total wall-clock time (6663.77 s) compare to ABC (9024.42 s), indicating that ABC-HBA achieve a more efficient optimization process while maintain a compact feature subset. These results demonstrate that the ABC-HBA optimisation not only enhances classification accuracy but also substantially reduces computational cost. The improvement can be attributed to the hybrid algorithm's balanced exploration–exploitation mechanism, which accelerates convergence and enables the selection of a more compact and effective feature subset. Therefore, the hybrid ABC-HBA model is both computationally efficient and practically suitable for real-time intrusion detection applications.

4.3. Overall classification performance

Table 1 presents a comparative evaluation of the proposed ABC–HBA model against representative state-of-the-art hybrid metaheuristic and DL–based intrusion detection approaches reported for the KDD Cup 1999 dataset, across key classification metrics, including accuracy, precision, recall, and F1-score. The base model (RF + Pseudo labelling) achieved an accuracy of 99.94%, while the ABC model recorded a slightly lower value of 99.88%. The proposed ABC-HBA model attained the highest accuracy of 99.95%, indicating marginal yet consistent improvement in overall predictive performance. In terms of F1-score, the ABC-HBA model achieved 97.54%, outperforming both the base model (94.15%) and the ABC model (94.98%) by approximately 3.35% and 2.52%, respectively. Similarly, the recall and precision values of the

Table 1
Comparative evaluation of the proposed ABC–HBA model and existing methods on the KDD cup 1999 dataset.

Reference	Technique	Accuracy	Precision	Recall	F1-Score
Alomari and Othman (2012)	Bees Algorithm + SVM	-	95.75%	-	-
Eesa et al. (2015)	Cuttle Fish Algorithm + DT	-	92.05%	-	-
Ogundokun et al. (2021)	PSO + DT	98.6%	75.3%	89.6%	81.8%
	PSO + KNN	99.6%	88.5%	96.2%	92.2%
	PSO + ANN	99.78%	90.1%	97.1%	94.2%
Shah and Trivedi (2015)	Back Propagation	96.7%	99.97%	97.27%	98.57%
Choudhary and Kesswani (2020)	Deep Neural Network (DNN)	96.3%	-	-	-
Bhati et al. (2020)	XGBoost	99.95%	-	-	-
Proposed Method	Base Line (RF)	99.93%	95.61%	90.71%	92.78%
	Base Line (RF + pseudo-labelling)	99.94%	93.23%	95.19%	94.15%
	RF + ABC	99.88%	93.50%	96.63%	94.98%
	RF + HBA	99.92%	95.81%	95.18%	94.47%
	RF + ABC - HBA	**99.95%**	**96.93%**	**98.16%**	**97.54%**

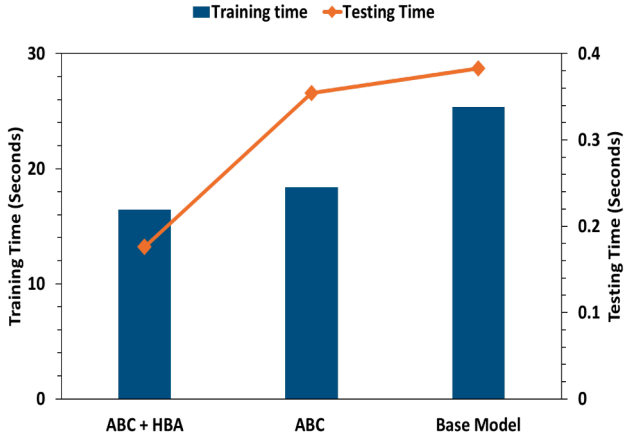

Fig. 9. Comparison of training and testing times.

hybrid model (98.16% and 96.93%) surpassed those of the other two models, confirming its ability to correctly identify attack instances while maintaining a low false-positive rate. These results demonstrate that the hybridisation of the ABC algorithm with the HBA effectively enhances both exploration and exploitation during optimisation, leading to better feature selection and improved classification capability. The consistent performance improvement across all evaluation metrics highlights the robustness and reliability of the proposed ABC-HBA model for intrusion detection tasks.

4.4. Class-wise performance on KDD cup 1999

Table 3 presents the class-wise performance of the base RF model, the ABC-optimised model, and the proposed hybrid ABC-HBA model across three key metrics: Precision, Recall, and F1-Score, for the five major categories: DoS, Probe, R2L, U2R, and Normal. The comparative evaluation reveals that the hybrid optimisation method consistently achieves superior or comparable performance across all classes, with particularly notable improvements in detecting minority and low-frequency attack types.

For the DoS class, precision improved from 99.99% in the baseline model (RF + Pseudo labelling) to 100% in the hybrid ABC-HBA model. The minor reduction observed in the ABC model (99.97%) indicates a slight weakness in its exploitation capability, as the standard ABC algorithm primarily emphasises global exploration and can underperform in refining near-optimal feature subsets. Integrating the HBA enhances the exploitation phase through adaptive intensity control, enabling the hybrid model to achieve complete elimination of false positives and stronger generalisation for frequent attack categories such as DoS.

For the Probe class, the precision and recall values fluctuate slightly between models, with the ABC model showing reduced precision (97.22%) compared to the baseline (RF + Pseudo labelling) (99.53%), but this improves substantially in the hybrid model (99.07%). Similarly, the F1-score increases from 97.75% in the ABC model to 99.46% in the ABC-HBA model. The enhanced local search capability of the hybrid algorithm refines the feature subset more effectively, reducing misclassification among closely related network behaviours and improving the overall balance between precision and recall.

In the R2L class, which represents low-frequency but complex attack types, the hybrid ABC-HBA model again demonstrates superior performance, achieving a precision of 98.95% and an F1-score of 98.95%, outperforming both the baseline and ABC models. The ABC model alone shows slightly inconsistent behaviour due to limited local refinement, while the hybrid model stabilises classification boundaries by effectively combining exploration and exploitation mechanisms, resulting in improved detection consistency.

The U2R class, the rarest and most challenging attack category, reveals the most significant performance. The baseline model (RF + Pseudo labelling) recorded an F1-score of 73.33%, which increased to 80.00% under the ABC model and further to 89.70% with the hybrid ABC-HBA model. The substantial improvement illustrates the hybrid model's enhanced ability to discover discriminative features for minority classes. By improving local search efficiency, the hybrid approach avoids overfitting to dominant classes and increases sensitivity to subtle feature variations that characterise U2R attacks.

For the Normal class, all models perform nearly identically, achieving precision and recall values above 99.9%. The hybrid model maintains this strong performance without overfitting to normal patterns, ensuring a balanced classification across both attack and non-attack traffic.

Overall, the minor variations observed in the ABC model's performance can be attributed to its limited exploitation capability, which sometimes restricts fine-grained optimisation of feature subsets. The incorporation of the HBA effectively mitigates this limitation by improving the local search process and enhancing convergence toward optimal feature combinations. Consequently, the proposed hybrid ABC-HBA model achieves a more balanced trade-off between exploration and exploitation, resulting in consistently superior classification performance across all attack categories.

4.5. ROC–AUC analysis

Fig. 10 presents the receiver operating characteristic (ROC) curves and corresponding area under the curve (AUC) values for the five attack classes: DoS, Probe, R2L, U2R, and Normal, under the three evaluated models. The AUC–ROC metric provides an overall assessment of each model's ability to discriminate between positive and negative instances across different threshold values. All three models exhibit very high and

Table 2
Stability analysis across five independent runs.

Metric	Mean	Standard Deviation
F1-score	0.99946	0.00012
Jaccard Similarity	0.651	0.042

closely overlapping ROC curves, with AUC values approaching 1.0 for all classes. This indicates that each model achieves an excellent balance between the true positive rate (TPR) and false positive rate (FPR). The similarity among the curves suggests that all models, even without optimisation, are capable of effectively distinguishing between normal and attack traffic in the KDD Cup dataset. This is due to the dataset's well-separated feature patterns and the RF's inherent ensemble structure, which enhances stability and prevents overfitting.

The hybrid ABC-HBA model exhibits ROC curves nearly identical to those of the baseline model (RF + Pseudo labelling), yet with a slightly sharper curvature near the top-left corner, indicating a more stable and confident classification boundary. The corresponding AUC values are: 1.0000 (DoS), 0.9993 (Probe), 0.99998 (R2L), 0.99993 (U2R), and 0.99995 (Normal). This improvement stems from the integration of the HBA, which enhances the exploitation phase of ABC by refining the best feature subsets through adaptive search pressure. As a result, the hybrid model achieves minor yet consistent performance gains, notably for U2R and R2L, which are the most infrequent and complex attack categories. The combination of ABC's exploration and HBA's exploitation ensures better convergence and improved detection consistency across iterations.

4.6. Precision–recall analysis

To ensure a more reliable evaluation in the presence of class imbalance, precision–recall (PR) curves Fig. 11 are generated for each class, along with the corresponding Average Precision (AP) scores. The proposed ABC-HBA model achieves AP values of 1.0000 for DoS, 0.9983 for Probe, 0.9908 for R2L, 0.8488 for U2R, and 1.0000 for Normal. In comparison to the baseline (RF + Pseudo labelling) and ABC model, the proposed method demonstrates improved performance, particularly for the minority class U2R, highlighting its effectiveness in imbalanced intrusion detection scenarios.

4.7. Feature selection stability analysis

To evaluate the robustness of the proposed ABC-HBA framework, five independent runs were conducted using different random seeds while keeping all other parameters fixed. For each run, the selected feature subsets and corresponding F1-scores were recorded.

The stability of the method was assessed in terms of both classification performance and feature selection consistency. Table 2 summarizes the mean and standard deviation of the F1-score and Jaccard similarity across the five runs.

The results indicate that the proposed method achieves highly consistent performance, with a mean F1-score of 0.99946 and a very low standard deviation of 0.00012. Furthermore, the mean Jaccard similarity of 0.651 (standard deviation = 0.042) demonstrates a moderate-to-high level of consistency in the selected feature subsets, despite the stochastic nature of the optimization process. These findings confirm the robustness and stability of the proposed approach.

4.8. External validation on NSL-KDD

To assess the generalisation ability of the developed models, an unseen sample from the NSL-KDD dataset was used for external evaluation. Although both datasets share similar features and attack categories, NSL-KDD provides a more balanced and less redundant data distribution, allowing a realistic examination of the models' adaptability to real-world network traffic scenarios. The validation subset of NSL-KDD used in this study consisted of 34 samples, distributed as follows: DoS = 11, Probe = 5, R2L = 3, U2R = 2, and Normal = 13. Given the extremely small number of samples in the minority classes, the observed accuracy is highly sensitive to even a few misclassification. For example, misclassifying a single U2R sample changes class accuracy by 50%, and overall accuracy by approximately 2.94% per sample. Therefore, the observed 50% validation accuracy does not indicate a failure of the RF model but rather reflects statistical instability caused by the limited sample size and class imbalance.

As summarised in Table 4, the performance of five models, baseline (RF only), baseline (RF + Pseudo labelling), ABC-optimised, HBA-optimised and ABC-HBA model, was assessed using four evaluation metrics: validation accuracy, precision, recall, and F1-score. The results reveal a clear hierarchy in generalisation performance. The baseline (RF + Pseudo labelling) model, despite achieving exceptionally high accuracy during the training phase, recorded a validation accuracy of only 50.00%, with relatively low precision (28.67%), recall (27.27%), and F1-score (22.76%) on the unseen NSL-KDD sample. This significant drop demonstrates that the base model is prone to overfitting, capturing dataset-specific regularities instead of generalizable attack patterns. The ABC-optimised model exhibited moderate improvement, achieving 70.59% validation accuracy with corresponding gains in precision (66.67%), recall (60.89%), and F1-score (57.70%). The enhancement indicates that the ABC algorithm contributed to better exploration of the search space and produced a more effective feature subset than the base model. However, its limited local refinement capability led to suboptimal exploitation, preventing the model from fully adapting to the novel data characteristics.

The proposed ABC-HBA hybrid model outperformed both counterparts across all evaluation metrics, achieving 94.12% validation accuracy, 95.24% precision, 86.67% recall, and 87.44% F1-score. These substantial improvements confirm that the hybridisation with the HBA effectively enhances the exploitation process, refining the feature subset selection and classifier optimisation. The balanced exploration–exploitation mechanism allows the hybrid model to capture underlying patterns that remain consistent across related datasets, thereby reducing sensitivity to data distribution shifts. As a result, the ABC-HBA model demonstrates superior robustness and adaptability, making it better suited for real-world intrusion detection scenarios where data variability and unseen attack patterns are common.

4.9. Class-wise validation performance

The results, presented in Table 5, highlight substantial performance differences between the baseline (RF only), baseline (RF + Pseudo labelling), ABC-optimised, HBA-optimised and the proposed hybrid ABC-HBA models. These differences are particularly evident in the detection of rare and low-frequency attack types, which are often the most challenging to identify in IDS.

The baseline model, although achieving perfect precision for frequent classes such as DoS (100%), failed to maintain balanced detection performance across all categories. Its recall for DoS dropped to only 36%, and it completely failed to detect Probe, R2L, and U2R attacks, producing 0% recall and F1-scores for those categories. These results indicate severe overfitting to the dominant patterns in the training data, with weak generalisation to unseen samples. The model's strong bias toward frequent classes suggests that it primarily learned dataset-specific patterns rather than underlying discriminative relationships that generalise across data distributions.

The ABC-optimised model introduced noticeable improvements, particularly for the Probe and R2L categories, where the F1-scores increased to 50% and 80%, respectively. It also achieved modest gains in DoS recall (45%) and in the Normal class (96% F1-score). These improvements reflect the enhanced exploration capability of the ABC algorithm, which helps discover more relevant feature subsets. However, the model still

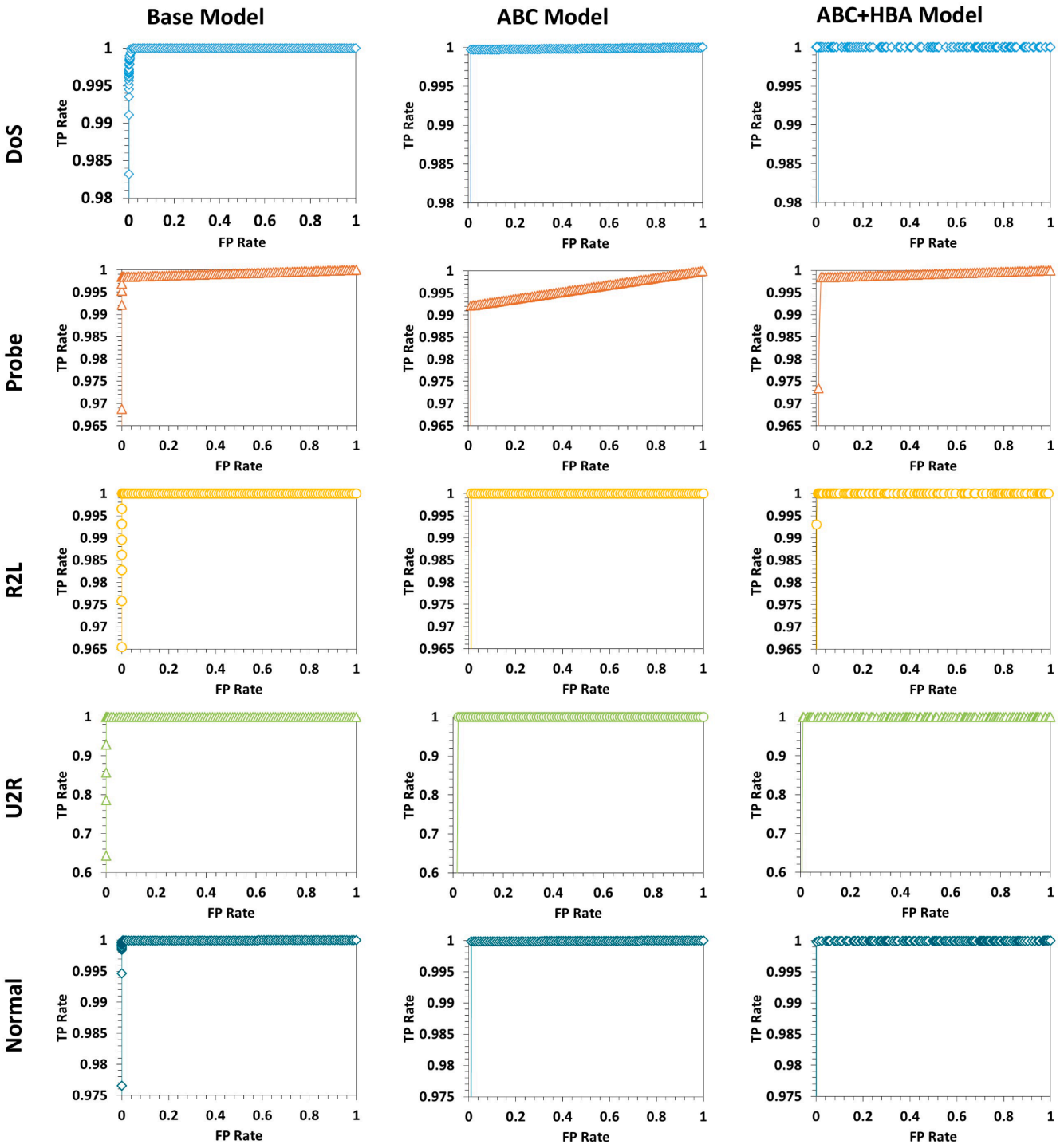

Fig. 10. ROC curves of the Base, ABC, and ABC-HBA models for attack and normal traffic classes.

struggled with consistent detection of rare attack types such as U2R (0% across all metrics). This limitation stems from the restricted exploitation and local refinement capability of the ABC algorithm, which prevented it from converging toward optimal decision boundaries for sparse or minority classes. The ABC model thus partially mitigates overfitting but lacks robustness in handling class imbalance and rare event detection.

The proposed hybrid ABC-HBA model achieved significant improvements across all classes, demonstrating strong generalisation capability and superior adaptability to unseen data. It recorded 100% precision, recall, and F1-score for DoS and U2R, 91% F1-score for Probe, and notably higher F1-scores for R2L (50%) and-Normal (96%). The successful detection of U2R and R2L attacks, both rare and infrequent in the dataset, highlights the model's ability to generalise effectively beyond the training distribution. This improvement is primarily due to the strong exploitation mechanism of HBA, which enhances the refinement of local optima identified by the ABC phase. By combining the global search strength of ABC with the adaptive local intensification of HBA, the hybrid model achieves a balanced exploration–exploitation trade-off, enabling it to capture subtle behavioural patterns associated with minority attack classes.

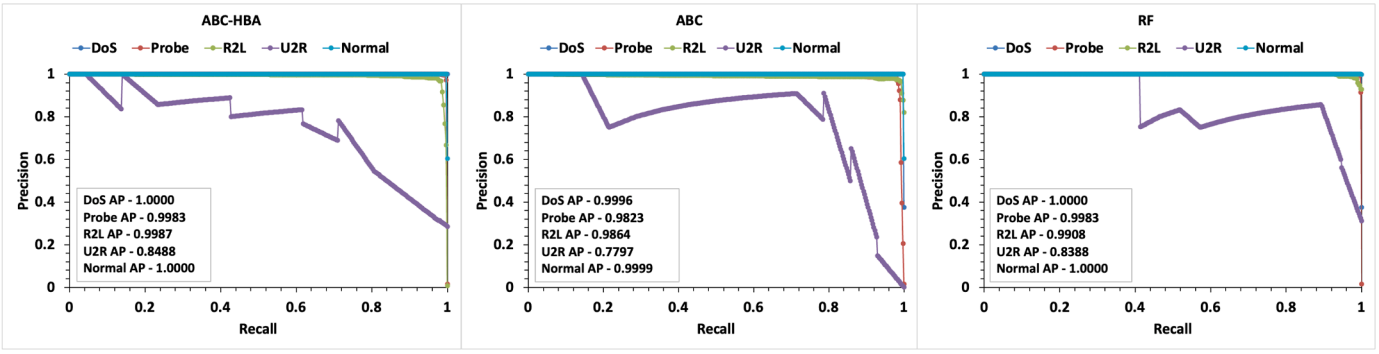

Fig. 11. Precision-recall curves.

Table 3
Detailed class-wise performance comparison of different models.

Class	Metric	Baseline RF Model	Base + Pseudo labeling	ABC + RF Model	HBA + RF Model	ABC-HBA Model
DoS	Precision	99.99%	99.99%	99.97%	99.99%	**100.00%**
	Recall	99.53%	**99.99%**	99.91%	**99.99%**	**99.99%**
	F1-Score	**99.99%**	**99.99%**	99.94%	**99.99%**	**99.99%**
Probe	Precision	**99.53%**	**99.53%**	97.22%	98.92%	99.07%
	Recall	99.22%	99.53%	98.28%	**99.84%**	**99.84%**
	F1-Score	99.37%	**99.53%**	97.75%	99.38%	99.46%
R2L	Precision	98.60%	97.92%	95.35%	95.59%	**98.95%**
	Recall	97.23%	97.92%	**99.31%**	97.58%	98.96%
	F1-Score	97.91%	97.92%	97.29%	96.58%	**98.95%**
U2R	Precision	80.00%	68.75%	75.00%	84.62%	**86.76%**
	Recall	57.14%	78.57%	85.71%	78.57%	**92.86%**
	F1-Score	66.67%	73.33%	80.00%	81.48%	**89.70%**
Normal	Precision	99.93%	**99.96%**	**99.96%**	99.95%	**99.96%**
	Recall	**99.97%**	99.95%	99.91%	99.91%	99.96%
	F1-Score	99.95%	99.95%	99.94%	99.93%	**99.96%**

Table 4
Comparison of model generalization performance of the Base(RF), Base(RF + Pseudo Labeling) ABC, HBA and ABC-HBA models.

Method	Accuracy	Precision	Recall	F1-score
RF (Baseline)	74.00%	46.00%	46.00%	44.00%
RF + Pseudo Labeling	50.00%	28.67%	27.27%	22.76%
RF + ABC	70.59%	66.67%	60.89%	57.70%
RF + HBA	71.00%	64.00%	56.00%	57.00%
RF + ABC–HBA	**94.12%**	**95.24%**	**86.67%**	**87.44%**

Importantly, although the training-phase performance of all models appeared relatively similar, the validation results reveal a considerable divergence. This observation indicates that the ABC-HBA model does not merely memorise training distributions but instead learns more generalised and transferable decision boundaries. The consistent detection across both common and rare classes of the hybrid model confirms its potential for robust, real-world deployment in adaptive and data-diverse environments.

4.10. External validation on CIC-IDS2017 dataset

The ABC–HBA approach was evaluated using the CIC-IDS 2017 dataset, a modern benchmark developed by the Canadian Institute for Cybersecurity. The dataset includes realistic benign traffic and 14 contemporary attack scenarios generated from diverse user behaviors, making it suitable for assessing intrusion detection models.

To enable consistent comparative analysis with the KDD99-based experiments, the CIC-IDS2017 attack categories were mapped into the four standard KDD99 classes based on their functional characteristics. Specifically, benign traffic was mapped to the Normal class. DoS attacks, including DoS Hulk, DDoS, DoS GoldenEye, DoS slowloris, and DoS Slowhttptest, were grouped under the DoS category due to their shared objective of exhausting system resources. Probe attacks were represented by surveillance activities such as PortScan. R2L attacks included FTP-Patator, SSH-Patator, and web-based attacks such as Web Attack–Brute Force, Web Attack–XSS, and Web Attack–SQL Injection, where an external attacker attempts to gain unauthorized access. U2R attacks were mapped to Infiltration and Heartbleed, as they involve privilege escalation or sensitive information extraction. Additionally, botnet traffic (Bot) was assigned to the DoS category due to its disruptive and service-flooding behaviour. This mapping ensures a consistent semantic alignment between CIC-IDS2017 and KDD99 attack taxonomies.

As shown in Tables 6 and 7, the proposed ABC–HBA framework achieved an overall accuracy of 99.88%, demonstrating strong precision and recall across dominant attack categories. Importantly, despite the severe class imbalance in the dataset, the model successfully identified rare and challenging attack types, including U2R (0.006%), which are typically difficult to detect in existing intrusion detection systems. These results highlight the robustness and generalisation capability of the proposed framework across both legacy and modern benchmark datasets.

Table 5
Class-wise Performance of RF, ABC, HBA, and ABC-HBA models on Validation Data.

Class	Metric	Baseline RF Model	Base + Pseudo labeling	ABC + RF Model	HBA + RF Model	ABC-HBA Model
DoS	Precision	100%	**100%**	**100%**	**100%**	**100%**
	Recall	91%	36%	45%	55%	**100%**
	F1-Score	95%	53%	62%	71%	**100%**
Probe	Precision	67%	0%	33%	60%	**83%**
	Recall	40%	0%	**100%**	60%	**100%**
	F1-Score	50%	0%	50%	60%	**91%**
R2L	Precision	0%	0%	**100%**	**100%**	**100%**
	Recall	0%	0%	67%	67%	33%
	F1-Score	0%	0%	**80%**	**80%**	50%
U2R	Precision	0%	0%	0%	0%	**100%**
	Recall	0%	0%	0%	0%	**100%**
	F1-Score	0%	0%	0%	0%	**100%**
Normal	Precision	62%	43%	**100%**	62%	93%
	Recall	100%	**100%**	92%	100%	**100%**
	F1-Score	76%	60%	**96%**	76%	**96%**

Table 6
Overall Performance Metrics of the ABC–HBA Intrusion Detection Model on the CIC-IDS 2017 Dataset.

Accuracy	Precision	Recall	F1-Score
99.88%	97.82%	98.78%	98.27%

Table 7
Class-wise Performance Metrics of the ABC–HBA Intrusion Detection Model on the CIC-IDS 2017 Dataset.

Attack	Precision	Recall	F1-Score
DoS	99.78%	99.59%	99.68%
Probe	89.68%	96.58%	93.00%
R2L	99.70%	97.79%	98.73%
U2R	100.00%	100.00%	100.00%
Normal	99.92%	99.95%	99.93%

5. Conclusion

With the increasing complexity and scale of modern network infrastructures, the demand for a generalised, adaptive, and highly accurate IDS has become paramount. Despite the availability of numerous IDS frameworks, challenges such as low detection accuracy, high false-positive rates, and limited generalisation to unseen network behaviour remain inadequately addressed. This study proposed a hybrid ABC–HBA model to enhance intrusion detection performance, particularly in imbalanced and heterogeneous network traffic with the primary objective of increasing overall detection accuracy while significantly improving detection rates for extremely rare attack types. The proposed method integrates meta-heuristic feature optimisation and a self-training phase using simulated unlabelled data to improve the discrimination capability of the model across both frequent and rare attack categories. Initially, feature selection was performed using the ABC algorithm. However, recognising ABC's limited local exploitation capacity, it was integrated with the HBA to achieve better local refinement and an improved exploration–exploitation balance. The optimised feature set was subsequently utilised to train an RF classifier, while pseudo-labelling was employed to incorporate simulated unlabelled data for self-training based model enhancement.

Experimental results demonstrate that the ABC–HBA hybrid model substantially outperforms both the baseline RF and ABC-optimised models. The proposed model achieved 96.93% precision, 98.16% recall, 97.54% F1-score, and 99.95% overall accuracy, reflecting its capability in achieving robust classification and minimising false alarms. More importantly, when evaluated on an external validation subset derived from the NSL-KDD dataset and CIC-IDS-2017 dataset, the proposed model maintained consistent performance, indicating enhanced generalisation and adaptability to real-world traffic variations. The ABC–HBA hybrid model presents a computationally efficient, feature-optimised, and generalizable IDS framework capable of accurately identifying both frequent and infrequent attack types. Overall, the findings of this study demonstrate that strategically combining complementary swarm intelligence algorithms enables the development of a more resilient and effective intrusion detection mechanism. The ABC–HBA hybrid model therefore represents a promising direction for future IDS development, as it can effectively overcome the individual limitations of each algorithm, resulting in improved feature optimisation and intrusion detection performance. As future work, the proposed ABC-HBA framework can be extended for real-time intrusion detection using streaming network data to evaluate its scalability and adaptability in dynamic environments, while integrating representation learning and temporal behaviour modelling to capture latent network patterns for improved detection of evolving and zero-day attacks (Yu et al., 2020, 2025).

CRediT authorship contribution statement

Sasangi Harischandra: Software, Data curation, Investigation, Validation, Visualization, Writing – original draft; **U.U. Samantha Rajapaksha:** Supervision, Writing – review & editing; **Bhagya Nathali Silva:** Supervision, Writing – review & editing; **Chandimal Jayawardena:** Conceptualization, Methodology, Supervision, Project administration, Writing – review & editing.

Data availability

Data will be made available on request.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Sasangi Harischandra reports administrative support, equipment, drugs, or supplies, and writing assistance were provided by Sri Lanka Institute of Information Technology. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Supplementary material

Supplementary material associated with this article can be found in the online version at 10.1016/j.eswa.2026.132661.

References

Abraham, J. A., & Bindu, V. R. (2021). Intrusion detection and prevention in networks using machine learning and deep learning approaches: A review. In *2021 International conference on advancements in electrical, electronics, communication, computing and automation (ICAECA)* (pp. 1–4). IEEE. https://doi.org/10.1109/icaeca52838.2021.9675595

Aghdam, M. H., & Kabiri, P. (2016). Feature selection for intrusion detection system using ant colony optimization. *International Journal of Network Security*, *18*, 420–432. https://api.semanticscholar.org/CorpusID:2573593.

Ahmad, I., Basheri, M., Iqbal, M. J., & Rahim, A. (2018). Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection. *IEEE Access*, *6*, 33789–33795. https://doi.org/10.1109/access.2018.2841987

Al-Mamory, S. O., & Jassim, F. S. (2013). Evaluation of different data mining algorithms with KDD CUP 99 data set. *Journal of Babylon University/Pure and Applied Sciences*, *21*(8), 2663–2681.

Alatas, B. (2010). Chaotic bee colony algorithms for global numerical optimization. *Expert Systems with Applications*, *37*(8), 5682–5687. https://doi.org/10.1016/j.eswa.2010.02.042

Alomari, O., & Othman, Z. A. (2012). Bees algorithm for feature selection in network anomaly detection. *Journal of Applied Sciences Research*, *8*(3), 1748–1756.

Alsaleem, M. Y. A. (2025). A swarm- optimized hybrid approach to feature selection in IDS. *Journal of Al-Qadisiyah for Computer Science and Mathematics*, *17*(4). https://doi.org/10.29304/jqcsm.2025.17.42565

Ambusaidi, M. A., He, X., Nanda, P., & Tan, Z. (2016). Building an intrusion detection system using a filter-based feature selection algorithm. *IEEE Transactions on Computers*, *65*(10), 2986–2998. https://doi.org/10.1109/tc.2016.2519914

Amini, M.-R., Feofanov, V., Pauletto, L., Hadjadj, L., Devijver, Ã., & Maximov, Y. (2025). Self-training: A survey. *Neurocomputing*, *616*, 128904. https://doi.org/10.1016/j.neucom.2024.128904

Amiri, F., Rezaei Yousefi, M., Lucas, C., Shakery, A., & Yazdani, N. (2011). Mutual information-based feature selection for intrusion detection systems. *Journal of Network and Computer Applications*, *34*(4), 1184–1199. https://doi.org/10.1016/j.jnca.2011.01.002

Araujo, N., de Oliveira, R., Ferreira, E., Shinoda, A. A., & Bhargava, B. (2010). Identifying important characteristics in the KDD99 intrusion detection dataset by feature selection using a hybrid approach. In *2010 17th international conference on telecommunications* (pp. 552–558). IEEE. https://doi.org/10.1109/ictel.2010.5478852

Aung, Y. Y., & Min, M. M. (2017). An analysis of random forest algorithm based network intrusion detection system. In *2017 18th IEEE/ACIS international conference on software engineering, artificial intelligence, networking and parallel/distributed computing (SNPD)* (p. 127–132). IEEE. https://doi.org/10.1109/snpd.2017.8022711

Balasaraswathi, V. R., Sugumaran, M., & Hamid, Y. (2017). Feature selection techniques for intrusion detection using non-bio-inspired and bio-inspired optimization algorithms. *Journal of Communications and Information Networks*, *2*(4), 107–119. https://doi.org/10.1007/s41650-017-0033-7

Battiti, R. (1994). Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, *5*(4), 537–550. https://doi.org/10.1109/72.298224

Bhati, B. S., Chugh, G., Al-Turjman, F., & Bhati, N. S. (2020). An improved ensemble based intrusion detection technique using < scp > XGBoost < /scp > . *Transactions on Emerging Telecommunications Technologies*, *32*(6). https://doi.org/10.1002/ett.4076

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/a:1010933404324

Chandrashekhar, A. M., & Raghuveer, K. (2012). Performance evaluation of data clustering techniques using KDD cup-99 intrusion detection data set. *International Journal of Information and Network Security (IJINS)*, *1*(4). https://doi.org/10.11591/ijins.v1i4.821

Choudhary, S., & Kesswani, N. (2020). Analysis of KDD-cup'99, NSL-KDD and UNSW-NB15 datasets using deep learning in iot. *Procedia Computer Science*, *167*, 1561–1573. https://doi.org/10.1016/j.procs.2020.03.367

Devi Priya, V. S., Sethuraman, S. C., & Khan, M. K. (2025). Blockchain-based deep learning models for intrusion detection in industrial control systems: Frameworks and open issues. *Journal of Network and Computer Applications*, *243*, 104286. https://doi.org/10.1016/j.jnca.2025.104286

Djellali, H., Djebbar, A., Zine, N. G., & Azizi, N. (2018). Hybrid artificial bees colony and particle swarm on feature selection. In *Computational Intelligence and Its Applications* , pp. 93–105). Springer International Publishing. https://doi.org/10.1007/978-3-319-89743-1_9

Eesa, A. S., Orman, Z., & Brifcani, A. M. A. (2015). A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems. *Expert Systems with Applications*, *42*(5), 2670–2679. https://doi.org/10.1016/j.eswa.2014.11.009

Farnaaz, N., & Jabbar, M. A. (2016). Random forest modeling for network intrusion detection system. *Procedia Computer Science*, *89*, 213–217. https://doi.org/10.1016/j.procs.2016.06.047

Farsi, A. A., Khan, A., & Bait-Suwailam, M. M. (2024). A practical evaluation of intrusion detection in iot networks using random forest and network intrusion detection dataset. In *2024 2nd international conference on computing and data analytics (ICCDA)* (pp. 1–6). IEEE. https://doi.org/10.1109/iccda64887.2024.10867402

Gao, M., Tian, J., & Xia, M. (2009). Intrusion detection method based on classify support vector machine. In *2009 second international conference on intelligent computation technology and automation* (pp. 391–394). IEEE. https://doi.org/10.1109/icicta.2009.330

Ghanem, W. A. H. M., Ghaleb, S. A. A., Jantan, A., Nasser, A. B., Saleh, S. A. M., Ngah, A., Alhadi, A. C., Arshad, H., Saad, A.-M. H. Y., Omolara, A. E., El-Ebiary, Y. A. B., & Abiodun, O. I. (2022). Cyber intrusion detection system based on a multiobjective binary bat algorithm for feature selection and enhanced bat algorithm for parameter optimization in neural networks. *IEEE Access*, *10*, 76318–76339. https://doi.org/10.1109/access.2022.3192472

Guan, Z., Li, J., Wu, L., Zhang, Y., Wu, J., & Du, X. (2017). Achieving efficient and secure data acquisition for cloud-supported internet of things in smart grid. *IEEE Internet of Things Journal*, *4*(6), 1934–1944. https://doi.org/10.1109/jiot.2017.2690522

Hasan, M. A. M., Nasser, M., Ahmad, S., & Molla, K. I. (2016). Feature selection for intrusion detection using random forest. *Journal of Information Security*, *07*(03), 129–140. https://doi.org/10.4236/jis.2016.73009

Hashim, F. A., Houssein, E. H., Hussain, K., Mabrouk, M. S., & Al-Atabany, W. (2022). Honey badger algorithm: New metaheuristic algorithm for solving optimization problems. *Mathematics and Computers in Simulation*, *192*, 84–110. https://doi.org/10.1016/j.matcom.2021.08.013

Hassan, I. H., Abdullahi, M., Isuwa, J., Yusuf, S. A., & Aliyu, I. T. (2024). A comprehensive survey of honey badger optimization algorithm and meta-analysis of its variants and applications. *Franklin Open*, *8*, 100141. https://doi.org/10.1016/j.fraope.2024.100141

Hernandez-Ramos, J. L., Karopoulos, G., Chatzoglou, E., Kouliaridis, V., Marmol, E., Gonzalez-Vidal, A., & Kambourakis, G. (2025). Intrusion detection based on federated learning: A systematic review. *ACM Computing Surveys*, *57*(12), 1–65. https://doi.org/10.1145/3731596

Hoque, N., Bhattacharyya, D. K., & Kalita, J. K. (2014). Mifs-nd: A mutual information-based feature selection method. *Expert Systems with Applications*, *41*(14), 6371–6385. https://doi.org/10.1016/j.eswa.2014.04.019

Ibrahim, A. O., Elfadel, E. M. E., Hashem, I. A. T., Syed, H. J., Ismail, M. A., Osman, A. H., & Ahmed, A. (2025). The artificial bee colony algorithm: A comprehensive survey of variants, modifications, applications, developments, and opportunities. *Archives of Computational Methods in Engineering*, *32*(6), 3499–3533. https://doi.org/10.1007/s11831-025-10269-w

Iftikhar, N., Rehman, M. U., Shah, M. A., Alenazi, M. J. F., & Ali, J. (2025). Intrusion detection in NSL-KDD dataset using hybrid self-organizing map model. *Computer Modeling in Engineering & Sciences*, *143*(1), 639–671. https://doi.org/10.32604/cmes.2025.062788

Jose, J., & Jose, D. V. (2023). Deep learning algorithms for intrusion detection systems in internet of things using CIC-IDS 2017 dataset. *International Journal of Electrical and Computer Engineering (IJECE)*, *13*(1), 1134–1141. https://doi.org/10.11591/ijece.v13i1.pp1134-1141

Kamarudin, M. H., Maple, C., Watson, T., & Safa, N. S. (2017). A logitboost-based algorithm for detecting known and unknown web attacks. *IEEE Access*, *5*, 26190–26200. https://doi.org/10.1109/access.2017.2766844

Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization. Technical Report.

Karaboga, D. (2010). Artificial bee colony algorithm. *Scholarpedia*, *5*(3), 6915. https://doi.org/10.4249/scholarpedia.6915

Karaboga, D., & Akay, B. (2011). A modified artificial bee colony (ABC) algorithm for constrained optimization problems. *Applied Soft Computing*, *11*(3), 3021–3031. https://doi.org/10.1016/j.asoc.2010.12.001

Karaboga, D., & Basturk, B. (2008). On the performance of artificial bee colony (ABC) algorithm. *Applied Soft Computing*, *8*(1), 687–697. https://doi.org/10.1016/j.asoc.2007.05.007

Kathiresan, V., Karthik, S., Divya, P., & Rajan, D. P. (2022). A comparative study of diverse intrusion detection methods using machine learning techniques. In *2022 international conference on computer communication and informatics (ICCCI)* (pp. 1–6). IEEE. https://doi.org/10.1109/iccci54379.2022.9740744

Kausar, N., Belhaouari Samir, B., Abdullah, A., Ahmad, I., & Hussain, M. (2011). A review of classification approaches using support vector machine in intrusion detection. In *Informatics Engineering and Information Science* , pp. 24–34). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-25462-8_3

Kauser, A.P., & Senthil, K.N. (2018). A comprehensive review of nature-inspired algorithms for feature selection. In *Handbook of Research on Modeling, Analysis, and Application of Nature-Inspired Metaheuristic Algorithms* , pp. 331–345). IGI Global. https://doi.org/10.4018/978-1-5225-2857-9.ch016

Keerthi Vasan, K., & Surendiran, B. (2016). Dimensionality reduction using principal component analysis for network intrusion detection. *Perspectives in Science*, *8*, 510–512. https://doi.org/10.1016/j.pisc.2016.05.010

Lazar, C., Taminau, J., Meganck, S., Steenhoff, D., Coletta, A., Molter, C., de Schaetzen, V., Duque, R., Bersini, H., & Nowe, A. (2012). A survey on filter techniques for feature selection in gene expression microarray analysis. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *9*(4), 1106–1119. https://doi.org/10.1109/tcbb.2012.33

Lee, K., Joo, J., Yang, J., & Honavar, V. (2006). Experimental comparison of feature subset selection using GA and ACO algorithm. In *Advanced Data Mining and Applications*, pp. 465–472). Springer Berlin Heidelberg. https://doi.org/10.1007/11811305_51

Li, J.-h. (2018). Cyber security meets artificial intelligence: a survey. *Frontiers of Information Technology & Electronic Engineering*, *19*(12), 1462–1474. https://doi.org/10.1631/fitee.1800573

Liao, Y., & Vemuri, V. R. (2002). Use of k-nearest neighbor classifier for intrusion detection. *Computers & Security*, *21*(5), 439–448. https://doi.org/10.1016/s0167-4048(02)00514-x

Louvieris, P., Clewley, N., & Liu, X. (2013). Effects-based feature identification for network intrusion detection. *Neurocomputing*, *121*, 265–273. https://doi.org/10.1016/j.neucom.2013.04.038

Mukherjee, S., & Sharma, N. (2012). Intrusion detection using naive bayes classifier with feature reduction. *Procedia Technology*, *4*, 119–128. https://doi.org/10.1016/j.protcy.2012.05.017

Mukkamala, S., Janoski, G., & Sung, A. (2002). Intrusion detection using neural networks and support vector machines. In *Proceedings of the 2002 international joint conference on neural networks. IJCNN'02 (cat. no.02CH37290)* IJCNN-02 (pp. 1702–1707). IEEE. https://doi.org/10.1109/ijcnn.2002.1007774

Nimbalkar, P., & Kshirsagar, D. (2021). Feature selection for intrusion detection system in internet-of-things (iot). *ICT Express*, *7*(2), 177–181. https://doi.org/10.1016/j.icte.2021.04.012

Ogundokun, R. O., Awotunde, J. B., Sadiku, P., Adeniyi, E. A., Abiodun, M., & Dauda, O. I. (2021). An enhanced intrusion detection system using particle swarm optimization feature extraction technique. *Procedia Computer Science*, *193*, 504–512. https://doi.org/10.1016/j.procs.2021.10.052

Padmaja, D. L., & Vishnuvardhan, B. (2016). Comparative study of feature subset selection methods for dimensionality reduction on scientific data. In *2016 IEEE 6th international conference on advanced computing (IACC)* (p. 31–34). IEEE. https://doi.org/10.1109/iacc.2016.16

Pandithurai, O., Venkataiah, C., Tiwari, S., & Ramanjaneyulu, N. (2024). Ddos attack prediction using a honey badger optimization algorithm based feature selection and bi-lstm in cloud environment. *Expert Systems with Applications*, *241*, 122544. https://doi.org/10.1016/j.eswa.2023.122544

Pathak, A., & Pathak, S. (2020). Study on decision tree and KNN algorithm for intrusion detection system. *International Journal of Engineering Research and*, *V9*(05). https://doi.org/10.17577/ijertv9is050303

Pawana, I. W. A. J., Abella, V., Lastre, J. K., Ko, Y., & You, I. (2025). Enhancing roaming security in cloud-native 5g core network through deep learning-based intrusion detection system. *Computer Modeling in Engineering & Sciences*, *145*(2), 2733–2760. https://doi.org/10.32604/cmes.2025.072611

Protić, D. (2018). Review of KDD cup '99, NSL-KDD and kyoto 2006+ datasets. *Vojnotehnicki glasnik*, *66*(3), 580–596. https://doi.org/10.5937/vojtehg66-16670

Resende, P. A. A., & Drummond, A. C. (2018). A survey of random forest based methods for intrusion detection systems. *ACM Computing Surveys*, *51*(3), 1–36. https://doi.org/10.1145/3178582

Saini, O., & Sharma, S. (2018). A review on dimension reduction techniques in data mining. *Computer Engineering and Intelligent Systems*, *9*(1), 7–14.

Sarasamma, S. T., Zhu, Q. A., & Huff, J. (2005). Hierarchical kohonenen net for anomaly detection in network security. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *35*(2), 302–312. https://doi.org/10.1109/tsmcb.2005.843274

Shah, B., & Trivedi, B. H. (2015). Reducing features of KDD CUP 1999 dataset for anomaly detection using back propagation neural network. In *2015 fifth international conference on advanced computing & communication technologies* (p. 247–251). IEEE. https://doi.org/10.1109/acct.2015.131

Sharma, K., & Silva, B. N. (2026). Data-centric single teacher guided knowledge distillation for alleviating sub-optimal supervision in image classification. *Applied Soft Computing*, *193*, 114897. https://doi.org/10.1016/j.asoc.2026.114897

Song, J., Zhu, Z., Scully, P., & Price, C. (2014). Modified mutual information-based feature selection for intrusion detection systems in decision tree learning. *Journal of Computers*, *9*(7). https://doi.org/10.4304/jcp.9.7.1542-1546

Songma, S., Chimphlee, W., Maichalernnukul, K., & Sanguansat, P. (2012). Classification via k-means clustering and distance-based outlier detection. In *2012 tenth international conference on ICT and knowledge engineering* (p. 125–128). IEEE. https://doi.org/10.1109/ictke.2012.6408540

Tan, Z., Jamdagni, A., He, X., & Nanda, P. (2010). Network intrusion detection based on LDA for payload feature selection. In *2010 IEEE globecom workshops* (p. 1545–1549). IEEE. https://doi.org/10.1109/glocomw.2010.5700198

Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the KDD CUP 99 data set. In *2009 IEEE symposium on computational intelligence for security and defense applications* (p. 1–6). IEEE. https://doi.org/10.1109/cisda.2009.5356528

UCI Machine Learning Repository (1999). Kdd cup 1999 data. https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

Ullah, F., Ullah, S., Srivastava, G., & Lin, J. C.-W. (2024). Ids-int: Intrusion detection system using transformer-based transfer learning for imbalanced network traffic. *Digital Communications and Networks*, *10*(1), 190–204. https://doi.org/10.1016/j.dcan.2023.03.008

Vapnik, V. N. (2000). The Nature of Statistical Learning Theory. Springer New York. https://doi.org/10.1007/978-1-4757-3264-1

Yang, J., Ye, Z., Yan, L., Gu, W., & Wang, R. (2018). Modified naive bayes algorithm for network intrusion detection based on artificial bee colony algorithm. In *2018 IEEE 4th international symposium on wireless systems within the international conferences on intelligent data acquisition and advanced computing systems (IDAACS-SWS)* (p. 35–40). IEEE. https://doi.org/10.1109/idaacs-sws.2018.8525758

Yang, L., Chen, J., Wang, Z., Wang, W., Jiang, J., Dong, X., & Zhang, W. (2021). Semi-supervised log-based anomaly detection via probabilistic label estimation. In *2021 IEEE/ACM 43rd international conference on software engineering (ICSE)* (p. 1448–1460). IEEE. https://doi.org/10.1109/icse43902.2021.00130

Yang, M.-h., & Wang, R.-c. (2008). Ddos detection based on wavelet kernel support vector machine. *The Journal of China Universities of Posts and Telecommunications*, *15*(3), 59–94. https://doi.org/10.1016/s1005-8885(08)60108-9

Yu, H., Lei, X., Song, Z., Liu, C., & Wang, J. (2020). Supervised network-based fuzzy learning of EEG signals for alzheimer's disease identification. *IEEE Transactions on Fuzzy Systems*, *28*(1), 60–71. https://doi.org/10.1109/tfuzz.2019.2903753

Yu, H., Zeng, F., Liu, D., Wang, J., & Liu, J. (2025). Neural manifold decoder for acupuncture stimulations with representation learning: An acupuncture-brain interface. *IEEE Journal of Biomedical and Health Informatics*, *29*(6), 4147–4160. https://doi.org/10.1109/jbhi.2025.3530922

Zebari, R., Abdulazeez, A., Zeebaree, D., Zebari, D., & Saeed, J. (2020). A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction. *Journal of Applied Science and Technology Trends*, *1*(1), 56–70. https://doi.org/10.38094/jastt1224

Zeng, X. (2024). Unmasking intruders: An in-depth analysis of anomaly detection using the KDD cup 1999 dataset. In *2024 3rd international conference on artificial intelligence and computer information technology (AICIT)* (pp. 1–4). IEEE. https://doi.org/10.1109/aicit62434.2024.10729979

Zhang, H., & Li, J. (2020). A new network intrusion detection based on semi-supervised dimensionality reduction and tri-lightGBM. In *2020 international conference on pervasive artificial intelligence (ICPAI)* (pp. 35–40). IEEE. https://doi.org/10.1109/icpai51961.2020.00014

Zhang, J., & Zulkernine, M. (2006). A hybrid network intrusion detection technique using random forests. In *First international conference on availability, reliability and security (ARES'06)* (pp. 8– 269). IEEE. https://doi.org/10.1109/ares.2006.7

Zhang, J., Zulkernine, M., & Haque, A. (2008). Random-forests-based network intrusion detection systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *38*(5), 649–659. https://doi.org/10.1109/tsmcc.2008.923876

Zhu, X. J. (2005). Semi-supervised learning literature survey. University of Wisconsin–Madison.