Magna Scientia Advanced Research and Reviews
eISSN: 2582-9394
Cross Ref DOI: 10.30574/msarr
Journal homepage: https://msarr.magnascientiapub.com/

(RESEARCH ARTICLE)

Check for updates

ZERO-DAY ATTACK DETECTION MODEL FOR NETWORK INTRUSION USING THE CUCKOO SEARCH ALGORITHM AND MACHINE LEARNING CLASSIFIERS

IBRAHIM SULTAN ABIOLA [1], BAKARE ABOLORE RALIAT [2,*] and ODEYALE KEHINDE MUSILIUDEEN [1]

[1] Al-Hikmah University, Department of Computer Science, Faculty of Computing, Engineering and Technology. Ilorin, Kwara.
[2] University of North Texas, Department of Advanced Data Analytics, Texas, USA.

ORCID Details
IBRAHIM SULTAN ABIOLA: https://orcid.org/0009-0001-3927-3351
BAKARE ABOLORE RALIAT: https://orcid.org/0009-0004-3433-7813
ODEYALE KEHINDE MUSILIUDEEN: https://orcid.org/0009-0004-2462-3752

Magna Scientia Advanced Research and Review, 2026, 18(01), 236–245

Publication history: Received on 03 September 2026; revised on 10 October 2026; accepted on 12 October 2026

Article DOI: https://doi.org/10.30574/msarr.2026.18.1.0200

Copyright © 2026 Author(s) retain the copyright of this article. This article is published under the terms of the Creative Commons Attribution License 4.0

ABSTRACT

Cybersecurity threats, especially zero-day attacks, present a major challenge to modern digital infrastructures because they exploit previously unknown vulnerabilities before security patches are released. Traditional intrusion detection systems (IDSs) often fail to detect these novel threats because they rely on predefined rules and known signatures. This study presents a zero-day attack detection model for network intrusion that combines the Cuckoo Search Algorithm (CSA) for feature optimization with five supervised machine-learning classifiers Artificial Neural Network (ANN), Decision Tree (C4.5), K-Nearest Neighbor (KNN), Naïve Bayes (NB), and Support Vector Machine (SVM). The publicly available TII-SSRC-23 and CIC-DDoS2019 datasets were used to train and evaluate the model, with data acquisition, preprocessing, CSA-based feature selection, classification, and performance evaluation implemented in Python. Model performance was assessed using accuracy, precision, recall, F1-score, kappa statistic, and time taken to build the model. Results show that CSA-based feature selection improved detection accuracy for most classifiers on the TII-SSRC-23 dataset, with ANN consistently achieving the best performance (99.03% accuracy, 0.9903 precision, recall and F1-score) while Naïve Bayes recorded the lowest performance overall. The developed model provides a scalable and efficient framework for enhancing network security against emerging zero-day exploits, and offers a benchmarked basis for future real-time and deep-learning extensions.

Keywords: Zero-Day Attack, Intrusion Detection System (IDS), Machine Learning, Cuckoo Search Algorithm, Cybersecurity.

1 INTRODUCTION

The increasing interconnectedness of digital systems has made network security an issue of growing importance for individuals, organizations and governments alike [1], [2]. As cyber threats continue to evolve in complexity, conventional rule-based intrusion detection systems (IDSs) are finding it increasingly difficult to identify and respond to emerging attacks, particularly zero-day exploits [3]. These attacks take advantage of previously unknown vulnerabilities in software, hardware or firmware before developers have an opportunity to detect and address them. Since zero-day attacks have no existing signatures or known behavioural patterns, they are able to evade traditional signature-based detection mechanisms. Cybercriminals frequently exploit these vulnerabilities to launch distributed denial-of-service (DDoS) attacks, disrupting network operations by overwhelming systems with malicious traffic and preventing legitimate users from accessing services [4]. The challenge is even more pronounced in Internet of Things (IoT) environments, where the vast number of interconnected and diverse devices significantly expands the potential attack surface, making networks more vulnerable to sophisticated cyberattacks [5].

Traditional intrusion detection systems (IDSs) rely heavily on predefined rules and known attack signatures, making them ineffective against previously unseen threats. As a result, they often produce high rates of both false positives and false negatives and are unable to detect novel exploits effectively [4]. To address these limitations, machine learning (ML) has become an increasingly important approach in cybersecurity. By learning complex patterns from large volumes of network traffic, ML models can identify anomalous behavior and detect potential threats, including zero-day attacks that have not been encountered previously [11], [12]. Supervised learning techniques, such as Support Vector Machines, Random Forests and Neural Networks, have demonstrated considerable potential for zero-day attack detection [13]. Nevertheless, many ML-based intrusion detection models continue to face challenges associated with the high dimensionality of network traffic data. The large number of features not only increases computational complexity but can also reduce classification performance [2], [14]. Despite growing interest in ML-based detection methods, relatively few studies have explored the use of metaheuristic feature-selection techniques to reduce data dimensionality while simultaneously improving detection accuracy and computational efficiency.

This study addresses this gap by developing a zero-day attack detection model that combines the Cuckoo Search Algorithm (CSA), a nature-inspired metaheuristic optimization technique, with five supervised machine learning classifiers: Artificial Neural Network (ANN), C4.5, K-Nearest Neighbour (KNN), Naïve Bayes (NB) and Support Vector Machine (SVM). The proposed model is trained and evaluated using two widely recognized benchmark datasets, TII-SSRC-23 and CIC-DDoS2019, to assess its ability to generalize across diverse network traffic environments. The impact of undetected cyberattacks extends well beyond financial losses. Security breaches can undermine organizational reputation, compromise sensitive and intellectual property, and, at the national level, threaten the resilience of critical infrastructure, including energy grids and communication networks [15]. Consequently, the development of an accurate and computationally efficient zero-day attack detection model is important not only for advancing research but also for strengthening real-world cybersecurity practices.

The main contributions of this study are summarised as follows:

The main contributions of this study are as follows:

- A comprehensive review of the literature on zero-day attack detection, machine learning classifiers and metaheuristic optimisation techniques, highlighting existing limitations in feature-selection methods for intrusion detection.
- The development of a zero-day attack detection framework that combines Cuckoo Search Algorithm (CSA)-based feature selection with five supervised machine learning classifiers.
- An experimental evaluation of the proposed framework using two widely adopted benchmark datasets, TII-SSRC-23 and CIC-DDoS2019. The evaluation compares the performance of the classifiers with and without CSA-based feature selection using accuracy, precision, recall, F1-score, Cohen's kappa statistic and model build time.
- A comparative performance analysis to determine the most effective classifier–dataset combination and to examine its potential for practical real-time deployment in network intrusion detection systems.

2 BACKGROUND

2.1 Zero-Day Attacks and Network Security

A zero-day attack exploits a previously unknown vulnerability in software or hardware, leaving the vendor with no opportunity to develop or deploy a security patch before the vulnerability is exploited [5], [16]. As no known attack signature exists at the time of exploitation, conventional intrusion detection systems (IDSs), which depend on matching network traffic against databases of known attack patterns, are inherently incapable of identifying such attacks before

they occur [4]. More broadly, network security refers to the policies, technologies and practices implemented to protect networks, applications, data and computing infrastructure from unauthorised access, misuse, disruption or destruction. It encompasses a wide range of security domains, including perimeter, endpoint, application, behavioural, cloud and data security.

2.2 Machine-Learning Classifiers for Intrusion Detection

Support Vector Machine (SVM) is a supervised machine learning algorithm that identifies an optimal decision boundary by constructing a maximum-margin hyperplane between classes. By projecting data into a higher-dimensional feature space, SVM can effectively model complex, non-linear decision boundaries while minimising classification errors [22]. Artificial Neural Networks (ANNs) are inspired by the structure and functioning of the human brain. They consist of interconnected processing units, or neurons, that learn from labelled training data and, once trained, can generalise their learned patterns to previously unseen instances [23]. Naïve Bayes (NB) is a probabilistic classification algorithm based on Bayes' theorem. It assumes that input features are conditionally independent, an assumption that simplifies computation and enables the classifier to perform efficiently, particularly when irrelevant or redundant features have been removed through feature selection [25]. Decision Tree algorithms, particularly C4.5, classify data by recursively partitioning the feature space according to attribute values, producing an interpretable tree structure that represents the decision-making process. In contrast, K-Nearest Neighbour (KNN) is an instance-based, or "lazy", learning algorithm that assigns a class label based on a majority vote among the k closest training instances. Because KNN makes no assumptions about the underlying data distribution, it is well suited to problems with complex decision boundaries [24].

2.3 Metaheuristic Feature Selection and the Cuckoo Search Algorithm

Metaheuristic optimisation algorithms are population-based search methods that iteratively explore the solution space to identify near-optimal solutions to complex optimisation problems. They achieve this by balancing exploration of new search regions with the exploitation of promising candidate solutions. The Cuckoo Search Algorithm (CSA) is one such metaheuristic, inspired by the brood-parasitic breeding behaviour of certain cuckoo species. In CSA, candidate solutions, represented as "eggs", are generated using Lévy flight-based random walks. During each iteration, high-quality solutions are retained, while a proportion of poorer solutions, analogous to abandoned host nests, are replaced with new candidates. This search strategy helps maintain population diversity and reduces the likelihood of the algorithm becoming trapped in local optima [19]. Previous studies have demonstrated that CSA performs competitively when compared with other well-established metaheuristic algorithms, including Genetic Algorithms (GA), Particle Swarm Optimisation (PSO) and Tabu Search (TS), across a variety of continuous and discrete optimisation problems [20]. In addition, several hybrid variants of CSA have been proposed to further enhance its convergence speed and optimisation performance [21]. In the present study, CSA is employed as a feature-selection technique for network traffic data. By eliminating redundant and non-informative features before classification, the algorithm reduces data dimensionality, lowers computational overhead and has the potential to improve the predictive performance of machine learning classifiers.

3 RELATED WORKS

Larry and Smith [9] proposed an artificial intelligence (AI)-driven framework for zero-day attack detection that analysed network traffic, system logs and historical threat intelligence using supervised, unsupervised and reinforcement learning techniques. Their framework demonstrated the feasibility of predicting zero-day vulnerabilities and recommending appropriate remediation strategies using real-world datasets. However, it did not incorporate a dedicated feature-optimisation stage, which may have limited its computational efficiency and predictive performance. Parmar and Tak [8] developed a hybrid approach that integrated anomaly detection with supervised classification, achieving lower false-positive rates and faster detection times than conventional intrusion detection systems (IDSs). Similarly, Touré et al. [7] proposed a hybrid supervised–unsupervised framework that first clustered network traffic flows before applying an online learning mechanism to validate detected anomalies. Their approach achieved classification accuracies of 98.4% on the IBM dataset and 96.6% on the NSL-KDD dataset. Nevertheless, the framework required continuous model retraining, resulting in considerable computational overhead and limiting its suitability for resource-constrained or real-time environments.

Ibraheem and Tosho [10] compared the performance of Gradient Boosting, Random Forest, Decision Tree and Support Vector Machine (SVM) classifiers for zero-day attack detection using the CSE-CIC-IDS2018 dataset. Their findings identified Random Forest as the most effective classifier; however, the study did not incorporate a metaheuristic feature-selection technique to optimise the input features, leaving a gap that the present study seeks to address. Zhen et al. [6] proposed a hybrid framework that combined autoencoders with Extreme Gradient Boosting (XGBoost) and Random Forest for zero-day attack detection on the CIC-MalMem-2022 dataset. Although the proposed approach achieved near-perfect classification accuracy, its reliance on autoencoder pre-training introduced additional

computational overhead, and its evaluation was limited to a single benchmark dataset. Azarudheen et al. [17] employed dynamic application security testing to identify previously unknown cross-site scripting (XSS) vulnerabilities in WordPress plugins, demonstrating that zero-day threats are not confined to network traffic but also pose significant risks at the application layer. Similarly, Shamshair et al. [3] presented a comprehensive review of artificial intelligence (AI)-based techniques for zero-day attack detection and concluded that, despite the growing body of research in this area, there remained a lack of comprehensive and up-to-date comparative studies examining the effectiveness of feature-selection-enhanced machine learning classifiers.

Table 1: Summary of Related Work on Zero-Day Attack Detection

Author(s) & Year	Focus	Method / Technology	Key Findings	Limitations
Larry & Smith (2025)	AI-augmented zero-day vulnerability prediction	Supervised, unsupervised & reinforcement learning on network/log/threat data	Improved speed, accuracy and proactivity of zero-day detection	No explicit feature-selection / dimensionality-reduction stage
Parmar & Tak (2025)	Hybrid zero-day defense	Anomaly detection combined with supervised classification	Reduced false positives and detection latency versus conventional IDS	Evaluated mainly on synthetic + limited real-world data
Touré et al. (2024)	Zero-day exploits in network flows	Hybrid supervised + unsupervised clustering, online learning	98.4% accuracy (IBM dataset), 96.6% (NSL-KDD)	High computational cost of continual online retraining
Ibraheem & Tosho (2024)	ML defense against zero-day attacks	Gradient boosting, random forest, decision tree, SVM (ensemble)	Random Forest gave the best detection rate on CSE-CIC-IDS2018	No metaheuristic feature optimisation; high ensemble complexity
Zhen et al. (2024)	Zero-day detection with autoencoders	Autoencoder + XGBoost / Random Forest hybrid	Random Forest-AE reached ~100% accuracy on CIC-MalMem-2022	Single-dataset evaluation; autoencoder training overhead
Azarudheen et al. (2024)	Zero-day web-plugin vulnerabilities	Dynamic application security testing (DAST)	Uncovered previously unknown XSS zero-days (3 WordPress plugins)	Application-layer scope only; not a network-traffic IDS
Shamshair et al. (2022)	Comparative AI review	Survey of ML/DL zero-day detection techniques	No prior comprehensive comparison existed across recent AI methods	Review only — no new model proposed

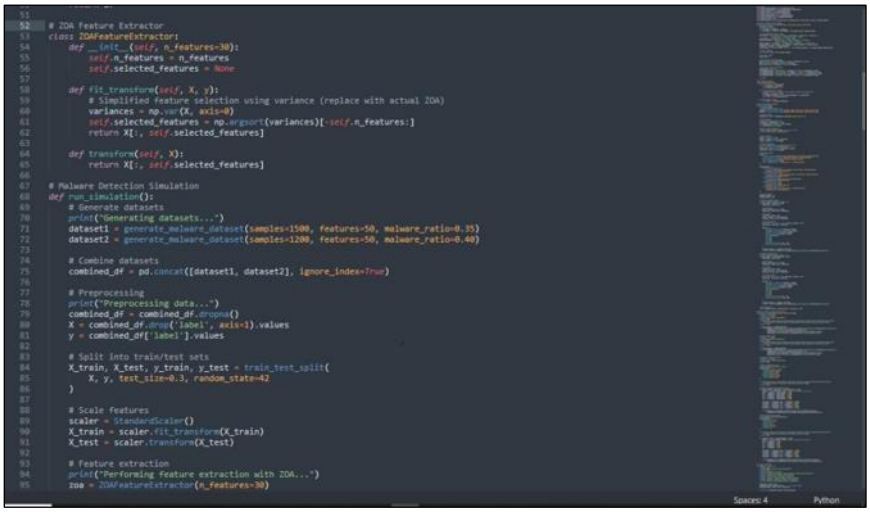
Taken together, these studies demonstrate that machine learning approaches offer significant advantages over traditional signature-based methods for detecting zero-day attacks. However, the existing literature also highlights a persistent gap. Relatively few studies have combined a systematic metaheuristic feature-selection approach with a comprehensive evaluation of multiple classical supervised learning classifiers across different benchmark datasets. To address this limitation, the present study integrates Cuckoo Search Algorithm (CSA)-based feature selection with five supervised classifiers—Artificial Neural Network (ANN), C4.5, K-Nearest Neighbor (KNN), Naïve Bayes (NB) and Support Vector Machine (SVM). The proposed framework is evaluated using two independent benchmark datasets, TII-SSRC-23 and CIC-DDoS2019, to assess its effectiveness and generalizability across diverse network traffic environments.

4 METHODOLOGY

4.1 System Overview

The proposed zero-day attack detection model was implemented using a four-stage pipeline comprising dataset acquisition, data preprocessing, CSA-based feature selection, classification and performance evaluation. To examine the impact of feature selection, two versions of the pipeline were developed and evaluated for each dataset. In the first version, the classifiers were trained using the complete preprocessed feature set without applying feature selection. In the second version, the Cuckoo Search Algorithm (CSA) was employed to identify a reduced set of informative features before classification. Comparing the performance of these two approaches made it possible to isolate and evaluate the contribution of CSA-based feature optimization to the overall effectiveness of the detection model.

Figure 1 illustrates the overall framework.



```
# ZOA Feature Extractor
class ZOAFeatureExtractor:
    def __init__(self, n_features=30):
        self.n_features = n_features
        self.selected_features = None

    def fit_transform(self, X, y):
        # Simplified feature selection using variance (replace with actual ZOA)
        variances = np.var(X, axis=0)
        self.selected_features = np.argsort(variances)[-self.n_features:]
        return X[:, self.selected_features]

    def transform(self, X):
        return X[:, self.selected_features]

# Malware Detection Simulation
def run_simulation():
    # Generate datasets
    print("Generating datasets...")
    dataset1 = generate_malware_dataset(samples=1500, features=50, malware_ratio=0.35)
    dataset2 = generate_malware_dataset(samples=1200, features=50, malware_ratio=0.40)

    # Combine datasets
    combined_df = pd.concat([dataset1, dataset2], ignore_index=True)

    # Preprocessing
    print("Preprocessing data...")
    combined_df = combined_df.dropna()
    X = combined_df.drop('label', axis=1).values
    y = combined_df['label'].values

    # Split into train/test sets
    X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=0.3, random_state=42
    )

    # Scale features
    scaler = StandardScaler()
    X_train = scaler.fit_transform(X_train)
    X_test = scaler.transform(X_test)

    # Feature extraction
    print("Performing feature extraction with ZOA...")
    zoa = ZOAFeatureExtractor(n_features=30)
```

Figure 1: Framework of the Zero-Day Attack Detection Model

4.2 Dataset Acquisition

Two publicly available network traffic datasets were used to develop and evaluate the proposed zero-day attack detection model. The first, TII-SSRC-23, is a comprehensive benchmark dataset designed for intrusion detection research. It provides labelled network traffic in both extracted tabular feature format (CSV) and raw packet capture (PCAP) format, offering flexibility for different stages of data analysis and model development. The second dataset, CIC-DDoS2019, is a widely recognised benchmark that contains an extensive collection of distributed denial-of-service (DDoS) attack scenarios, most of which employ reflection-based amplification techniques. In addition, CIC-DDoS2019 shares the same feature schema as the related CIC-IDS2017, CIC-IDS2018 and CIC-DoS2017 datasets, making it suitable for comparative studies and facilitating the transfer of feature engineering and machine learning models across these benchmark datasets.

4.3 Preprocessing

Before model training, the raw network traffic data were cleaned and preprocessed to improve data quality and ensure consistency across both datasets. Missing values were handled using either imputation or row deletion, depending on the nature and extent of the missing data. In addition, irrelevant and redundant features were removed through correlation-based filtering, resulting in a consistent and well-structured feature matrix for both the baseline pipeline without feature selection and the CSA-based feature-selection pipeline.

4.4 Feature Selection Using the Cuckoo Search Algorithm

The Cuckoo Search Algorithm (CSA) was applied to the preprocessed feature set to identify a subset of informative features that minimised classification error while reducing data dimensionality. The feature-selection process involved six main steps. First, an objective function was defined to evaluate the quality of candidate feature subsets based on classification errors. Next, an initial population of candidate solutions, referred to as host nests, was generated randomly, together with the specification of key algorithm parameters, including the discovery probability (pa), the maximum number of iterations and the Lévy flight step size. During each iteration, a new candidate solution was generated through a Lévy flight random walk, and its fitness was evaluated. The newly generated solution was then compared with a randomly selected host nest and replaced it if it achieved a better fitness value. To maintain diversity

within the population and reduce the risk of premature convergence, a proportion of the least-fit host nests were discarded and replaced with newly generated random solutions. This iterative process continued until the stopping criterion, defined by either the maximum number of iterations or the desired fitness level, was satisfied, with the best-performing feature subset retained as the final solution.

4.5 Classification Algorithms

Five supervised machine learning classifiers were applied to both the CSA-optimized feature set and the original preprocessed feature set without feature selection. These classifiers comprise Artificial Neural Network (ANN), Decision Tree (C4.5), K-Nearest Neighbour (KNN), Naïve Bayes (NB) and Support Vector Machine (SVM), as described in Section 2.2. Each classifier was trained and evaluated independently using both the TII-SSRC-23 and CIC-DDoS2019 datasets under two experimental conditions: with CSA-based feature selection and without feature selection. This experimental design resulted in four evaluation scenarios, enabling a direct assessment of the impact of CSA-based feature optimisation on the performance of each classifier across both datasets.

4.6 Experimental Setup and Evaluation Metrics

The proposed pipeline was implemented in Python within a reproducible notebook environment. NumPy and Pandas were used for data preprocessing and manipulation, Scikit-learn was employed to develop and evaluate the machine learning classifiers, and Matplotlib and Seaborn were used to generate data visualizations. Each dataset was divided into training and testing subsets before model development. The performance of each classifier was then assessed using six evaluation metrics: accuracy, precision, recall, F1-score, Cohen's kappa statistic and model build time. In addition to measuring predictive performance, model build time was included as an important practical metric for assessing the suitability of the proposed approach for near-real-time deployment in resource-constrained network environments.

5 RESULTS

5.1 Zero-Day Detection Performance — TII-SSRC-23 Dataset

Table 2: Detection Performance on TII-SSRC-23 (Without vs. With CSA Feature Selection)

Algorithm	Acc. no-CSA (%)	Acc. CSA (%)	F1 no-CSA	F1 CSA	Time CSA (s)
ANN	98.90	99.03	0.9890	0.9903	10.20
KNN	97.70	97.70	0.9770	0.9770	0.63
C4.5	92.53	92.80	0.9253	0.9280	0.47
NB	89.37	93.40	0.8936	0.9340	0.21
SVM	98.30	98.37	0.9830	0.9837	0.98

On the TII-SSRC-23 dataset, the Artificial Neural Network (ANN) consistently delivered the best classification performance, achieving an accuracy of 98.90% without CSA-based feature selection and improving slightly to 99.03% after feature optimisation. The CSA-enhanced ANN model also recorded the highest overall performance across all evaluation metrics, with an accuracy of 99.03%, precision of 0.9903, recall of 0.9903, an F1-score of 0.9903 and a Cohen's kappa statistic of 0.9807. Among the five classifiers, Naïve Bayes (NB) benefited most from CSA-based feature selection, with its accuracy increasing from 89.37% to 93.40%. This improvement suggests that feature optimization is particularly effective for classifiers whose performance is more susceptible to irrelevant or redundant features. In contrast, the C4.5 Decision Tree classifier recorded the lowest classification accuracy under both experimental conditions. Furthermore, Naïve Bayes achieved the shortest model build time of 0.21 seconds, demonstrating that effective feature selection can substantially reduce computational overhead while maintaining competitive predictive performance.

5.2 Zero-Day Detection Performance — CIC-DDoS2019 Dataset

On the CIC-DDoS2019 dataset, the Artificial Neural Network (ANN) achieved the highest classification accuracy without CSA-based feature selection, recording 97.46%. Following the application of CSA, ANN remained the best-performing classifier, achieving an accuracy of 95.17% and an F1-score of 0.9514. However, the performance of ANN, K-Nearest Neighbour (KNN) and Support Vector Machine (SVM) declined slightly after feature selection on this dataset. By contrast, both Naïve Bayes (NB) and the C4.5 Decision Tree classifier benefited from CSA-based feature optimization. In particular, the accuracy of NB increased from 80.60% to 85.33%, indicating that classifiers with stronger modelling assumptions tend to benefit more from the removal of redundant and less informative features. Conversely, more

complex models, such as ANN and SVM, appear to be less reliant on feature optimization to achieve strong predictive performance. Overall, the TII-SSRC-23 dataset produced consistently better detection results than CIC-DDoS2019 across all evaluation metrics. This difference is likely attributable to variations in network traffic characteristics, attack patterns and class distribution between the two benchmark datasets.

Table 3: Detection Performance on CIC-DDoS2019 (Without vs. With CSA Feature Selection)

Algorithm	Acc. no-CSA (%)	Acc. CSA (%)	F1 no-CSA	F1 CSA	Time CSA (s)
ANN	97.46	95.17	0.9746	0.9514	15.55
KNN	94.40	92.63	0.9431	0.9250	0.52
C4.5	84.47	84.67	0.8443	0.8463	0.42
NB	80.60	85.33	0.8081	0.8483	0.02
SVM	96.50	94.50	0.9648	0.9444	2.28

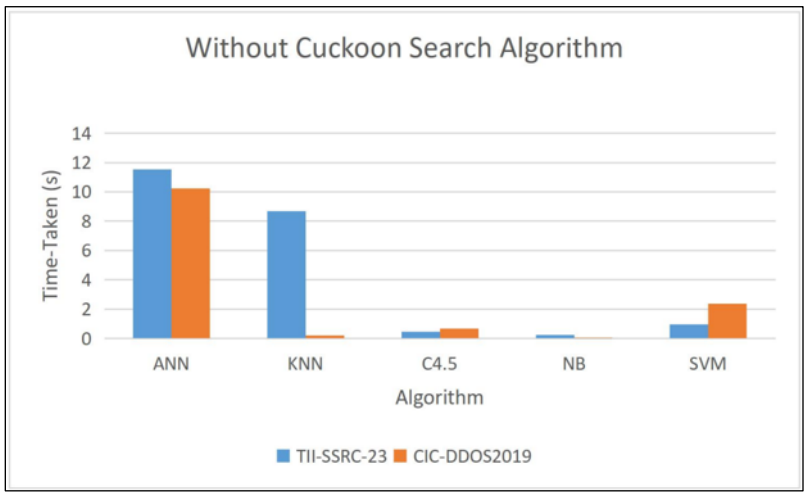


Figure 2: Comparative Accuracy of the Five Classifiers on Both Datasets (With CSA Feature Selection)

6 DISCUSSION

The results demonstrate that integrating Cuckoo Search Algorithm-based feature selection with conventional supervised classifiers is an effective strategy for zero-day attack detection, particularly for classifiers whose performance is sensitive to feature redundancy. Naïve Bayes showed the largest relative improvement from CSA on both datasets, consistent with its conditional-independence assumption, which is violated more severely when redundant; correlated attributes are present. ANN consistently delivered the best absolute performance, but its accuracy and build time were also the most dataset-dependent, improving under CSA on TII-SSRC-23 while declining slightly on CIC-DDoS2019 a reminder that feature selection is not universally beneficial and should be validated per deployment context rather than assumed.

Compared with the related works reviewed in Section 3, the proposed model offers a distinguishing contribution: whereas Ibraheem and Tosho [10] relied purely on ensemble learning without a dedicated feature-optimisation stage, and Touré et al. [7] achieved strong accuracy only at high computational cost from continual online retraining, the CSA-based approach in this study reduced both dimensionality and, for most classifiers, build time, while matching or improving accuracy an important property for near-real-time deployment on network hardware with limited processing capacity. The consistently strong performance of ANN also echoes the findings of Zhen et al. [6], who likewise found that combining a learned feature representation with a strong base classifier yields the best zero-day detection outcomes.

Two practical implications follow. First, organizations adopting ML-based IDSs should not assume that more features are always better: applying a lightweight metaheuristic feature-selection stage, such as CSA, before classification can simultaneously reduce computational overhead and, for simpler classifiers, improve detection accuracy. Second, because relative classifier performance differed between the two datasets, any deployment should be benchmarked on

traffic representative of the target environment rather than assuming that results from one benchmark dataset transfer directly to another.

6.1 THREATS TO VALIDITY

6.1.1 Internal Validity

The Cuckoo Search Algorithm parameters (discovery probability, population size, number of iterations) were set empirically rather than optimized exhaustively, and different parameter choices could alter the selected feature subset and, consequently, the reported accuracy gains. The classifiers were also used with largely default hyperparameters; systematic hyperparameter tuning could change the relative ranking of the five algorithms.

6.1.2 External Validity

The model was evaluated on two benchmark datasets, TII-SSRC-23 and CIC-DDoS2019, which while widely used do not capture the full diversity of real-world network traffic, device types, and attack strategies encountered in production environments. The consistently higher performance observed on TII-SSRC-23 versus CIC-DDoS2019 suggests that generalisation to other traffic distributions should not be assumed without further empirical testing.

6.1.3 Construct Validity

Performance was assessed using accuracy, precision, recall, F1-score, kappa statistic and build time, which are standard but not exhaustive IDS evaluation criteria. These metrics do not directly capture detection of latency under live traffic load or resilience to adversarial evasion, both of which are relevant to zero-day defence in practice and represent an opportunity for more comprehensive future evaluation.

6.1.4 Conclusion Validity

All experiments were conducted offline on static, pre-labelled benchmark data rather than in a live network environment. Conclusions about real-time detection latency and behaviour under adversarial or evolving zero-day conditions are therefore inferred from offline batch performance rather than measured directly and should be validated through online or streaming deployment in future work.

7 CONCLUSION

This study developed and evaluated a zero-day attack detection model that integrates Cuckoo Search Algorithm-based feature selection with five supervised machine-learning classifiers ANN, C4.5, KNN, NB and SVM, on two publicly available network-traffic datasets, TII-SSRC-23 and CIC-DDoS2019. The results show that CSA-based feature selection reduces dimensionality and, for most classifiers, particularly Naïve Bayes improves detection accuracy, while ANN consistently delivered the strongest overall performance, reaching 99.03% accuracy, 0.9903 precision, recall and F1-score, and a 0.9807 kappa statistic on TII-SSRC-23. These findings confirm the practical value of combining metaheuristic feature optimization with conventional supervised learning for zero-day network intrusion detection, offering a scalable and computationally efficient alternative to purely signature-based or unoptimized ML-based IDSs.

Future work should extend the model to real-time streaming environments and IoT or cloud-native network infrastructures, incorporate deep-learning architectures (e.g., CNNs, LSTMs) alongside CSA-based feature selection, explore hybrid metaheuristics that combine CSA with other optimizers such as particle swarm optimization or genetic algorithms, and validate the model on larger, more diverse and continuously updated traffic datasets to better reflect the evolving nature of zero-day threats.

STATEMENTS ON COMPLIANCE WITH ETHICAL STANDARDS

Acknowledgments

The author acknowledges the Department of Computer Science, Faculty of Computing, Engineering and Technology, Ilorin, Kwara State, Nigeria, and the University of North Texas, Department of Advanced Data Analytics, Denton, Texas, USA, for their academic contributions and institutional support.

Disclosure of conflict of interest

The author declares no conflicts of interest regarding this manuscript.

REFERENCES

[1] E. S. M. Kala, "Critical Role of Cyber Security in Global Economy," Open Journal of Safety Science and Technology, vol. 13, pp. 231–248, 2023.

[2] L. Diana and P. Dini, "Networking Security," Computers, vol. 14, pp. 1–44, 2025.

[3] S. U. R. Shamshair, A. Imran, and G. Adeem, "Comparative Evaluation of AI-Based Techniques for Zero-Day Attacks Detection," Electronics, vol. 11, pp. 1–25, 2022.

[4] O. Harazeem, A. Taha, A. Tchakoucht, and Y. Kayode, "A systematic literature review for network intrusion detection system (IDS)," International Journal of Information Security, pp. 1–38, 2023. https://doi.org/10.1007/s10207-023-00682-2

[5] A. Armijos and E. Cuenca, "Zero-day attacks: review of the methods used based on intrusion detection and prevention systems," in Proc. IEEE, 2023, pp. 1–7. https://doi.org/10.1109/C358072.2023.10436218

[6] D. Zhen, L. Yee, et al., "An intrusion detection model to detect zero-day attacks in unseen data using machine learning," PLoS ONE, pp. 1–25, 2024. https://doi.org/10.1371/journal.pone.0308469

[7] A. Touré, Y. Imine, A. Semnont, T. Delot, and A. Gallais, "A framework for detecting zero-day exploits in network flows," Computer Networks, vol. 248, pp. 1–15, 2024. https://doi.org/10.1016/j.comnet.2024.110476

[8] M. Parmar and P. Tak, "Proactive Cyber Defense: Using Machine Learning to Detect and Mitigate Zero-Day Attacks in Real-Time Environments," International Journal of Contemporary Resolution in Multidisciplinary, vol. 4, no. 2, pp. 228–234, 2025.

[9] A. J. Larry and J. Smith, "AI-Augmented Threat Intelligence for Zero-Day Vulnerability Detection," International Journal of Machine Learning Research in Cybersecurity and Artificial Intelligence, vol. 16, no. 1, pp. 1–15, 2025.

[10] I. O. Ibraheem and A. U. Tosho, "Zero Day Attack Vulnerabilities: Mitigation using Machine Learning for Performance Evaluation," Journal of Computers for Society, vol. 5, no. 1, pp. 43–58, 2024. https://doi.org/10.17509/jcs.v5i1.70795

[11] H. O. Bello, A. B. Ige, and M. N. Ameyaw, "Adaptive machine learning models: Concepts for real-time financial fraud prevention in dynamic environments," World Journal of Advanced Engineering Technology and Sciences, vol. 12, no. 2, pp. 22–34, 2024.

[12] L. Ali, K. Thakur, S. Schmeelk, J. Debello, and D. Dragos, "Deep Learning vs. Machine Learning for Intrusion Detection in Computer Networks: A Comparative Study," Applied Sciences, vol. 15, pp. 1–19, 2025.

[13] A. Kumar and J. A. Gutierrez, "Impact of Machine Learning on Intrusion Detection Systems for the Protection of Critical Infrastructure," Information, vol. 16, no. 515, pp. 1–42, 2025.

[14] B. R. Kikissagbe and M. Adda, "Machine Learning-Based Intrusion Detection Methods in IoT Systems: A Comprehensive Review," Electronics, vol. 13, no. 3601, pp. 1–23, 2024.

[15] C. Alcaraz and S. Zeadally, "Critical infrastructure protection: Requirements and challenges for the 21st century," International Journal of Critical Infrastructure Protection, vol. 8, pp. 1–35, 2017. https://doi.org/10.1016/j.ijcip.2014.12.002

[16] R. Zaib and K. Zhou, "Zero-Day Vulnerabilities: Unveiling the Threat Landscape in Network Security," Mesopotamian Journal of Cybersecurity, pp. 57–64, 2022.

[17] M. Azarudheen, M. Mohideen, M. S. Nadeem, J. Hardy, H. Ali, U. U. Tariq, F. Sabrina, M. Waqar, and S. Ahmed, "Behind the Code: Identifying Zero-Day Exploits in WordPress," Future Internet, vol. 16, no. 256, pp. 1–22, 2024.

[18] V. Babaey and R. Fargardi, "Detecting Zero-Day Web Attacks with an Ensemble of LSTM, GRU, and Stacked Autoencoders," Computers, vol. 14, pp. 1–29, 2025.

[19] A. Mohamad, A. M. Zain, and N. Erne Bazin, "Cuckoo Search Algorithm for Optimization Problems – A Literature Review," Applied Mechanics and Materials, vol. 421, pp. 502–506, 2013. https://doi.org/10.4028/www.scientific.net/AMM.421.502

[20] M. Mareli and B. Twala, "An adaptive Cuckoo search algorithm for optimisation," Applied Computing and Informatics, vol. 14, no. 2, pp. 107–115, 2017. https://doi.org/10.1016/j.aci.2017.09.001

[21] M. Shehab, A. T. Khader, and M. Laouchedi, "A hybrid method based on cuckoo search algorithm for global optimization problems," Journal of ICT, vol. 3, no. 3, pp. 469–491, 2018.

[22] D. K. Srivastava and L. Bhambhu, "Data classification using support vector machine," Journal of Theoretical and Applied Information Technology, vol. 12, no. 1, pp. 1–7, 2010.

[23] H. Kukreja and S. Kuldeep, “An Introduction to Artificial Intelligence,” IJARIIE, vol. 1, no. 5, pp. 27–30, 2016.

[24] S. B. Imandoust and M. Bolandraftar, “Application of K-Nearest Neighbor (KNN) Approach for Predicting Economic Events: Theoretical Background,” International Journal of Engineering Research and Applications, vol. 3, no. 5, pp. 605–610, 2013.

[25] A. Oluwaseun and M. S. Chaubey, “Data Mining Classification Techniques,” Global Scientific Journals, vol. 7, no. 4, pp. 79–95, 2019.

[26] H. Sebestyen and D. E. Popescu, “A Literature Review on Security in the Internet of Things: Identifying and Analysing Critical Categories,” Computers, vol. 14, no. 61, pp. 1–45, 2025.

[27] M. Ozkan-Okay, A. A. Yilmaz, E. Akin, A. Aslan, and S. S. Aktug, “A Comprehensive Review of Cyber Security Vulnerabilities,” Electronics, vol. 12, no. 1333, pp. 1–42, 2023.

[28] H. M. Rai and J. Yoo, “The Improved Network Intrusion Detection Techniques Using the Feature Engineering Approach with Boosting Classifiers,” Mathematics, vol. 12, pp. 1–35, 2024.