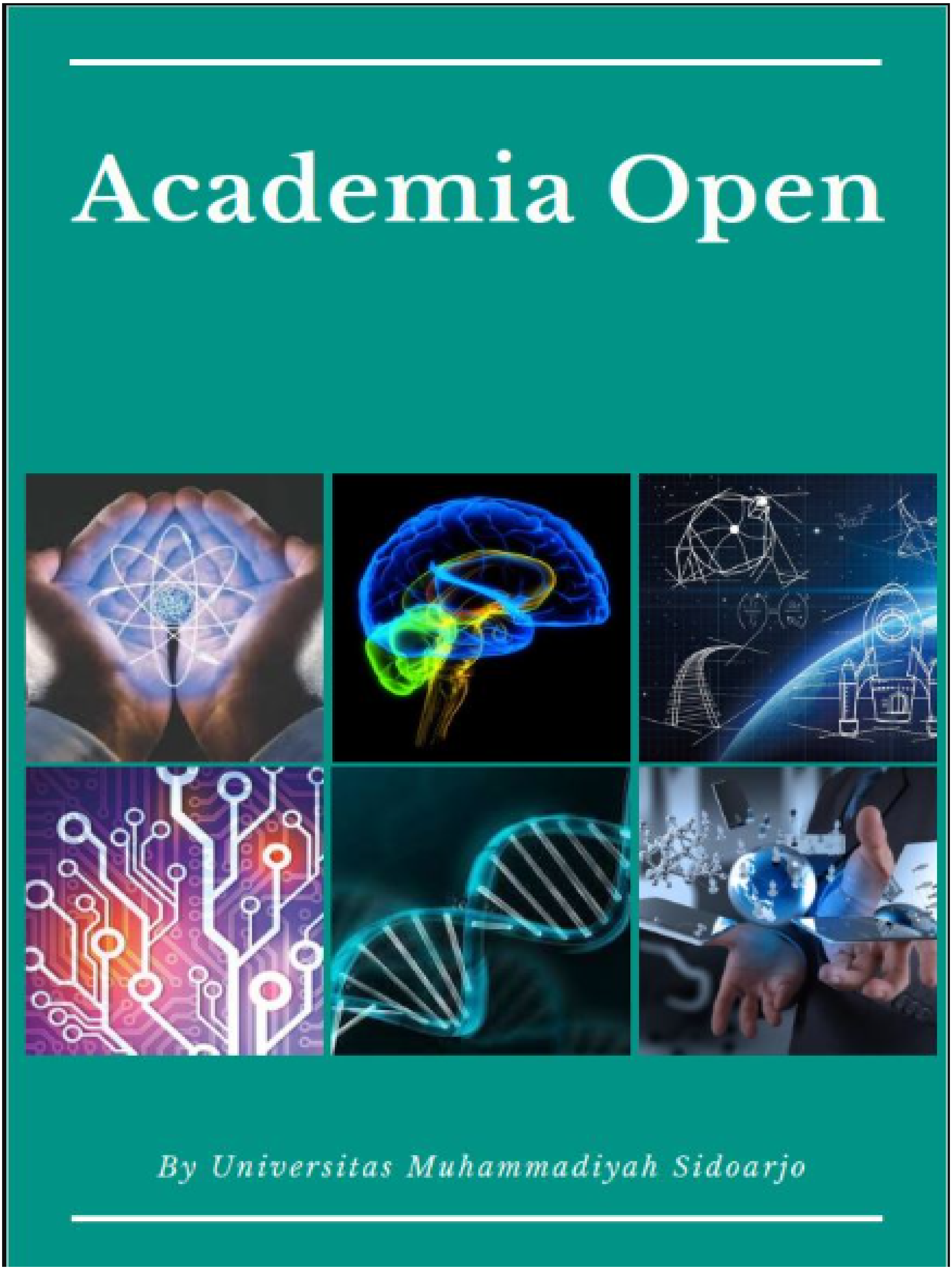
Academia Open

Vol. 10 No. 1 (2025): June
DOI: 10.21070/acopen.10.2025.13019



ISSN 2714-7444 (online), https://acopen.umsida.ac.id, published by Universitas Muhammadiyah Sidoarjo

Academia Open

Vol. 10 No. 1 (2025): June
DOI: 10.21070/acopen.10.2025.13019

Table Of Contents

Originality Statement

The author[s] declare that this article is their own work and to the best of their knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the published of any other published materials, except where due acknowledgement is made in the article. Any contribution made to the research by others, with whom author[s] have work, is explicitly acknowledged in the article.

Conflict of Interest Statement

The author[s] declare that this article was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright Statement

Academia Open

Vol. 10 No. 1 (2025): June
DOI: 10.21070/acopen.10.2025.13019

EDITORIAL TEAM

ISSN 2714-7444 (online), https://acopen.umsida.ac.id, published by Universitas Muhammadiyah Sidoarjo

Academia Open

Vol. 10 No. 1 (2025): June
DOI: 10.21070/acopen.10.2025.13019

Article information

Check this article update (crossmark)

Check this article impact [(*)]

Save this article to Mendeley

[(*)] Time for indexing process is various, depends on indexing database platform

ISSN 2714-7444 (online), https://acopen.umsida.ac.id, published by Universitas Muhammadiyah Sidoarjo

Academia Open

Vol. 10 No. 1 (2025): June
DOI: 10.21070/acopen.10.2025.13019

Enhancing Intrusion Detection in IoT Networks via Enhanced Flower Pollination and Ensemble Classification: Peningkatan Deteksi Intrusi pada Jaringan IoT Melalui Peningkatan Polinasi Bunga dan Klasifikasi Ensemble

Abeer Gabbar Abed, alkhaqanyhb@gmail.com (*)

Islamic Azad University Research Sciences Branch, Indonesia

(*) Corresponding author

Abstract

General Background: The rapid expansion of Internet of Things networks has intensified cybersecurity challenges, particularly distributed denial-of-service attacks targeting interconnected devices. **Specific Background:** Intrusion Detection Systems based on machine learning often suffer from high computational complexity and reduced performance when processing large-scale datasets due to manual feature extraction and dimensionality issues. **Knowledge Gap:** Existing approaches lack efficient hybrid frameworks that integrate advanced optimization algorithms with ensemble classification to address large modern intrusion datasets such as NSL-KDD and UNSW-NB15. **Aims:** This study proposes an Intrusion Detection System integrating the Enhanced Flower Pollination Algorithm for optimal feature selection with an ensemble classification framework combining Random Forest, Decision Tree, and Support Vector Machine. **Results:** Experimental evaluation achieved accuracy rates of 99.67% on NSL-KDD and 99.32% on UNSW-NB15, demonstrating reduced computational complexity and improved detection capability across multiple attack categories. **Novelty:** The study introduces a hybrid EFPA-based feature selection strategy integrated with majority voting ensemble classification for IoT security environments. **Implications:** The proposed framework supports scalable, high-accuracy intrusion detection suitable for real-time IoT deployments and provides a foundation for future integration with advanced security infrastructures.

Keywords: Intrusion Detection System, Internet of Things, Enhanced Flower Pollination Algorithm, Ensemble Classification, Network Security

Key Findings Highlights:

1. Hybrid optimization reduced dimensionality while preserving critical attack indicators.
2. Majority voting integration increased model generalization across attack categories.
3. High detection performance achieved on two benchmark cybersecurity datasets.

Published date: 2026-02-10

ISSN 2714-7444 (online), https://acopen.umsida.ac.id, published by Universitas Muhammadiyah Sidoarjo

Introduction

The extensive adoption of digital technologies and their incorporation into everyday processes has rendered network security progressively more critical in recent years. the Internet of Things (IoT) ecosystem, comprising billions of interconnected gadgets. The vast scale and variety of this ecosystem have heightened the potential of cyberattacks on networks. especially during distributed denial of service (DDoS) assaults, which seek to disrupt services, obtain data, or engage in espionage. A primary mechanism for enhancing intrusion detection systems (IDS) is their capacity to identify intrusions. Cybersecurity. They scrutinize network traffic and monitor for anomalous patterns that may indicate an attack. Notwithstanding significant progress in the application of Machine Learning Numerous conventional systems in this domain exhibit fundamental problems, particularly those arising from machine learning (ML) and deep learning (DL) technologies [1]

1. System complexity resulting from the need to manually extract features.
2. Poor performance with large data sets, such as modern attack databases.
3. Limited generalization capabilities when faced with new or changing attack patterns.

To address these challenges, researchers are turning to optimization algorithms that help select the best features and reduce the data size while maintaining its relevance. In this context, the Enhanced Flower Pollination Algorithm (EFPA) stands out as an effective tool due to its ability to strike a balance between exploration and exploitation in a search space, making it suitable for selecting optimal features. [2]

Accordingly, this study aims to:

1. Improve attack detection accuracy by using EFPA with ensemble classification techniques.
2. Reduce system complexity by selecting optimal features.
3. Test the model on modern and widely used databases such as NSL-KDD and UNSW-NB15 to demonstrate its effectiveness.

In Section 2, the relevant literature is reviewed. In Section 3, the methodology is introduced. In Section 4, the results and discussion are evaluated. Finally, in Section 5, the research concludes with conclusions and recommendations for future studies.

Previous Studies

Recent years have witnessed a significant increase in research efforts to develop intrusion detection systems (IDS) based on machine learning (ML) and deep learning (DL) techniques, given the urgent need to protect networks against complex and advanced cyberattacks. Previous studies can be categorized into several main areas:

1. Traditional Intrusion Detection Systems (IDS)

The first generation of intrusion detection systems used signature-based detection, which involved comparing incoming data to previously identified patterns of attacks. While this approach works well against known assaults, it can't handle emerging threats (also known as zero-day attacks) or ones that employ obfuscation tactics [3], [4]

Later, anomaly-based detection emerged, which relies on identifying normal traffic patterns and then detecting any deviations from them. Although it is more flexible, it suffers from high rates of false alarms [5], [6]

2. Using Machine Learning Techniques

Researchers have resorted to ML methods like Decision Trees, k-Nearest Neighbors (KNN), and Support Vector Machines (SVM) due to the continuous evolution of data. Take Lee and Stolfo (2000) as an example. They proved that classification approaches can make it easier to spot serious assaults. Dealing with big or imbalanced data, however, decreases performance, according to studies [7] , [8]

3. Deep Learning in Intrusion Detection

Researchers have developed RNNs, CNNs, and autoencoders for deep learning.

Kim et al. (2016) used a CNN model to extract features directly from raw data and achieved promising results. [9], [10]

Shone et al. (2018) demonstrated that combining Autoencoders with Random Forests reduces complexity and improves performance.

However, these models require high computational resources and long training times, and they do not handle imbalanced data efficiently. [11]

4. Feature Selection Techniques

Studies have shown that selecting appropriate features plays a pivotal role in improving the performance of IDS systems. For example, Kira & Rendell (1992) used traditional methods such as Information Gain, but these methods are limited when dealing with high-dimensional data. [12]

In recent years, optimization algorithms such as Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) have been introduced. Xue et al. (2015) demonstrated that combining these algorithms with ML classifiers improves accuracy and reduces model complexity. [13] , [14]

5. Ensemble Classification Techniques

One recent trend is to combine more than one classifier to improve performance. For example:

Breiman (2001) used Random Forest, which has proven effective in combating overfitting.

Other studies have shown that combining SVM + Decision Tree + Random Forest increases generalization and reduces errors [15] .

6. Contribution of this research compared to previous studies[16]

Through the literature review, the research gaps can be summarized as follows:

1. Heavy reliance on manually extracted features, which increases complexity.
2. Poor performance on large databases such as UNSW-NB15.
3. The need for hybrid techniques that combine optimization algorithms and ensemble classification.

This research contributes to bridging this gap by:

1. Introducing the Enhanced Flower Pollination Algorithm (EFPA) for selecting optimal features.
2. Combining multiple classifiers (Random Forest, Decision Tree, SVM) via Ensemble Classification.
3. Testing the model on modern, internationally recognized databases (NSL-KDD and UNSW-NB15).

Methodology

1. Research Design

An Intrusion Detection System (IDS) model utilizing the Enhanced Flower Pollination Algorithm (EFPA) for feature selection in an Ensemble Classification (EC) setting with Random Forest (RF), Decision Tree (ID3), and Support Vector Machine (SVM) is the primary goal of this research. The concept seeks to balance computational complexity with detection accuracy in IoT contexts.

2. Datasets Used

A. NSL-KDD Database

- This is an improved version of the traditional KDD'99 database, purged of duplicate records.
- It contains four main attack types: Denial of Service (DoS), User to Root (U2R), Remote to Local (R2L), and Probe.
- It is widely used for evaluating IDS systems.

B. UNSW-NB15 Database

- Recently developed to cover multiple types of contemporary threats.
- It includes various records of normal and malicious traffic.
- It has features that better reflect modern network behaviors than NSL-KDD.

3. Data Preprocessing

Before building the model, several basic steps were implemented:

1. Data Cleaning: Removing missing values and non-significant fields.
2. Data Splitting: Dividing the data into Training (70%) and Testing (30%).
3. Data Normalization: Converting values to a standard range (0–1) to speed up the training process.
4. Class Balancing: In the event of class imbalance, techniques such as SMOTE (Synthetic Minority Over-sampling) were used.

4. Feature Selection with EFPA

The Flower Pollination Algorithm (FPA) is based on the principle of cross-pollination and self-pollination in nature, while the Enhanced FPA (EFPA) offers improvements to the scaling factor to adjust the balance between exploration and exploitation.

EFPA steps:

- Generate random initial solutions representing sets of features.
- Evaluate each solution using a fitness function based on classification accuracy.
- Update solutions via a polling mechanism (local and global).
- Select the optimal set of features with the highest importance in attack detection.

Result: Significantly reduce the number of features without losing important information, which reduces computational complexity and increases training speed.

5. Ensemble Classification

After selecting the optimal features, an Ensemble Classifier was used, based on:

1. Random Forest (RF): Relies on combining multiple decision trees with a voting method.
2. Decision Tree (ID3): A simple classifier based on partitioning data based on attribute values.
3. Support Vector Machine (SVM): Effective at class separation using hyperplane optimization.

These classifiers are trained in parallel, and their results are then combined via Majority Voting to produce the final decision.

6. Simulation Environment

- Python 3. 9 is the language that is used for programming.
- Libraries such as NumPy, Matplotlib, pandas, and Scikit-learn were utilized in this project.
- Intel Core i7 processor, 16 gigabytes of random access memory, and Windows 10 as the operating system

7. Evaluation Standards

The model's performance was assessed using a variety of criteria, including:

• Accuracy: The percentage of samples that were correctly put into the right category.• Accuracy: The share of correct samples among those that were found to be positive. • Remember: The percentage of good samples that were actually found. • The F1 score is the steady middle ground between accuracy and memory. • AUC-ROC: A way to measure how well the model can tell the difference between classes. False Alarm Rate (FAR): The proportion of false alarms.

Conclusion and Recommendations

The proposed model, which is based on the Enhanced Flower Pollination Algorithm (EFPA) and contains Ensemble Classifier techniques (such as Random Forest, Decision Tree (ID3), and Support Vector Machine), is intended to address this issue. When it came to finding network threats, the Vector Machine (SVM) worked better than other methods.

Using the best feature selection on the UNSW-NB15 and NSL-KDD datasets, the model got accuracy rates of 99.32% and 99.67%, respectively. These results show that the system is better prepared for large-scale, real-time deployments by improving feature selection, which decreases computing complexity and improves accuracy.One of the most important things this study adds is the use of Ensemble Classification, which combines many classifiers to make the model better at generalization and making fewer mistakes. It gives fewer false results and works better against all types of attacks. The suggested way did better than other Machine Learning (ML) and Deep Learning (DL) approaches, especially in terms of Accuracy and Recall.

Despite these promising results, there are still some challenges that need to be addressed in the future. The model has been evaluated on limited standard datasets, opening the door to testing it on newer, more complex datasets such as CIC-IDS2017 and IoT network data.

In the future, feature selection methods could be developed by incorporating additional techniques such as autoencoder, PCA, and Information Gain to enhance classification efficiency.

Another important recommendation is to work on improving the model's ability to process streaming data so that it can operate in real time, and to integrate it with other security systems such as firewalls and blockchain-based security to achieve a more integrated protection system.

References

[1] J. Healey, “The US government and zero-day vulnerabilities: From pre-heartbleed to shadow brokers,” *J. Int. Affairs*, vol. 1, pp. 1–15, 2016.

[2] D. O’Brien, “Internet Security Threat Report—Ransomware 2017,” *Symantec*, vol. 11, pp. 203–214, 2017.

[3] P. Anantharaman, et al., “Going Dark: A Retrospective on the North American Blackout,” in *Proc. Paradigms Workshop (NSPW)*, 2018, pp. 1–10.

[4] J. Hawdon, et al., “Cybercrime victimization among Virginia businesses: Frequency, vulnerabilities, and consequences of cybervictimization,” *Criminal Justice Studies*, pp. 1–23, 2023.

[5] H. Abdi and L. J. Williams, “Partial least squares methods: Partial least squares correlation and partial least square regression,” in *Computational Toxicology, vol. II*, New York, NY: Springer, 2013, pp. 549–579.

[6] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, “Anomaly-based network intrusion detection: Techniques, systems and challenges,” *Comput. Secur.*, vol. 28, no. 1–2, pp. 18–28, 2009.

[7] S. Mukkamala, G. Janoski, and A. Sung, “Intrusion detection using neural networks and support vector machines,” in *Proc. Int. Joint Conf. Neural Netw. (IJCNN’02)*, 2002, pp. 1702–1707.

[8] D. E. Denning, “An intrusion-detection model,” *IEEE Trans. Softw. Eng.*, vol. SE-13, no. 2, pp. 222–232, 1987.

[9] E. Vasilomanolakis, S. Karuppayah, M. Mühlhäuser, and M. Fischer, “Taxonomy and survey of collaborative intrusion detection,” *ACM Comput. Surv.*, vol. 47, no. 4, pp. 1–33, 2015.

[10] A. H. Sung and S. Mukkamala, “Identifying important features for intrusion detection using support vector machines and neural networks,” in *Proc. Symp. Applications and the Internet (SAINT’03)*, 2003, pp. 209–216.

[11] M. A. Siddiqi, W. Pak, and M. A. Siddiqi, “A study on the psychology of social engineering-based cyberattacks and existing countermeasures,” *Appl. Sci.*, vol. 12, no. 12, p. 6042, 2022.

[12] O. I. Falowo, et al., “Threat actors’ tenacity to disrupt: Examination of major cyberattacks,” *J. Cybersecurity*, vol. 9, no. 1, pp. 1–12, 2023.

[13] S. Roy, et al., “A lightweight supervised intrusion detection mechanism for IoT networks,” *Future Gener. Comput. Syst.*, vol. 127, pp. 276–285, 2022.

[14] A. Ponmalar and V. Dhanakoti, “An intrusion detection approach using ensemble support vector machine based chaos game optimization algorithm in big data platform,” *Appl. Soft Comput.*, vol. 116, p. 108295, 2022.

[15] T. Verwoerd and R. Hunt, “Intrusion detection techniques and approaches,” *Comput. Commun.*, vol. 25, no. 15, pp. 1356–1365, 2002.

[16] P. Kabiri and A. A. Ghorbani, “Research on intrusion detection and response: A survey,” *Int. J. Netw. Secur.*, vol. 1, no. 2, pp. 84–102, 2005.

References

1. [1] J. Healey, “The US Government and Zero-Day Vulnerabilities: From Pre-Heartbleed to Shadow Brokers,” Journal of International Affairs, vol. 70, no. 1, pp. 1–15, 2016.
2. [2] D. O’Brien, “Internet Security Threat Report Ransomware 2017,” Symantec, vol. 11, pp. 203–214, 2017.
3. [3] P. Anantharaman et al., “Going Dark: A Retrospective on the North American Blackout,” in Proc. New Security Paradigms Workshop, 2018, pp. 1–10.
4. [4] J. Hawdon et al., “Cybercrime Victimization Among Virginia Businesses: Frequency, Vulnerabilities, and Consequences,” Criminal Justice Studies, vol. 36, no. 1, pp. 1–23, 2023.
5. [5] H. Abdi and L. J. Williams, “Partial Least Squares Methods: Partial Least Squares Correlation and Partial Least Square Regression,” in Computational Toxicology, vol. II, New York, NY, USA: Springer, 2013, pp. 549–579.
6. [6] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Macia-Fernandez, and E. Vazquez, “Anomaly-Based Network Intrusion Detection: Techniques, Systems and Challenges,” Computers and Security, vol. 28, no. 1–2, pp. 18–28, 2009.
7. [7] S. Mukkamala, G. Janoski, and A. Sung, “Intrusion Detection Using Neural Networks and Support Vector Machines,” in Proc. IEEE International Joint Conference on Neural Networks, 2002, pp. 1702–1707.
8. [8] D. E. Denning, “An Intrusion-Detection Model,” IEEE Transactions on Software Engineering, vol. SE-13, no. 2, pp. 222–232, 1987.
9. [9] E. Vasilomanolakis, S. Karuppayah, M. Muhlhauser, and M. Fischer, “Taxonomy and Survey of Collaborative Intrusion Detection,” ACM Computing Surveys, vol. 47, no. 4, pp. 1–33, 2015.
10. [10] A. H. Sung and S. Mukkamala, “Identifying Important Features for Intrusion Detection Using Support Vector Machines and Neural Networks,” in Proc. IEEE SAINT, 2003, pp. 209–216.
11. [11] M. A. Siddiqi, W. Pak, and M. A. Siddiqi, “Psychology of Social Engineering-Based Cyberattacks and Countermeasures,” Applied Sciences, vol. 12, no. 12, p. 6042, 2022.
12. [12] O. I. Falowo et al., “Threat Actors Tenacity to Disrupt: Examination of Major Cyberattacks,” Journal of Cybersecurity, vol. 9, no. 1, pp. 1–12, 2023.
13. [13] S. Roy et al., “Lightweight Supervised Intrusion Detection Mechanism for IoT Networks,” Future Generation Computer Systems, vol. 127, pp. 276–285, 2022.

14. [14] A. Ponmalar and V. Dhanakoti, “Intrusion Detection Using Ensemble Support Vector Machine Based Chaos Game Optimization in Big Data Platform,” Applied Soft Computing, vol. 116, p. 108295, 2022.
15. [15] T. Verwoerd and R. Hunt, “Intrusion Detection Techniques and Approaches,” Computer Communications, vol. 25, no. 15, pp. 1356–1365, 2002.
16. [16] P. Kabiri and A. A. Ghorbani, “Research on Intrusion Detection and Response: A Survey,” International Journal of Network Security, vol. 1, no. 2, pp. 84–102, 2005.