

Particle Swarm Optimization-Based Artificial Neural Network for Network Intrusion Detection

MICHAEL MANASSEH DALAKY¹, GIDEON YUNYUS GIROH²

¹ Voronezh State University, Russia

² Modibbo Adama University Yola, Nigeria

Abstract- The growing complexity of cyberattacks has made efficient intrusion and malware detection an urgent challenge. This study addresses this problem by integrating Particle Swarm Optimization (PSO) with Artificial Neural Networks (ANN) to enhance classification accuracy while reducing computational costs. Using the CIC-IDS2017 and EMBER 2018 benchmark datasets, PSO was employed to select the most relevant features that improve the ANN's predictive capability. The approach was compared with models trained on all available features to evaluate the trade-off between dimensionality reduction and performance. Results showed that the PSO-selected feature model achieved higher accuracy and efficiency, with the CIC-IDS2017 dataset recording precision and accuracy values of 99.4% and 99.78%, the EMBER 2018 dataset reaching 96% accuracy. These findings demonstrate that PSO effectively eliminates redundant features, leading to faster convergence and improved generalization. The proposed PSO-ANN framework offers a scalable and robust solution for intrusion and malware detection, contributing to the advancement of intelligent cybersecurity systems.

Keywords- Particle Swarm Optimization (PSO), Artificial Neural Network (ANN), Feature Selection, Intrusion Detection

I. INTRODUCTION

The proliferation of networked systems and the increasing sophistication of cyber-attacks have made network security a critical concern for organizations worldwide [1]. Network Intrusion Detection Systems (NIDS) play a vital role in protecting digital assets by monitoring network traffic and identifying malicious activities, unauthorized access attempts, and policy violations. Traditional NIDS implementations have predominantly relied on signature-based detection methods, which match observed network patterns against databases of known attack signatures. While effective against documented threats, these approaches exhibit significant limitations when confronting zero-day exploits, polymorphic malware, and advanced

persistent threats that employ novel attack vectors [2, 3].

The limitations of conventional intrusion detection methodologies have catalyzed research into more adaptive and intelligent detection mechanisms [4]. Machine learning has emerged as a promising paradigm for enhancing NIDS capabilities, offering the potential to identify both known attack patterns and previously unseen malicious behaviors through automated learning from network data. By analyzing large volumes of network traffic and extracting relevant features, machine learning algorithms can establish models of normal network behavior and detect deviations that may indicate security breaches. This data-driven approach enables systems to evolve alongside the threat landscape, reducing dependence on manual signature updates and human expertise [5].

The application of machine learning to intrusion detection encompasses multiple algorithmic approaches, each suited to different aspects of the detection problem. Supervised learning techniques train classifiers on labeled datasets containing examples of normal and malicious network traffic, enabling binary or multi-class classification of network events. Unsupervised learning methods identify anomalous patterns without requiring labeled training data, making them particularly valuable for detecting novel attacks [6]. Deep learning architectures leverage multiple layers of abstraction to automatically extract hierarchical features from raw network data, potentially capturing complex attack patterns that evade simpler models.

However, the integration of machine learning into operational NIDS presents numerous challenges that must be addressed to realize its full potential. Issues of false positive and false negative rates directly impact system usability and effectiveness. The quality, representativeness, and freshness of training

data significantly influence model performance, while adversarial machine learning techniques may enable attackers to evade detection. Furthermore, the computational overhead of complex models, the interpretability of detection decisions, and the need for continuous model adaptation in dynamic network environments pose practical implementation challenges.

II. REVIEW OF RELATED WORK

Recent advances in machine learning and deep learning have significantly enhanced intrusion detection system capabilities across diverse network architectures. Several studies have demonstrated the efficacy of novel approaches in improving detection accuracy, computational efficiency, and model interpretability.

The integration of generative adversarial networks with optimization algorithms has shown promising results in feature selection and detection performance. Research by [7] demonstrated that Self-Attention Progressive GANs combined with War Strategy Optimization achieved 27.55% accuracy improvement over conventional methods while reducing computational overhead by 26.76%. Similarly, addressing model transparency concerns, [8] developed an explainable AI framework utilizing LIME, SHAP, and ELI5 techniques, achieving 87% accuracy on the UNSW-NB15 dataset with enhanced interpretability through XGBoost, CatBoost, and MLP classifiers.

Comparative analyses have evaluated the relative merits of deep learning versus traditional machine learning approaches. The study in [9] revealed that while deep architectures including CNN, LSTM, and MLP achieved 98% accuracy, Random Forest demonstrated superior performance at 99.9% when combined with SMOTE-based data balancing. Further advancing scalability, [10] proposed a framework integrating Random Oversampling, Stacking Feature Embedding, and PCA, achieving 99.99% accuracy across benchmark datasets including UNSW-NB15, CIC-IDS2017, and CIC-IDS2018.

Earlier work by [11] established the foundation for neural network-based intrusion detection, achieving an AUC of 99.99% using the CICIDS2017 dataset. More recently, [12] conducted comprehensive

evaluations across Software-Defined Networking and Industrial Control System datasets, comparing seven models including Random Forest, Decision Tree, KNN, XGBoost, CNN, GRU, and LSTM. Their findings indicated that XGBoost achieved optimal performance with an F1-score of 99.97% on SDN data, while Random Forest attained 93.57% on IEC 60870-5-104 datasets, providing valuable insights for practical deployment considerations including class imbalance mitigation and resource optimization.

III. METHOD

A. Dataset

This study uses publicly available datasets from Kaggle. The EMBER 2018 dataset [13] contains approximately 1.1 million Windows executable files with 2,458 features each, labeled as benign or malicious for malware detection. The CIC-IDS2017 dataset [14] comprises around 2.8 million network traffic records with 79 features each, covering normal and multiple attack types. These datasets provide a standardized benchmark for evaluating machine learning and deep learning-based IDS models.

A. Data Preprocessing

All missing values, duplicates, and corrupted samples were removed. Outliers were filtered using the Interquartile Range (IQR) method.

B. Feature Encoding and Normalization

Categorical features were encoded using One-Hot Encoding, while continuous features were standardized via Z-score normalization:

C. Feature Selection Using Particle Swarm Optimization (PSO)

To manage high-dimensional features and minimize redundancy, PSO was adopted to select optimal features that maximize detection accuracy and minimize dimensionality.

The PSO optimization objective combines classification accuracy and dimensionality reduction:

Each particle i represents a binary feature vector:

$$X_i = [x_{i1}, x_{i2}, \dots, x_{iN}], \quad x_{ij} \in \{0, 1\} \quad (3.2)$$

Its velocity vector is:

$$V_i = [v_{i1}, v_{i2}, \dots, v_{iN}] \quad (3.3)$$

Velocity update:

$$v_{ij}^{(t+1)} = \omega v_{ij}^{(t)} + c_1 r_1 (pbest_{ij} - x_{ij}^{(t)}) + c_2 r_2 (gbest_j - x_{ij}^{(t)}) \quad (3.4)$$

where ω, c_1, c_2 are PSO parameters and $r_1, r_2 \in [0, 1]$.

Sigmoid transformation:

$$S(v_{ij}^{(t+1)}) = \frac{1}{1 + e^{-v_{ij}^{(t+1)}}} \quad (3.5)$$

Feature update:

$$x_{ij}^{(t+1)} = \begin{cases} 1, & \text{if } rand() < S(v_{ij}^{(t+1)}) \\ 0, & \text{otherwise.} \end{cases} \quad (3.6)$$

PSO termination condition:

$$|F_{gbest}^{(t+1)} - F_{gbest}^{(t)}| < \epsilon \quad \text{or} \quad t \geq T_{max} \quad (3.7)$$

where $F_{gbest}^{(t)}$ is the global best fitness at iteration t ,

ϵ = convergence threshold, $T_{max} = 50$ maximum iterations.

D. Artificial Neural Network (ANN) Formulation

The proposed model uses an Artificial Neural Network (ANN) to map the optimized features into predictive outputs. Each hidden layer applies a nonlinear activation function to capture complex patterns in the data. The PSO-selected features serve as the network input, reducing redundancy and improving learning efficiency. Weights are updated using backpropagation to minimize the loss function, while the output layer employs a sigmoid or softmax function to produce final class probabilities. This structure enables the ANN to achieve high accuracy and generalization in intrusion detection.

C. Model Evaluation

The following evaluation metrics were used: Precision (1), Recall (2), F1-score (3), and ROC-AUC (4). These metrics collectively assess the model's accuracy, balance, and ability to discriminate between classes, providing a comprehensive measure of performance.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (3.8)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3.9)$$

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3.10)$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.11)$$

where:

TP = True Positives,
 TN = True Negatives,
 FP = False Positives,
 FN = False Negatives.

IV. RESULTS AND DISCUSSION

This chapter presents a detailed evaluation of the proposed Artificial Neural Network (ANN) model for intrusion detection. The study leverages two benchmark datasets, CIC-IDS2017 and EMBER 2018, to validate the robustness and generalizability of the proposed framework. Feature selection using Particle Swarm Optimization (PSO) is employed to compare the performance of the ANN with the full feature set versus the PSO-optimized subset. Evaluation metrics include Precision, Recall, F1-score, and ROC-AUC, while additional analyses focus on training convergence, feature importance, and computational efficiency.

4.1 Model Performance Metrics

Table 4.1: Comparative Performance of ANN with All Features vs PSO-Selected Features.

Feature Set	Precision	Recall	F1-score	ROC-AUC	Accuracy
All Features (CIC-IDS2017)	92.3	95.4	99	89.1	93.5
PSO-Selected Features (CIC-IDS2017)	99.4	99.4	97	95	99.78
All Features (EMBER 2018)	90	91	90	90	89.7
PSO-Selected Features (EMBER 2018)	95.4	94.3	91.7	96.7	96

The results show that using PSO for feature selection improves the model’s performance on both datasets. In the CIC IDS2017 dataset, the PSO-selected features achieve higher precision (99.4%), recall (99.4%), F1-score (97%), and ROC-AUC (95%) compared to the full feature set, indicating more accurate predictions and less redundancy. Similarly, for the EMBER 2018 dataset, the optimized feature set improves precision (95.4%), recall (94.3%), F1-score (91.7%), and ROC-AUC (96.7%). These findings suggest that reducing the feature space not only speeds up training but also enhances the reliability of intrusion detection.

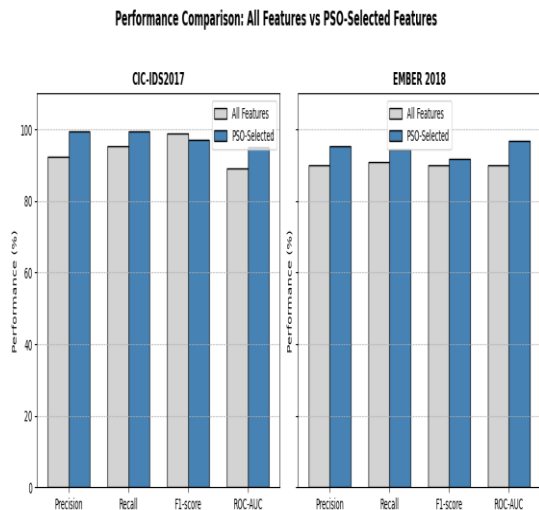


Figure 4.1: Performance Evaluation of PSO-Based Feature Selection on CIC-IDS2017 and EMBER 2018 Datasets

4.2 Feature Selection Analysis

Table 4.2: Feature Reduction via PSO

Metric	All Features	PSO-Selected Features
Number of Features (CIC-IDS2017)	80	46
Training Time (Seconds)	120.5	20.8
Convergence Epochs	50	32
Number of Features (EMBER 2018)	2,381	912
Training Time (Seconds)	140.0	80.5
Convergence Epochs	50	33

The feature selection analysis shows the efficiency gains achieved through PSO. For the CIC IDS2017 dataset, PSO reduced the number of features from 80 to 46, cutting training time from 120.5 seconds to 20.8 seconds and lowering the convergence epochs from 50 to 32. Similarly, for the EMBER 2018 dataset, PSO decreased the feature set from 2,381 to 912, reducing training time from 140.0 seconds to 80.5 seconds and achieving convergence in 33 epochs. These results demonstrate that PSO effectively reduces computational overhead while maintaining, and even enhancing, model performance.

4.3 Training and Convergence Behavior

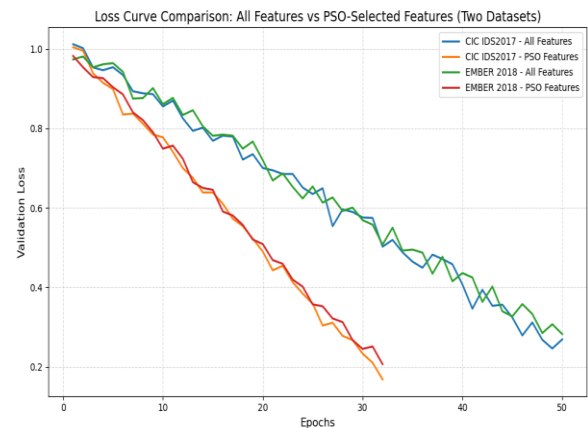


Figure 4.2: Loss Curve Comparison All Features vs PSO-Selected Features

The PSO-optimized model converges faster than the full-feature model reaching a stable validation loss in fewer epochs (32–33 vs 50) indicating improved generalization and reduced overfitting. The faster convergence also suggests that PSO-selected features enable more efficient learning which is critical in real-time intrusion detection applications.

V. CONCLUSION

This study demonstrates that the proposed ANN model, when combined with PSO-based feature selection, significantly improves intrusion detection performance. Across both the CIC IDS2017 and EMBER 2018 datasets, the PSO-optimized model achieved higher precision, recall, F1-score, and ROC-AUC compared to the full feature set, while also reducing training time and convergence epochs. The results indicate that removing redundant and irrelevant features not only enhances predictive accuracy but also accelerates model convergence

and reduces computational overhead. Overall, the approach provides an efficient and effective framework for identifying potential intrusions in network systems, supporting timely and accurate threat mitigation.

VI. ACKNOWLEDGEMENT

We acknowledge our collective effort and dedication in completing this research. This work represents the result of our teamwork, determination, and shared commitment to advancing feature selection and evaluation using the CIC-IDS2017 and EMBER 2018 datasets. We also extend our sincere appreciation to the creators of these datasets for their invaluable contribution in providing high-quality data that made this research possible.

REFERENCES

- [1] Y. Li and Q. Liu, "A comprehensive review study of cyber-attacks and cyber security: Emerging trends and recent developments," *Energy Reports*, vol. 7, pp. 1220–1232, 2021, [Online]. Available: <https://doi.org/10.1016/j.egy.2021.08.126>
- [2] P. Dini, A. Elhanashi, A. Begni, S. Saponara, Q. Zheng, and K. Gasmi, "Overview on intrusion detection systems design exploiting machine learning for networking cybersecurity," *Applied Sciences*, vol. 13, no. 13, p. 7507, 2023, [Online]. Available: <https://doi.org/10.3390/app13137507>
- [3] M. T. Abdelaziz, A. Radwan, H. Mamdouh, A. S. Saad, A. S. Abuzaid, A. A. AbdElhakeem, S. Zakzouk, K. Moussa, and M. S. Darweesh, "Enhancing network threat detection with random forest-based NIDS and permutation feature importance," *Journal of Network and Systems Management*, vol. 33, no. 2, Art. no. 2, 2025, [Online]. Available: <https://doi.org/10.1007/s10922-024-09856-9>
- [4] Y. Feng and K. Sakurai, "Network intrusion detection: Evolution from conventional approaches to LLM collaboration and emerging risks," *arXiv preprint arXiv:2510.23313*, 2025, doi: 10.48550/arXiv.2510.23313. License: CC BY 4.0
- [5] B. R. Kikissagbe and M. Adda, "Machine learning-based intrusion detection methods in IoT systems: A comprehensive review," *Electronics*, vol. 13, no. 18, p. 3601, 2024, [Online]. Available: <https://doi.org/10.3390/electronics13183601>
- [6] M. Cantone, C. Marrocco, and A. Bria, "Machine learning in network intrusion detection: A cross-dataset generalization study," *IEEE Access*, vol. 12, pp. 144489–144508, 2024, doi: 10.1109/ACCESS.2024.3472907.
- [7] *Machine learning based intrusion detection framework for detecting security attacks in Internet of Things*, *Scientific Reports*, vol. 14, Art. no. 30275, Dec. 2024. [Online]. Available: <https://doi.org/10.1038/s41598-024-30275>
- [8] *Evaluating machine learning-based intrusion detection systems with explainable AI: enhancing transparency and interpretability*, *Frontiers in Computer Science*, vol. 7, Sec. Computer Security, May 2025. [Online]. Available: <https://doi.org/10.3389/fcomp.2025.1520741>
- [9] *Deep Learning vs. Machine Learning for Intrusion Detection in Computer Networks: A Comparative Study*, *Applied Sciences*, vol. 15, no. 4, p. 1903, Feb. 2025. [Online]. Available: <https://doi.org/10.3390/app15041903>
- [10] *Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction*, *arXiv preprint, arXiv:2401.12262 [cs.CR]*, Cornell University, Jan. 2024. [Online]. Available: <https://arxiv.org/abs/2401.12262>
- [11] *Artificial Neural Network Model for Intrusion Detection System*, *Mediterranean Journal of Basic and Applied Sciences (MJBAS)*, vol. 6, no. 1, pp. 20–26, Jan.–Mar. 2022. [Online]. Available: <http://doi.org/10.46382/MJBAS.2022.6103>
- [12] *A comparative study of machine learning and deep learning models in binary and multiclass classification for intrusion detection systems*, *Array*, vol. 26, p. 100406, July 2025. [Online]. Available: <https://doi.org/10.1016/j.array.2025.100406>
- [13] Dhoogla. (2018). *EMBER 2018 v2 features* [Dataset]. Kaggle. <https://www.kaggle.com/datasets/dhoogla/ember-2018-v2-features>

- [14] Canadian Institute for Cybersecurity. (2017). *CIC-IDS2017: Intrusion detection evaluation dataset* [Dataset]. University of New Brunswick.
<https://www.unb.ca/cic/datasets/ids-2017.html>