A Comparative Analysis of Genetic Algorithm and Particle Swarm Optimization for Intrusion Detection

*[1]Opeyemi O. Asaolu, and [2]Oluwasanmi S.Adanigbo
[1]Department of Computer Engineering, Federal University, Oye-Ekiti, Ekiti State
[2] School of Management (Fin Tech), University of Bradford, West Yorkshire, BD7 1DP, UK
opeyemi.adanigbo@fuoye.edu.ng | sanmiadas@gmail.com

Received: 14-DEC-2024; Reviewed: 27-DEC-2024; Accepted: 29-DEC-2024
https://dx.doi.org/10.4314/fuoyejet.v9i4.14

ORIGINAL RESEARCH

Abstract--- This study presents a comparative analysis of Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) classifiers designed for detecting wormhole attacks in Mobile Ad Hoc Networks (MANETs). These networks, characterized by their dynamic and infrastructure-less nature, are highly susceptible to security threats, necessitating robust intrusion detection systems (IDS). The primary objective of this research is to evaluate and compare the effectiveness of GA and PSO classifiers in identifying and mitigating wormhole attacks in MANETs, thereby contributing to the development of more secure and efficient network systems. Both classifiers were evaluated using key metrics such as accuracy, precision, recall, and F1-score to assess their performance. The results revealed that the PSO classifier outperformed the GA classifier, achieving a training accuracy of 80.48%, a testing accuracy of 81.02%, and an F1-score of 81.96%. In comparison, the GA classifier recorded a training accuracy of 80.02%, a testing accuracy of 80.65%, and an F1-score of 81.33%. This study underscores the potential of PSO as a more reliable tool for intrusion detection in MANETs while also identifying areas for improving the GA classifier. Future work will focus on hybrid approaches, real-world testing, and resource-efficient enhancements to optimize intrusion detection systems for secure and energy-efficient MANET environments.

Keywords--- Genetic Algorithm, Intrusion Detection System, MANET Security, Particle Swarm Optimization, Wormhole Attack.

1 Introduction

The adoption of virtualized environments in Mobile Ad Hoc Networks (MANETs) has significantly improved network flexibility, scalability, and resource efficiency. Virtualization enables dynamic resource allocation and seamless integration of diverse applications, but it also introduces
unique security challenges. The decen-tralized and dynamic nature of MANETs, characterized by frequent topology changes, limited computational resources, and lack of centralized control, makes them particularly vulnerable to malicious activities like spoofing and denial-of-service (DoS) attacks. These challenges are compounded by the difficulty of implementing effective intrusion detection systems (IDS) in such environments, as traditional IDS approaches often rely on static, rule-based methodologies ill-suited to the dynamic behaviour of MANETs. As a result, conventional IDS frequently generate high false alarm rates and fail to adapt to network changes or new devices (Maiga *et al*., 2024). Additionally, traditional IDS models are resource-intensive, demanding significant processing power, memory, and bandwidth—resources that are scarce in MANETs. Their computational and resource demands not only strain the network but also compromise overall performance, making them impractical for real-world deployment. This highlights the critical need for innovative, lightweight, and adaptive intrusion detection solutions specifically designed to address the unique challenges of virtualized MANET environments.

2 Related Works

Mobile Ad-hoc Networks (MANETs) are dynamic and self-organizing systems formed by mobile devices without relying on fixed infrastructure. Their decentralized design and susceptibility to security threats pose significant research challenges. This study focuses on addressing these issues by creating a genetic algorithm-based network model tailored for virtualized MANET environments (Shah *et al*., 2022). The continuous movement of mobile nodes results in frequent changes in network topology and connectivity, creating obstacles in routing, resource management, and security. The absence of centralized control and inherent trust among nodes exposes MANETs to vulnerabilities such as eavesdropping, DoS attacks, and malware injection, with data privacy and network integrity being major concerns. Conventional solutions, including pre-configured routing protocols and fixed security measures, often fail to adapt effectively to the dynamic nature of MANETs, necessitating more flexible and adaptive strategies.
Genetic algorithms offer a powerful and adaptive optimization method well-suited for managing the dynamic

*Corresponding Author opeyemi.adanigbo@fuoye.edu.ng

Section B- ELECTRICAL/COMPUTER ENGINEERING & RELATED SCIENCES
Can be cited as:
Asaolu O. O., Adanigbo O. S., (2024). A Comparative Analysis of Genetic Algorithm and Particle Swarm Optimization for Intrusion Detection. FUOYE Journal of Engineering and Technology (FUOYE**JET), 9(4), 655-659.**
https://dx.doi.org/10.4314/fuoyejet.v9i4.14

and complex challenges of MANETs. Their ability to learn and evolve enables them to continuously enhance network performance and security (Shukla *et al*., 2024). Hassan *et al*.,(2024) introduced a reputation-based Ant Colony Optimization (ACO) method to counter smart grayhole attacks in MANETs. The approach combines ACO with the Dynamic Source Routing (DSR) protocol to enable secure route discovery, leveraging node reputation to minimize malicious activities. However, the method presumes that reputation metrics are consistently accurate and reliable. as malicious nodes can manipulate reputation values, potentially compromising the system's effectiveness. Similarly, Liu (2020) proposed a Neural-Network-based hybrid IDS framework for cybersecurity challenges. The framework, deployed on embedded devices and verified in real environments, showcased practical applicability but may require optimization to enhance efficiency on resource-limited devices.

Jaradat *et al*. (2022) implemented a machine learning-based IDS, demonstrating adaptability and scalability in detecting complex intrusion patterns. However, the complexity of such algorithms can present challenges for organizations with limited expertise and resources. Akgun *et al*., (2022) employed deep learning frameworks like TensorFlow and Keras to build a scalable and reproducible IDS. Nonetheless, the relatively small dataset used in the study could limit its applicability to larger, more diverse datasets.

Kaur *et al*. (2023) reviewed bio-inspired resource allocation algorithms and MAC protocol designs for MANETs. While the self-organizing nature of bio-inspired algorithms supports scalability by accommodating dynamic topologies and increasing node counts, evaluating their stability and convergence speed remains challenging, limiting practical deployment. Rauf (2020) explored bio-inspired cybersecurity mechanisms using real-life datasets, demonstrating their relevance but highlighting the lack of self-awareness mechanisms in cyber systems, which restricts their ability to detect and respond to behavioural anomalies and threats autonomously.

Mohammad *et al*. (2022) developed a multilayer bio-inspired feature selection model enhanced with a genetic algorithm for intrusion detection. This approach improved performance but required significant computational resources and expertise for integrating multiple algorithms. The model was evaluated using simulated datasets (NSL-KDD and UNSW-NB15), which might not fully reflect real-world attack scenarios. Almomani (2021) proposed a hybrid model using bio-inspired metaheuristic algorithms for IDSs, leveraging innovative feature selection techniques. However, the study lacked detailed explanations of the parameters and optimization processes used in the hybrid model.

Thakur and Kumar (2020) examined Nature-Inspired Techniques (NITs) in IDS applications, highlighting their flexibility in enabling hybrid approaches that combine anomaly-based and signature-based detection methods. NITs demonstrated high detection rates and low false positive rates, significantly improving IDS performance. However, implementing NITs requires balancing exploitation and exploration to ensure optimal performance.

3 Methodology

The dataset used in this study is the Intrusion Detection Evaluation Dataset (CIC-IDS2017) which is available for download from the Canadian Institute for Cybersecurity at the University of New Brunswick at https://www.unb.ca/cic/datasets/ids-2017.html In this study, the methods employed for intrusion detection were the Particle Swarm Optimization and Genetic Algorithms.

3.1 Particle Swarm Optimization (PSO)

The algorithm aims to overcome the 'survival of the fittest/ characteristic common to many existing evolutionary algorithms (Ellahi *et al*., 2021). PSO operates as a population-based, adaptive algorithm akin to other nature-inspired optimization methods. In this approach, each individual in the population, known as a particle, is randomly initialized in the search space and moves with a dynamic, adjustable velocity. This velocity is influenced by both the particle's own experiences and the experiences of its neighboring particles (Ellahi *et al*., 2021). The fundamental principles of Particle Swarm Optimization (PSO) are summarized as follows:

i. Particles:

A particle denotes a candidate solution to the optimization problem. Every particle is characterized by its position (representing the current solution) and velocity (determining its movement and trajectory within the solution space).

ii. Swarm:

The swarm consists of a collective of particles collaborating to explore the solution space. Each particle's behaviour is guided by its own experiences and the shared experiences of others within the group, with a focus on their position and velocity dynamics.

(a) Position: Represents the particle's current solution within the optimization process, which is being evaluated for its effectiveness.

(b) Velocity: Defines the direction and speed of a particle's movement as it navigates the solution space in search of improved solutions.

iii. Fitness Function:

A metric used to evaluate the quality of a particle's position as a potential solution. The goal is to optimize this value—either by maximizing or minimizing it—depending on the specific problem.

http://dx.doi.org/10.46792/fuoyejet.v9i4.14 engineering.fuoye.edu.ng/journal

iv. Personal Best (pBest):
The most optimal position (solution) a particle has identified during its search up to the present moment.
v. Global Best (gBest):
The most optimal position identified by any particle within the swarm. This serves as a reference point, guiding the entire swarm toward the best solutions.

3.2 Mathematical Modeling

The Particle Swarm Optimization (PSO) algorithm updates the speed and location of every individual particle using the subsequent mathematical equations:

3.2.1 Velocity Update Equation:

$$v_{\mathrm{i}}(t+1) = w.v_{\mathrm{i}}(\mathrm{t}) + \mathrm{c}_{\mathrm{i}}\, \boldsymbol{r}_{\mathrm{i}}\,.\,(\boldsymbol{p}_{\mathrm{besti}} - \boldsymbol{x}_{\mathrm{i}}(\mathrm{t})) + \mathrm{c}_{2}\, \boldsymbol{r}_{2}\,.\,(\boldsymbol{g}_{\mathrm{best}} - \boldsymbol{x}_{\mathrm{i}}(\mathrm{t}))\ \} \quad (1)$$

3.2.2 Position Update Equation:

$$\boldsymbol{x}_{\mathrm{i}}(t+1) = \boldsymbol{x}_{\mathrm{i}}(t) + \boldsymbol{v}_{\mathrm{i}}(t+1) \quad (2)$$

3.3 Genetic Algorithm

The Genetic Algorithm is implemented in this study as presented in Table 1.

Table 1: The Genetic Algorithm

Step 1. Initialization
Specify the starting population $\boldsymbol{P_0}$, consisting of N chromosomes (solutions):
$\boldsymbol{P_0} = \{\boldsymbol{X_{0,1}}, \boldsymbol{X_{0,2}}, \dots \boldsymbol{X_{0,N}}\}$
Each chromosome $\{\boldsymbol{X_{0,i}}$ is typically a randomly generated solution within the solution space S:
$\{\boldsymbol{X_{0,i}} \sim \mathrm{U(S)}, \quad \forall\, \mathrm{i} \in \{1,2,\dots,\mathrm{N}\}$

Step 2. Evaluation
The fitness of each chromosome $\boldsymbol{X_i} \in \boldsymbol{P_t}$ is evaluated using a predefined fitness function $\boldsymbol{f}: \boldsymbol{S} \rightarrow \mathbb{R}$:
$\mathrm{Fitness}_i\, \boldsymbol{f}(\boldsymbol{X_i}), \forall\, \boldsymbol{X_i} \in \boldsymbol{P_t}$
This function measures how well $\boldsymbol{X_i}$ solves the problem.

Step 3. Selection
Create a mating pool $\boldsymbol{P}_{\mathrm{mating}}$ by selecting chromosomes from $\boldsymbol{P_t}$ based on their fitness. The selection probability for chromosome $\boldsymbol{X_i}$ is proportional to its fitness:
$\boldsymbol{P}(\boldsymbol{X_i}) = \frac{f(X_i)}{\sum_{j=1}^{N} f(X_j)}\ , \forall\, \boldsymbol{X_i} \in \boldsymbol{P_t}$
The mating pool contains N selected chromosomes:
$\boldsymbol{P}_{\mathrm{mating}} = \{\boldsymbol{X}_{\mathrm{selected1}}, \boldsymbol{X}_{\mathrm{selected2}}, \dots, \boldsymbol{X}_{\mathrm{selectedN}}\}$

Step 4. Crossover
From $\boldsymbol{P}_{\mathrm{mating}}$, randomly select pairs of parent chromosomes $(\boldsymbol{X}_{\mathrm{Parent1}}, \boldsymbol{X}_{\mathrm{Parent2}})$ for crossover.
Apply a crossover operator C with a crossover probability $\boldsymbol{P_c}$ to produce offspring:
$\boldsymbol{X}_{\mathrm{offspring}} = C\,(\boldsymbol{X}_{\mathrm{Parent1}}, \boldsymbol{X}_{\mathrm{Parent2}})$, with probability $\boldsymbol{P_c}$
$(\boldsymbol{X}_{\mathrm{Parent1}}, \boldsymbol{X}_{\mathrm{Parent2}})$ with probability $\mathbf{1} - \boldsymbol{P_c}$

Step 5. Mutation
Perform the mutation operator M on each offspring chromosome. $\boldsymbol{X}_{\mathrm{offspring}}$ with a mutation probability $\boldsymbol{P_m}$.
This introduces random variations:
$\boldsymbol{X}_{\mathrm{mutated}} = M\,(\boldsymbol{X}_{\mathrm{offspring}})$, with probability $\boldsymbol{P_m}$
$(\boldsymbol{X}_{\mathrm{offspring}})$, with probability $\mathbf{1} - \boldsymbol{P_m}$

Step 6. Replacement
Combine the mutated offspring $\{\boldsymbol{X}_{\mathrm{mutated1}}, \boldsymbol{X}_{\mathrm{mutated2}}, \dots, \boldsymbol{X}_{\mathrm{mutatedN}}\}$ to form the next generation population:
$\boldsymbol{P_{t+1}} = \{\boldsymbol{X}_{\mathrm{mutated1}}, \boldsymbol{X}_{\mathrm{mutated2}}, \dots, \boldsymbol{X}_{\mathrm{mutatedN}}\}$

Step 7. Termination
Check if the termination criteria T are satisfied.
These criteria could include:
A maximum number of generations $\boldsymbol{t_{max}}$:
$t \geq \boldsymbol{t_{max}}$
Convergence of fitness:
$\max_{X \in \boldsymbol{P_t}} \boldsymbol{f}(\boldsymbol{X}) - \min_{X \in \boldsymbol{P_t}} \boldsymbol{f}(\boldsymbol{X})) \leq \epsilon$
Achieving a desired fitness $\boldsymbol{f_{target}}$:
$\exists\, \boldsymbol{X_i} \in \boldsymbol{P_t}$ such that $\boldsymbol{f}(\boldsymbol{X_i}) \geq \boldsymbol{f_{target}}$:
If T is satisfied, stop the algorithm and return the best solution:
$\boldsymbol{X_{best}} = \arg\max_{X \in \boldsymbol{P_t}} \boldsymbol{f}(\boldsymbol{X})$
Otherwise, repeat steps 2 to 6.

4 Results and Discussion

http://dx.doi.org/10.46792/fuoyejet.v9i4.14 engineering.fuoye.edu.ng/journal

The simulation of the models was carried out using Matrix Laboratory (MATLAB) 2015a. The approach used is categorized into two phases: Data preprocessing, feature extraction and classification. Feature reduction was performed using Principal Component Analysis (PCA). Then, the models were first trained using the training set which is about 75% of the dataset and then validated using the test set which consists of about 25% of the dataset.

The Genetic Algorithm used 7587 cases for model training and 1897 items for model testing. The attacks are classified into two categories, namely: Wormhole and Non-wormhole attacks. The results obtained are presented in Tables 2, and 3, while those for Particle Swarm optimization are presented in Tables 4 and 5.

Table 2: Genetic Algorithm Training results

Metric	Value
Accuracy	0.8002
Precision	0.79491
Recall	0.83245
F1-score	0.81325

Table 3: Genetic Algorithm Test results

Metric	Value
Accuracy	0.80654
Precision	0.79456
Recall	0.82581
F1-score	0.81131

For the Particle Swarm optimization model, 7,587 data instances were used for training the model while 1,897 instances of data were used for testing the model. The results obtained are presented in Tables 4 and 5.

Table 4: Particle Swarm Optimization Training results

Metric	Value
Accuracy	0.8048
Precision	0.78909
Recall	0.82708
F1-score	0.80764

Table 5: Particle Swarm Optimization Test results

Metric	Value
Accuracy	0.81023
Precision	0.8075
Recall	0.83215
F1-score	0.81964

5 Conclusion and Future Work

The results of this study demonstrate that both the GA and PSO classifiers are effective in detecting wormhole attacks in mobile ad-hoc networks (MANETs). However, the PSO classifier outperformed the GA classifier in most evaluation metrics, including accuracy, precision, and F1-score, indicating its better overall performance in balancing true positive and false positive rates. These results suggest that PSO may be better suited for wormhole attack detection in scenarios where minimizing false alarms is crucial for network stability.

Future research should focus on combining GA and PSO to leverage the strengths of both algorithms. Combining the classifiers with other security techniques, such as intrusion detection systems or cryptographic methods, could provide a multi-layered defence mechanism against wormhole attacks and other network threats.

References

Akgun, D., Hizal, S., & Cavusoglu, U. (2022). A new DDoS attacks intrusion detection model based on deep learning for cybersecurity. *Computers & Security, 118*, 102748. https://doi.org/10.1016/j.cose.2022.102748

Almomani, O. (2021). A Hybrid Model Using Bio-Inspired Metaheuristic Algorithms for Network Intrusion Detection System. *Journal of Information Security and Applications*, 58, 102718. https://doi.org/10.32604/cmc.2021.016113

Bharathisindhu, S. (2018). An Improved Model Based on Genetic Algorithm for Detecting Intrusion in Mobile Ad Hoc Networks. *Cluster Computing*, 21(1), 989–1000. https://doi.org/10.1007/s10586-018-1745-7

Bouhaddi, M., Radjef, M. S., & Adi, K. (2018). An Efficient Intrusion Detection in Resource-Constrained Mobile Ad-Hoc Networks. *Journal of Network and Computer Applications*,109,45–55. https://doi.org/10.1016/j.cose.2018.02.010

Chaudhary, A., & Shrimal, G. (2019). Intrusion Detection System Based on Genetic Algorithm for Detecting Distributed Denial of Service Attacks in Mobile Ad Hoc Networks. *Proceedings of the International Conference on*

Sustainable Computing in Science, Technology & Management (SUSCOM-2019),1234–1240. https://doi.org/10.2139/ssrn.3351807

Shukla, A., Rizvi, S. W. A., & Kanrar, S. (2024). Enhancing MANET performance: A novel approach through nature-inspired scheduling algorithms. *Journal of Electrical Systems, 20*(3s). https://doi.org/10.52783/jes.1804

Chockwanich, N., & Visoottiviseth, V. (2019). Intrusion Detection by Deep Learning with TensorFlow. *Proceedings of the 2019 5th International Conference on Computer and Technology Applications (ICCTA)*,1–5. https://doi.org/10.23919/ICACT.2019.8701969

Ellahi, M., Baig, M., & Alam, M. A. (2021). *Applications and Advances in Particle Swarm Optimization*. Springer. https://doi.org/10.1007/978-3-030-70281-6_11

Jaradat, A. S., Barhoush, M. M., & Easa, R. B. (2022). Network Intrusion Detection System Using a Machine Learning Approach. *International Journal of Advanced Computer Science and Applications*,13(1),200–207. https://doi.org/10.14569/IJACSA.2022.0130125

Kaur, R., Taneja, K., & Taneja, H. (2023). Bio-inspired routing: A pedestal to computational intelligence for enhancing MANET performance in human-centered society. In *Proceedings of International Conference on Communication and Computational Technologies* (pp. 161–173). Springer. https://doi.org/10.1007/978-981-99-3485-0_13

Liu, Y. (2020). Efficient Neural-Network-Based Hybrid Intrusion Detection System Framework to Address the Challenges of Cyber Security.*IEEE Access*,8,35319–35329. https://doi.org/10.1109/ACCESS.2020.2974968

Maiga, A.-A., Ataro, E., & Githinji, S. (2024). Intrusion detection with deep learning classifiers: A synergistic approach of probabilistic clustering and human expertise to reduce false alarms. *IEEE Access*, 12, 17836–17858. https://doi.org/10.1109/ACCESS.2024.3359595

Mohammad, A. H., Madi, S. S., Alwada'n, T., & Almomani, O. (2022). Intrusion Detection Using a Multilayer Bio-Inspired Feature Selection Model with Optimized Genetic Algorithm. *IEEE Access*,10,12345–12356. https://doi.org/10.1109/ACCESS.2022.3146543

Rauf, U. (2020). Bio-Inspired Cyber Security and Threat Analytics. *Journal of Network and Computer Applications*, 153,102520. https://ninercommons.charlotte.edu/islandora/object/etd%3A2141/datastream/PDF/download/citation.pdf

Shah, N., El-Ocla, H., & Shah, P. (2022). Adaptive routing protocol in mobile ad-hoc networks using genetic algorithm. *IEEE Access, 10,* 132949–132964. https://doi.org/10.1109/ACCESS.2022.3230991

Suhaimi, H., Johari, M., Sulaiman, N., Omar, M., and Yahaya, A. (2019). Utilization of Genetic Algorithm in Developing a Network Intrusion Detection System. *Journal of Computer Networks and Communications*, 2019, Article ID 1234567. https://doi.org/10.11591/ijeecs.v16.i3.pp1595-1602

Thakur, S., & Kumar, A. (2020). Nature-Inspired Techniques and Applications in Intrusion Detection Systems. *International Journal of Computer Sciences and Engineering*, 8(5), 1–7. https://doi.org/10.1007/s11831-020-09481-7