Rishav Kumar Mishra, 2025, 13:6
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

IOT Network Malicious Session Detection by Genetic Feature Optimization Algorithm

Rishav Kumar Mishra, Prof. Sujeet Gautam, Prof. S Vishwakarma
Department of Computer Science and Engineering Patel College of Science and Technology, Bhopal, MP, India

Abstract- The rapid growth of Internet of Things (IoT) networks has significantly improved human comfort and quality of life. However, this expansion has also increased vulnerability to cyber intrusions, making IoT security a critical concern. This work proposes an IoT network intrusion detection system that classifies network sessions into normal and attack categories. A Genetic Algorithm (GA) is employed for optimal feature selection, enabling the identification of the most representative session attributes for accurate classification. The selected features are then utilized by the K-Nearest Neighbour (KNN) classifier to detect intrusions effectively. Experiments conducted on a real-world dataset demonstrate that the proposed GA-based IoT Network Security model significantly enhances detection accuracy and optimizes key evaluation performance metrics.

Index Terms - Machine Learning, IOT Network Optimization, Intrusion Detection.

I. INTRODUCTION

Security and privacy concerns related to computer networks are growing rapidly worldwide due to the extensive integration of information technology into everyday activities. The increasing number of Internet-based applications, along with the emergence of advanced technologies such as the Internet of Things (IoT), has created new opportunities for cyberattacks targeting networks and computing systems [1].

The IoT refers to a collection of interconnected devices capable of communicating autonomously without human intervention. These devices, often embedded with sensors, include smart appliances, lighting systems, vehicles, medical equipment, and agricultural tools, and are widely deployed in sectors such as healthcare, farming, transportation, and smart environments [2].

While IoT technologies improve efficiency, reduce operational costs, and enable intelligent decision-making, they also inherit all security vulnerabilities associated with the Internet, as it serves as the core communication backbone [3]. Furthermore, IoT devices generally possess limited computational power, storage capacity, and energy resources, and often function without direct human supervision. The rapid growth and large-scale deployment of IoT devices in daily life significantly amplify security challenges, emphasizing the need for robust network-based security solutions [4]. Although existing systems can detect certain types of attacks, identifying sophisticated and evolving threats remains difficult.

As cyberattacks become more frequent and network traffic volumes continue to increase, there is a growing demand for faster and more intelligent intrusion detection mechanisms [5].

In this regard, machine learning (ML) has emerged as an effective approach for enhancing IoT security by enabling adaptive and data-driven decision-making. ML techniques have been widely applied in network traffic analysis [6], intrusion detection [7], and botnet identification [8]. Attack detection using ML is generally performed through signature-based or anomaly-based methods. Signature-based techniques are particularly effective for detecting known attacks with low false alarm rates, making them a valuable component of IoT security frameworks [9].

II. RELATED WORK

The authors in [12] propose an intrusion detection model based on a genetic algorithm and a deep belief network. They use the NSL-KDD dataset for detecting four types of attacks: DoS, R2L, Probe and U2R. This paper, in comparison with our work, uses an old dataset difficult to be applicable to modern IoT networks and does not implement blockchain in their solution as an integrated mechanism for monitoring and securing IIoT networks.

In [13], an intrusion detection technique based on statistical flow features is proposed for protecting the network traffic of Internet of Things applications. The authors in this work use three machine learning techniques to detect malicious traffic events: Decision Tree, Naive Bayes and Artificial Neural Network (ANN). They use the same dataset employed by us, the UNSWNB15 dataset; however, they do not implement blockchain in their solution as an integrated mechanism for monitoring and securing IIoT networks.

A machine learning security framework for IoT systems is proposed in [15]. They built a dataset based on the NSL-KDD dataset and evaluated their proposal in a real smart building scenario. As we said in the previous related works, an old dataset may not be suitable for modern IoT networks. They use one-class SVM (Support Vector Machine) technique for detecting four types of attacks: DDoS, Probe, U2R and R2L. However, they do not use a blockchain approach for supervising IIoT networks.

The authors in [16] developed an algorithm for detecting denial-of-service (DoS) attacks using a deep-learning algorithm. They use three approaches for detecting DoS attacks: Random Forests, a Multilayer Perceptron and a Convolutional Neural Network. They use the same dataset employed by us, but they just aim to detect one attack (DoS) and do not integrate blockchain in their solution.

The authors in [20] propose a model using a machine learning algorithm to detect and mitigate botnet-based distributed denial of service (DDoS) attacks in IoT networks. The use different machine learning algorithms such as K- Nearest Neighbour (KNN), Naive Bayes model and Multi-layer Perception Artificial Neural Network (MLP ANN). They use the same dataset employed by us, but they just aim to detect one attack (DoS) and do not integrate blockchain in their solution.

In [21], the authors propose an intrusion and cyber attacks traffic identification model using Machine Learning (ML) algorithms for IoT security analysis. The authors in this work use four machine learning techniques to detect malicious traffic events: Random Forest, Random Tree, Decision Tree, Naive Bayes and BayesNet. They use the same dataset employed by us, but they do not integrate blockchain in their solution.

III. METHODOLOGY

This section presents an overview of the proposed Genetic Algorithm–based IoT Network Security framework. The overall architecture of the model is illustrated in Fig. 1 through a block diagram comprising key stages such as dataset preprocessing, feature dimensionality reduction, and model training. Each functional block is described in detail under separate subsections to clearly explain the workflow of the proposed approach.

Dataset Cleaning The input dataset contains a large number of features, not all of which contribute equally to intrusion detection. Therefore, an initial data cleaning step is performed to eliminate irrelevant and redundant attributes. In this process, non-informative fields such as session identifiers, connection types, and communication protocol details are removed, as they do not significantly influence attack classification. This refinement reduces data complexity and prepares the dataset for effective feature selection using the Genetic Algorithm.

CD‣Dataset_Cleaning(RD) ----------Eq. 1
In Equation (1), RD represents the raw dataset and CD denotes the cleaned dataset matrix. After preprocessing, the dataset is organized in a two-dimensional matrix where each row corresponds to

a network session and each column represents a specific feature associated with that session.

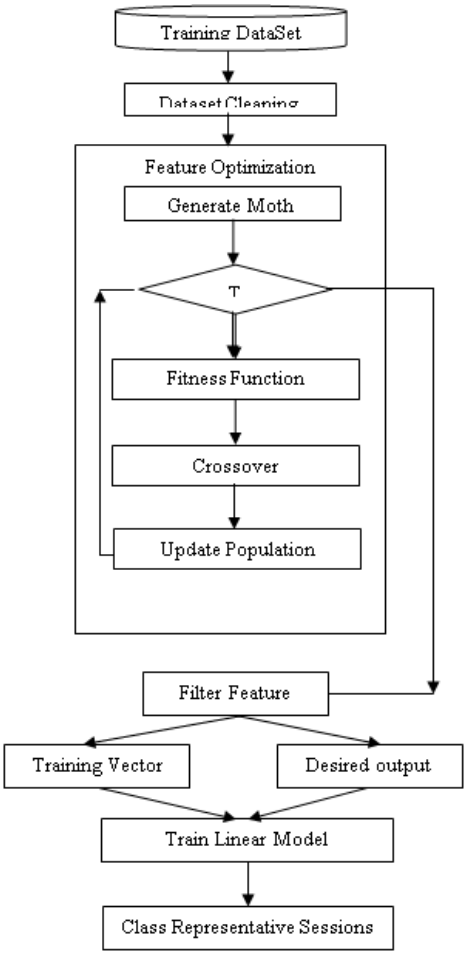

Fig. 1 Block diagram of MFOCMSD network intrusion detection.

Feature Optimization Using Genetic Algorithm

The cleaned dataset matrix (CD) is further processed using a Genetic Algorithm–based optimization strategy to minimize the dimensionality of the training feature set and enhance classification performance.

Genetic Algorithm Representation

In the proposed model, each chromosome represents a candidate solution corresponding to an optimized subset of features. The primary objective of the algorithm is to identify an optimal chromosome that maximizes classification accuracy while minimizing redundant features.

Population Initialization The initial population consists of multiple chromosomes, where each chromosome is represented as a binary vector of length n, equal to the number of features in the cleaned dataset. A value of 1 in the chromosome indicates that the corresponding feature is selected for training, whereas 0 signifies feature exclusion. If p chromosomes are generated, the population matrix M has dimensions p × n. The selection of f features within each chromosome is carried out using a Gaussian-based random initialization function.

M←Generate_Population(p,n,f)(Eq. 2)M \leftarrow Generate_Population(p, n, f) \quad \text{(Eq. 2)}M←Generate_Population(p,n,f)(Eq. 2)

Fitness Function Evaluation Each chromosome in the population is evaluated using a fitness function that measures its classification capability. The selected feature subset of a chromosome is used to construct training vectors, which are then fed into a K-Nearest Neighbour (KNN) classifier to identify representative clusters and compute detection accuracy. The obtained accuracy serves as the fitness value for the chromosome.

Input: Population matrix M, Clean dataset CD
Output: Fitness vector F

- For each chromosome w=1w = 1w=1 to WWW
- For each training session s=1s = 1s=1 to CDCDCD
- Generate training vector using selected features
- Assign desired output labels
- Train the KNN classifier
- Predict class labels for training sessions
- Increment fitness score for each correct prediction
- End loops

Here, TV represents the training vector and DO denotes the desired output.

Chromosome Update Strategy Once fitness values are computed, chromosomes are sorted in descending order based on their fitness scores. The best-performing chromosome is identified as the elite solution and retained for the next generation.

Crossover and Mutation To ensure diversity and avoid premature convergence, crossover and mutation operations are applied to non-elite chromosomes. A predefined number of feature positions are randomly altered, switching binary values from 0 to 1 or vice versa, guided by the elite chromosome. Newly generated offspring chromosomes are evaluated, and only those with improved fitness replace their parent chromosomes. This evolutionary process continues until the maximum number of iterations is reached.

Feature Selection Output After completion of all iterations, the chromosome with the highest fitness value is selected as the final optimal solution. Features corresponding to binary value 1 in the chromosome are retained as the optimized feature set, while remaining features are discarded. The selected features are then used to construct the final training and desired output matrices for classification.

KNN Based Cluster Representativ

The optimized feature set obtained from the Genetic Algorithm is further utilized to determine cluster representatives using the K-Nearest Neighbour (KNN) classification model.

In this approach, KNN analyzes the similarity between feature vectors by computing distance measures in the feature space. For each network session, its distance from the nearest cluster representative is evaluated to determine its class membership. The identified cluster representatives act as reference points that capture the intrinsic characteristics of normal and attack sessions. By comparing incoming session feature vectors with these representatives, the model effectively classifies sessions based on proximity, thereby improving detection accuracy and reducing misclassification. This representative-based learning strategy enhances the robustness and reliability of session-level intrusion identification.

IV. EXPERIMENT AND RESULTS

Experimental setup of proposed model and comparing model was developed on MATLAB software. Experimental machine having 4 GB ram, i3 6th generation processor. IO dataset was taken from [15]. Comparison of proposed model was done with cloud malicious session detection model proposed in [16].

Evaluation Parameter To test our results, this work usesthe following measures Precision, Recall, and F-score. These parameters are dependent on the TP (True Positive), TN True Negative), FP (False Positive), and FN (False Negative).

Results

The precision performance of IoT intrusion detection models was evaluated across varying dataset sizes, as presented in Table 1.

The results clearly indicate that the proposed model consistently outperforms the existing approach. On average, the proposed method achieves an improvement of approximately 5–6% in precision compared to the baseline model reported in [16]. This enhancement is primarily attributed to the effective feature reduction achieved through the Moth Flame Optimization–based Genetic Algorithm, which eliminates redundant attributes and enables more accurate clustering by the KNN classifier.

Table 1. Precision value based comparison of IOT network intrusion detection models.

Dataset Size	Existing Model	Proposed Model
5,000	0.9214	0.9865
10,000	0.9189	0.9827
15,000	0.9176	0.9819
20,000	0.9182	0.9808
25,000	0.9190	0.9796

Table 2. Recall value based comparison of IOT network intrusion detection models.

Dataset Size	Existing Model	Proposed Model
5,000	0.8457	0.9858
10,000	0.8423	0.9834
15,000	0.8436	0.9821
20,000	0.8441	0.9812
25,000	0.8462	0.9809

Table 2 presents the recall performance comparison between the existing and proposed intrusion detection models. The proposed approach demonstrates a significant increase in recall values across all dataset sizes, showing an average improvement of nearly 14–15% over the previous work. The improved recall indicates that the model is more effective in correctly identifying attack sessions. This gain is achieved through KNN-based learning on optimally selected features, which strengthens the model's ability to capture attack patterns.

Table 3. F-Measure value based comparison of IOT network intrusion detection models.

Dataset Size	Existing Model	Proposed Model
5,000	0.8831	0.9861
10,000	0.8796	0.9824
15,000	0.8802	0.9817
20,000	0.8810	0.9809
25,000	0.8824	0.9802

The F-measure, which represents the harmonic mean of precision and recall, is reported in Table 3. The proposed model achieves consistently higher F-measure values compared to the existing approach. These results confirm that the integration of the Moth Flame Optimization–based Genetic Algorithm for feature selection leads to a balanced improvement in both precision and recall, thereby enhancing the overall detection performance of the IoT intrusion detection system.

Table 4. Accuracy value based comparison of IOT network intrusion detection models.

Dataset Size	Existing Model	Proposed Model
5,000	0.7986	0.9739
10,000	0.7924	0.9668
15,000	0.7931	0.9655
20,000	0.7943	0.9641
25,000	0.7960	0.9629

The accuracy comparison results are summarized in Table 4. The proposed intrusion detection framework exhibits a notable accuracy improvement of approximately 6–7% over the existing model. This performance gain is largely due to the reduced feature space obtained through the optimization process, which enhances the discriminatory capability of the KNN classifier. As a result, the proposed model delivers more reliable and consistent classification of normal and malicious IoT network sessions across different dataset sizes.

V. CONCLUSION

The proposed framework processes an IoT intrusion dataset containing multiple network features, each carrying different levels of relevance for attack detection. To eliminate redundant and irrelevant attributes, a Moth Flame Optimization–based Genetic Algorithm is employed to divide features into selected and rejected groups. This optimization strategy reduces dimensionality and enhances the quality of the training data. The selected features are then used to identify representative feature patterns that serve as cluster centers for intrusion and non-intrusion classes using a K-Nearest Neighbour (KNN) approach. Classification is performed by measuring the distance between incoming sessions and these cluster representatives. Experimental results obtained on a real IoT dataset demonstrate that the proposed model achieves improved precision in intrusion detection compared to existing approaches. Future work may incorporate advanced learning models to further enhance detection accuracy and scalability.

REFERENCES

1. J. Deogirikar and A. Vidhate, "Security attacks in iot: A survey," International Conference on I-SMAC (I-SMAC), pp. 32–37, 2017.
2. T. Bodstrom and T. H ¨ am¨ al¨ ainen, "State of the art literature review ¨ on network anomaly detection with deep learning," Internet of Things, Smart Spaces, and Next Generation Networks and Systems, pp. 64–76, 2018.
3. I. Arnaldo, A. Cuesta-Infante, A. Arun, M. Lam, C. Bassias, and K. Veeramachaneni, "Learning representations for log data in cybersecurity," International Conference on Cyber Security Cryptography and Machine Learning, pp. 250–268, 2017.
4. M. Du, F. Li, G. Zheng, and V. Srikumar, "Deeplog: Anomaly detection and diagnosis from system logs through deep learning," Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 1285–1298, 2017.
5. B. J. Radford, B. D. Richardson, and S. E. Davis, "Sequence aggregation rules for anomaly detection in computer network traffic," arXiv preprint arXiv:1805.03735, 2018.
6. I. Lambert and M. Glenn, "Security analytics: Using deep learning to detect cyber attacks," 2017.
7. M. Stevanovic and J. M. Pedersen, "Detecting bots using multi-level traffic analysis." IJCSA, vol. 1, no. 1, pp. 182–209, 2016.
8. Zhang, Y.; Li, P.; Wang, X. Intrusion Detection for IoT Based on Improved Genetic Algorithm and Deep Belief Network. IEEE Access 2019, 7, 31711–31722.
9. Moustafa, N.; Turnbull, B.; Choo, K.R. An Ensemble Intrusion Detection Technique based on proposed Statistical Flow Features for Protecting Network Traffic of Internet of Things. IEEE Internet Things J. 2018, 6, 4815–4830.
10. Bagaa, M.; Taleb, T.; Bernal, J.; Skarmeta, A. A machine learning Security Framework for Iot Systems. IEEE Access 2020, 8, 114066–114077.
11. Susilo, B.; Sari, R. Intrusion Detection in IoT Networks Using Deep Learning Algorithm. Information 2020, 11, 279.
12. Liu, J.; Kantarci, B.; Adams, C. Machine Learning-Driven Intrusion Detection for Contiki-NG-Based IoT Networks Exposed to NSL-KDD Dataset. In Proceedings of the ACM Workshop on Wireless Security and Machine Learning, Linz, Austria, 13 July 2020.
13. Pokhrel, S.; Abbas, R.; Aryal, B. IoT Security: Botnet detection in IoT using Machine learning. arXiv 2021, arXiv:2104.02231.
14. Shafiq, M.; Tian, Z.; Sun, Y.; Du, X.; Guizani, M. Selection of effective machine learning algorithm and Bot-IoT attacks traffic identification for internet of things in smart city. Future Gener. Comput. Syst. 2020, 107, 433–442.
15. Ullah I., Mahmoud Q.H. (2020) A Scheme for Generating a Dataset for Anomalous Activity Detection in IoT Networks. In: Goutte C., Zhu X. (eds) Advances in Artificial Intelligence. Canadian AI 2020.
16. A. Fatani, M. Abd Elaziz, A. Dahou, M. A. A. Al-Qaness and S. Lu, "IoT Intrusion Detection System Using Deep Learning and Enhanced Transient Search Optimization," in IEEE Access, vol. 9.