IOT Network Malicious Session Detection by KNN and Moth Flame Optimization Algorithm

P.Ashwini [1], V.Chiranjeevi [2], S.Nagamani [3]

[1]Assistant Professor, Swarna Bharathi Institute of Science & Technology, India, E-mail: ashwini.podila@gmail.com.
[2]Assistant Professor, Swarna Bharathi Institute of Science & Technology, India, E-mail: chiru508@gmail.com.
[3]Assistant Professor, Swarna Bharathi Institute of Science & Technology, India, E-mail: nagamanikunchipudi@gmail.com.

Abstract-

The Internet of Things (IoT) network improves people's quality of life in many ways. Nowadays, the security of IoT devices is a big worry since this expanding network attracts a lot of hackers that want to attack the system. An intrusion detection system for IoT networks that can distinguish between attack and regular sessions is presented in this research. The study used a moth flame optimization genetic algorithm to choose characteristics for determining which sessions were typical of the class. K-Nearest Neighbor was used to identify the class session. Results from an experiment on a real-world dataset demonstrate that the suggested model, Moth Flame based IOT Network Security (MFIOTNS), significantly boosts productivity by fine-tuning a number of evalautin parameters.

Key Words: KNN, Clustering, Genetic Algorithm, Intrusion Detection.

I. INTRODUCTION

As information technology becomes more embedded in people's everyday lives, concerns about privacy and network security are growing globally. Computer security is becoming a must. Attacks on computer networks and systems have recently increased in tandem with the proliferation of new technologies like the Internet of Things (IoT) and the number of apps hosted on the Internet. A network of interconnected computing devices that may establish connections autonomously is known as the Internet of Things (IoT). A great deal of equipment in various fields, including medicine, agriculture, transportation, and more, may be linked to the web via the Internet of Things (IoT) [1]. This includes coffee makers, lights, bicycles, and countless more. Our work and lifestyles are being transformed by IoT applications that save time and resources. Additionally, it offers limitless benefits and opens up a plethora of chances for development, innovation, and the sharing of information. Since the Internet lies at the very heart of the Internet of Things (IoT), every security risk that exists on the Internet also exists on the IoT. Internet of Things (IoT) nodes lack human controls, have limited resources, and have poor capacity in comparison to other conventional networks. Security challenges with the Internet of Things (IoT) are becoming more pressing as their use becomes more pervasive in people's everyday lives, necessitating the creation of network-based security solutions. The detection of some assaults is still a challenge for existing systems, despite their effective performance against some.

There is no question that there is room for more innovative ways to enhance network security, and the need for quicker and more efficient attack detection techniques is growing in tandem with the exponential growth in the volume of data stored in networks [2]. Here, Machine Learning (ML) stands up as a top computational paradigm for delivering embedded intelligence in the IoT setting.

Network traffic analysis, intrusion detection, botnet identification, and other network security activities have all made use of machine learning algorithms [3, 4, 5, 6]. An essential component of any Internet of Things (IoT) solution is machine learning, which may be defined as the capacity of an intelligent device to automate or alter a state or behavior based on knowledge. Machine learning (ML) is useful for tasks like classification and regression because it can infer useful information from data produced by devices or people. Similarly, ML may be used to provide security services in an IoT network. There is a growing interest in using machine learning to attack detection issues, and ML is finding more and more uses throughout the cybersecurity industry. While several studies have used ML approaches to identify effective threat detection methods, there is a dearth of research about detection methods that are well-suited to Internet of Things (IoT) settings. Two primary forms of cyber-analysis—signature-based (also known as misuse-based) and anomaly-based—can be used to apply machine learning to the attack detection process. By analyzing known attacks for certain patterns of traffic (also called "signatures"), signature-based approaches may identify them. This kind of detection method has the benefit of

efficiently detecting all known assaults without producing an excessive amount of false alarms.

II. RELATED WORK

In their proposal for an intrusion detection model, the authors of [13] use a deep belief network and a genetic algorithm. The four attack types—DoS, R2L, Probe, and U2R—are detected using the NSL-KDD dataset. When compared to our work, this study makes use of an outdated dataset that isn't relevant to current IoT networks and doesn't use blockchain as an integrated method for IIoT network monitoring and security. The article suggests a method for securing the system against intrusions using statistical flow characteristics [14]. data transfer over the network for IoT applications. Decision Tree, Naive Bayes, and Artificial Neural Network (ANN) are the three machine learning approaches used by the authors of this study to identify fraudulent traffic occurrences. While they do make use of the UNSWNB15 dataset, which we also utilize, they fail to include blockchain technology into their system as a means of integrating IIoT network monitoring and security. In [15], the author suggests a framework for the security of IoT systems using machine learning. They tested their idea in a real-life smart building situation after creating a dataset using the NSL-KDD dataset as a foundation. An outdated dataset could not be appropriate for contemporary IoT networks, as mentioned in earlier similar publications. To identify DDoS, Probe, U2R, and R2L assaults, they use a one-class Support Vector Machine (SVM) method. But they aren't managing IIoT networks using blockchain technology.

Using a deep-learning algorithm, the authors of [16] created a system that can identify DoS assaults. To identify denial-of-service assaults, they use a Convolutional Neural Network, a Multilayer Perceptron, and Random Forests. They use the same information as we do, but their solution does not include blockchain technology and focuses only on detecting denial-of-service attacks.

The authors of [20] provide a strategy for detecting and mitigating botnet-based distributed denial of service (DDoS) assaults in IoT networks by using a machine learning algorithm. K-Nearest Neighbor (KNN), the Naive Bayes model, and Multi-layer Perception Artificial Neural Network (MLP ANN) are among the machine learning methods that are used. They use the same information as we do, but their approach does not include blockchain technology and focuses only on detecting a single attack (DoS). One approach to analyzing the security of the Internet of Things (IoT) is the model proposed in [21], which uses Machine Learning (ML) techniques to detect intrusions and cyberattacks. In order to identify harmful traffic occurrences, the writers of this study use four machine learning techniques: BayesNet, Naive Bayes, Decision Tree, Random Forest, and Random Tree. They use the same information as we do, but their solution does not include blockchain technology.

III.METHODOLOGY

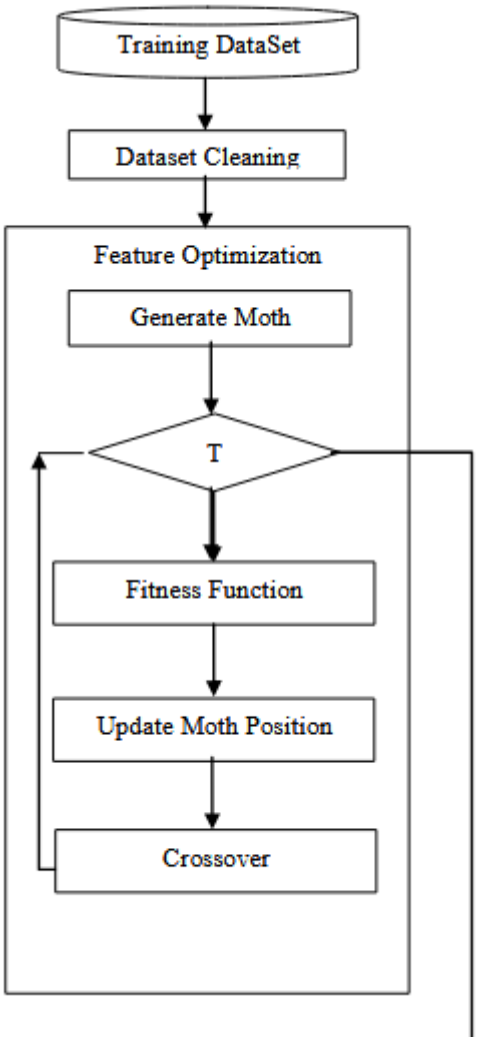
A concise description of the Moth Flame based Internet of Things Network Security (MFIOTNS) is given in this section. Processing datasets, reducing dimensions, and training blocks are shown in Fig. 1 of the proposed model's block design. Here, under several titles, you'll find an explanation of each block.

1. Cleaning the dataset:

Input data contains several characteristics. and With its own unique significance, this process purges the data collection of any extraneous information.For example,

The input dataset used in this study has n fields. The initial feature values, such as session ID, connection type, transferring protocol, etc., were eliminated from the analysis.

Dataset entries may be deleted.



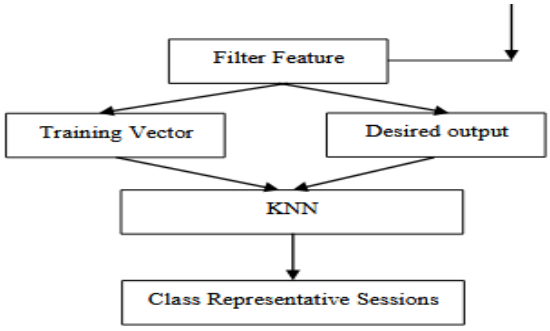


Fig. 1 Block diagram of MFOCMSD network intrusion detection.

CD←Dataset_Cleaning(RD) ----------Eq. 1

Raw dataset (RD) and clean dataset (CD) are the matrices in equation 1.

Each session is represented by a row in the processed dataset, and each column contains the collection of features for that session.

2.Feature optimization

To improve learning accuracy and decrease training vector values, the Moth Flame Optimization Algorithm was applied to the input CD matrix.

Moth Flame Optimization Algorithm, Third Edition: This paper's approach treats each chromosome as a moth.

Finding a moth flame on its way to the moon was the goal of this method. The work's chromosomes are moth flames.

Fourth, make moth flames.

A chromosome is one potential solution to an optimum set of features, while moth flames are groups of chromosomes.

It follows that a Moth Flame is an n-element vector, where n is the number of columns in the CD. In the Moth Flame vector, every element is a binary value. A feature is considered for training if its value is one, and not picked for the population if its value is zero. If there are p moth flames, then the moth flame population matrix M has pxn dimensions. The Gaussian random value generator function is used to choose f number of features in the vector.

M←Generate_Moth Flame(p, n, f)----------Eq. 2

5.Workout Routine

Distance was used to rate each moth flame. The fitness value is used for the assessment of distance. The Moth Flame feature vector is fed into the KNN (K-Nearest Neighbor) training vector in order to locate cluster representatives and test the work's detection accuracy [11]. The work's distance parameter is this detection accuracy number.

Input: M, CD
Output: F

1. Loop w=1:W // for w Moth Flames
2. Loop s=1:CD // for s training session
3. TV[s]←Training_Vector(W[w], CD[s])
4. DO[s]←Desired_Output(W[w], CD[s])
5. EndLoop
6. TNN←Train_Neural_Network(TV, DO)
7. Loop s=1:CD // for s training session
8. TV←Training_Vector(W[w], CD[s])
9. O←Predict(TV,TNN)
10. If DO[s] equals O
11. F[w]←Increment F by 1
12. EndIf
13. EndLoop
14. Endloop

In above algorithm TV is training vector, DO is desired output.

6. change the location of the moth flame Sort f in decreasing order according on the F value obtained by the fitness function. Then, choose the best Moth Flame chromosome from the population.

Because chromosomal changes are essential to the Crossover Genetic method, we adjusted the number of Moth Flames' random positions according on the value of parameter X. The finest local moth flame was not used for this procedure. Here, according to the finest local Moth Flame feature set, we randomly changed the X-number of places for each Moth Flame from zero to one or one to zero. If the offspring Moth Flame performed better on the path distance test, the parent Moth Flame would be kept; otherwise, the offspring would be eliminated.

Proceed to the filter feature block if you've reached the maximum iteration step; otherwise, assess the fitness value of every Moth Flame Moth Flame.

8. A Filtering Function

Locate the optimal Moth Flame from the most recent population update when iteration is finished. A feature is considered chosen for the training vector if it has a value of one in the chromosome, and it is considered unselected otherwise. In this part, we also created the desired output matrix.

Clustering Modeled on K-Nearest Neighbors

The KNN model was utilized to locate the cluster representative using the feature set generated from the aforementioned technique.

Locating such a representative efeature set is useful for determining the session class using a distance vector.

IV. EXPERIMENT AND RESULT

To set up the experiment, we used MATLAB to create an MFOCMSD and a comparison model.

Experimental PC with an i3 6th gen CPU and 4 GB of RAM. The IO dataset was sourced from ([15]). The cloud malicious session detection methodology suggested in [16] was compared with MFIOTNS. **1. Measurement Criteria** The accuracy, recall, and F-score are the metrics used to evaluate our findings in this study. The True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) values determine these parameters.

V.RESULTS

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

Dataset Size	Previsous Work	MFIOTNS
5000	0.9359	0.987
10000	0.9322	0.9814
15000	0.9312	0.9812
20000	0.9322	0.9806
25000	0.9323	0.9793

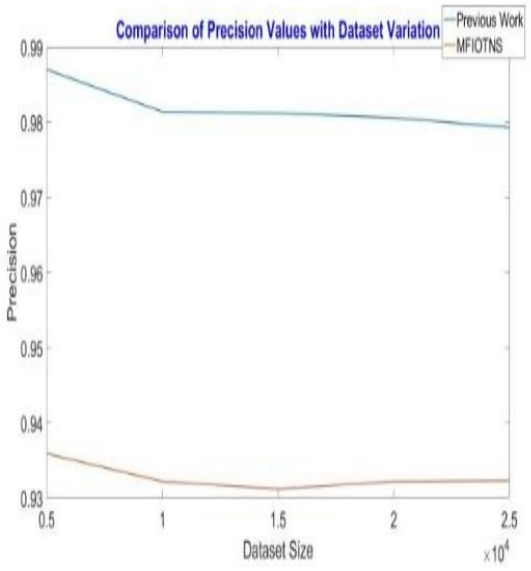


Fig. 2 Precision value based comparison.

Results from comparing IOT network intrusion detection models on datasets of varying sizes reveal that the prospeod model outperforms the prior model by 5.004% [16]. Because the clustering of the KNN model was enhanced with less features, the precision value was discovered to have increased by using the moth flame feature optimization approach in the suggested model.

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

Dataset Size	Previsous Work	MFIOTNS
5000	0.8623	0.9862
10000	0.8568	0.9838
15000	0.8582	0.9825
20000	0.8586	0.9816
25000	0.8606	0.9816

On table 2 you can see the results of the comparison of the recall value parameters.

That was found that the suggested model enhanced the IoT compared to the values derived from the prior model in [16], the intrusion detection recall parameter increased by %. KNN using feature-based learning, which has enhanced the recall detection.

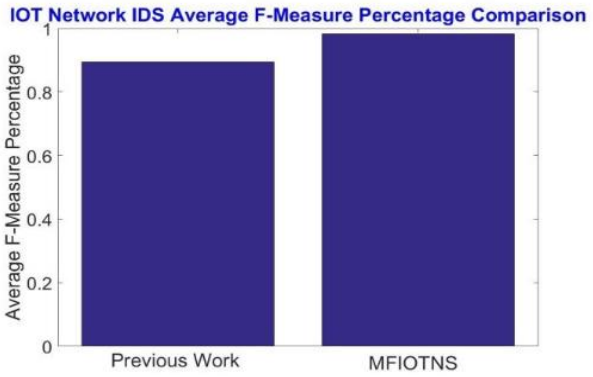


Fig. 3 F-measure value based comparison.

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

Dataset Size	Previsous Work	MFIOTNS
5000	0.8976	0.9866
10000	0.8929	0.9826
15000	0.8932	0.9819
20000	0.8939	0.9811
25000	0.895	0.9805

The f-measure parameter is the inverse of the recall and accuracy squared. The f-measure values of IOT network intrusion detection have been improved with the application of moth flame optimization genetic algorithm for feature selection, as shown in Table 3.

Table 4. Accuracy value based comparison of IOT

network intrusion detection models.

Dataset Size	Previsous Work	MFIOTNS
5000	0.8148	0.9748
10000	0.8074	0.9673
15000	0.8079	0.966
20000	0.8090	0.9646
25000	0.8109	0.9634

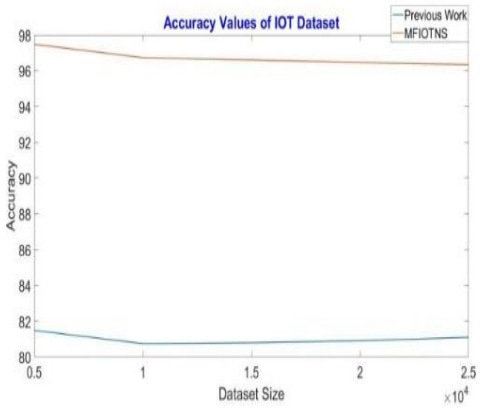


Fig. 4 Average accuracy value based comparison.

VI.CONCLUSION

Organizations, hotels, and small businesses all benefit greatly from IoT networks. However, several types of attacks may exploit its lack of security mechanisms. An approach to such a network's intrusion detection is presented in this article. Since the input dataset contains a collection of features that include a set of vulnerabilities. The characteristics are grouped into two categories: chosen and rejected, according to the moth flame optimization process.

Using the KNN method, we were able to identify the feature values that serve as the cluster centers for intrusion and non-intrusion class detection based on the features that were chosen.

The suggested model outperformed the state-of-the-art models in terms of intrusion detection accuracy on the IoT dataset by a significant margin. To improve the work's detection accuracy in the future, researchers may utilize a different training model.

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 preprintarXiv: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 proposedStatistical 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. 21. 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 ActivityDetection in IoT Networks. In: Goutte C., Zhu X. (eds) Advances in Artificial Intelligence. Canadian AI 2020.