Fuzzy Systems and Soft Computing
ISSN : 1819-4362

A FRAMEWORK FOR IDS IN IOT USING AN IMPROVED MULTI-OBJECT CAPUCHIN SEARCH ALGORITHM

Dr. A. Jayasudha Assistant Professor, Department of Computer Applications and Technology, SRM Arts and Science College, Tamil Nadu, India jsudhaannamalai@gmail.com

Abstract

Background: Security for these interconnected systems has become more critical as the number of devices connected to the Internet of Things (IoT) has increased. This research adds substantially to the current conversation about effective cybersecurity defences by addressing new risks in the ever-changing Internet of Things (IoT) environment. **Methods**: An improved Multi-Object Capuchin Search Algorithm (MOCSA), demonstrates a new approach to building an Intrusion Detection System (IDS) for the Internet of Things (IoT). The study highlights the effectiveness of MOCSA in detecting anomalies, especially when used with the NSL-KDD and TONNE-IoT datasets. **Findings**: The exceptional results achieved by MOCSA, especially on the NSL-KDD and TONNE-IoT datasets. The results of the studies were encouraging; MOCSA achieved a higher accuracy of 0.9013. The Random Forest (RF) method for classification, together with MOCSA's efficient feature selection and extraction operations, is responsible for this remarkable accuracy. When looking at metrics like recall, accuracy, precision, and F1-score, it is clear that MOCSA perform exceptionally well in both the training and testing stages. **Novelty and applications**: The paper delves deep into the efficacy of MOCSA, highlighting their adaptability and resilience across different assessment criteria. This study demonstrates a new approach to building an Intrusion Detection System (IDS) for the Internet of Things (IoT).

Keywords:
Intrusion Detection System (IDS), Internet of Things (IoT), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Multi-Object Capuchin Search Algorithm (MOCSA).

1. Introduction

Transportation, automated homes, medical care, production, warehouse management, safety, and Blockchain monitoring are just a few of the many industries that have been greatly affected by the new computer era brought about by the Internet of Things (IoT). The versatility of IoT devices is evident in their diverse applications, and it is projected that their number will reach 30.9 billion by 2025, indicating a remarkable growth rate exceeding 55% from the 13.8 billion devices in use in 2021, as forecasted [10]. This rapid expansion has attracted attention from researchers, inventors, entrepreneurs, and cybercriminals alike, all recognizing the immense commercial potential and reliance on IoT devices. The market demand for these devices is substantial due to their utility, prompting increased interest from prospective investors. Entrepreneurs and innovators are continually enhancing the appeal of the field by introducing novel applications that improve and simplify daily life [14].

However, the surge in IoT adoption has also created opportunities for cybercriminals to exploit security vulnerabilities in these devices. The exponential increase in cyberattacks on IoT devices is attributed to the sector's rapid growth and potential financial gains, as highlighted in research by various sources [15]. R. Williams et al.'s research reveals vulnerabilities in many fully functional IoT devices [16]. The broader threat landscape surrounding IoT devices is exacerbated by their common integration with other systems and appliances. This connectivity often serves as an entry point for malicious actors seeking access to associated resources. A notable example is the 2016 Dyn attack, where imaging IoT devices caused an internet outage affecting major platforms like Amazon, Twitter, and Netflix [11].

Most Internet of Things (IoT) gadgets and applications have inadequate security in their architecture. The creation of malware networks and the distribution of various forms of harmful software are both made possible by hacker groups taking advantage of this weakness. Due to their inherent complexity

and restricted funds, typical techniques for identifying intrusions are not suitable for protecting IoT systems [12]. Within intelligent platforms, which predominantly utilise Wi-Fi networks to connect IoT devices, there is a critical requirement for heightened security in wireless transmission [1].
Certain devices lack crucial hardware security support, which results in their inability to detect cyberattacks. As a consequence, IoT devices become inefficient at defending against advanced threats [2]. The security issue in the IoT stems from the wide range of gadgets and detectors, wireless connections, insufficient device security design, limitations in resources, and the intrinsic intricacy of the IoT. The ability to communicate and share data inside systems powered by the IoT relies heavily on protocol routing. [13]. One well-known IPv6 networking method is the RPL protocol, which is ideal for low-power, low-loss networks.
However, this protocol encounters difficulties in dealing with advanced cyber threats [8]. The RPL system has several vulnerabilities that make it susceptible to attacks at the communication layer, including insufficient processing power, power reserve capacity, and targeted security frameworks [3]. The Sybil assault exemplifies how these networks are particularly prone to DoS attacks. The Sybil attack changes the way DODAG Information Objects (DIOs) are sent in RPL by breaking implicit assumptions on purpose, spreading fake identity information, and rendering core nodes useless while flooding the network with DIOs [4]. Aside from IDS, there are various countermeasures to address cyber-attacks on IoT devices [9].
The monitor-based technique entails the observation and recording of network traffic and communication activities for the purpose of detecting potential routing faults. In a given instance [5], individual nodes gauge the frequency of lost packets by monitoring the network traffic of adjacent nodes in both the downstream and upstream directions. The objective is to detect instances of selectively forwarding threats in wireless mesh networks (WMNs). These nodes across the forwarded path that act as intermediaries communicate with the source node in an acknowledgment-based mechanism to confirm packet receipt or indicate improper routing behaviour.
Many academics have utilised meta-heuristic techniques to tackle the problem of decreasing features in datasets with a high number of dimensions. A novel approach to anomaly detection in the IoT was presented in [6]. Their plan is to combine the random forest technique with two more optimisation algorithms, GWO and PSO. Although machine learning algorithms have achieved success in detecting anomalies, their effectiveness has been slightly hindered by the growing number of features and the volume of data. In contrast, multiple research suggest that deep learning approaches surpass machine learning methods, especially when applied to large datasets [7]. IDS for the IoT, also known as IoT, can greatly benefit from deep learning approaches because of the massive amounts of data generated by diverse devices with complicated properties.
In this study, a more efficient method of identifying abnormalities in the IoT has been introduced with an improved version that uses the MOCSA methodology. It demonstrating an improved MOCSA algorithm for feature selection: a binary multi-objective approach that takes many criteria into account to improve anomaly detection accuracy.

2. Improved MOCSA

2.1 Architecture

We have created a system for detecting anomalies that classifies IoT abnormalities according to the kind of attacks they represent. The system makes use of a convolutional neural network in conjunction with an improved multi-objective CSA. We created a CNN with hybrid levels dubbed the IoT Feature Mining CNN to enhance the detection of oddities in the IoT. Both global and local properties are efficiently extracted by this network. We introduced quantitative MOCSA technique for choosing features is an enhanced multifaceted binary variant of the Capuchin method. We have created a novel hybrid approach, MOCSA, which integrates CNN_IDS and MOCSA to blend these approaches. Figure 1 depicts the four-stage design of the Internet of Things anomaly detection system. A MOCSA and a CNN form its basis.

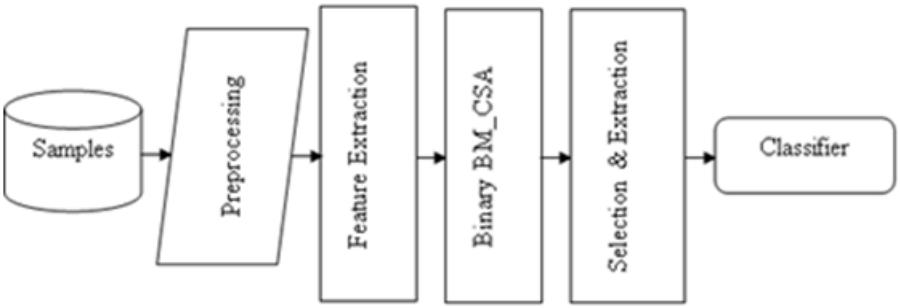

Figure 1: Architecture

Table 1: NSL-KDD and TONNE-IoT data set classifications

Datasets	Traffics	Training	Testing
NSL-KDD	Non attacked Data	75434	102322
	Attacked Data	64345	23564
TONNE-IoT	Non attacked Data	500000	
	Attacked Data	534765	

Two datasets, NSL-KDD and TONNE-IoT, are used to exanimated the suggested technique. Below, we outline the characteristics of each dataset, which are organised into two types, and each dataset contains samples. There are two parts to the NSL-KDD dataset, KDD Train and KDD Test, which together contain 139,779 samples. Table 1 details the two classes that make up the NSL-KDD dataset: normal samples and abnormal samples. The features that make up dataset number 41.

In contrast, the TONNE-IoT dataset includes operating systems and network traffic, and it is a next-generation collection of IoT/IIoT statistics. There are 1034765 records in this collection, including both normal and attack data. To be more precise, 461,043 records were culled from IoT traffic, with 534765 records representing attack traffic and 500,000 records representing regular traffic. Also, the TONNE-IoT dataset contains 88 attributes and includes 16 categories of attacks, including normal and aberrant samples.

2.3 Pre-processing

The main techniques used to prepare the NSL-KDD and TONNE-IoT datasets for analysis. Data cleaning is the first step in pre-processing; it entails removing any noisy data found while sampling the dataset. The second part of the pre-processing stage is filling in missing data with suitable alternatives. For numerical features, the average value is used for replenishment, whereas for string or batch data, the most frequent technique is used. The third stage is to use the label-encoder method to transform string or batch characteristics into their respective numerical kinds. In the last stage, data normalisation is addressed. All features are adjusted to fall within the range of [0, 1] using the Min-Max normalisation approach, as shown in Equation 1:

$$S_{nm} = \frac{S - S_{min}}{S_{max} - S_{min}} \qquad (1)$$

Each feature's lowest value, S_{min}, is represented by the variable *S*, while the highest value, S_{max}, is denoted by the variable *S*. Both datasets are prepared to be used with ML and DL techniques after the preparation procedures are finished.

2.2 Feature Extraction

2.2.1 Configuration

The first step in executing instantaneous feature extraction is setting up the network. A Linux station with two interfaces for networks has been installed across the router that serves as the LAN to monitor packets of information as they go in and out of the network. This was accomplished by effectively bridging the two network ports on the Linux computer, which allowed it to intercept data

packets going in and out of the workstation. Here, the C++ programming language serves as the foundation for the packet sniffing method.

2.2.2 Feature Extraction

The feature extraction process made use of the well-known and resource-efficient packet sniffing technology. In a network with two or more interfaces, this strategy can be applied to any machine. The LIBPCAP package and the C++ programming language were used to code the packet-capturing system. Key to this process is the LIBPCAP package, which offers an API for collecting data from networks at an advanced level. Using a text buffer, which is the programming language of C++ software, captures the packets. Not only that, but ICMP, TCP, and UDP packets were all taken care of by a newly-made packet analysis function.

2.4 Methodology

In order to address the issue of picking out features utilizing variables that are generated in a binary format, a binary-based, multifaceted, improved CSA known as MOCSA was developed. Some changes have been made to the recommended MOCSA method: (1) it now uses documents for storing Pareto remedies; (2) the binary variant incorporates binary and inherited employees; (3) Levy's flight is added to boost CSA outcomes and investigation space; (4) the algorithm's efficiency is increased by deploying the best population size inside the repository for multiple objectives mode; and (5) evaluating and the roulette drive choices are used to choose the most effective solution within the preservation.

Algorithm A: Improved Multi-Object Capuchin Search Algorithm (MOCSA)

Initialize parameters.
Initialize binary velocity and positions
Calculate fitness for each capuchin.
Create a grid, grid indices, and save Pareto to CapSAArchiv.
Select leaders based on CapSAArchive
while (iteration < MaxIterations) do
Update parameter n
Update velocity using leaders Pareto-optimal
for each capuchin (k from 1 to target) do
if (k <= target/2) then
if (random value >= 0.1) then
if (random event P occurs) then
if (random value ≤ 0.2) then
Update position of leaders leaping on trees
else if (0.2 < random value < 0.30) then
Update position of leaders using Levy flight
else
Update position of leaders walking on the ground
end if
else if (0.5 < random value < 0.75) then
Update position of leaders swinging on tree branches
else if (0.75 < random value < 1.0) then
Update position of leaders climbing on trees
end if
else
Update position of leaders using t and mutation
end if
else
Update position of followers
end if
end for

32. Calculate fitness of each solution.
33. Add Non-Dominated capuchin to CapSAArchive.
34. Update CapSAArchiv using new Pareto-optimal
35. Select leaders based on CapSAArchive
36. Update Grid and Grid Indices.
37. end while
38. The final optimal solution is the location of capuchins.

2.5 Storing in the archive

In order to optimize for multiple objectives simultaneously, it is necessary to establish an archive for storing statistically optimal outcomes. To save Pareto optimum answers, the MOCSA algorithm proposes a special archive named MOCSA_Archive. Grading is used to describe the objective function space in MOCSA_Archive, with non-dominated solutions included along with feature count and classifier error rate.

2.6 Updating velocity with elitism

Changing the velocities of all population solutions is the second phase of the MOCSA algorithm. When the objective function is singular and the best solution xbest can be found using a sort operator, the usual technique uses Equation (2) for this purpose. It becomes much more difficult to determine the optimal solution when there are numerous unique objectives, especially in a non-dominated solution scenario. To solve this problem, we search Cap-SA Archive for all the non-dominated solutions and pick the best one. The selection process is usually conducted at random using methods such as the roulette wheel or based on grades. Feature selection takes a back seat to improving accuracy, which is why we prioritise reducing the categorization error rate in our work. When feature selection either increases accuracy or, if nothing changes, does not significantly decrease, then it is meaningful.

3. Results and Discussion

The investigation of the proposed methods has proven that MOCSA can classify and extract features from anomaly detection datasets. As a reliable method for anomaly detection, MOCSA has proven its worth.

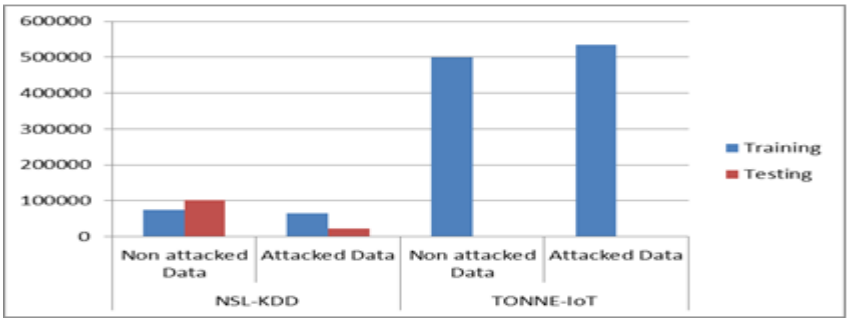

Figure 2. Comparison of dataset volumes

Figure 2 displays a summary of the results achieved MOCSA methodology to the NSL-KDD and TONNE-IoT sample datasets.

As seen in the curves that follow, the following metrics were assessed as part of the DNN model's effectiveness evaluation: recall, precision, accuracy, the F1-s, and the confusion matrix (CM). Performance analysis in the presence of skewed class distributions is when these measures really shine. The formulas for recall, accuracy, precision, and F1-score are given in Equations 2 to 5 below.

$$\text{Accuracy} = \frac{TP+TN}{FP+TP+TN+FN} \quad (2)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (3)$$

$$\text{Recall} = \frac{TP}{FN+FP} \quad (4)$$

$$\text{F1} = 2 * \frac{Precision*Recall}{Precision+Recall} \quad (5)$$

Table 2. Performance comparison of proposed and existing methods.

Techniques	Accuracy	Precision	Recall	F1-score
XGBoost-SMOTE	0.9243	0.9800	0.8970	0.92370
SVM	0.8763	0.9324	0.7564	0.8376
K-Means	0.8543	0.9345	0.6453	0.7864
NB	0.7413	0.6098	0.7144	0.6577
OCSVM	0.7354	0.9543	0.6543	0.7864
MOCSA	0.9013	0.9931	0.8632	0.9023

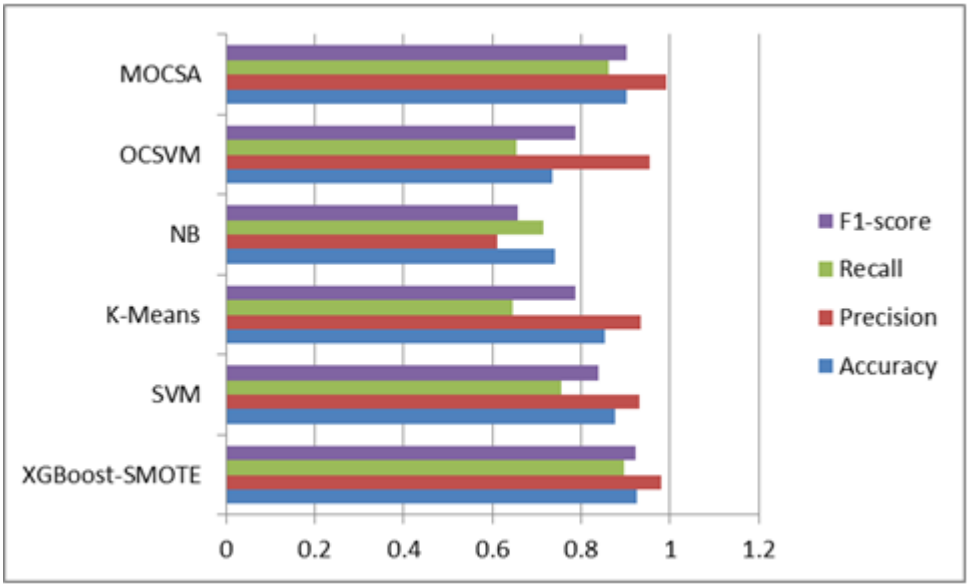

Figure 3. Performance comparison of proposed and existing methods

On the datasets NSL-KDD and TONNE-IoT the MOCSA techniques, according to their proposals, demonstrated rather strong performance with accuracies of 0.9013, respectively. Because of its efficient feature selection, feature extraction, and RF algorithm-based classification, MOCSA stood out with an impressive accuracy of 0.9013. All factors analysed highlight MOCSA's persistent brilliance, as seen in the entire comparison. When the F1-score was higher, both techniques showed remarkable outcomes, but they performed well across the board. The accuracy, precision, recall, and F1-score for the training and testing processes with the NSL-KDD and TONNE-IoT datasets are shown in Table 2.

The suggested MOCSA stood out from the rest of the methods when looking at Table 2, Figure 3, outperforming them all in terms of F1-score, recall, accuracy, and precision. On the other hand, some algorithms had poor recall values, which led to more false notifications.

4. Conclusion

A lot of research has been done on the suggested methods, especially on MOCSA, and it is clear that they work for sorting and extracting features from datasets for anomaly detection. The study showcases MOCSA's outstanding performance on the NSL-KDD and TONNE-IoT datasets, revealing it as a standout and trustworthy solution for anomaly detection. The experimental results showed that the MOCSA procedure performed well, with MOCSA reaching 0.9013, in line with their respective methodologies. The remarkable accuracy of 0.9013 achieved by MOCSA is largely attributable to its implementation of the RF (random forest) method for classification as well as its efficient feature selection and feature extraction processes. Over the course of the comparison, it becomes clear that all of these factors work together to make MOCSA so brilliant. By looking at several measures like recall, accuracy, precision, and F1-score, we can see that MOCSA did a great job. An in-depth analysis of these methods' consistent and impressive successes is shown in Table 2. This analysis was carried out during the training and testing phases using the NSL-KDD and TONNE-IoT datasets. For the NSL-KDD and TONNE-IoT datasets in particular, the results show that MOCSA is effective in anomaly detection. Not only are these approaches quite accurate, but

they also perform well across a variety of criteria, suggesting they could be useful tools for anomaly recognition and intrusion detection. The research adds significantly to the continuing discussion on efficient methods for strengthening cybersecurity defences in response to new and changing threats.

5. References

1. Apostolos Gerodimos, Leandros Maglaras, Mohamed Amine Ferrag, Nick Ayres, Ioanna Kantzavelou, IoT: Communication protocols and security threats, Internet of Things and Cyber-Physical Systems, Volume 3, 2023;1-13, ISSN 2667-3452. Available from: https://doi.org/10.1016/j.iotcps.2022.12.003.
(https://www.sciencedirect.com/science/article/pii/S2667345222000293).
2. Mukhtar, B.I.; Elsayed, M.S.; Jurcut, A.D.; Azer, M.A. IoT Vulnerabilities and Attacks: SILEX Malware Case Study. Symmetry, 2023. Available from: https://doi.org/10.3390/sym15111978
3. Hussain, Muhammad Zunnurain, and Zurina Mohd Hanapi, "Efficient Secure Routing Mechanisms for the Low-Powered IoT Network: A Literature Review" Electronics 2023;12, 3: 482. Available from: https://doi.org/10.3390/electronics12030482
4. A. K. Mishra, D. Puthal and A. K. Tripathy, "A Secure RPL Rank Computation and Distribution Mechanism for Preventing Sinkhole Attack in IoT-based Systems," IEEE INFOCOM 2023 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Hoboken, NJ, USA, 2023;1-6, Available from: https://doi: 10.1109/INFOCOMWKSHPS57453.2023.10225895
5. Alazab, A.; Khraisat, A.; Singh, S.; Bevinakoppa, S.; Mahdi, O.A. Routing Attacks Detection in 6LoWPAN-Based Internet of Things. Electronics, 2023;12;1320. Available from: https://doi.org/10.3390/electronics12061320
6. Amiri, Z., Heidari, A., Navimipour, N.J. et al. Adventures in data analysis: a systematic review of Deep Learning techniques for pattern recognition in cyber-physical-social systems. Multimedia Tools Applications;22909-22973, 2023. Available from: https://doi.org/10.1007/s11042-023-16382-x
7. Bhavsar, M., Roy, K., Kelly, J. et al. Anomaly-based intrusion detection system for IoT application. Discov Internet Things, 2023;3;5. Available from: https://doi.org/10.1007/s43926-023-00034-5
8. Abed AK, Anupam A. Review of security issues in Internet of Things and artificial intelligence-driven solutions. Security and Privacy. 2023;6(3):e285. Available from: https://doi:10.1002/spy2.285.
9. Xiaoya Xu, Yunpeng Wang, Pengcheng Wang, "Comprehensive Review on Misbehavior Detection for Vehicular Ad Hoc Networks", Journal of Advanced Transportation, vol. 2022, Article ID 4725805, 27 pages, 2022. Available from: https://doi.org/10.1155/2022/4725805
10. O. Altay, Chaotic slime Mould optimization algorithm for global optimization, Artif. Intell. Rev. 55(5), 2022;3979:4040. Available from: https://doi.org/10.1007/s10462-021-10100-5
11. C. Bulla, M.N. Birje, Anomaly detection in industrial IoT applications using deep learning approach, in: Artificial Intelligence in Industrial Applications, Springer, 2022;127:147. Available from: http://dx.doi.org/10.1007/978-3-030-85383-9_9
12. T. Saba, A. Rehman, T. Sadad, H. Kolivand, S.A. Bahaj, Anomaly-based intru-sion detection system for IoT networks through deep learning model, Comput. Electr. Eng. 99 2022;107810. Available from: https://doi.org/10.1016/j.compeleceng.2022.107810
13. F. Hussain, R. Hussain, S. A. Hassan and E. Hossain, "Machine Learning in IoT Security: Current Solutions and Future Challenges," in IEEE Communications Surveys & Tutorials, 2022;22,1686:1721. Available from: https://doi: 10.1109/COMST.2020.2986444
14. N. Balakrishnan, A. Rajendran, D. Pelusi, V. Ponnusamy, Deep Belief Network enhanced intrusion detection system to prevent security breach in the Internet of Things, Int. Things 14, 2021;100112. Available from: https://doi.org/10.1016/j.iot.2019.100112
15. M. Braik, A hybrid multi-gene genetic programming with capuchin search algorithm for modeling a nonlinear challenge problem: modeling industrial winding process, case study, Neural Process. Lett. 53(4), 2021;2873:2916. Available from: https://doi.org/10.1007/s11063-021-10530-w

16. V. Thamilarasi, P. K. Naik, I. Sharma, V. Porkodi, M. Sivaram and M. Lawanyashri, "Quantum Computing - Navigating the Frontier with Shor's Algorithm and Quantum Cryptography," 2024 International Conference on Trends in Quantum Computing and Emerging Business Technologies, Pune, India, 2024, pp. 1-5, doi: 10.1109/TQCEBT59414.2024.10545283.
17. L. Breiman, Random Forests, Machine Learning, 45(1) 2001;5-32. Available from: https://doi.org/10.1023/A:1010933404324