

Artificial Intelligence (AI) Based Internet-of-Things (IoT)-Botnet Attacks Identification Techniques to Enhance Cyber security

Mani Gopalsamy

Senior Cyber Security Specialist

Louisville, KY, USA- 40220

Abstract—The IoT is a promising technology that, when implemented properly, may have enormous advantages. Nevertheless, since IoT devices are not secure, it has led to a rise in cybersecurity threats. Botnets, for example, have emerged as a serious danger in the IoT space, but there aren't many thorough and systematic studies examining the significance of botnet detection techniques. Thus, this paper explores the application of artificial intelligence (AI)-based techniques to identify and mitigate botnet attacks within Internet-of-Things (IoT) environments, aiming to enhance cybersecurity. Botnets pose a significant danger to digital systems because they allow attackers to remotely manipulate networks of compromised devices. The research utilises the Bot-IoT dataset, comprising over 72 million records of benign and malicious network traffic, to develop and evaluate machine learning models for botnet detection. Several machine learning models, including MLP, and DT, are used to classify IoT botnet attacks. With an impressive 99.97% accuracy rate, the DT model surpasses all other models in important measures like as F1-score, recall, and precision. According to the results of the comparison, DT outperformed other models, and DT is now the best option for detecting botnets in the IoT. The study demonstrates that AI-driven approaches can significantly improve IoT cybersecurity by detecting botnet activities with high accuracy and reliability.

Keywords—IoT-Botnet, Cyber security, BoT-IoT dataset, Decision Tree, Botnet detection.

I. INTRODUCTION

The term "botnet" refers to a group of infected computers that are managed remotely by an individual known as a "Botmaster" via a C&C company. After malware is installed, it develops a backdoor that allows complete remote access to these systems without the user's awareness. The command and control (C&C) infrastructure is what really sets botnets apart from other forms of malware. It allows the Botmaster to remotely administer infected devices while hiding their identity [1].

In this evolving landscape, intelligent learning systems can analyse user actions and behaviours in the cyber realm, easily detecting behavioural patterns across social media platforms. Nevertheless, the black hat group pursues its own interests by spreading harmful activities; botnets, in particular, pose a serious danger to our increasingly digital society [2]. A botnet consists of zombie networks that continually propagate bots—malicious programs acting on the commands of a "botherder," who executes bot attacks for personal gain[3]. These attacks are challenging to mitigate due to the rapid proliferation of botnets, which evade detection by dynamically altering their behaviour[4], causing the value of botnet information to degrade quickly[5].

Cybercriminals may use botnets, which are collections of many internet-connected devices purposefully infected with malware, to launch DDoS assaults, steal data, or provide unauthorised access to equipment. A botnet assault is a

malevolent effort to target and interfere with networks, devices, websites, or whole IT environments by using a collection of linked computers[6][7]. Cybersecurity analysts employ various methods to identify and manage botnets[8], including monitoring packets received from attackers. The structure of botnets is continually evolving, with analysts utilising p2p structures, network traffic analysis, behavioural analysis, signature detection, and DNS tracking to detect these threats[9][10].

However, the unique nature of IoT devices complicates the collection of common normal data, making it difficult to identify botnet activity. Machine learning-based detection systems offer a promising solution, ensuring the identification of not only known attacks but also their variants. In order to overcome these obstacles, we suggest a feature selection approach in our ML-based botnet attack detection architecture, which lowers the processing resource requirements for deployment on devices with limited resources. Experimental results demonstrate that our proposed system achieves sufficiently high detection accuracy to effectively identify botnet attacks[11].

This study's goal is to investigate and create effective artificial intelligence (AI)-based techniques for detecting and mitigating botnet attacks in IoT environments. As the number of IoT devices rises, so does their susceptibility to botnet assaults, which have the potential to seriously disrupt systems and networks. This research aims to enhance cybersecurity by applying ML and DL models to IoT traffic data, improving detection accuracy, and providing a comprehensive framework for safeguarding IoT networks from evolving cyber threats. The study's primary contributions are listed below:

- Proposed an ML-based architecture for IoT botnet detection, which effectively classifies both known and variant attacks with high accuracy.
- Developed a comprehensive preprocessing pipeline that enhances the quality of input data by eliminating duplicates, applying label encoding, and normalising data to improve model performance.
- Introduced feature importance analysis, which highlights the most relevant features for predicting botnet attacks, enabling more efficient and accurate detection.
- Compared multiple machine learning models (MLP, and Decision Trees), identifying the DT model as the most reliable in terms of F1-score, recall, accuracy, and precision.
- Provided a detailed analysis of a Bot-IoT dataset, demonstrating its effectiveness in training and testing IoT botnet detection models while offering insights into the most critical features and attack patterns.

A. Structure of paper

The remainder of the paper is organised in this manner. In Section II, we provide a synopsis of the literature on strategies for detecting botnets in the IoT. Section III outlines the study's methodology, which includes the steps used to prepare the data and the ML models that were used. A thorough examination and explanation of the experimental findings are given in Section IV.

The research's findings and next steps are finally presented in Section V.

II. LITERATURE REVIEW

This section reviews previous research on IoT botnet attack detection employing ML and DL techniques.

In this paper, Pradeepthi and Kannan, (2018) provide a novel approach that uses neuro-fuzzy classification techniques to identify botnet traffic. An application was installed on Eucalyptus Cloud, and it was attacked using a variety of open-source botnet simulation tools to build the dataset used for the experiment. Using 15,000 instances and 56 characteristics, the system's accuracy was 94.78%[12].

In this paper, Liu, Liu and Zhang, (2019) suggest a DL-based method for detecting IoT botnets. After getting to know the dataset, build a CNN, and then use it to detect traffic. Final test results show that our system has a 99.57% success rate in differentiating between safe traffic and attack traffic of different kinds[13].

In this study, Fernández-Peña and Zurita-Amores (2019), a botnet' detection method based on machine learning formalized and evaluated. This proposal makes use of Splunk, a tool that

allowed us to use the Random Forest algorithm to analyse DNS logs in order to detect connections to C&C servers. The achieved results showed an error margin of +/- 5.44 for 18,748,713 events which were analysed. This way, the validity of this proposal was proved[14].

In this paper, Nguyen, Ngo and Le, (2018) provide a unique method for Linux IoT botnet detection that combines a CNN classifier with a PSI graph. A total of 10033 ELF files, comprising 6031 harmless files and 4002 samples from IoT botnets, were used in the experiment. With an F-measure of 94% and an accuracy rate of 92%, the PSI graph CNN classifier does well in the evaluation[15].

In this paper, Esmaeili and Shahriari, (2019) a technique has been presented that uses a tool known as POD Bot. The detection process is carried out using both network traffic analysis and application characteristics. Using a collection of botnets of popular varieties, POD Bot was tested and found that it could correctly identify approximately 87% of high-risk and 96% of very-high-risk botnets[16].

The following Table I provide the existing work comparative analysis IOT- Botnet Attack identification Techniques is given below:

TABLE I. COMPARATIVE ANALYSIS OF IOT-BOTNET ATTACK BASED ON MACHINE LEARNING TECHNIQUES.

Reference	Dataset	Proposed models	Techniques used	Performance Metrics	Key Finding
[12]	15,000 instances, 56 attributes	Neuro-Fuzzy Classification	Eucalyptus cloud setup, open-source botnet simulation tools	Accuracy: 94.78%	Reduced false positives due to fuzzy rules introduction.
[13]	Damped incremental statistics for IoT traffic	Convolutional Neural Network (CNN)	Z-Score normalisation, TAM-based multivariate correlation analysis (MCA)	Accuracy: 99.57%	Effective in distinguishing benign traffic from attack traffic.
[14]	DNS logs (18,748,713 events)	Random Forest	Splunk tool for analysing connections to C&C servers	Error margin: +/- 5.44	Validated machine learning approach with improved results through data source verification.
[15]	10,033 ELF files (4,002 IoT botnet samples, 6,031 benign files)	PSI Graph CNN Classifier	Combination of PSI graph and CNN	Accuracy: 92%, F-measure: 94%	Effective in detecting Linux IoT botnets.
[16]	High-risk and very high-risk botnets	POD Bot	Application features and network traffic analysis	Accuracy: 87% (high risk), 96% (very high risk)	Improved detection through combined methods compared to previous approaches.

III. METHODOLOGY

A goal of this project is to improve cybersecurity by creating AI-based methods for identifying and stopping botnet assaults in IoT systems. By applying ML and DL models to IoT data, a goal is to improve detection accuracy and provide a robust framework to protect IoT networks from malicious activities. The results of this evaluation will help improve the ability to identify botnet attacks, which starts with data collection, utilising the Bot-IoT dataset from UNSW Canberra's Cyber Range Lab, which contains over 72 million records of both benign and malicious traffic, used for IoT botnet detection. The data undergoes preprocessing, including removing duplicates, label encoding, and normalisation, followed by splitting into training (80%) and testing (20%) sets. Finding the most crucial features for a model's accuracy is a goal of feature importance analysis. Classifying IoT botnet assaults involves the use of many ML models, including MLP, and DT. Decision trees employ the Gini coefficient to optimise node splitting, enhancing classification accuracy. The experiments, executed on a high-

performance system using Python, Jupyter Notebook, and relevant libraries, generate confusion matrices for both training and testing, facilitating performance evaluation. The flowchart of botnet methods of attack to improve cyber security is shown in Figure 1 below.

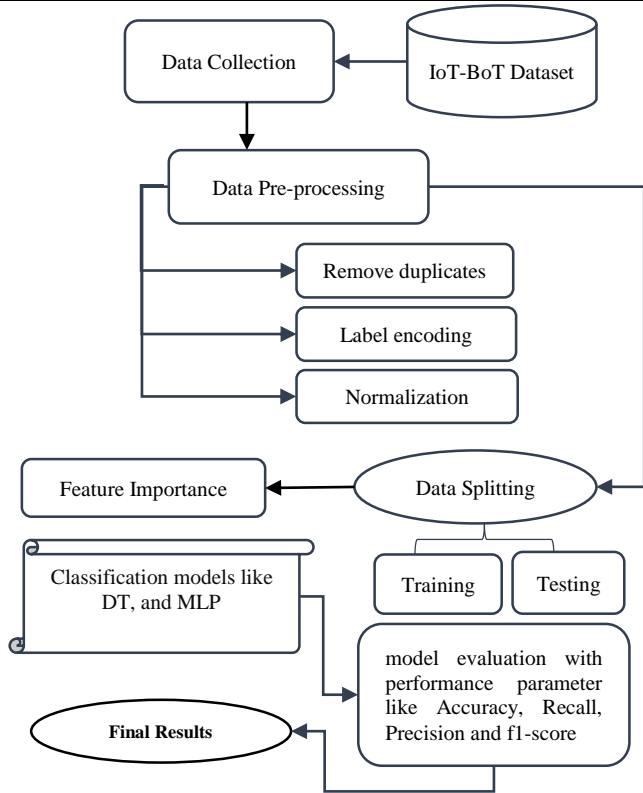


Fig. 1. Flowchart of the IoT Botnet Attack Detection System.

The steps outlined in the flow chart are described in detail below:

A. Data Collection

Data gathering is an essential part of creating complete datasets for efficient analysis. An invaluable resource for testing IoT botnet detection algorithms is the Bot-IoT dataset, created by the Cyber Range Lab at UNSW Canberra. This dataset collects network traffic using PCAP files and contains more than 72 million records of good and bad actions.

B. Data Preprocessing

Data preprocessing is an essential stage that ensures the greatest possible foundation for accurate and effective machine learning model performance by converting unstructured, raw data into clean, organised, and optimised input. Below is a list of the essential pre-processing steps:

1) Remove duplicates

This step ensures that redundant or repeated entries are eliminated from the dataset, providing clean and unique data for model training, which helps improve accuracy and reduce overfitting.

2) Label encoding

ML models that rely on numerical input may work with data that has been label encoded, as it transforms category characteristics into numerical values.

3) Normalization

In order to guarantee that every feature contributes equally throughout the model training process, normalisation adjusts numerical features to a consistent range, usually between 0 and 1.

C. Data Splitting

Data splitting is the process of separating the dataset into two sections: 20% for testing and 80% for training. The training set is used to train the ML model, while the testing set is put aside to evaluate the model's performance on unknown data and ensure it generalises well to new data.

D. Feature Importance

To improve model performance by zeroing down on the main drivers of accuracy, feature significance provides a quantitative measure of how each feature contributes to a model's predictions. This helps in identifying the most significant features.

E. Classification Models

This section presents a range of classification models employed for detecting and classifying IoT botnet attacks within cybersecurity, emphasising a comparative analysis of their performance.

1) Multilayer Perceptron Classifier (MLP)

MLP are feedforward ANN models that translate input data into a collection of suitable outputs. The intake, production, and hidden layers are its three layers. The processing signal is sent to the input layer. During MLP processing, the input and output have an infinite number of hidden layers [17].

2) Decision Tree (DT)

The DT is a tree-like classifier that belongs to the category of nonlinear supervised classification models. Connecting nodes between branches stand in for discriminating criteria, while leaf nodes at the ends of branches reflect each record's category. Depending on whether the data meets the conditions shown at the node, we choose a number of branches to continue with and repeat the earlier steps until we reach a leaf node in the DT categorisation process.

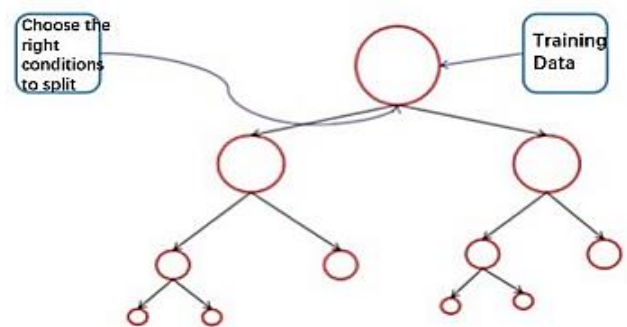


Fig. 2. Decision tree generation.

The process of creating a DT is shown in Figure 2, where the main goal is to differentiate data that falls into several categories as clearly as possible. To achieve this, the Gini coefficient is introduced as a loss function, which plays a crucial role in constructing the Classification and Regression Tree (CART) model.

$$Gini_{(p)} = \sum_{k=1}^K P_k (1 - p_k) = 1 - \sum_{k=1}^K P_k^2 \quad (1)$$

Among the Gini expressions, K stands for the entire number of categories, or Equ 1. In category k , p_k is the ratio of the number of records in category k that fulfil the forward discrimination criteria to the number of records in category k that satisfy the current discriminant criterion.

IV. RESULTS AND DISCUSSION

This section provides a summary of experiments and arguments conducted utilising the IOT Botnet dataset's files for ML training and testing. This displays the experimental outcomes of ML and DL models to detect IoT Botnet attacks. The findings are showcased through a variety of figures, graphs, and tables to illustrate the performance and comparisons of the models. The experiment's results are displayed as a confusion matrix, f1-score, recall, accuracy, and precision.

A. Dataset Description

The Cyber Range Lab at UNSW Canberra developed the Bot-IoT dataset to test methods for detecting botnets in the IoT by simulating a real-world network. It is a useful tool for cybersecurity research as it contains a mix of benign traffic and traffic produced by ten distinct kinds of botnet attacks. The

dataset is derived from PCAP files, with the full version containing over 72 million records, providing a detailed and extensive amount of data for analysis. Furthermore, there is a smaller group that includes over 3 million records, or 5% of the initial traffic. This subset has extra features that are created to help with ML tasks like feature selection and classification. This dataset is widely used to develop and test models for identifying botnet threats in IoT networks, offering researchers a comprehensive resource for evaluating the effectiveness of various detection techniques. The 5% dataset's sample sizes for malicious and benign traffic are shown in Table II.

TABLE II. CLASS DISTRIBUTION OF NORMAL AND ATTACK TRAFFIC IN THE BOT-IOT DATASET.

	Class	Samples
1	Service Scanning	73,168
2	OS Fingerprinting	17,914
3	DDoS TCP	977,380
4	DDoS UDP	948,255
5	DDoS HTTP	989
6	DoS TCP	DoS TCP
7	DoS UDP	DoS UDP
8	DoS HTTP	1485
9	Normal	477
10	Keylogging	73
11	Data Exfiltration	6

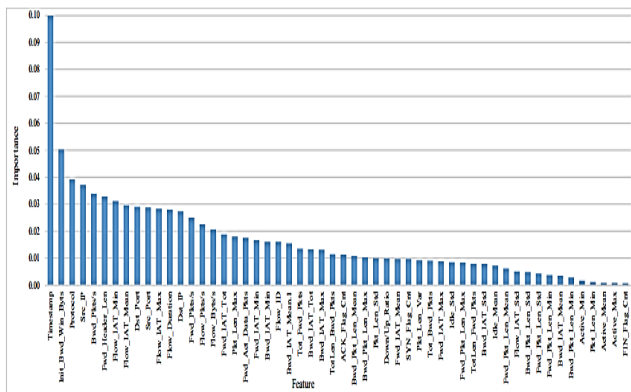


Fig. 3. Feature importance for all features IoT Botnet Dataset.

The significance of the characteristics included in the classification model is displayed in Figure 3. An x-axis shows characteristics' rankings, while a y-axis shows significance ratings, which range from 0 to 1. The chart shows a clear descending order of feature importance, with the first few features displaying significantly higher importance than the rest, indicating their stronger predictive power in the model.

B. Performance Measures

This section evaluates the model's performance using essential metrics like F1-score, recall, accuracy, and precision. To gain a deeper understanding of a model's effectiveness, a confusion matrix is also computed for further analysis and validation.

1) Confusion Matrix

A confusion matrix compares expected classes with actual classes and displays them in tabular form. Each quadrant's sample count is shown. This aids comprehension of a model's expected True Negatives, False Negatives, True Positives, and False Positives. These described as;

- **True Positive:** The model's prediction was also positive, and the actual result was positive.
- **False Positive:** It is false, and your forecast is positive.
- **False Negative:** You have made a negative prediction, which is also incorrect.
- **True Negative:** Both the actual and projected values were negative according to the model.

2) Accuracy

A classifier's accuracy may be used as a means of performance evaluation. This measure presents the rate of all instances correctly categorised as Eq. (2).

$$Accuracy = \frac{TP+TN}{TP+TN+FN+FP} \tag{2}$$

3) Precision

A proportion of positive events that are successfully anticipated is shown by the measure of precision. As in Eq. (3) the metric here stands for the model's predictive power for the class:

$$Precision = \frac{TP}{TP+FP} \tag{3}$$

4) Recall

A classifier's recall indicates its accuracy in spotting positive instances. Equation (4) provides the recall formula:

$$Recall = \frac{TP}{TP+FN} \tag{4}$$

5) F1-score

One common metric for measuring a classifier's efficacy is the F-score, as Eq. (5). An F-score, which is sometimes described as the harmonic mean of recall and precision, is a statistic that accounts for both of these variables:

$$F1\text{-score} = 2 * \frac{precision*recall}{precision+recall} \tag{5}$$

A following measure evaluate the performance of proposed IoT Botnet attack identification.

TABLE III. RESULTS FOR DECISION TREE MODEL.

Performance Measures	Decision Tree Model
Accuracy	99.97
Precision	99.97
Recall	99.97
F1-score	99.97

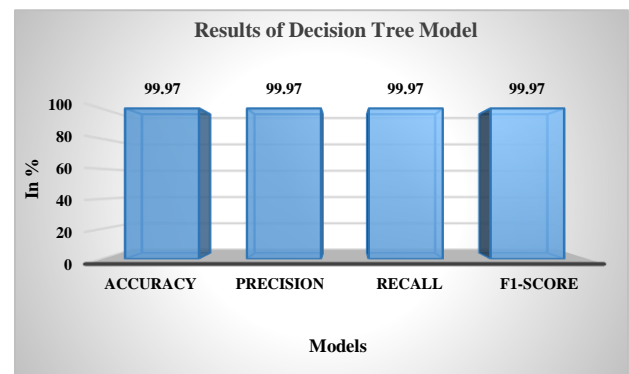


Fig. 4. Results of Decision Tree Model.

Figure 4 presents a result of the DT model, which demonstrates the highest accuracy, achieving an impressive score of 99.97%, respectively for IoT-botnet detection.

Service_Scanning	21887	21	1	0	3	1	0	0	1
OS_Fingerprinting	17	5454	2	0	0	0	0	0	0
DDoS_TCP	0	0	292681	0	0	214	0	0	0
DDoS_UDP	0	1	0	284802	0	0	0	0	0
DDoS_HTTP	0	0	0	0	318	0	0	0	0
DoS_TCP	1	0	11	1	0	184815	2	0	0
DoS_UDP	0	0	0	3	0	0	309749	0	0
DoS_HTTP	1	3	0	0	3	0	0	403	0
Normal	1	1	0	0	0	0	2	1	133
Source_Scanning									
OS_Fingerprinting									
DDoS_TCP									
DDoS_UDP									
DDoS_HTTP									
DoS_TCP									
DoS_UDP									
DoS_HTTP									
Normal									

Fig. 5. Confusion matrix of classification results for the decision tree model.

Figure 5 shows a confusion matrix, a crucial visual for evaluating the performance of classification classifiers on the Bot-IoT dataset. The matrix's rows show occurrences in real classes while its columns show instances in anticipated classes. The diagonal cells, which display the number of right predictions for each class, are highlighted and have high values, which means the model is running well. Off-diagonal cells show the misclassifications between different attack types and benign traffic, providing insights into the types of errors made by the model.

C. Comparative analysis

Table IV details the outcomes of a comparison of ML algorithms for detecting IoT botnets. This comparison mainly focuses on accuracy, precision, recall, and F1-score, all of which were evaluated using datasets from the IoT Botnet.

TABLE IV. COMPARISON BETWEEN VARIOUS MODELS BASED ON IOT BOTNET ATTACKS USING IOT-BOT DATASET.

Models	Accuracy	Precision	Recall	F1- score
MLP[18]	84	88	84	83
DT	99.97	99.97	99.97	99.97

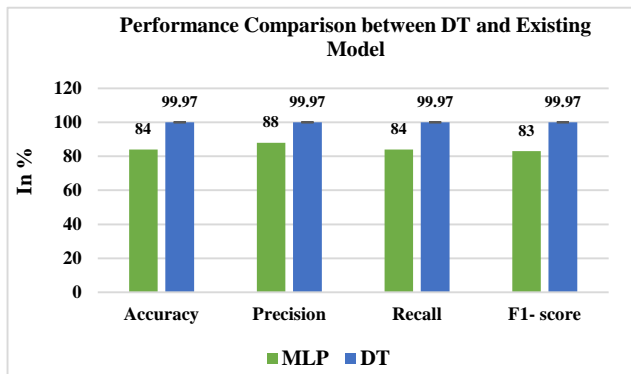


Fig. 6. Bar Graph of comparison of models.

Table IV and Figure 6 demonstrate that when it came to IoT botnet assaults, the DT model performed better than any other ML model that was evaluated. With an F1 score of 99.97% and near-perfect accuracy, precision, and recall, it proved to be the most trustworthy model for botnet classification tasks. A Multi-Layer Perceptron (MLP) model shows slightly lower results, with an accuracy 84% and an F1-score 83%. Overall, the DT model proves to be the most robust, while MLP exhibits weaker performance across key metrics.

V. CONCLUSION AND FUTURE SCOPE

This study aimed to improve cybersecurity in IoT settings by detecting and classifying IoT botnet attacks using multiple ML models. The outcomes demonstrate that the DT model outperforms other models, achieving an almost perfect F1-score, recall, accuracy, and precision with 99.77% performance. A comparative analysis highlights the effectiveness of the DT model in identifying botnet attacks; other models, such as Multilayer Perceptron displayed moderate results, with the latter showing weaker performance in handling complex attack scenarios. Overall, this study contributes to the development of AI-based techniques that can enhance IoT security by improving the detection and classification of botnet threats.

Future studies should investigate more sophisticated DL methods, such as CNNs and RNNs, to enhance real-time botnet identification. Hybrid models combining multiple approaches may also enhance performance. Additionally, applying reinforcement learning could create adaptive systems to address evolving threats, and testing on larger, diverse datasets will help generalise the results across various IoT networks.

REFERENCES

- [1] A. Azab, M. Alazab, and M. Aiash, "Machine learning based botnet identification traffic," in *Proceedings - 15th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, 10th IEEE International Conference on Big Data Science and Engineering and 14th IEEE International Symposium on Parallel and Distributed Proce*, 2016. doi: 10.1109/TrustCom.2016.0275.
- [2] V. V. Kumar and F. T. S. Chan, "A superiority search and optimisation algorithm to solve RFID and an environmental factor embedded closed loop logistics model," *Int. J. Prod. Res.*, 2011, doi: 10.1080/00207543.2010.503201.
- [3] V. Kumar, V. V. Kumar, N. Mishra, F. T. S. Chan, and B. Gnanasekar, "Warranty failure analysis in service supply Chain a multi-agent framework," in *SCMIS 2010 - Proceedings of 2010 8th International Conference on Supply Chain Management and Information Systems: Logistics Systems and Engineering*, 2010.
- [4] S. Dixit, P. Pathak, and S. Gupta, "A novel approach for gray hole and black hole detection and prevention," in *2016 Symposium on Colossal Data Analysis and Networking, CDAN 2016*, 2016. doi: 10.1109/CDAN.2016.7570861.
- [5] V. V. Kumar, M. K. Pandey, M. K. Tiwari, and D. Ben-Arieh, "Simultaneous optimization of parts and operations sequences in SSMS: A chaos embedded Taguchi particle swarm optimization approach," *J. Intell. Manuf.*, 2010, doi: 10.1007/s10845-008-0175-4.
- [6] V. V. Kumar, S. R. Yadav, F. W. Liou, and S. N. Balakrishnan, "A Digital Interface for the Part Designers and the Fixture Designers for a Reconfigurable Assembly System," *Math. Probl. Eng.*, vol. 2013, pp. 1–13, 2013, doi: 10.1155/2013/943702.
- [7] K. Mullangi, V. K. Yarlagadda, N. Dhameliya, and M. Rodriguez, "Integrating AI and Reciprocal Symmetry in Financial Management: A Pathway to Enhanced Decision-Making," *Int. J. Reciprocal Symmetry Theor. Phys.*, vol. 5, no. 1, pp. 42–52, 2018.
- [8] S. G. Priya Pathak, Akansha Shrivastava, "A survey on various security issues in delay tolerant networks," *J Adv Shell Program.*, vol. 2, no. 2, pp. 12–18, 2015.
- [9] S. K. R. A. Sai Charan Reddy Vennapusa, Takudzwa Fadziso, Dipakkumar Kanubhai Sachani, Vamsi Krishna Yarlagadda, "Cryptocurrency-Based Loyalty Programs for Enhanced Customer Engagement," *Technol. Manag. Rev.*, vol. 3, no. 1, pp. 46–62, 2018.
- [10] V. K. Yarlagadda and R. Pydipalli, "Secure Programming with SAS: Mitigating Risks and Protecting Data Integrity," *Eng. Int.*, vol. 6, no. 2, pp. 211–222, Dec. 2018, doi: 10.18034/ei.v6i2.709.
- [11] S. G. Darshan Meena, Priya Pathak, "Cryptography Bases Solution FOR Distributed Denial of Service Attack in Manet," *Int. J. Grid Distrib. Comput.*, vol. 9, no. 6, pp. 219–234, 2016.
- [12] K. V. Pradeepthi and A. Kannan, "Detection of Botnet traffic by using Neuro-fuzzy based Intrusion Detection," in *2018 10th International Conference on Advanced Computing, ICoAC 2018*, 2018. doi: 10.1109/ICoAC44903.2018.8939109.

- [13] J. Liu, S. Liu, and S. Zhang, "Detection of IoT botnet based on deep learning," in *Chinese Control Conference, CCC*, 2019. doi: 10.23919/ChiCC.2019.8866088.
- [14] F. Fernández-Peña and A. Zurita-Amores, "Botnets Detection in DNS logs using machine learning | Detección de Botnets en logs DNS utilizando Aprendizaje Automático," *Iber. Conf. Inf. Syst. Technol. Cist.*, 2019.
- [15] H. T. Nguyen, Q. D. Ngo, and V. H. Le, "IoT Botnet Detection Approach Based on PSI graph and DGCNN classifier," in *2018 IEEE International Conference on Information Communication and Signal Processing, ICICSP 2018*, 2018. doi: 10.1109/ICICSP.2018.8549713.
- [16] S. Esmacili and H. R. Shahriari, "PodBot: A New Botnet Detection Method by Host and Network-Based Analysis," in *ICEE 2019 - 27th Iranian Conference on Electrical Engineering*, 2019. doi: 10.1109/IranianCEE.2019.8786432.
- [17] V. V. Kumar, F. T. S. Chan, N. Mishra, and V. Kumar, "Environmental integrated closed loop logistics model: An artificial bee colony approach," in *SCMIS 2010 - Proceedings of 2010 8th International Conference on Supply Chain Management and Information Systems: Logistics Systems and Engineering*, 2010.
- [18] J. Alsamiri and K. Alsubhi, "Internet of things cyber attacks detection using machine learning," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 12, pp. 627–634, 2019, doi: 10.14569/ijacsa.2019.0101280.