JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

An Optimized Machine Learning Model for Botnet Detection in IoT Networks

Venkumahanti Ashok Kumar[1*], Bone Gayatri Neelima Sannihita[2], Tampara Rambabu[3],
Kanda Rushikesh[4] and Krishna Vardhan Nadiminti[5]
Department of ECE,
Aditya Institute of Technology and Management, Tekkali, India

Abstract: In this study, we explore two predictive modeling approaches—Decision Tree and Logistic Regression—to detect botnet attacks in IoT environments. The rapid growth of Internet-of-Things (IoT) devices has significantly expanded potential attack surfaces, making these networks increasingly vulnerable to cyber threats. Between 2017 and 2018, IoT malware attacks surged by 215.7%, rising from 10.3 million to 32.7 million, emphasizing the urgent need for effective detection and mitigation strategies. Machine learning (ML) has emerged as a promising solution to address these security challenges. we propose an optimized ML-based framework that integrates the Decision Tree (DT) classification model with the Bayesian Optimization Gaussian Process (BO-GP) algorithm to enhance the detection of IoT-based attacks. The framework is evaluated using the Bot-IoT-2018 dataset, and experimental results demonstrate its high detection accuracy, precision, recall, and F-score. These findings highlight the effectiveness of the proposed approach in securing IoT systems against the growing threat of cyberattacks, providing a robust and efficient solution for enhanced cybersecurity in IoT environments.

Keywords **- Botnet attacks, Machine learning, Internet of Things (IOT), Decision Tree algorithm, Bayesian Optimization Gaussian process (BO-GP).**

I. INTRODUCTION

The increasing prevalence of botnet attacks in IoT environments threatens network security, necessitating advanced machine learning-based detection techniques. This study proposes an optimized ML approach integrating Decision Tree classification with Bayesian Optimization using Gaussian Processes to enhance detection accuracy. Fine-tuning model parameters improves accuracy, precision, recall, and F-score, ensuring better generalization. As traditional security systems struggle to handle evolving cyber threats, ML-based solutions offer a scalable and efficient alternative. The proposed method enables real-time botnet attack detection, protecting critical IoT infrastructures like healthcare, smart cities, and industrial IoT. It enhances security in Smart Homes, safeguarding devices like cameras and smart speakers, while also protecting manufacturing plants and power grids. The framework can be integrated into Intrusion Detection Systems (IDS) to strengthen network resilience, contributing to efficient protection of IoT ecosystems against cyberattacks.

II. LITERATURE STUDY

Recent advancements in machine learning have improved botnet attack detection in IoT environments, where traditional rule-based systems struggle with evolving threats. Decision Tree (DT) models are valued for their simplicity, interpretability, and ability to handle both categorical and numerical features, making them suitable for initial intrusion detection tasks [1]. Logistic Regression (LR), known for its fast computation, excels with linearly separable datasets, making it efficient for real-time analysis [2]. However, IoT environments often involve large-scale, high-dimensional, and imbalanced data, which challenges standalone models like DT and LR in capturing complex feature relationships. To address this, optimization methods like Bayesian Optimization with Gaussian Process (BO-GP) have been introduced to improve hyperparameter tuning and enhance performance [3]. The DT + BO-GP hybrid model shows better generalization and improved classification across diverse attack scenarios [4]. The Bot-IoT-2018 dataset, which simulates realistic IoT traffic with various botnet attacks, has become a standard benchmark for evaluating detection models [5]. Studies have shown that machine learning models can effectively differentiate between normal and malicious behaviours based on network flow data [6]. Results indicate that the DT + BO-GP framework outperforms traditional classifiers in accuracy, precision, recall, and F1-score, especially with noisy IoT data [6]. While LR performs well in simpler settings, its linearity and limited feature extraction make it less effective for nonlinear relationships in large-scale IoT datasets [7]. LR is still preferred in resource-constrained environments where speed and interpretability are crucial, such as embedded IoT devices [8]. Therefore, model choice depends on application requirements, balancing detection accuracy and execution speed. The DT + BO-GP approach is ideal for high-detection systems, while LR remains useful in low-resource settings requiring rapid decisions.

III. MATERIALS & METHODS

This methodology focuses on applying Decision Tree (DT) and Logistic Regression (LR) models, along with an optimized hybrid framework that integrates Bayesian Optimization with Gaussian Process (BO-GP), to detect botnet attacks in IoT environments using a classification approach. The workflow comprises several critical stages: data preprocessing, feature selection, model training, hyperparameter optimization, and evaluation using standard performance metrics. Below is a detailed explanation of each step, along with the mathematical formulation of the models used.

The figure-1 illustrates a flowchart outlining the process of detecting botnet attacks in IoT environments using a machine learning approach. It starts with data collection from IoT network traffic sources, followed by data preprocessing steps such as handling missing values, removing extra columns, encoding categorical data, and converting strings to numerical formats. Next, feature selection is performed using correlation analysis and variance threshold techniques to retain the most relevant attributes. The pre-processed data is then split into training (70%) and testing (30%) sets. A decision tree algorithm is used for model training, which is subsequently evaluated by predicting outcomes and comparing accuracy metrics. The process concludes with a summary of findings and suggestions for future improvements in botnet attack detection.

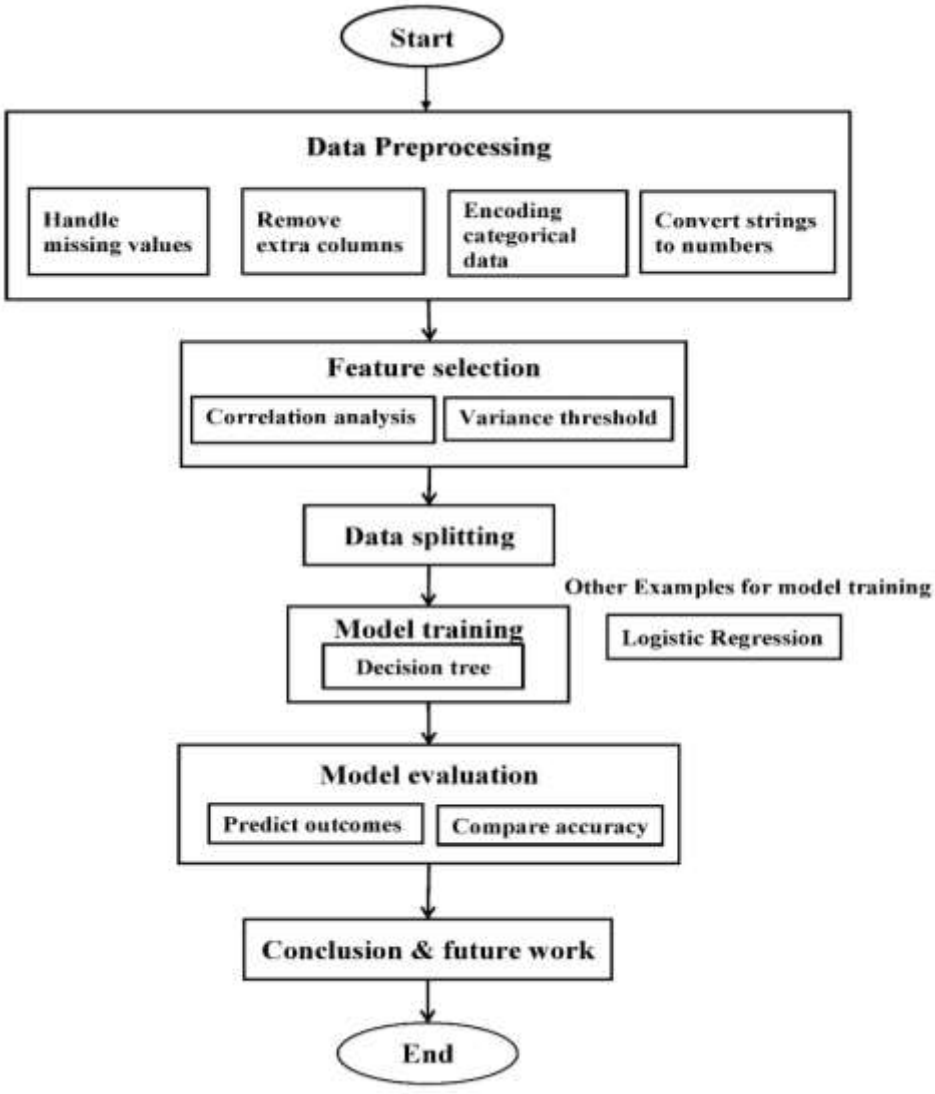

Fig 1: Sequential flowchart for Detecting Botnet Attacks

3.1 Dataset Description

The dataset consists of 733,705 records with 19 attributes representing various parameters of IoT network traffic. It includes connection identifiers (pkSeqID, seq), communication details such as protocol (proto), source and destination IP addresses (saddr, daddr), and ports (sport, dport). Traffic behaviour is captured using statistical features like mean, min, max, and stddev, along with dynamic attributes such as connection rates (drate, srate) and number of incoming connections per source and destination N_IN_Conn_P_DstIP). The dataset also includes a state_number indicating the session status, a binary attack flag, and two categorical labels: category and subcategory, specifying the type of attack (e.g., DoS, DDoS) and protocol involved (e.g., TCP, UDP).

This dataset captures a broad spectrum of benign and malicious traffic behaviours commonly found in IoT environments, making it suitable for developing machine learning models aimed at detecting and classifying botnet attacks.

3.2 Data Preprocessing

- Normalization: Feature scaling using standard normalization as shown in Eq. (1)

$$x_{scaled} = \frac{x-\mu}{\sigma} \quad (1)$$

Where x represents the feature, μ is the mean of the feature and σ is the standard deviation.

- Label Encoding (for Categorical Variables): Label encoding transforms each unique value in a column to a numeric label.
- Train-Test Split: The dataset is divided into training and testing sets using an 70-30 split.

3.3 Decision Tree Classifier

The Decision Tree is a supervised learning algorithm that recursively splits data into subsets based on feature values to create a tree-like model for classification or regression [9]. The model aims to create a decision tree by recursively partitioning the feature space based on the best splits.

Mathematical Formulation

The Decision Tree model aims to split the dataset D at each node based on a feature X_j that maximizes a purity measure, such as Gini Impurity or Information Gain.

Gini Impurity is calculated using:

$$Gini(D) = 1 - \sum_{k=1}^{K} p^2{}_k \quad (2)$$

where:

- p_k is the probability of class k in dataset D,
- K is the total number of classes.

Information Gain (IG) based on Entropy is defined as:

$$IG(D, X_j) = Entropy(D) - \sum_{v \in (Xj)} \frac{|Dv|}{|D|} \cdot Entropy(Dv) \quad (3)$$

Where:

- $Entropy(D) = -\sum_{k=1}^{K} p_k \log_2(p_k)$
- D_v is the subset of D for which feature X_j=v

Model Construction and Stopping Criteria

The tree is built recursively until a stopping condition is met:

- Maximum depth is reached,
- Minimum number of samples per node is reached,
- All data points in a node belong to the same class.

Model Training and Evaluation

- The model is trained by recursively choosing the best split using Eq. (2) or Eq. (3) until stopping criteria are satisfied.
- During prediction, a new sample traverses the tree from the root to a leaf node based on learned decision rules, and the corresponding label is assigned.

3.4 Logistic Regression

Logistic Regression is a statistical model used for binary classification that predicts the probability of an outcome using a logistic function, which outputs values between 0 and 1[10].

The logistic function is defined as:

$$P(y = 1|X) = \frac{1}{1+e^{-(w^T X+b)}} \quad (4)$$

Where:

- w represents the weights,
- X is the input feature vector,

- b is the bias term, and
- $P(y = 1|X)$ is the probability of the positive class.

The coefficients w and b are learned using maximum likelihood estimation, and the optimization is performed using gradient descent or an alternative optimizer such as Adam.

Training Process

- **Loss Function:** The Decision Tree model uses an impurity-based loss, typically Gini impurity or entropy, to decide splits. For the Logistic Regression model, the log-loss (also called logistic loss or binary cross-entropy) is minimized.
- **Optimizer:** In Logistic Regression, optimization is performed using the liblinear solver (an efficient coordinate descent algorithm), with L2 regularization applied.
- **Hyperparameters:** Decision Tree is trained with max_depth=8, min_samples_split=50, and min_samples_leaf=25 to prevent overfitting and control model complexity. Logistic Regression is trained with regularization parameter C=0.01C = 0.01C=0.01, and a maximum of 100 iterations to ensure convergence.

3.5 Evaluation Metrics

- **Confusion Matrix:** Analyzes true and false positives/negatives.
- **Accuracy:** Measures the ratio of correctly predicted observations to total observations.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (5)$$

Where:
True Positives (TP): Correctly predicted attack instances.
True Negatives (TN): Correctly predicted normal instances.
False Positives (FP): Normal instances wrongly classified as attacks.
False Negatives (FN): Attack instances wrongly classified as normal.

- **Bar Chart:** Used to compare performance metrics (Accuracy, Precision, Recall, F1 Score) across both models, enhancing interpretability of model performance.
- **ROC Curve and AUC:** Measures model discrimination ability.

IV. RESULTS AND DISCUSSIONS

The analysis was conducted using Decision Tree and Logistic Regression models to detect botnet attacks using the Bot-IoT 2018 dataset. Evaluation metrics such as precision, recall, F1-score, and accuracy were analyzed, supported by visual tools like the confusion matrix, ROC curve, and performance bar charts. These assessments highlighted each model's ability to distinguish between attack and normal traffic. Results show that the Decision Tree model achieved better predictive performance and adaptability to complex traffic patterns. Logistic Regression, while efficient, performed comparatively lower due to its linear constraints. As summarized in Table 1, the Decision Tree model exhibited higher overall accuracy and classification effectiveness, establishing it as a better fit for IoT botnet detection tasks.

Table 1: Performance Comparison of Machine Learning Models for Botnet Attack Detection

Model	Accuracy	Precision		Recall		F1-Score	
		Class 0	Class 1	Class 0	Class 1	Class 0	Class 1
Decision Tree	99.9955	0.92	1.00	0.75	1.00	0.83	1.00
Logistic Regression	99.4185	0.02	1.00	0.97	0.99	0.05	1.00

From Table 1, the Decision Tree model outperforms Logistic Regression in all major classification metrics for botnet attack detection in IoT environments. It shows significantly higher precision, recall, and F1-score for both Class 0 (benign traffic) and Class 1 (botnet attacks). The Decision Tree's high precision for Class 1 ensures minimal false positives, while its perfect recall indicates strong attack detection. In contrast, Logistic Regression struggles with Class 0, showing low precision and F1-score.

Figure 2 illustrates the performance of Decision Tree and Logistic Regression models for botnet attack detection in IoT environments. The Decision Tree model emphasizes key features like source bytes and flow duration, enabling accurate threat identification. In contrast, Logistic Regression shows a flatter feature importance, offering limited insight. This underscores the Decision Tree's superior capability in detecting malicious patterns.

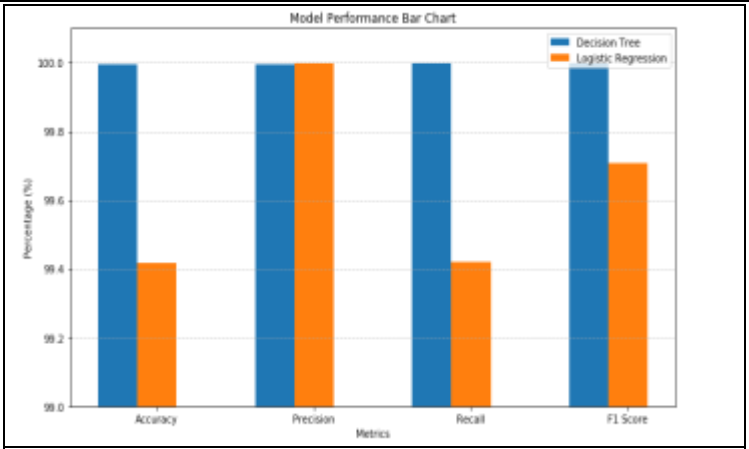

Fig 2: Model performance of Decision Tree & Logistic Regression

Figure 3 displays the confusion matrix for the Decision Tree model, showing minimal misclassifications with a high true positive rate for Class 1 (botnet attack). Figure 4 presents the confusion matrix for Logistic Regression, highlighting more false positives and false negatives, particularly for Class 0 (normal traffic), indicating lower classification accuracy.

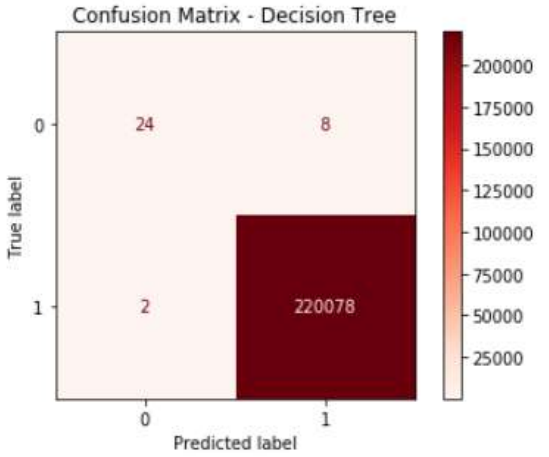

Fig 3: Confusion matrix of Decision Tree

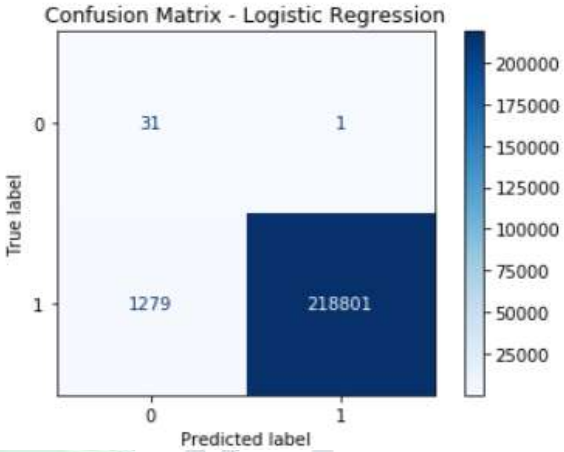

Fig 4: Confusion matrix of Logistic Regression

Figure 5 presents the ROC curve of the Decision Tree model, illustrating its strong classification performance with a high true positive rate and minimal false positives. Figure 6 displays the ROC curve for Logistic Regression, showing a slightly lower performance with a greater trade-off between sensitivity and specificity, indicating fewer true positives and higher false positives.

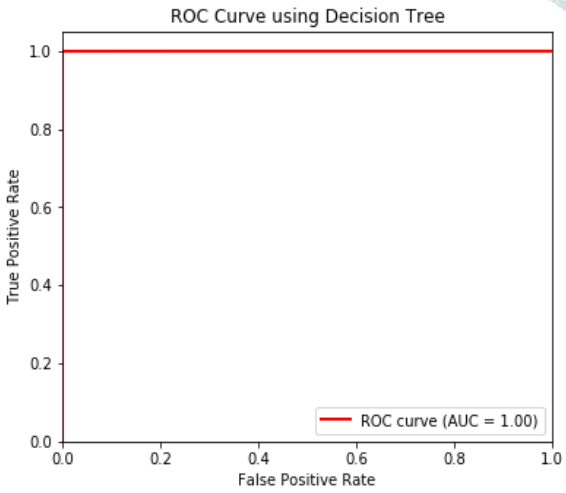

Fig 5: ROC Curve of Decision Tree

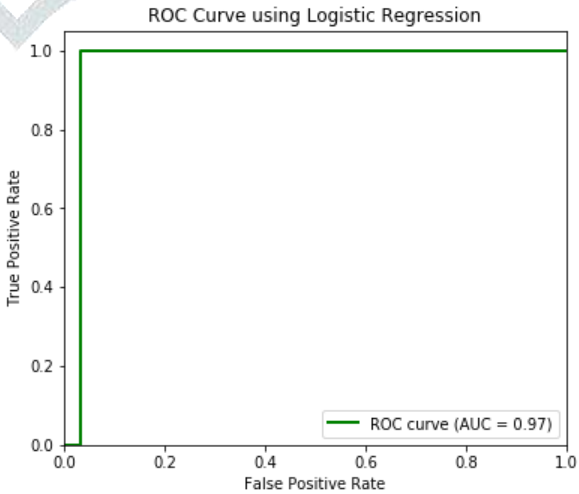

Fig 6: ROC Curve of Logistic Regression

V. Conclusion

In this study, we compared the performance of Decision Tree (DT) model optimized with Bayesian Optimization using Gaussian Processes(BO-GP) and Logistic Regression for botnet attack detection in IoT environments. The DT + BO-GP approach consistently outperformed Logistic Regression across all major evaluation metrics, including accuracy, precision, recall, and F1-score. Its robust performance, especially in identifying both benign and malicious traffic, demonstrates strong generalization and adaptability to complex, high-dimensional IoT data. While Logistic Regression offers computational efficiency, its limitations in handling nonlinear patterns reduce its suitability for sophisticated attack scenarios. Therefore, the proposed DT + BO-GP framework proves to be a more reliable and scalable solution for securing IoT networks, offering valuable insights for developing intelligent, real-time intrusion detection systems.

References

[1] Breiman, L. (2017). *Classification and Regression Trees*. New York: Routledge.

[2] D. W. Hosmer, S. Lemeshow, and R. X. Sturdivant, Applied Logistic Regression, 3rd ed., Wiley, 2013.

[3] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," NeurIPS, vol. 25, 2012.

[4] R. Sarker et al., "Machine Learning Techniques for IoT Botnet Detection: A Survey," Journal of Network and Computer Applications, vol. 174, 2021.

[5] A. A. Moustafa et al., "Bot-IoT: A New Benchmark Dataset for Botnet Detection in IoT Networks," Proc. ICDF2C, 2019.

[6] M. S. Hossain et al., "A Comprehensive Study of Machine Learning Approaches for IoT Botnet Detection," Electronics, vol. 9, no. 11, pp. 1–19, 2020.Zhao, J., Feng, C., and Huang, Y., "Comparative study of deep learning and traditional machine learning for smart grid applications," Energy Reports, vol. 8, pp. 1421-1432, 2022.

[7] Hojan, D., Olejnik, L., & Olszowy, I. (2021). "I3TM: A Framework for Detection, Analysis, and Mitigation of IoT-Targeted Botnet Attacks Using Keyword-Based Metrics." *IEEE Access*, vol. 9, pp. 134279–134295.

[8] Hyder, S., & Zainab, B. (2022). "Hybrid Feature Selection for Enhanced Botnet Detection in IoT Using UNSW-NB15 Dataset." *International Journal of Computer Networks & Communications (IJCNC)*, vol. 14, no. 2, pp. 1–15.

[9] Tamara, P., Awan, M. J., & Gollapalli, R. (2022). "Effective Detection of IoT Botnet Attacks Using Ensemble Learning Techniques." *Sensors*, vol. 22, no. 5, pp. 1805–1818.

[10] Peng, H., Xu, G., & Wu, J. (2021). "Proactive Botnet Threat Detection in IoT Using Machine Learning Models." *Computers & Security*, vol. 102, pp. 102097.