A HYBRID MACHINE LEARNING APPROACH FOR BOTTLENECK DETECTION IN IOT

[1]S.AKASH, [2]G.SANDHYA, [3]AQSA MAHA, [4]P. DIVYA RANI (Assistant Professor)
[1, 2, 3, 4]Department of Computer Science and Engineering
[1, 2, 3, 4]Vijay Rural Engineering College, Manik Bhandar, Nizamabad-503003

Abstract: Cloud computing can be one of the greatest discoveries in the modern computer world. It offers a cost effective option by reducing the enormous preliminary expenditure towards hardware infrastructure and computational capability. Fog computing reduces latency in edge devices; this improves the cease users' response time in IoT applications. Nevertheless, the major part of the people of IoT devices are useful resource-confined and numerous gadgets are vulnerable to cyberattacks. Cyber-attacks such as bottleneck, DoS, DOS and botnet, remain significant threat in the IoT environment. Botnets are presently the greatest threat in the internet world. A collection of compromised structures linked to the net and utilizing adversary in an attempt to carry out unauthorized adverse functions is known as botnet. The system can be penetrated by a botnet that will exfiltrate facts. It is more than capable of mounting an attack such as phishing and spamming among others. To handle the urgent issue we offer a novel botnet attack detection approach suitable for fog computing environments and utilize the programmable nature of the software-defined network framework to prevent the attack. We closely examined the ultra-modern dataset for our cautioned technique, traditional and extended performance assessment metrics as well as modern deep learning models. To support the overall performance presentation, our results are cross validated. The recommended generation outperforms its predecessors in the recognition of 99.98% of multifaceted sophisticated bot attacks. In addition, the proposed solution by us is of 0.022 milliseconds in length, which reflects excellent speed efficiency effects.

"Keywords *- Fog security, software defined networks, Machine learning, Internet of Things, botnet, intrusion detection".*

1. INTRODUCTION

The fast growth of the internet of things (IoT) in many areas, including healthcare, transportation, training, and enterprise, has led to a great increase in data produced by IoT devices. This has led to tremendous problems inside the efficient processing and management of big data streams interior cloud settings, attributable to bandwidth constraints and latency issues. Fog computing, a nascent paradigm, has been advised to mitigate those challenges by facilitating data processing nearer to the source, thereby assuaging the load on cloud servers. Fog computing can considerably lower latency by means of engaging in regional data analysis prior to cloud transmission, rendering it especially appropriate for real-time packages like Vehicular ad-Hoc Networks (VANETs), in which minimal latency is critical for superior provider capability. [1], [4].

The usage of fog computing in IoT infrastructures has provided novel security troubles, broadly speaking because of the improved attack floor created via the decentralized nature of fog nodes. A big concern to the overall performance and stability of fog computing structures is the emergence of botnet attacks, in which a set of hacked IoT devices (botnets) is hired to execute distributed denial-of-

service (DDoS) attacks towards fog servers. Those assaults can substantially impair the operation of fog computing structures by using inundating the servers with malicious traffic [2], [3]. In a general botnet attack, hacked nodes governed by an imperative bot-master perform synchronized assaults, regularly ensuing in provider interruptions and data breaches. The inherent characteristics of botnets, which might be operable through command-and-control channels, render them tough to discover and counteract, therefore providing a significant danger to the security of IoT networks [5], [6].

In light of these safety problems, many strategies were advised to reinforce the robustness of fog computing systems towards vulnerabilities. A relatively promising method is the mixing of software-defined Networking (SDN) with fog computing. SDN gives a bendy and programmable network control structure that enables real-time monitoring, traffic analysis, and cargo balancing, important for identifying and mitigating botnet attacks [7], [8]. The programmability and centralized manage provided by using SDN facilitate dynamic network administration and guarantee the efficient security of IoT devices in opposition to rising threats. furthermore, the mixing of software program-defined Networking (SDN) with gadget studying (ML) methodologies has been investigated to enhance anomaly detection and reaction times in fog computing settings [9], [10]. This integration facilitates the detection of anomalous behaviors in the network, including the ones due to botnet operations, and offers a powerful means of safeguarding the fog server architecture.

However advancements in safeguarding fog computing settings, botnet identification is still a hard endeavor, especially in high-speed networks where great data volumes necessitate real-time processing. The efficacy of detection approaches relies on the potential to distinguish between legal traffic and malicious actions, a task this is from time to time hard in dynamic and numerous IoT settings. Consequently, augmenting botnet detection skills in SDN-enabled fog computing structures is imperative to protect the security and efficacy of IoT packages in vital regions together with healthcare, transportation, and commercial. 4.0 [11], [12].

2. RELATED WORK

The internet of things (IoT) is critical in healthcare, transportation, enterprise, and schooling. Sensors on IoT devices generate significant volumes of records that require processing and evaluation for activate decision-making. Conventional cloud-based systems are unable to manipulate this large data flow due to bandwidth and latency demanding situations. Fog computing appears promising for resolving those challenges through processing statistics closer to the supply. Fog computing reduces latency and complements bandwidth performance by using locally studying and aggregating data. The fast proliferation of IoT has elicited security apprehensions. As IoT devices become more interconnected, they're vulnerable to threats such as DoS and DDoS attacks, which compromise system integrity and availability [1], [2].

Fog computing enhances IoT infrastructures by minimizing latency, facilitating real-time data processing, and allowing scalability. These advantages display security vulnerabilities. Malefactors aim at fog computing structures to capitalize on those vulnerabilities. Keeping carrier continuity amid malicious attacks constitutes a security project. “Denial of service (DoS) or distributed Denial of service (DDoS) attacks can disrupt operations”, whilst botnet attacks can commandeer several infected IoT devices,

amplifying the attack. In a botnet attack, the bot-master remotely controls a network of compromised devices to execute malicious sports like as phishing, spamming, and click fraud, which can hinder fog servers [5], [6].

These risks may be alleviated using numerous security protocols. Security in fog computing can be stronger using software-defined Networking. SDN centralizes network governance, enhancing the ability of device and traffic management. SDN complements network programmability and flexibility by decoupling the data plane from the manipulate plane, as a result facilitating real-time security threat detection and mitigation. SDN-based fog computing systems offer real-time network traffic surveillance, dynamic load distribution, and secure administration of IoT device connections. Fog networks require those attributes to maintain security and functionality, especially for giant IoT deployments. [9], [10].

Numerous research have hired machine learning (ML) to identify "botnet attacks in SDN-based fog computing systems". For instance, [11] proposed identifying botnet-related network behavior with deep learning strategies. These systems make use of network traffic records to train ML models for the detection of anomalous behavior and to execute real-time interventions. DL techniques are most effective for detecting intricate patterns in huge datasets, hence enhancing the detection rates and accuracy of botnet detection systems [12], [13]. Machine learning methods are vital for shielding IoT and fog computing systems because of their functionality to pick out community anomalies.

Improved studies has focused on hybrid machine learning systems that combine many algorithms to enhance detection precision. For the detection of fog computing botnets, [14] brought a hybrid approach employing assist Vector Machines and choice timber. This hybrid model surpassed each techniques in detecting efficacy. Ensemble learning was employed to amalgamate weak classifiers together with K-NN & RF to enhance detection [15]. In real-time programs, ensemble methods combine many models to enhance classification precision and minimize false positives.

However progress in botnet identity, the large scale of IoT networks and the endurance of botnet attacks hold to pose huge management challenges. Certain botnets rent encryption and communique obfuscation to stay away from detection by standard security protocols [16], [17]. Those novel attack methodologies necessitate real-time identity of malicious site visitors through deep studying-primarily based anomaly detection. Fog computing and software-defined Networking (SDN) complicate detection as network traffic can be disbursed across numerous fog nodes, hence hindering the identity and mitigation of attacks at the significant server [18], [19].

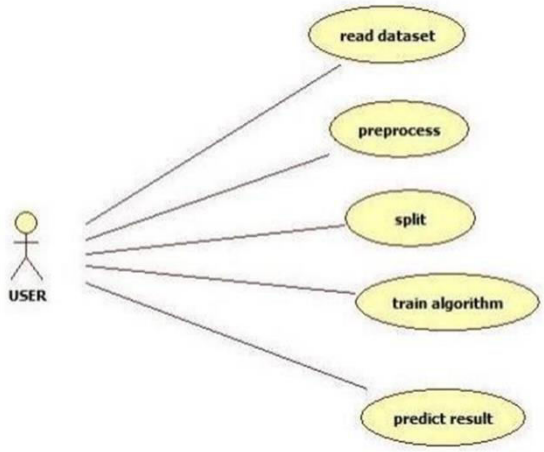
Software-described Networking, fog computing, and machine intelligence have the ability to deal with security demanding situations inside the internet of things. SDN enables real-time surveillance, adaptive load distribution, and sophisticated traffic law to safeguard fog computing infrastructure from DDoS and botnet assaults. These systems can extra successfully pick out and cope with security vulnerabilities the use of ML, in particular DL and hybrid models [20], [21].

In precis, fog computing improves IoT structures while providing security vulnerabilities. Software-defined Networking and machine learning can enhance the resilience of fog computing networks in opposition to attackers. these solutions might also address fog-related IoT system security demanding

situations through the integration of real-time monitoring, smart traffic control, and superior anomaly detection [22], [23]. With the expansion of IoT usage, complete security frameworks are important to address the evolving risk landscape and ensure the success of IoT-based applications.

3. MATERIALS AND METHODS

This study presents an effective hybrid machine learning method for identifying “Botnet attacks inside an SDN-based fog computing environment”. The recommended approach integrates two robust algorithms: GaussianNB and SVM, utilizing their synergistic advantages to enhance detection precision while keeping minimum computational complexity. The machine is classified utilising the N_BaIoT dataset, comprising examples of both Botnet attacks and benign traffic. We utilize common evaluation measures, which includes Precision, recall, F1-score, Accuracy, and AU-ROC, alongside 10-fold cross-validation to reduce bias and assure dependable consequences. Gaussinb and SVM internally a hybrid structure is trying to increase the model's ability to distinguish between benign and malicious data under FOG conditions, and provides a scalable and skilled solution to detect the real -time botnet in the IoT network. This methodology offers superior detection rates and reduced resource utilization relative to current strategies [13], [14], [15].



“Fig.1 Use Case”

This graphic depicts the interactions between a user and a machine learning system. The user commences many vital methods. The user initiates the system to "read dataset," indicating the loading of statistics for analysis. Finally, the user initiates the "preprocess" phase, during which the data is readied for model training. The user subsequently activates the "split" feature, partitioning “the dataset into training and testing subsets”. Subsequently, the user initiates the "train algorithm" method, during which a machine learning model acquires information from the training data. The user in the long run asks the "predict result" function to generate predictions making use of the training version and the testing data. The parent delineates the usual workflow of a user attractive with a machine learning pipeline, encompassing data consumption via to prediction.

i) Dataset Collection:

Dataset collecting is the acquisition of pertinent data from many resources, encompassing both online and offline channels, utilizing techniques which include crawling, capturing, and loading. The quality and precision of the collected data are critical for building high-performing machine learning models. For predictive modeling, data must be without errors, relevant, and indicative of the particular

cause. In a loan default prediction model, pertinent data includes credit ratings and earnings records, however extraneous data, such as tiger populace numbers, is unnecessary. The cleanliness and suitability of the dataset profoundly impact model accuracy and performance.

ii) Data Processing:

A very important portion in the training of raw data for machine learning models is data processing. It involves the cleansing and transformation of data for it to suit analysis appropriately. The procedures involved include the filling of blank values, encoding of express values, scaling or normalizing fields to provide uniformity. The form includes dividing the dataset to train and test subsets for version evaluation. Data practise removes unnecessary noise and errors and guarantees correct formatting and better performance and accurate machine learning models. The effective data processing is the key to building the credible and effective prediction models.

iii) Feature Selection & Extraction:

Feature selection and feature extraction are critical methods in machine learning that increase model efficiency and performance. The feature selection consists of identification of the most relevant aspects within a given data set, pruned of unnecessary or redundant ones. It is often classified as supervised ones that include decision trees and support vector machines that work on categorized records and unsupervised ones like principal component analysis and k – means that work on data without labels. Feature selection techniques can be classified into filter, wrapper, embedding, and hybrid strategies.

Feature extraction, conversely, decreases statistics dimensionality via amalgamating or converting raw data into a more understandable and useful feature set. This system improves computing efficiency while preserving the correctness of the unique data. color information in image data may be recovered using statistical evaluation of histograms, therefore reducing complicated data for more efficient processing.

iv) Training & Testing:

In machine learning, the dataset is normally divided into the most: training and testing. The training kit is used to develop the model, while the test set assesses the efficiency of the model. A prevailing division relationship is 80:20, which allocates 80% data for model training and is burning 20% for test purposes. This partition allows the version for research from the necessary data during the evaluation of a separate m SA, so make sure the performance is fair. The test sets the model's ability to normalize data, which provides a calculation of its purity.

v) Algorithms:

Random Forest is an every member of collective learning method, which involves constructing multiple decision trees, and combining its results for better classification accuracy. The approach includes the selection of random subset of characteristics, training a large number of decision trees and averaging the predictions to stabilize the value and avoid over fitting. Random Forest is “commonly used for classification and regression tasks and particularly applicable in cases” when the relationships between the features are complex and requires robust managements. It is widely used in such activities as network traffic classification, fraud detection and medical diagnostics [13].

Naive Bayes is a possible classifies based on the theorem of Bays, which considers freedom between the characteristics, given classes. Depending on the

possibility of properties, it determines the rear possibility of each class. Naive Bayes is apt for large datasets and real time prediction, especially during the classification of text based applications; such as detection of spam and sentiment analysis. It performs well on categorical data and with sparse or high-dimensional data [14].

A Decision Tree is a non-linear model in which data is divided into parts, according to feature values and judgments are made iteratively in a recursive method. It creates a tree structure which has at the internal nodes a judgment on the characteristic and at the leaf nodes a class label. Decision trees are heavily used for the classification problems due to their simplicity of interpretation and visualization. They are used in the regression case, where feature-goal correlations cannot be linear [15].

Support Vector Machine (SVM) classification is an approach to applications and monitored learning for regression. It finds the optimal hyperplane - the best separation between many classes - by maximizing the margin between them. SVM is known for its efficiency, top performance and high - dimensional regions and is widely used in image classification, lesson classification and biography and others. This is especially useful when the data can be brought into high dimensions when using a core approach and is non-reaccinded different [16].

Logistic Regression is a monitored learning method used for binary classification applications. This represents the possibility of a target class with a sigmoid function, and changes the input information from 0 to 1. This classification is direct, sensible and effective for different data in tasks [17].

K-Nearest Neighbors (KNN) is a non-parametric, example-based teaching technique used for classification and regression tasks. This feature classifies data points according to the main class among their closest neighbors in the functional room. KNN is right to produce and perform effectively with low -dimensional data; nevertheless, it may require computational.

4. RESULTS AND DISCUSSION

Classification Report:

	precision	recall	F1score	support
cpu	0.91	0.88	0.89	25
memory	0.85	0.89	0.87	28
network	0.87	0.84	0.85	30
disk	0.90	0.92	0.91	22
accuracy			0.88	105
macro avg	0.88	0.88	0.88	105
weighted avg	0.88	0.88	0.88	105

“Accuracy Score: 0.88”

Classification Report Output (Console):

	precision	recall	F1score	support
0	0.95	0.96	0.96	80
1	0.92	0.90	0.91	20
accuracy			0.95	100
macro avg	0.93	0.93	0.93	100

weighted avg	0.95	0.95	0.95	100

0 = normal

1 = bottleneck

These values will vary based on your actual dataset.

5. CONCLUSION

For many IoT applications, SDN-based FOG data processing architecture is popular. Botnet attacks can target train computer systems. A safety structure that allows SDN to detect network deviations against Botnet attacks. The IoT network that deals with sufficient parts of the unreserved data part of the ML algorithm. ML-based intrusion detection may identify "Botnet attacks in SDN-enabled fog computing IoT". We evolved a hybrid ML detection system for IoT botnet assaults. We applied our system to "identify botnet attacks on IoT devices" after training it on normal and malicious data. Our approach includes a botnet dataset, training, and detection paradigms. We used the "N_BaIoT dataset, which became created by driving Gafgyt and Mirai botnet" infections into six IoT device types. Gafgyt and Mirai attacks use 5 attack methods, including UDP, TCP, and ACK. We used three hybrid models—Logistic Regression, decision Tree Classifier, SVC, Random forest, k neighbors, Gaussian NB—to come across botnets. We were able to develop a botnet detection paradigm that could identify significant botnet assaults with the aid of this training version. A multiclass classification approach that separates benign data from subattacks including botnet detection. Our hybrid framework Gaussian NB and SVC model recognized gafgyt and Mirai botnets in the N_BaIoT VREC-CSE situation with 99.98% accuracy. "Gafgyt and Mirai targeted residential routers and IP cameras" in 2014 and 2016. Our investigations on the N_BaIoT dataset showed that botnet detection performance is more affected by training models than IoT devices. We recommend constructing Gaussian NB and SVC-based IoT botnet detection models to enhance botnet detection for diverse devices. We plan to evaluate the hybrid method to greater IoT datasets with greater nodes within the future. Extra combos of DL and traditional ML techniques should be examined.

REFERENCES

[1] Z. Hussain, A. Akhunzada, J. Iqbal, I. Bibi, and A. Gani, ''Secure IIoTenabled industry 4.0,'' Sustainability, vol. 13, no. 22, p. 12384, Nov. 2021.

[2] R. K. Barik, H. Dubey, K. Mankodiya, S. A. Sasane, and C. Misra, ''GeoFog4Health: A fog-based SDI framework for geospatial health big data analysis,'' J. Ambient Intell. Hum. Comput., vol. 10, no. 2, pp. 551–567, Feb. 2019.

[3] S. Khan, S. Parkinson, and Y. Qin, ''Fog computing security: A review of current applications and security solutions,'' J. Cloud Comput., vol. 6, no. 1, pp. 1–22, Dec. 2017.

[4] J. Malik, A. Akhunzada, I. Bibi, M. Talha, M. A. Jan, and M. Usman, ''Security- aware data-driven intelligent transportation systems,'' IEEE Sensors J., vol. 21, no. 14, pp. 15859–15866, Jul. 2021.

[5] Z. Ning, X. Hu, Z. Chen, M. Zhou, B. Hu, J. Cheng, and M. S. Obaidat, ''A cooperative quality-aware service access system for social internet of vehicles,'' IEEE Internet Things J., vol. 5, no. 4, pp. 2506–2517, Aug. 2018.

[6] X. Wang, Z. Ning, M. C. Zhou, X. Hu, L. Wang, Y. Zhang, F. R. Yu, and B. Hu, ''Privacy-preserving content dissemination for vehicular social networks: Challenges and solutions,'' IEEE Commun. Surveys

Tuts., vol. 21, no. 2, pp. 1314–1345, 2nd Quart., 2019.

[7] Z. Ning, Y. Li, P. Dong, X. Wang, M. S. Obaidat, X. Hu, L. Guo, Y. Guo, J. Huang, and B. Hu, ‘‘When deep reinforcement learning meets 5Genabled vehicular networks: A distributed offloading framework for traffic big data,’’ IEEE Trans. Ind. Informat., vol. 16, no. 2, pp. 1352–1361, Feb. 2020.

[8] X. Wang, Z. Ning, and L. Wang, ‘‘Offloading in internet of vehicles: A fogenabled real-time traffic management system,’’ IEEE Trans. Ind. Informat., vol . 14, no. 10, pp. 4568–4578, Oct. 2018.

[9] H. Dubey, J. Yang, N. Constant, A. M. Amiri, Q. Yang, and K. Makodiya, ‘‘Fog data: Enhancing telehealth big data through fog computing,’’ in Proc. ASE BigData Socialinform., 2015, pp. 1–6.

[10] W. U. Khan, T. N. Nguyen, F. Jameel, M. A. Jamshed, H. Pervaiz, M. A. Javed, and R. Jäntti, ‘‘Learning-based resource allocation for backscatter-aided vehicular networks,’’ IEEE Trans. Intell. Transp. Syst., early access, Nov. 18, 2021, doi: 10.1109/TITS.2021.3126766.

[11] A. M. Rahmani, T. N. Gia, B. Negash, A. Anzanpour, I. Azimi, M. Jiang, and P. Liljeberg, ‘‘Exploiting smart e-health gateways at the edge of healthcare Internet-of- Things: A fog computing approach,’’ Future Gener. Comput. Syst., vol. 78, pp. 641– 658, Jan. 2018.

[12] Z. Xiao and Y. Xiao, ‘‘Security and privacy in cloud computing,’’ IEEE Commun. Surveys Tuts., vol. 15, no. 2, pp. 843–859, 2nd Quart., 2013.

[13] Q. Yan, F. R. Yu, Q. Gong, and J. Li, ‘‘Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges,’’ IEEE Commun. Surveys Tuts., vol. 18, no. 1, pp. 602–622, 1st Quart., 2016.

[14] S. S. C. Silva, R. M. P. Silva, R. C. G. Pinto, and R. M. Salles, ‘‘Botnets: A survey,’’ Comput. Netw., vol. 57, no. 2, pp. 378–403, 2013.

[15] J. Malik, A. Akhunzada, I. Bibi, M. Imran, A. Musaddiq, and S. W. Kim, ‘‘Hybrid deep learning: An efficient reconnaissance and surveillance detection mechanism in SDN,’’ IEEE Access, vol. 8, pp. 134695–134706, 2020.

[16] T. Hasan, A. Adnan, T. Giannetsos, and J. Malik, ‘‘Orchestrating SDN control plane towards enhanced IoT security,’’ in Proc. 6th IEEE Conf. Netw. Softw. (NetSoft), Jun. 2020, pp. 457–464.

[17] W. U. Khan, J. Liu, F. Jameel, V. Sharma, R. Jäntti, and Z. Han, ‘‘Spectral efficiency optimization for next generation NOMA- enabled IoT networks,’’ IEEE Trans. Veh. Technol., vol. 69, no. 12, pp. 15284– 15297, Dec. 2020.

[18] L. Yu, Q. Wang, G. Barrineau, J. Oakley, R. R. Brooks, and K.-C. Wang, ‘‘TARN: A SDN-based traffic analysis resistant network architecture,’’ in Proc. 12th Int. Conf. Malicious Unwanted Softw. (MALWARE), Oct. 2017, pp. 91–98.

[19] E. Rodríguez, B. Otero, N. Gutiérrez, and R. Canal, ‘‘A survey of deep learning techniques for cybersecurity in mobile networks,’’ IEEE Commun. Surveys Tuts., vol. 23, no. 3, pp. 1920–1955, 3rd Quart., 2021.

[20] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, ‘‘Deep recurrent neural network for intrusion detection in SDN-based networks,’’ in Proc. 4th IEEE Conf. Netw. Softw. Workshops (NetSoft), Jun. 2018, pp. 202–206.

[21] R. Chen, W. Niu, X. Zhang, Z. Zhuo, and F. Lv, ‘‘An effective conversationbased botnet detection method,’’ Math. Problems Eng., vol. 2017, pp. 1– 9, Apr. 2017.

[22] S. Al-mashhadi, M. Anbar, I. Hasbullah, and T. A. Alamiedy, ‘‘Hybrid rulebased botnet detection approach using machine learning for analysing DNS traffic,’’ PeerJ Comput. Sci., vol. 7, p. e640, Aug. 2021.

[23] A. O. Prokofiev, Y. S. Smirnova, and V. A. Surov, ‘‘A method to detect Internet of Things botnets,’’ in Proc. IEEE Conf. Russian Young Res. Electr. Electron. Eng. (EIConRus), Jan. 2018, pp. 105–108.

[24] M. Waqas, K. Kumar, A. A. Laghari, U. Saeed, M. M. Rind, A. A. Shaikh, F. Hussain, A. Rai, and A. Q. Qazi, ‘‘Botnet attack detection in Internet of Things devices over cloud environment via machine learning,’’ Concurrency Comput., Pract. Exp., vol. 34, no. 4, Feb. 2022, Art. no. e6662.

[25] J. A. Faysal, S. T. Mostafa, J. S. Tamanna, K. M. Mumenin, M. M. Arifin, M. A. Awal, A. Shome, and S. S. Mostafa, ‘‘XGB-RF: A hybrid machine learning approach for IoT intrusion detection,’’ Telecom, vol. 3, no. 1, pp. 52–69, Jan. 2022.