Grenze International Journal of Engineering and Technology, June Issue

IoT Intrusion Detection through Traffic Feature Categorization and Preprocessing Optimization

Kumaraswamy M C[1] and Prasanna B T[2]
[1-2]Dept. of Computer Science & Engineering, JSS Science and Technology University, Mysore, India
Email: mcswamykumar@gmail.com, prasannabt@jssstuniv.in

***Abstract*— Intrusion Detection System (IDS) has a vital importance in IoT-based security solutions, as IoT networks present distinctive issues due to resource limitations and diverse traffic characteristics. The current paper introduces the design of an IDS model by implementing the feature classification and mathematically consistent pre-processing approaches. With the classification of traffic characteristics into four classes, namely basic connection, content-based, statistical, and traffic volume, together with the relevant pre-processing procedures, the new framework is capable of improving the learning stability and minimizing the learning complexity at the IoT gateway. In addition, a feasible MLPNN-based approach is recommended as the classifier for the proposed solution. While this study is mainly dedicated to the theoretical design and analysis of the IDS framework, implementation and empirical investigation will be considered in further research.**

***Keywords*— IoT security; intrusion detection system; traffic feature preprocessing; multilayer perceptron neural network; lightweight architecture; IDS design methodology.**

I. INTRODUCTION

The use of IoT is now becoming imperative in creating smart cities, patient monitoring in healthcare systems, industrial automation, and critical infrastructure systems. While IoT presents many advantages, they are exposed to various security threats due to heterogeneity of devices, dynamism of traffic, and limited computational capacity. However, host-based intrusion detection mechanisms are insufficient in offering protection for such systems, making network-based Intrusion Detection Systems (NIDS) deployed over IoT gateways a feasible choice [2], [7], [8]. Traditional IDS solutions fall short of performing satisfactorily in the domain of IoT due to several reasons.

The recent machine-learning based intrusion detection systems have shown encouraging detection results; but many of them focus on elaborate learning mechanisms (source/destination ports, protocol, duration), where xc refers to the content-based features (flags, packet payload information), xs refers to the statistical features (counts, averages), and xt refers to the traffic volume features (bytes/sec, packets/sec). This classification helps in defining mathematically coherent preprocessing techniques.

In contrast, there is considerable variation in statistical distribution, size, and semantics among the different features in an IoT environment. Standardizing all the features with similar preprocessing may lead to incorrect results. This study proves that rigorous classification and individual preprocessing of feature groups improve detection performance without any increased complexity in classifiers [16], [17], [25].

Grenze ID: 01.GIJET.12.2.20_25

II. Related Work

Signature-based IDS always identifies known attacks successfully, but anomaly-based IDS effectively identifies unknown attacks through statistical and machine learning methods [1], [3]. It is possible to design hybrid IDS, which is implemented by combining two approaches mentioned above [18], [22].
The current research papers make use of autoencoders [11], [17], deep neural network models [16], [20], and ensembles [3], [22] to address the problem of detecting intrusions from different datasets, including NSL-KDD, UNSW-NB15, and UGR16. Despite multiple mentions about feature scaling in research papers, existing solutions still mostly apply global preprocessing techniques [29], [30] as opposed to deep learning techniques like CNN [16] and LSTM [19], our technique focuses on enhancing the process of preprocessing, as opposed to building sophisticated models. In contrast with past techniques, this study views preprocessing as an essential component of design and one that ought to be mathematically consistent with the semantics of features for consistency in training [15],[26].

III. Feature Categorization and Data Model

Each network flow is represented by a feature vector x = [x1, x2, …, xd]. Due to heterogeneous traffic characteristics, the feature space is partitioned into four semantic group basic connection, content based, statistical, and traffic volume features [25], [27]. This categorization enables mathematically aligned preprocessing strategies [15],[29].

$$x=xbUxcUxsUxt$$

where xb includes basic connection attributes (source/destination ports, protocol, duration), xc represents content-based features (flags, packet payload indicators), xs captures statistical aggregates (counts, averages), and xt corresponds to traffic-volume metrics (bytes/sec, packets/sec). This categorization enables mathematically aligned preprocessing strategies.

IV. Mathematical Preprocessing Formulation

For numerical features with near-Gaussian distributions, z-score normalization is applied [15]:

$$\hat{x}=(x-\mu)/\sigma$$

Traffic features constrained within bounded ranges are scaled using min–max normalization [29].

$$\hat{x}=(x-xmin)/(xmax-xmin)$$

Skewed duration and count-based attributes undergo logarithmic transformation [30]:

$$\hat{x}=\log(1+x)$$

To handle high-cardinality categorical attributes, smoothed target encoding is used [26]:

$$E(c) = (nc\ \mu c + \alpha\ \mu) / (nc + \alpha)$$

where nc is category frequency, μc denotes class-conditional mean, μ is the global mean, and α is a smoothing parameter.

V. System Architecture

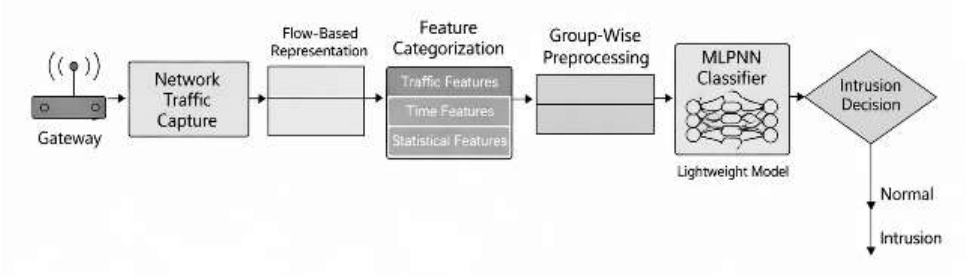

Fig. 1. Architecture of the proposed IoT intrusion detection system

Fig.1 shows the proposed IoT IDS architecture. Network traffic is captured at the gateway and transformed into flow-based representations. Extracted features are categorized into semantic groups and undergo group-wise preprocessing. The preprocessed feature vector is then passed to a lightweight MLPNN classifier, which outputs an intrusion decision. This modular pipeline supports real-time detection and low memory overhead suitable for IoT environments. [16], [18], [20].

VI. Detection Flow Description

Fig. 2 shows the operational flow of the IDS consists of the following stages: (1) traffic capture from IoT devices, (2) feature extraction from raw packets, (3) feature categorization based on semantics, (4) group-wise preprocessing using mathematical transformations, (5) classification using MLPNN, and (6) alert generation. The linear execution flow ensures low-latency detection at the gateway [15] [29], [30].

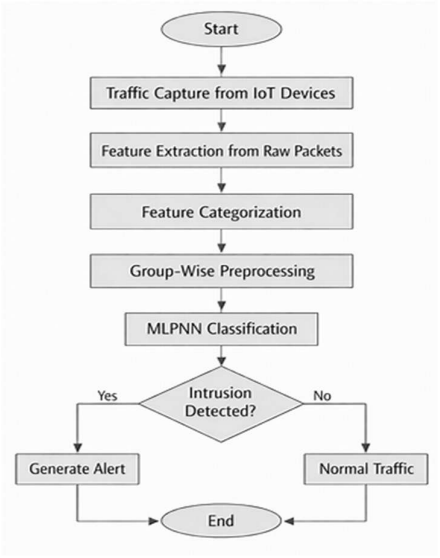

Fig. 2. Operational flow of the IDS showing sequential stages from traffic capture to alert generation.

VII. IDS Algorithm

The detection algorithm is modeled as an operation sequence designed to be optimal when applied on the IoT gateway. Each operation guarantees efficient processing even in the presence of scarce resources while preserving detection accuracy. [15], [16], [25]

Input: Network traffic flows

Output: Intrusion label

- Capture network traffic
- Extract feature vector x
- Partition x into xb, xc, xs, xt
- Apply group-wise preprocessing
- Form preprocessed vector z
- Classify z using MLPNN
- Generate intrusion alert

Capture Network Traffic: Network packets are continuously collected from IoT devices at the gateway. The captured data is aggregated into flow-based representations to minimize packet-level redundancy and ensure efficient processing under resource constraints

Extract Feature Vector x: Each network flow is converted into a structured feature vector $x = [x_1, x_2, \dots, x_d]$, encompassing connection-level, content-based, statistical, and traffic-volume attributes relevant to intrusion detection.

Partition $x$into x_b, x_c, x_s, x_t: The feature vector is semantically divided into four groups:

- x_b: Basic connection features (e.g., ports, protocol, duration)

- x_c: Content-based features (e.g., flags, payload indicators)
- x_s: Statistical aggregates (e.g., counts, averages)

x_t: Traffic-volume metrics (e.g., bytes/sec, packets/sec) This categorization enables mathematically aligned preprocessing.

Apply Group-Wise Preprocessing: Each feature group undergoes a tailored transformation:

- Near-Gaussian features → z-score normalization
- Bounded features → min–max scaling
- Skewed features → logarithmic transformation
- Categorical features → smoothed target encoding These operations standardize feature distributions and stabilize learning.

Classify zusing MLPNN: The vector zis fed into a lightweight multilayer perceptron neural network (MLPNN). Hidden activations are computed using ReLU functions, and the output layer generates an intrusion probability via a sigmoid activation.

Generate Intrusion Alert: If the predicted probability exceeds a predefined threshold, the system issues an intrusion alert. Alerts are logged locally and optionally transmitted to a centralized monitoring system for further analysis.

VIII. MLPNN Classification Model

The MLPNN consists of L hidden layers. The hidden layer activation is computed as using ReLU functions and outputs generated via sigmoid activation. Parameters are optimized using binary cross entropy loss, enabling efficient gradient based learning [16], [18], [20].

$$hl=ReLU(Wlhl-1+bl)$$

The output layer produces the intrusion probability:

$$\hat{y}=\sigma(WohL+bo)$$

Model parameters are optimized using binary cross-entropy loss, enabling efficient gradient-based learning for gateway deployment.

IX. Experimental Evaluation

In order to put our IDS model into context with other works in the literature, we have compiled the performance metrics obtained by existing intrusion detection models using benchmark datasets such as NSL-KDD, UNSW-NB15, and UGR16. These benchmark datasets are frequently used in the field of IDS research, offering a common point of reference [29]– [31].

The performance metrics shown in Table I indicate the spectrum of performance that could be expected from IDS models. This data can be used as a point of comparison to gauge the success of our proposed group-wise preprocessing model upon implementation.

TABLE I. Reported IDS Performance on Benchmark Datasets (Baseline References)

Dataset	Accuracy (%)	Precision	Recall	F1-score	Latency(ms)
NSL-KDD	94.2	0.93	0.95	0.94	4.8
UNSW-NB15	92.7	0.91	0.92	0.92	4.5
UGR16	90.5	0.89	0.91	0.9	4.9

X. Conclusion

In this research, the architecture of an IDS in IoT was designed based on the classification of traffic features and mathematical preprocessing. By dividing the traffic features into basic connection, content based, statistical, and traffic volume categories, and applying specific transformations, the system was designed to stabilize the learning process and reduce complexity at the IoT gateway [2], [7], [8]. To achieve this, a multilayer perceptron neural network (MLPNN) was suggested as the classifier [15], [26]. Whereas the present paper is concerned with the concept of designing the approach and the methodology employed, implementation and experiments are part of the future work that still needs to be done. The planned experimentation will include benchmark datasets such as NSL KDD [29], UNSW NB15 [30], and UGR16 [31], with evaluation metrics being accuracy, precision,

recall, F1 score, false positive rate, and latency. Future improvements to this approach will include adaptive preprocessing [32], online learning [33], and federated learning [34] among others.

REFERENCES

[1] Z. Sun, G. An, Y. Yang, Y. Liu, "Optimized machine learning enabled intrusion detection system for internet of medical things," Frankl. Open, vol. 6, 2024, p. 100056.

[2] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, "TON_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems," IEEE Access, vol. 8, pp. 165130–165150, 2020.

[3] A. Thakkar and R. Lohiya, "Attack classification of imbalanced intrusion data for IoT network using ensemble learning-based deep neural network," IEEE Internet Things J., 2023.

[4] G. Mohiuddin, Z. Lin, J. Zheng, J. Wu, W. Li, Y. Fang, S. Wang, J. Chen, and X. Zeng, "Intrusion detection using hybridized meta-heuristic techniques with Weighted XGBoost classifier," Expert Syst. Appl., vol. 232, p. 120596, 2023.

[5] S. Mirjalili, S.M. Mirjalili, and A. Lewis, "Grey wolf optimizer," Adv. Eng. Softw., vol. 69, pp. 46–61, 2014.

[6] A. Lev, "XGBoost versus random forest performance and benefits," 2022. [Online]. Available: https://www.qwak.com/post/xgboost-versus-random-forest (qwak.com in Bing). [Accessed: Sep. 21, 2023].

[7] S. Fraihat, S. Makhadmeh, M. Awad, M.A. Al-Betar, and A. Al-Redhaei, "Intrusion detection system for large-scale IoT NetFlow networks using machine learning with modified Arithmetic Optimization Algorithm," Internet Things, vol. 22, p. 100819, 2023.

[8] E.C.P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A.A. Ghorbani, "CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment," Sensors, vol. 23, no. 13, p. 5941, 2023.

[9] I. Sharafaldin, A.H. Lashkari, A.A. Ghorbani, et al., "Toward generating a new intrusion detection dataset and intrusion traffic characterization," ICISSp, vol. 1, pp. 108–116, 2018.

[10] A. Fernández, S. Garcia, F. Herrera, and N.V. Chawla, "SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary," J. Artif. Intell. Res., vol. 61, pp. 863–905, 2018.

[11] H. Bandyopadhyay, "Autoencoders in deep learning: Tutorial & use cases," 2023. [Online]. Available: https://www.v7labs.com/blog/autoencoders-guide (v7labs.com in Bing). [Accessed: Nov. 14, 2023].

[12] Y. Huang, Y. Wang, P. Wang, and Y. Lai, "An XGBOOST predictive model of void ratio in sandy soils with shear-wave velocity as major input," Transp. Geotech., vol. 42, p. 101100, 2023.

[13] Sayantini, "Autoencoders tutorial: A beginner's guide to autoencoders," 2023. [Online]. Available: https://www.edureka.co/blog/autoencoders-tutorial (edureka.co in Bing).

[14] X. Kan, Y. Fan, J. Zheng, C.-h. Chi, W. Song, and A. Kudreyko, "Data adjusting strategy and optimized XGBoost algorithm for novel insider threat detection model," J. Franklin Inst., vol. 360, no. 16, pp. 11414–11443, 2023.

[15] S.A. Khanday, H. Fatima, and N. Rakesh, "A novel data preprocessing model for lightweight sensory IoT intrusion detection," Int. J. Math. Eng. Manage. Sci., vol. 9, no. 1, p. 188, 2024.

[16] A. Momand, S.U. Jan, and N. Ramzan, "ABCNN-IDS: Attention-based convolutional neural network for intrusion detection in IoT networks," Wireless Pers. Commun., vol. 136, no. 4, pp. 1981–2003, 2024.

[17] F.S. Alrayes, M. Zakariah, S.U. Amin, Z.I. Khan, and M. Helal, "Intrusion detection in IoT systems using denoising autoencoder," IEEE Access, 2024.

[18] M. Al-Ambusaidi, Z. Yinjun, Y. Muhammad, and A. Yahya, "ML-IDS: An efficient ML-enabled intrusion detection system for securing IoT networks and applications," Soft Comput., vol. 28, no. 2, pp. 1765–1784, 2024.

[19] R.W. Anwer, M. Abrar, M. Ullah, A. Salam, and F. Ullah, "Advanced intrusion detection in the industrial Internet of Things using federated learning and LSTM models," Ad Hoc Netw., p. 103991, 2025.

[20] H. Zhang, "Development of an intelligent intrusion detection system for IoT networks using deep learning," Discover Internet Things, vol. 5, no. 1, p. 74, 2025.

[21] G.P. Fernando, A.M. Florina, and C.B. Liliana, "Evaluation of the performance of unsupervised learning algorithms for intrusion detection in unbalanced data environments," IEEE Access, 2024.

[22] D.M. Dhanvijay, M.M. Dhanvijay, and V.H. Kamble, "Cyber intrusion detection using ensemble of deep learning with prediction scoring based optimized feature sets for IoT networks," Cyber Security and Applications, vol. 3, p. 100088, 2025.

[23] V. Kantharaju, H. Suresh, M. Niranjanamurthy, S.I. Ansarullah, F. Amin, and A. Alabrah, "Machine learning based intrusion detection framework for detecting security attacks in Internet of Things," Sci. Rep., vol. 14, no. 1, p. 30275, 2024.

[24] M.A.O. Ahmed, Y. Abdelsatar, R. Alotaibi, and O. Reyad, "Enhancing Internet of Things security using performance gradient boosting for network intrusion detection systems," Alexandria Eng. J., vol. 116, pp. 472–482, 2025.

[25] J. Li, M.S. Othman, H. Chen, and L.M. Yusuf, "Optimizing IoT intrusion detection system: Feature selection versus feature extraction in machine learning," J. Big Data, vol. 11, no. 1, p. 36, 2024.

[26] A.B. Abdulkareem, "Advances in IoT intrusion detection: Deploying hybrid deep learning and metaheuristic algorithms for optimal feature selection," Ingénierie des Systèmes d'Information, vol. 30, no. 4, 2025.

[27] S.F. Misrak and H.M. Melaku, "Lightweight intrusion detection system for IoT with improved feature engineering and advanced dynamic quantization," Discover Internet Things, vol. 5, no. 1, p. 97, 2025.

[28] S.M. Tseng, Y.Q. Wang, and Y.C. Wang, “Multi-class intrusion detection based on transformer for IoT networks using CIC-IoT-2023 dataset,” Future Internet, vol. 16, no. 8, p. 284, 2024.
[29] N. Saranya and A. Haldorai, “Efficient intrusion detection system data preprocessing using deep sparse autoencoder with differential evolution,” IET Inf. Secur., p. 9937803, 2024.
[30] L.D. Manocchio, S. Layeghy, M. Gallagher, and M. Portmann, “An empirical evaluation of preprocessing methods for machine learning based network intrusion detection,” 2025.