LIGHTWEIGHT IDS-BASED FEATURE SELECTION ALGORITHM FOR CYBER-PHYSICAL SYSTEMS & IOE DEVICES

Sunil Kaushik[1], Akashdeep Bhardwaj[2], Saud Aljaloud[3], Naif Alsharabi[3]

[1]Indus Towers, Gurgaon 122001, Haryana, India
[2]Centre for Cybersecurity, School of Computer Science, UPES, Dehradun 248007, India
[3]College of Computer Science and Engineering, University of Hail, Hail 81451, Saudi Arabia

ABSTRACT

The quick spread of Internet connections has instigated the revolutionary age of Cyber-Physical Systems (CPS) and Internet of Everything (IoE) devices. The IOE and CPS devices are the cornerstone of Industry 4.0. which is centred on Machine-to-Machine (M2M) communication. IoE and CPS devices are used in hostile environments and have limited computing and energy resources. Criticality and dependence of the Internet have exposed IoE and CPS systems to cyber-attacks. Thus, to prevent any damage, these systems require a competent and lightweight intrusion detection system (IDS). The current research recommends a novel IDS built upon a new feature selection algorithm which can identify entropy reducing and highly statistical reliable features from a dataset. The proposed feature selection technique showed significant improvements in performance measures for several classifiers. Proposed IDS with the IOTID20 dataset demonstrated that the accuracy and performance metrics exceeded 99%. The trustworthiness of the proposed IDS is further supported by its constant efficacy on the NSLKDD dataset. The proposed IDS is found to be competitive with all previous studies in all performance areas. Thus, proposed IDS on novel and innovative feature selection techniques can protect the digital ecosystem and IoE landscapes from cyber-attacks to bolster Industry 4.0.

KEYWORDS

Smart Devices, Threat Intelligence, IoT Vulnerabilities, Intelligent Intrusion Detection, Connected Systems, Feature Selection, IoE Security.

1. INTRODUCTION

Industry 4.0 rely heavily on the Internet of Things (IoT), connecting devices and systems in ways that make daily operations smoother and more efficient [1]. Cybersecurity concerns grow with increasing devices [2,3], which may put data, accessibility, and system performance at risk. These threats have a likelihood of having detrimental impacts on everyday digital lives in addition to enterprises, with effects on both safety and the economy [4,5]. IoT devices, such as industrial equipment to intelligent sensors, are vulnerable to threats like malware, DDoS, and unauthorized access since these devices often run on basic operating systems with very little computational capacity [6.7]. Because these devices are so interconnected, a single breach can ripple across entire networks [8]. Additionally, the complication is strengthened through varied arrangements of connected and wireless communication techniques used by embedded systems with internet access. [9,10].

To mitigate the growing cybersecurity risks associated with resource-constrained IoT and IoE environments, enterprises increasingly rely on Intrusion Detection Systems (IDS) as a critical defensive mechanism. IDS enables the detection of unauthorized and anomalous activities, including zero-day attacks

[11,12] , by analyzing deviations in network behaviour [13]. However, due to the limited computational and storage capabilities of many IoE devices, deploying conventional IDS remains challenging, thereby necessitating lightweight and adaptive intrusion detection solutions [14,15]. Considering soaring cyber-attacks and the compounding inexplicability of Internet of Everything (IoE) devices, an IDS that can efficiently defend IoT systems whilst consuming the restricted processing resources and storage is required. [6] Suggest that the IDS system can be made lightweight if it has the right feature selection technique, which not only requires lesser computation to analyse but also differentiates between attacks and normal traffic, in other words, requires a lesser training time. Further [60] maintains that the computation time signifies the computational complexity and utilization of computational resources [58]. The proposed MIRCHI framework critically chooses attributes based on the statistical tools such as Chi-square (CHI) and information theory-based tool Mutual Information (MI), such that features and the label share, in addition to the predicted distribution of a feature within a class. This study is different from the other studies because of the following reasons.

- Most existing studies perform feature selection using standard libraries or by fusing multiple methods in separate iterations, increasing computational cost. The MIRCHI algorithm combines these procedures in a single pass by traversing the dataset only once and uses resources more efficiently.
- Additionally, MIRCHI removes redundant or correlated features, and the most relevant features are retained. This optimized feature set improves both efficiency and accuracy in any attack classification.

This research presents the following contributions to the field of IoT and IoE security:

- It proposes a highly accurate, scalable, and lightweight IDS that minimizes training time while consuming minimal computational resources.
- The study introduces a feature selection technique by merging principles from information theory using Mutual Information (MI) and statistics Chi-square (CHI) to improve detection efficiency.
- The study uses machine learning algorithms to develop a dynamic and effective IDS, achieving peak accuracies of 99.81% and 99.58% on the NSLKDD and IoTID20 datasets, respectively.
- The MIRCHI framework's performance is benchmarked against recent studies using these datasets, where it demonstrated comparable or superior accuracy with reduced detection time, underlining its real-world suitability for IoT and IoE environments.

The study is organized into five sections following the introduction. Section 2 analyses related work and showcases the nitty-gritties of feature selection and classification algorithms. Section 3 proposes a new algorithm called MIRCHI. Section 4 gives details of the datasets, methodology, and experimental setup. Section 5 evaluates the results and compares them with other recent studies and concludes with directions for future research.

2. RELATED WORK

In recent years, considerable hard work has been put into addressing cybersecurity challenges in IoT environments. Several studies have used ML and DL techniques for IoT security, and few have analytically studied feature selection to improve attack classification. Li et al. [16] proposed an AE-RF approach to remove irrelevant and redundant features, achieving evaluation on the CICIDS2017 dataset. Lu and Tian [17] used autoencoders to select optimal features, while Safaldin et al. [18] achieved 96% accuracy with a GWO-based filter on NSLKDD and 98% using simple correlation on UNSWNB15.

Liu and Du [25] employed a genetic algorithm for feature selection, showing high accuracy but with high computational cost and long training time. Mushtaq et al. [26] used AE-based techniques, achieving 89% accuracy, though AE methods are computationally heavy [27]. Kumar and Subba [28] applied PCA on the ADFA-WD dataset (accuracy 91%), while Bhayo et al. [29] reported 98% using statistical methods. PCA-based methods on NSLKDD [30] yielded lower accuracy, and GXGBoost with

Fisher-score/genetic methods [31] achieved 99%, but with heavy computation [27]. MOEFS-based selection [32] reached 96% on CICIDS2017, and PCA+SVM [33] obtained 96% on NSLKDD.

Ensemble methods without feature selection reached 77% [34], while hybrid AE-Isolation Forest [35] achieved 81% in ~1150 ms. ADASYN+RENN [36] and ADASYN+DL [37] achieved 86% and 89%, respectively. Chi-square + Bi-LSTM [38] yielded 97% in 156 ms. Information-theory models [39] achieved >99% on IoTID20 and NSLKDD but required 156 ms. Deep learning ensembles [40,41] ranged from 86.2% [42] to 90%, with heavy computation. XAI+RF [43] reached 98% in 34,000 ms, PCA+Bat Optimization [44] 99%, CNN+GRU [45] 98% in 98 ms, and CNN [46] 99.72%. Random Forest feature selection with KNN [47] achieved 98% but took >40,000 ms. Overall, although high accuracies are reported (up to 99.72%), many methods are found to be computationally heavy, stressing the need for lightweight, efficient feature selection and IDS frameworks for IoT and IoE environments.

State-of-the-art studies indicate that challenges related to high dimensionality and feature redundancy continue to be dominant. Recent studies establish that removing redundant features can increase accuracy but require high training time and computational power[48,49]. Many of these machine learning-based IDS interpret deviations from normal patterns as anomalies but are afflicted with misclassification because of redundant features [51,52,53]. These studies stress improvements in cybersecurity for complex IoE and CPS environments. For example, HIDIM [54] throws ordered dependencies and class imbalance in network intrusion detection, improving accuracy and reducing false positives. Blockchain-aided digital twin offloading and privacy-preserving mechanisms [55–56] ensure secure computation and efficient resource use in space-air-ground networks. CALRA [57] provide anonymous, leakage resilient authentication for vehicular crowdsensing. Federated learning incentives for AIoT [59], low-latency UAV communication [60], and the energy-efficient, low-latency EALLR routing model for mobile edge computing [61] demonstrate further innovations in secure, efficient, and optimized IoT systems.

Table 1 summarizes studies using various feature selection techniques. Analysis shows that few recent works validated IDS on IoT-specific datasets. Neural network-based methods [11,24,30,31] require high computational power due to many nodes, while wrapper-based algorithms [24,30] suffer from low convergence and local optima [12]. Filter-based methods [26,28,29] struggle with outliers and nonlinear feature relationships. In contrast, information-centric and statistical techniques [21] are lightweight, efficient, and identify relevant features with minimal computation and training time. Hence, the following gaps were found in the research:

- There are fewer studies around Information Theory and Simple Statistical Techniques to identify the features.
- The increasing attack surface area and inherent complexities require IDS that are less computation-intensive and hence lightweight.
- Very few existing studies are around highly accurate, agile, and lightweight IDS for IoT as well as normal networks.

Table 1: Analysis of recent studies on IDS

Study	Year	Technique Classifier	Dataset	Accuracy	Training Time	Lightweight
[19]	2022	GTO-Wrapper-NN	IoTID20	95.60%	154	No
[20]	2022	DEELM-Wrapper-	NSL-KDD	96.20%	-	No
[21]	2022	SE-Wrapper-DT	NSL-KDD	97.10%	821	No
[22]	2023	LPIO-Wrapper	NSL-KDD	98.60%	-	No
[23]	2023	XGBOOST-Wrapper-CNN	KDDCup99	99%	-	No
[24]	2023	LAAN-Wrapper-NN	IoTID20	98.16%	-	No

[25]	2023	GA-Wrapper-DT	NSL-KDD	99.80%	-	No
[32]	2022	MOEFS-Wrapper-	CICIDS2017	94.30%	-	No
[36]	2023	ADASYN+RENN-NN-CNN	NSLKDD	99.00%	1150	No
[37]	2022	Chi-Wrapper-NN	IoTID20	97.12%	156	No
[39]	2022	IG-GR-RF	IoTID20	99%	158	No
[43]	2023	XAI-Wrapper-RF	IoTID20	99%	34000	No
[45]	2023	GRU-Wrapper-CNN	IoTID20	98%	98	Yes
[47]	2024	RF-Wrapper-KNN	IoTID20	99%	40000	No
[49]	2024	GA-Wrapper-NN	IoTID20	99.41%	82	Yes
[50]	2025	NN-Wrapper-NN	IoTID20	99.00%	85	Yes
[51]	2024	RFL-Wrapper-NN	IoTID20	99.10%	89	Yes

Thus, authors propose a lightweight feature selection which is a hybrid of Information-centric with Statistical Techniques. Thus, it draws the benefits of both effectively, section 3 discusses these techniques in detail, and a new technique is proposed in section 4.

3. IMPORTANT FUNDAMENTAL CONCEPTS

3.1 Statistical Concepts

Mutual Information (MI) of two variables, chosen randomly, indicates the knowledge revealed or the intensity of linkage involved [51]. Assume L and M are two such variables in the specified space; each of these variables has its own pmf (probability mass function) denoted by p(l) and p(m), respectively. Therefore, the PMFs are divided by a gap that is determined by employing the Kullback-Leibler equation:

MI (L:M)=∑_(l∈L,m∈M)▒〖p(l,m) log (p(l, m))/(p(l)p(m))〗 …. (1)

Applying Naïve Bayes theorem, the equation can be arranged as
MI(L:M)=∑_(l∈L,m∈M)▒〖p(l,m) log (p(l|m))/p(l) … (2)
MI(L:M)=∑_(l∈L,m∈M)▒〖p(l,m) log (p(l|m))/p(l) 〗
MI(L;M)= -∑(l∈L,m∈M)▒〖p(l,m) log? 〖p(l)〗 -(-∑(l∈L,m∈M)▒〖p(l,m) log? 〖p(l|m)〗 〗) … (3)

Assuming p(l, m) ≈ p(l), the equation can be rewritten as follows
MI(L;M)= -∑(l∈L)▒〖p(l) log? 〖p(l)〗 -(-∑(l∈L,m∈M)▒〖p(l,m) log? 〖p(l|m)〗 〗) 〗 .... (4)

Entropy of a variable and the conditional entropy of two variables are given by equations 5 and 6
H(L)= -∑_(l∈L)▒〖p(l) log? 〖p(l)〗 ... (5)
H(L|M)= -∑_(l∈L,m∈M)▒〖p(l,m) log? 〖p(l|m)〗 ... (6)

Hence MI (L; M) = H(L)- H(L|M) … (7)

By symmetry, for random variables X and Y can be rewritten as MI (X;Y) = H(X) – H(X|Y) … (8)

Consider H(X) or H(Y) represent the entropy of random variables X or Y. Conditional and restricted entropies H(X|Y), or H(Y|X), describe their entropy in the presence of each other. Joint entropy H(X, Y) is signified as the merged entropy of X and Y as shown in Figure 1.

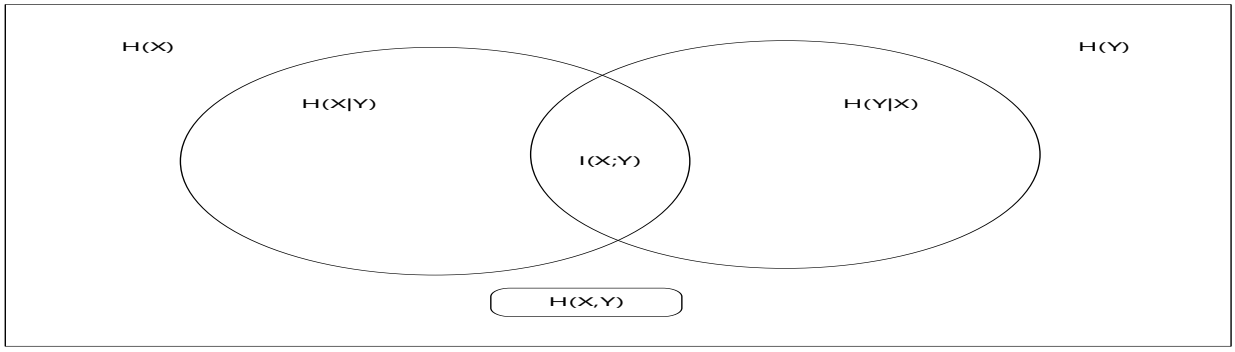

Figure 1: Schematic diagram of mutual information [51]

Mutual Information (MI) measures how well two random variables predict each other for each class, is model-independent, captures nonlinear dependencies, and identifies features with high MI relative to the label, which are informative, high in entropy, and important for accurate variable classification.

Linear Correlation (LC) quantifies the relationship between two variables, with the correlation coefficient R(A,B)R(A, B)R(A,B) ranging from -1 to 1, where values near ±1 indicate a strong linear association and values near 0 indicate a weak or unreliable relationship.

$$R(A,B) = \frac{\sum(A_i^1 - \bar{B})(B_j - \bar{B})}{\sqrt{\sum(A_i^1 - \bar{A})^2 X(B_j - \bar{B})}} \quad (9)$$

CHI indicator [27] measures the “degree of independence” of the attribute 'x' with group 'y'. The Chi-square indicator is arithmetically explained as below:

$$CHI(fi, yj) = \frac{N(AL - MN)^2}{(A+N)(A+L)(M+N)(M+L)} \quad (10)$$

In CHI statistics, A denotes the co-occurrence of fi and yj, N the occurrence of fi without yj, M the occurrence of yj without fi, and L when neither occurs. Together with Mutual Information (MI), which captures nonlinear dependencies, CHI is a lightweight and reliable choice for IDS feature selection, supporting fast, accurate classification with minimal computational cost.

3.2 CLASSIFICATION ALGORITHMS

In this research, a range of ML algorithms were judged with due thoroughness, considering the benefits of application in IDS systems. The authors created filtering criteria and classifiers, and the matching parameters of the criteria were selected. The criterion of selection is given below –

- High Accuracy
- Higher efficiency and faster training speed
- Ability to consider continuous and discrete data.
- Ability to handle outliers while working on large datasets.

Considering the advantages, limitations, and performance in recent studies for classifiers, ML algorithms (Table 2) were identified for more scrutiny. This part of the study objects to completely examine and expose the traits and convenience of the selected classifiers.

Linear Discriminant Analysis (LDA) applies a trajectory to boost distinction in illustrations fitting to dissimilar classes and dipping the gap among two illustrations within a class [15,17]. In LDA, the gap amongst vectors is decided using matrices Gb (among class), Ga (inside class), with vector k. These components exhibit a vital role in the computation of the projection vector. The projection vector in LDA is computed as follows:

$$a = \text{argmax}\frac{G_b k.k^T}{G_a k.k^T} \tag{11}$$

where $k \in R^{m \times n}$

Logistic regression (LR) is believed to be the basic classifier that is largely used for binary classification. It uses a sigmoid activation function. For a feature vector F = {A1, A2, ..., An}, the constrained probability of allocating the specified vector to a class is stated as follows [17,20].

$$P\ (F = 1|\text{A1}\ ,\text{A2}\ ,\dots\ ,\text{An}) = \frac{e(\ t0 + t1\ A1 + \cdots + tn\ An)}{1 + e(\ t0 + t1\ A1 + \cdots + tn\ An)} \tag{12}$$

where it signifies the component of regression

Gaussian Naïve Bayes (NB) categorization technique keeps the base of the traditional probability formula -Naïve Bayes. It undertakes to conclude the probability of absolute vector X being allocated to a given target category, B, while believing that every outcome is jointly unrelated [18,19].

$$P(B_i|X) = \frac{P(B_i).P(X|B_i)}{\sum_{i=1}^{n} P(B_i).P(X|B_i)} \tag{13}$$

Decision Tree (DT) is a widespread supervised technique that is principally employed in grouping outcomes. DT pulls the uncertainty in a category, termed entropy, to govern the ideal constraints that influence the effect on the arbitrariness when separating data. The entropy H(D) of a specific feature D with probability ai is calculated by means of the subsequent equation [21,22].

$$H(D) = \sum_{i=1}^{n} -a_i log a_i \tag{14}$$

Support Vector Machine Algorithm (SVM) deploys a hyperplane to separate numerous classes [23]. The hyperplanes are defined underneath:

$$f(x) = (\lambda + \mu x) \tag{15}$$

Hither, refer to the load and bias , respectively

The Gradient Boosting ML algorithm (GBM) exploits a group of weak and greedy classifiers like CART. Every single tree endeavours to reduce mean square error (MSE) involving monitored and assumed values, leading to added recapitulations for superior accuracy. Remarkably, GBM surpasses RF in much research [25,29].

Random Forest (RF) algorithm is built by packing DT so that the highly fitting tree which minimalizes faults is nominated. In tandem with other tree-based algorithms, RF exhibits superior accuracy with no overfitting [32,35].

4. MIRCHI FEATURE SELECTION FRAMEWORK

This section introduces a novel feature selection framework, **MIRCHI**, which fuses statistical and information-theoretic concepts. MIRCHI processes the feature vector B = {a1, a2, a3……., an}, where each attribute ai represents a feature, and the class of each attack vector is defined as uk∈{0,1}. The target space is denoted as A^T = {a1,a2…..,am}. MIRCHI aims to identify the optimal feature subset U⊆BU to maximize classification accuracy. Initially, each attribute is normalized using Min-Max scaling to reduce the effect of outliers. For each attribute ai, **Mutual Information (MI)** with the label B is calculated (Equation 8), and features with MI(ai, B)>1.3 are added to U1, representing highly relevant attributes. Next, **CHI statistics** (Equation 10) are computed, and features with CHI>2.22 are added to U2. The sets U1 and U2 are combined to form U. Highly correlated features (r>0. 75) are removed to create a lean, efficient feature set. MI and CHI are calculated in a single pass, which reduces computational cost and training time. The efficiency of the MIRCHI-selected features was gauged using multiple ML algorithms (Table 2), as described in Section, and the results are discussed in Section 6. A detailed algorithm is provided in Algorithm 1.

Algorithm I: MIRCHI Algorithm

INPUT

$B = \{b_i, b_2, \ldots\ldots. b_n\}$ // Features in the dataset.

$A^T = \{ a_1, a_2 \ldots.., a_m\}$ // Rows in the dataset.

OUTPUT

$R = \{r_i, r_2, \ldots\ldots. r_k\}$ // Reduced feature set where k<n.

BEGIN

{

//Initialization

$\Phi \leftarrow U_1, U_2, U$

While j ∈ J where J ={1,2,… n}

{

$b_j = (b_j – \text{argmin}(b_j) /(\text{argmax}(b_j)- \text{argmin}(b_j))$

j++

}

While j ∈ J

{

$MI (b_i:A) \leftarrow H(b_i) –-H(b_i |A)$ //Calculate MI for j^{th} features

IF argmax($MI(b_i:A)$

$U_1 \leftarrow U_1 \cup b_i$

ENDIF

$CHI2 (b_j: a_j)$ //Get CHI for j^{th} features

IF argmax($CHI2 (b_j: b_j)$)

$U_2 \leftarrow U_2 \cup u_j$

ENDIF

j++

}

$U \leftarrow U_1 \cup U_2$

While j ∈ J

{ For k >= j+1 : k ∈ { 2,3,… l}

{

argmax(r(bj : a_k)) >0.75

$U \leftarrow U-b_i$

```
            }
        }
        j++
        Return U
}
```

5. EXPERIMENTAL SETUP AND SCHEME

This section details the tests piloted to evaluate the MIRCHI algorithm's efficacy in picking crucial features for IDS [62]. The dataset was pre-processed by preprocessing to augment consistency. Eighty percent of the data was used for training, with the remaining 20% earmarked for validation. MIRCHI selected the significant features that were applied to ML classifiers (Table 2). The best-performing models were identified using **GridSearchCV**, and the optimal parameters of the models are listed in Table 2. L2 regularization was used with most classifiers to avoid overfitting and confirm robust performance on the validation set.

Table 2: List of Classifiers and Tuning parameters

Classifier	Tuning parameters	Penalty
LR	C=0.013	l2
LDA	Shrinkage= 0.685, Solver='lsqr',	l2
NB	alpha = 0.83, var_smoothing = 0.06236786	l2
DT	max_depth = 99	-
RF	criterion='entropy', maxfeatures = log2	-
SVM	kernel='sigmoid', C=0.036, gamma = 0.029	l2
GBM	learning_rate= 0.015, max_features=log2, max_depth=10 n_estimators=990	l2

Figure 2 shows the outline framework of the proposed IDS [63]. The full dataset is first preprocessed to remove duplicates, NaN values, and out-of-range entries (Section 5.2), followed by normalization (Section 5.3). Features with the highest Mutual Information (MI) and Chi-square (CHI) values are collected into lists T1 and T2, which are merged to form a reduced feature set. Pearson correlation is applied to remove highly correlated features (threshold 0.75), producing the final optimized feature set. Selected ML algorithms (Section 3.3) are then applied, and the model achieving the highest accuracy with minimal computation is proposed as the IDS solution.

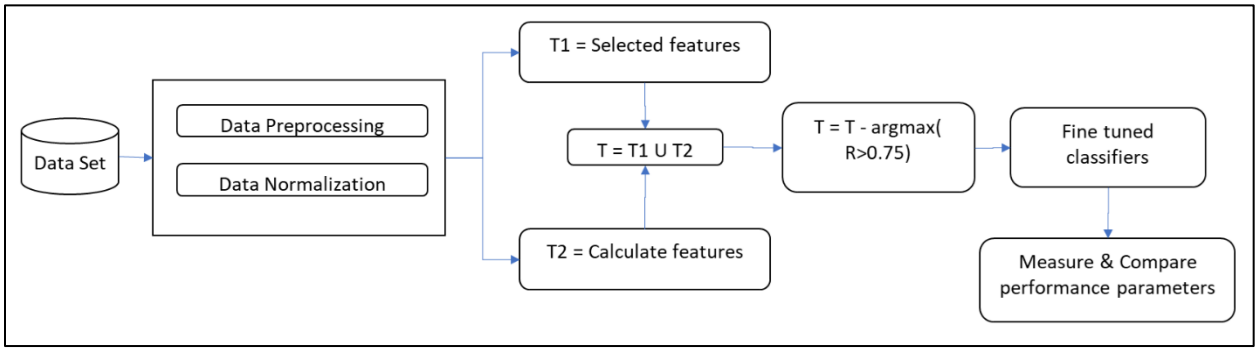

Figure 2: Wire-frame model of suggested IDS system

5.1 DEFINITIONS OF USED DATA SETS

Assessment of the fusion feature collection method engaged carrying out trials and experiments utilizing datasets: IoTID20 [52] and NSL-KDD [53]. The IoTID20 feature set incorporates a varied collection of vulnerabilities on IoT and benign network traffic. Gathered in the IoT [64, 65] gambit of the intelligent home, this dataset encompasses linked devices. The testbed set up is illustrated in Figure 3 [52].

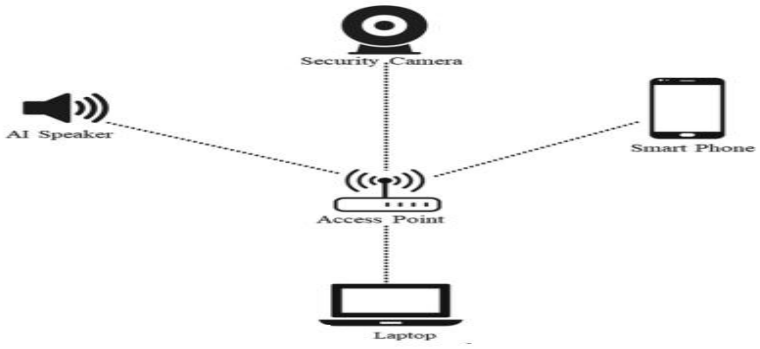

Figure 3: The IoTID20 dataset generation environment [52]

The IoTID20 dataset contains 83 features, with 585,710 attack records and 40,073 normal records. The other data set , called the NSLKDD dataset, is also used to validate this study and contains 41 features, including 9 nominal attributes, and addresses issues in the original KDDcup99 while maintaining IDS relevance. Figure 4 summarizes the dataset.

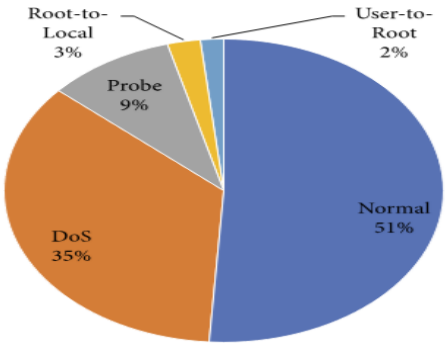

Figure 4: Taxonomy of NSLKDD feature-set [53]

5.2 DATA PROCESSING

NSLKDD and IoTID20 datasets were cleaned by handling missing values, converting non-numeric data, and removing duplicates. MIRCHI then selected the most important features, producing the final optimized dataset.

5.3 NORMALIZATION OF DATA

The range of the data is narrowed with the use of normalization. The values in each column were brought to a minimum value of 0 and a maximum value of 1 using the Min-Max normalization technique.

$$z = \frac{x - minimum(x)}{[\text{maximum(x)} - \text{minimum}(x)} \quad (16)$$

Every single attribute in the dataset got treated with normalization.

5.4 PERFORMANCE INDICATORS

Accuracy measures the model's effectiveness in correctly locating and categorizing benign cases in a dataset. In terms of math, it is displayed as

$$A = \frac{TP+TN}{TP+FP+TN+FN} \quad (17)$$

Precision centers on the correctness of the definite estimates nominated with classifier. It also called as Positive predicted value (PPV). Scientifically it is defined as

$$P = \frac{TP}{TP+FP} \quad (18)$$

The model's **recall** explains how insightful it is to assault detection. This is also known as the True Positive Rate (TPR) at times. In terms of math, it can be written as

$$R = \frac{TP}{TP+FN} \quad (19)$$

The **F1 score**, also known as the **F-measure**, shows that the classifier exhibits productively balanced recall and precision, allowing it to identify assaults with the fewest possible false positives and false negatives. In terms of math, it can be written as

$$\text{F1 Score} = 2.\frac{PX\,R}{P+R} \quad (20)$$

The following section presents the determined performance of each ML classifier.

6. RESULTS AND DISCUSSIONS

The IoTID20 and NSLKDD datasets were fully pre-processed and normalized before applying selected classifiers. The classifiers were tuned with hyperparameter tuning, and the comparison of tuned and untuned classifiers with the IoTID20 dataset and NSLKDD dataset is given in Figure 5.

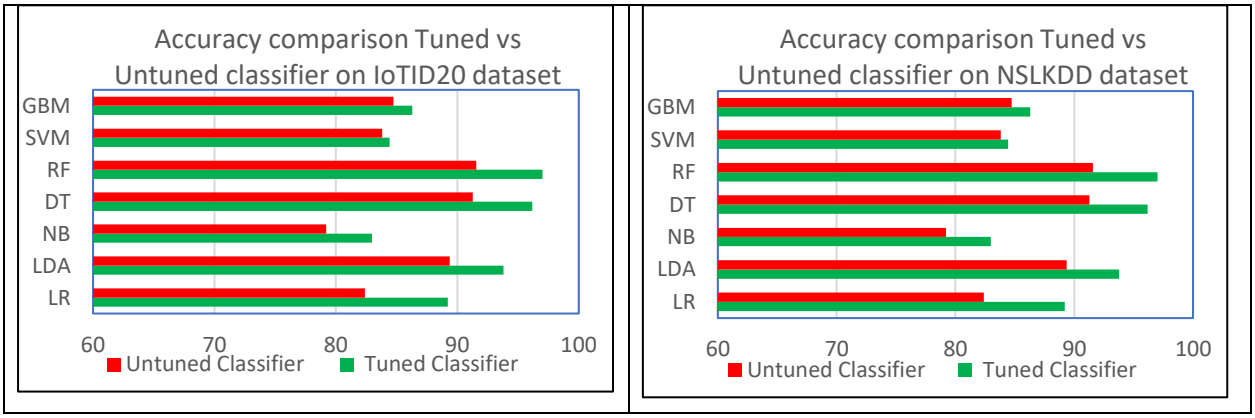

Figure 5: Accuracy comparison Tuned vs Untuned classifier on IoTID20 dataset & NSLKDD Dataset

Table 3: Performance metrics derived from the entire IOTIDS20 dataset

Clfr	TT	TsT	Acc	PPV	TPR	F1
LR	342.01±5.1	0.9	85.23±0.7	85.18±0.73	86.81±0.71	85.99±0.71
LDA	146.79±3.7	0.11	92.31±0.46	91.58±0.45	92.27±0.46	91.92±0.45
NB	142.34±3.9	0.56	79.88±0.98	79.76±0.99	78.57±1.02	79.16±0.99
DT	350.27±1.01	0.2	95.41±0.01	94.65±0.01	95.13±0.01	94.89±0.01
RF	414.02±0.92	0.88	95.87±0.01	95.65±0.01	95.13±0.01	95.39±0.01
SVM	919.53±16.3	23	71.67±1.01	71.51±1.09	71.53±1.01	71.52±1.02
GBM	870.12±2.1	19	77.19±0.08	77.1±0.09	77.26±0.06	77.18+0.06

On the IoTID20 dataset , Random Forest (RF) and Decision Tree (DT) classifiers achieved the highest attack classification accuracy, while SVM performed the lowest at 71%. SVM and GBM required the longest classification times, despite lower accuracy. LDA achieved 92% accuracy with a shorter processing time. On the NSLKDD dataset, classifiers showed similar accuracy and training patterns. RF and DT again achieved maximum accuracy, GBM and SVM outperformed NB but were slower, and Precision, Recall, and F1-Score followed consistent trends across both datasets.

Table 4: Performance metrics derived from the whole NSLKDD collection

Clfr	TT	TsT	Acc	PPV	TPR	F1
LR	292±5.78	0.9±0.07	89.21±0.86	89.01±0.82	88.89±0.81	88.95±0.81
LDA	127±3.21	0.11±0.0	93.79±0.43	94.01±0.52	93.71±0.49	93.86±0.51
NB	132±1.09	0.56±0.03	82.97±0.27	82.9±0.29	82.9±0.29	82.9±0.29
DT	181±0.11	0.2±0.001	96.17±0.03	96.1±0.03	96.1±0.03	96.1±0.03
RF	192±0.01	0.88±0.001	97.01±0.01	96.99±0.02	97.02±0.01	97±0.02
SVM	814±7.38	23±0.8	84.43±0.83	84.39±0.76	84.42±0.78	84.4±0.78
GBM	798±1.07	19±0.02	86.28±0.09	86.2±0.09	86.18±0.07	86.19±0.07

The MIRCHI algorithm identified 27 features from IoTID20 (Table 5) and 26 from NSLKDD (Table 6), establishing an improved MIRCHI feature set for each dataset.

Table 5: Chosen features of IOTIDS20 dataset

Flow_ID,Src_Port,Dst_Port,Protocol,Timestamp,TotLen_Fwd_Pkts,TotLen_Bwd_Pkts,Fwd_Pkt_Len_Max,Fwd_Pkt_Len_Min,Fwd_Pkt_Len_Mean,Fwd_Pkt_Len_Std,Flow_Pktss,Flow_IAT_Mean,Flow_IAT_Max,Bwd_IAT_Tot,Bwd_IAT_Std,Bwd_IAT_Max,Bwd_Pktss,Pkt_Len_Std,Pkt_Len_Var,ACK_Flag_Cnt,Pkt_Size_Avg,Fwd_Seg_Size_Avg,Subflow_Fwd_Byts,Subflow_Bwd_Byts,Init_Bwd_Win_Byts,Idle_Mean,Idle_Max

Table 6: Chosen attributes of NSLKDD dataset

List of features selected from NSLKDD Dataset
Flow_Duration,Fwd_Pkt_Len_Mean,Fwd_Pkt_Len_Std,Bwd_Pkt_Len_Min,Bwd_Pkt_Len_Mean,Flow_Byts/s,Flow_Pkts/s,Flow_IAT_Mean,Flow_IAT_Std,Flow_IAT_Max,Fwd_IAT_Mean,Pkt_Len_Mean,ACK_Flag_Cnt,Down/Up_Ratio,Pkt_Size_Avg,Fwd_Seg_Size_Avg,Active_Max,Active_Min,Idle_Mean,Idle_Max,Src_IP_oct3,Src_IP_oct4,Dst_IP_oct1,Dst_IP_oct2,Dst_IP_oct3

Applying specific attributes from the IoTID20 dataset to machine learning techniques improved its accuracy from 77.5% to 99.6%.

Table 7: Performance metrics derived from subset of features on IOTIDS20 dataset using MIRCHI.

Clfr	TT	TsT	Acc	PPV	TPR	F1
LR	33.71±1.07	0.05±0	89.81±0.11	89.38±0.16	89.63±0.21	89.5±0.19
LDA	3.86±0.8	0.06±0	97.82±0.13	97.84±0.09	97.81±0.1	97.82±0.1
NB	1.06±0.08	0.18±0.01	87.19±0.03	87.44±0.01	86.92±0.01	87.18±0.01
DT	44.18±0.04	0.085±0	99.68±0.02	99.58±0.02	99.57±0.01	99.57±0.01
RF	50.19±0.01	0.655±0.01	99.42±0.03	99.27±0.01	99.29±0.02	99.28±0.02
SVM	756±1.18	17±0.23	77.82±0.17	77.46±0.16	77.86±0.17	77.66±0.16
GBM	387±1.03	14±0.09	82.68±0.02	82.66±0.01	82.61±0.01	82.63±0.01

Figure 6 compares the accuracy of ML classifiers using the full IoTID20 feature set and the MIRCHI-selected features, referred to as MIRCHI IoTID20 dataset. DT and RF achieved roughly 99.5% accuracy.

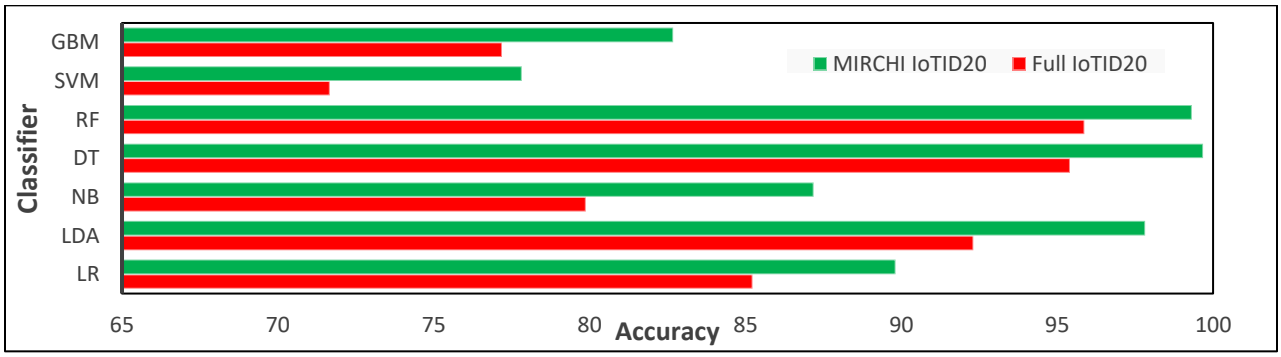

Figure 6: Comparing MIRCHI dataset of chosen features and baseline IoTID20 dataset for accuracy

Figure 7 establishes the parallel of the PPV of ML techniques beside complete attributes of IoTID20 and features of IOTID20 chosen utilizing the MIRCHI framework. As can be shown in Figure 8, recall for NB and SVM was significantly more prominent than that of the entire feature set, at 10.6% and 8.9%, respectively. The MIRCHI dataset yielded Recall values of 99.6%, 99.3%, and 97.8% for DT, RF, and LDA, respectively. These values were 4.3% to 6% greater than the entire IoTID20 feature set.

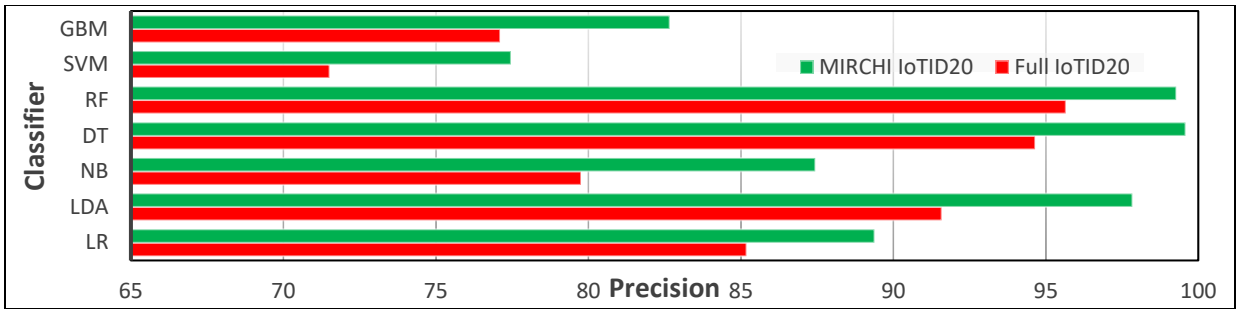

Figure 7: Comparing MIRCHI dataset of chosen features with IoTID20 starting dataset for precision

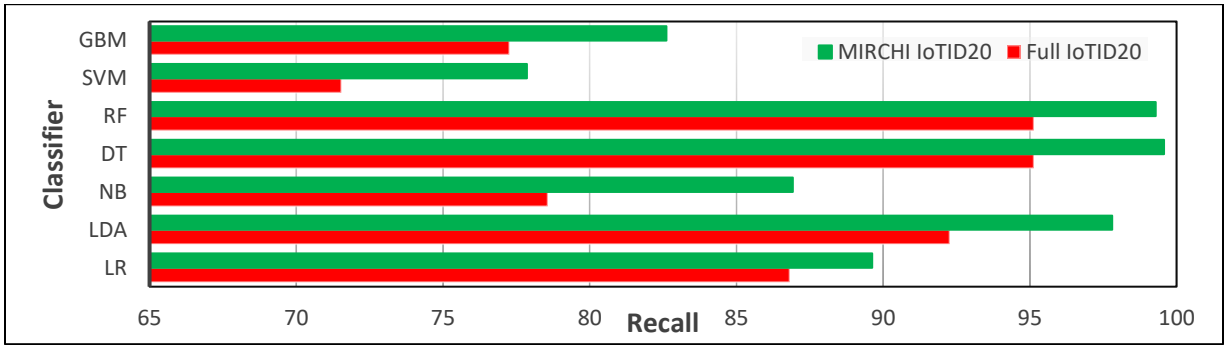

Figure 8: Comparing MIRCHI dataset of chosen features with IoTID20 starting dataset for Recall

The same pattern emerges in ML algorithms for Recall and F1Score as displayed in Figure 9. The range of the F1Score with the chosen features was between 77.66% and 99.6%.

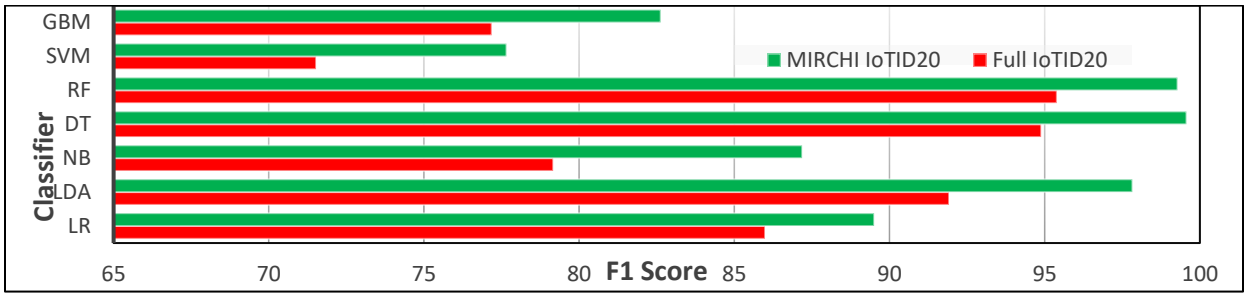

Figure 9: Comparing MIRCHI dataset of chosen features with IoTID20 starting dataset forF1-Score

The output of the programs is given in the form of a confusion matrix for IoTID20 selected features through MIRCHI, with classifiers as illustrated in Figure 10. The significance of the performance of MIRCHI over the initial dataset was tested using a t-test, and compared with the initial dataset using a t-test. The t-test showed the p-value of 0.0076 for accuracy, 0.0058 for PPV, and 0.0062 for TPR. All the p-values were less than 0.5. This helps to conclude that the MIRCHI feature helped to increase the performance of the IDS system.

LR LDA DT

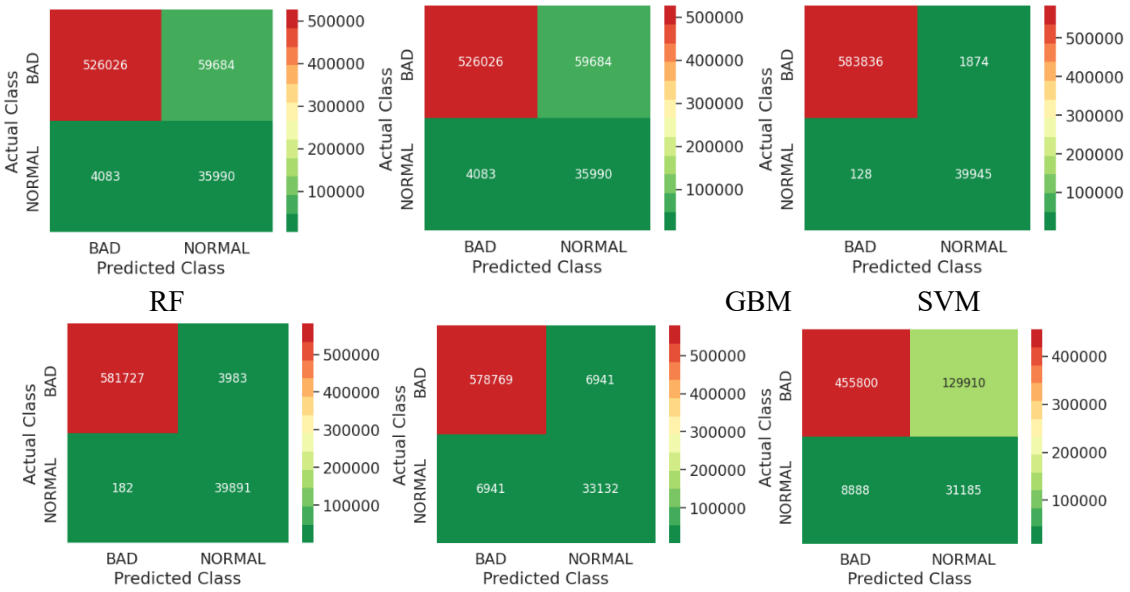

Figure 10: Confusion Matrix for MIRCHI dataset of chosen features from IoTID20

MIRCHI framework further discovered significant features which assisted in raising the performance parameters such as PPV, TPR, and accuracy of ML algorithms employed by the nominated attributes of NSLKDD dataset. The new dataset created after selecting the important features of NSLKDD using the MIRCHI technique is referred to as MIRCHI NSLKDD dataset in this study. The parameter performance is provided in Table 8. The comparison of the classification algorithm's accuracy using the complete record set and characteristics chosen by the MIRCHI framework is shown in Figure 11. DT employing the MIRCHI features enhanced by almost 3.78% and realized an accuracy of 99.81%. RF and LDA using attributes chosen by means of the MIRCHI dataset exhibited an accuracy of 99.62% and 97.29%, respectively. The greatest accuracy improvements were revealed by GBM and SVM, at 8.2% and 6.3%, respectively.

Table 8: Performance metrics obtained by MIRCHI-identified attributes on NSLKDD dataset

Clfr	TT	TsT	Acc	PPV	TPR	F1
LR	96±1.01	0.05±0.0	91.13±0.03	91.13±0.04	91.13±0.03	91.13±0. 03
LDA	45±0.01	0.06±.0	97.29±0.03	97.29±0.01	97.29±0.01	97.29±0.01
NB	44±0.03	0.18±0.0	88.2±0.011	88.21±0.02	88.21±0.02	88.21±0.02
DT	32±0.08	0.085±0.0	99.81±0.01	99.81±0.03	99.81±0.01	99.81±0.02
RF	41±0.01	0.655±0.01	99.62±0.02	99.62±0.01	99.62±0.02	99.62±0.01
SVM	256±0.32	17	89.09±0.09	89.09±0.08	89.09±0.07	89.09±0.08
GBM	271±0.01	14±0.01	93.35±0.01	93.35±0.02	93.35±0.02	93.35±0.02

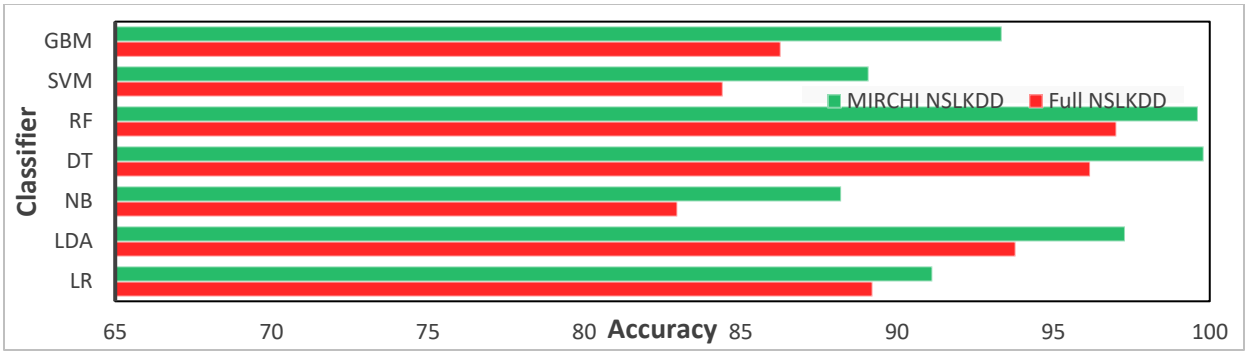

Figure 11: Comparing MIRCHI dataset of chosen features with NSLKDD starting dataset for Accuracy

As shown in Figure 12, the identified attributes also contributed to an enhance in precision The DT and RF classifiers had the highest level of precision. Both demonstrated a precision of greater than 99%.

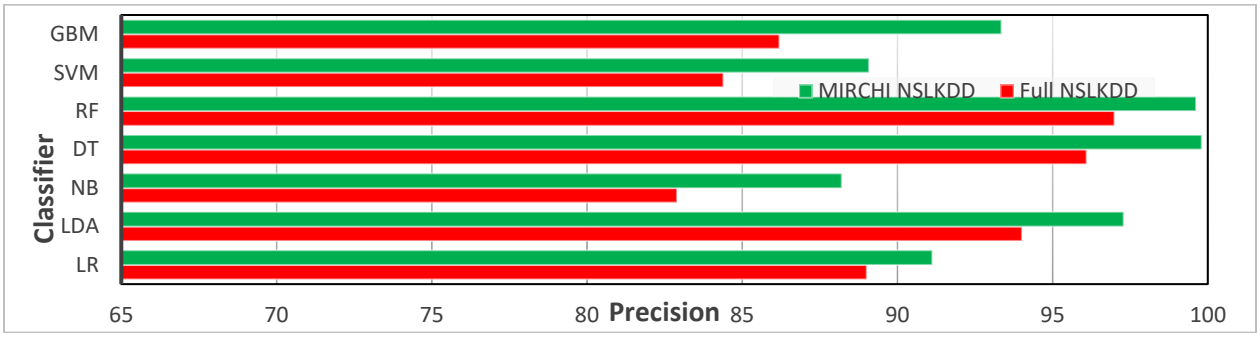

Figure 12: Comparing MIRCHI dataset of chosen features with NSLKDD starting dataset for Precision

Like IoTIDS20, MIRCHI chosen dimensions of NSLKDD qualified the classifiers to lower false negative classifications, adding to the system's robustness. Figure 13 shows that DT and RF had the superior TPR above 99%.

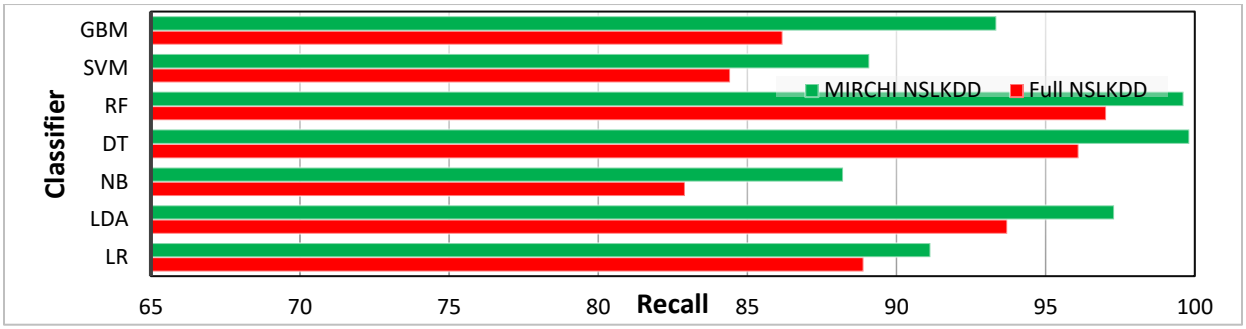

Figure 13: Comparing MIRCHI dataset of chosen features with NSLKDD starting dataset for F1-Score

All the classifiers displayed in Figure 14 show a rise in the F measure by the identified features over the F measure attained employing the full dataset. Subsequent the pattern of PPV and TPR, the DT and RF uncovered better and superior F-measure, higher than 99%.

Figure 14: Comparing MIRCHI dataset of chosen features with NSLKDD starting dataset for F1-Score

The output of the programs is given in the form of a confusion matrix for NSLKDD selected features through MIRCHI with various classifiers as illustrated in Figure 15. On analysis of Table 7 , the DT technique, when employed with the chosen attributes attained from the MIRCHI technique, presents maximum performance results across all parameters.

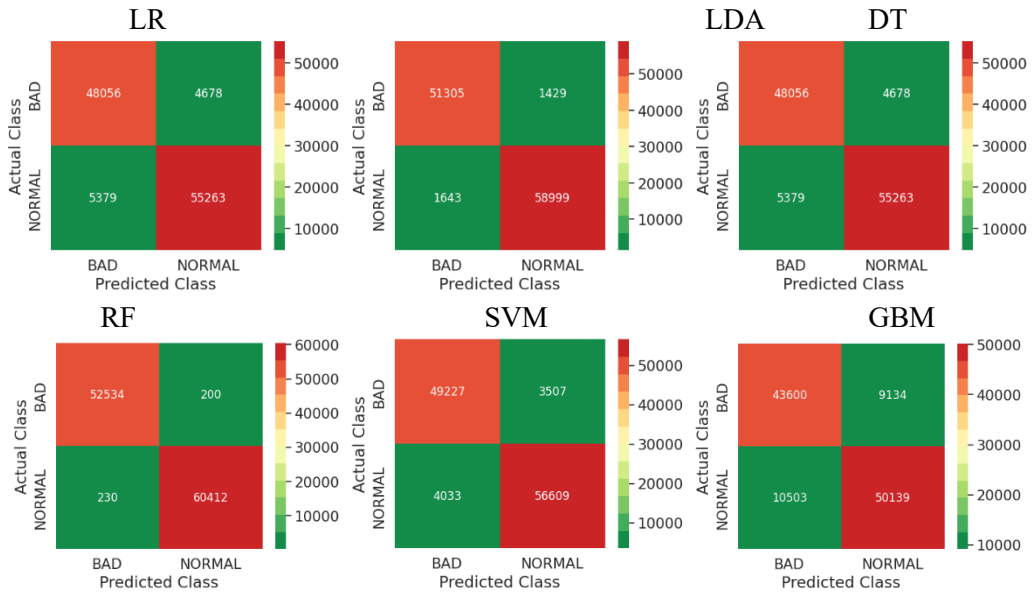

Figure 15: Confusion Matrix of various algorithms for MIRCHI dataset of chosen features from IoTID20

The classifiers demonstrated strong precision and accuracy, effectively distinguishing benign vectors from attacks with low false-positive rates. Decision Tree (DT) achieved high recall and balanced performance, avoiding underfitting or overfitting. Compared with recent studies, the proposed MIRCHI-DT system achieved 99.68% accuracy, precision, recall, and F1-score on IoTID20, performing comparably to DeSouza et al. [40,41], Albulayhi et al. [44,45], and Tubishat et al. [49] (Tables 9). Similarly, for NSLKDD, Table 10 indicates that only a limited number of studies concluded with 99% accuracy. Cao et al. [36] reported 99% accuracy, precision, recall, and F1-score, while Chu et al. [33] and Imrana et al. [38] achieved 97% accuracy. These findings accentuate the importance of efficient feature selection with optimized classifiers to achieve high-performance IDS in IoT and industrial systems.

Table 9: Evaluation of the recent investigations with the proposed research (IoTID20 dataset)

Study	Year	Accuracy	Precision	Recall	F1-Score	ROC
[42]	2021	87.00%	87	87	87	87
[40]	2022	99.79%		98.00%	98%	
[41]	2022	98.76%		98.00%	98%	
[39]	2022	99.98%	99.90%	99.90%	99.90%	
[43]	2023	99.51%	98.51%	99.63%	99.07%	
[44]	2024	99.96%	99.97%	99.60%	99.70%	
[45]	2024	99.59%	99.17%	99.02%	99.07%	
Proposed	2024	99.68%	99.58%	99.58%	99.58%	100.00%

Table 10: Evaluation of the recent investigations with the proposed research (NSLKDD dataset)

Study	Year	Accuracy	Precision	Recall	F1 Score
[33]	2019	97.20%	99.10%	-	-
[35]	2022	90.50%	91.72%	91.72	91.72
[36]	2022	99%	99%	99	99
[38]	2022	97.12%	98.45%	90.10%	93.48%
[19]	2022	95.60%	91.42%	97.37%	93.48%
[34]	2022	81.20%	80.4	88.8	84.2
[45]	2024	99.00%	99.00%	99.00%	99.00%
[47]	2024	98.24%	97.99%	97.91%	98.00%
Proposed	2024	99.92%	99.92%	99.89%	99.90%

Table 11 compares the computation time of latest published studies with the proposed IDS with IoTID20 and NSLKDD dataset. Analysis recommends that proposed IDS system consumed less time compared to latest published studies. Computation time implies the computational complexity and consumption of computational assets [37].

Table 11: Analysis of the computational complexity of suggested methodology and the current studies

Study	Dataset	Computation Time (ms)
[38]	IoTID20	153
[19]	IoTID20	154
[34]	IoTID20	122
[36]	NSLKDD	138
[37]	NSLKDD	168
Proposed IDS	IoTID20	62
Proposed IDS	NSLKDD	46

CONCLUSION

The current research work recommends an effective and efficient IDS system for the enterprise IoE environment, leveraging the state-of-the-art MIRCHI feature identification algorithm. The proposed IDS system achieves exemplary performance with 99.68% accuracy for IoTID20 and 99.81% for accuracy on the NSLKDD dataset. These prototypical results were achieved with only

62 milliseconds for IoTID20 and 46 milliseconds for the NSLKDD dataset. These results undoubtedly establish the superiority of the proposed IDS system compared with current IDS and emphasize its reduced training complexity, enhanced accuracy, and reduced computational complexity. Thus, the proposed IDS system can be considered a favourable system for application in IoE domains, such as smart cities, enterprise IoE, health IoE, communication IoE, and mobile network towers, that are vulnerable to data theft, corporate espionage, and cyber wars.

Conflicts of Interest

The authors declare no conflict of interest.

REFERENCES

[1] Asgharzadeh, H., Ghaffari, A., Masdari, M., & Gharehchopogh, F. S. (2023). An anomaly-based intrusion detection system on the Internet of Things using a convolutional neural network and multi-objective enhanced Capuchin Search Algorithm. Journal of Parallel and Distributed Computing, 175, 1-21.

[2] Wahab, O. A. (2022). Intrusion detection in the IoT under data and concept drift: Online deep learning approach. IEEE Internet of Things Journal, 9(20), 19706-19716.

[3] Wani, A. R., Gupta, S. K., Khanam, Z., Rashid, M., Alshamrani, S. S., & Baz, M. (2022). A novel approach for securing data against adversary attacks in UAV embedded HetNet using identity-based authentication scheme. IET Intelligent Transport Systems.

[4] Jayalaxmi, P. L. S., Saha, R., Kumar, G., Conti, M., & Kim, T. H. (2022). Machine and Deep Learning Solutions for Intrusion Detection and Prevention in IoTs: A Survey. IEEE Access.

[5] Lahasan, B., & Samma, H. (2022). Optimized deep autoencoder model for Internet of Things intruder detection. IEEE Access, 10, 8434-8448.

[6] Fatima, M., Rehman, O., and Rehman, I. M. (2023). Li-ids: An approach towards a lightweight ids for resource-constrained iot. In 2023 International Conference on Smart Applications, Communications and Networking (SmartNets), pages 1–6. IEEE.

[7] Sangaiah, A. K., Javadpour, A., Ja'fari, F., Pinto, P., Zhang, W., & Balasubramanian, S. (2023). A hybrid heuristics artificial intelligence feature selection for intrusion detection classifiers in a cloud of things. Cluster Computing, 26(1), 599-612.

[8] Meddeb, R., Jemili, F., Triki, B., & Korbaa, O. (2023). A deep learning-based intrusion detection approach for mobile Ad-hoc networks. Soft Computing, 1-15.

[9] Gyamfi, E., & Jurcut, A. (2022). Intrusion detection in the Internet of Things systems: a review on design approaches leveraging multi-access edge computing, machine learning, and datasets. Sensors, 22(10), 3744.

[10] Islam, N., Farhin, F., Sultana, I., Kaiser, M. S., Rahman, M. S., Mahmud, M., ... & Cho, G. H. (2021). Towards Machine Learning Based Intrusion Detection in IoT Networks. Computers, Materials & Continua, 69(2).

[11] Wu, X., Jin, Z., Zhou, J., & Duan, C. (2023). Quantum Walks-based Classification Model with Resistance for Cloud Computing Attacks. Expert Systems with Applications, 120894.

[12] Anita, T. M. (2023). An Intelligent Hybrid GA-PI Feature Selection Technique for Network Intrusion Detection Systems. International Journal of Intelligent Systems and Applications in Engineering, 11(7s), 718-731.

[13] Mohiuddin, G., Lin, Z., Zheng, J., Wu, J., Li, W., Fang, Y., ... & Zeng, X. (2023). Intrusion detection using hybridized meta-heuristic techniques with Weighted XGBoost Classifier. Expert Systems with Applications, 120596.

[14] Kumar, V., Das, A. K., & Sinha, D. (2021). UIDS: a unified intrusion detection system for IoT environment. Evolutionary Intelligence, 14, 47-59.

[15] Ni, C., & Li, S. C. (2024). Machine learning enabled Industrial IoT Security: Challenges, Trends and Solutions. Journal of Industrial Information Integration, 100549

[16] Li, X., Chen, W., Zhang, Q., & Wu, L. (2021). Building auto-encoder intrusion detection system based on random forest feature selection. Computers & Security, 95, 101851

[17] Lu, G., & Tian, X. (2021). An Efficient Communication Intrusion Detection Scheme in AMI Combining Feature Dimensionality Reduction and Improved LSTM. Security and Communication Networks, 2021

[18] Safaldin, M., Otair, M., & Abualigah, L. (2021). Improved binary gray wolf optimizer and SVM for intrusion detection system in wireless sensor networks. Journal of ambient intelligence and humanized computing, 12(2), 1559-1576.
[19] Kareem, S. S., Mostafa, R. R., Hashim, F. A., & El-Bakry, H. M. (2022). An effective feature selection model using hybrid metaheuristic algorithms for IoT intrusion detection. Sensors, 22(4), 1396.
[20] Yaseen, W. L., Idrees, A. K., & Almasoudy, F. H. (2022). Wrapper feature selection method based differential evolution and extreme learning machine for the intrusion detection system. Pattern Recognition, 132, 108912.
[21] Rashid, M., Kamruzzaman, J., Imam, T., Wibowo, S., & Gordon, S. (2022). A tree-based stacking ensemble technique with feature selection for network intrusion detection. Applied Intelligence, 52(9), 9768-9781.
[22] Alghanam, O. A., Almobaideen, W., Saadeh, M., & Adwan, O. (2023). An improved PIO feature selection algorithm for IoT network intrusion detection system based on ensemble learning. Expert Systems with Applications, 213, 118745.
[23] Talukder, M. A., Hasan, K. F., Islam, M. M., Uddin, M. A., Akhter, A., Yousuf, M. A.,& Moni, M. A. (2023). A dependable hybrid machine learning model for network intrusion detection. Journal of Information Security and Applications, 72, 103405.
[24] Jeyaselvi, M., Dhanaraj, R. K., Sathya, M., Memon, F. H., Krishnasamy, L., Dev, K. & Qureshi, N. M. F. (2023). A highly secured intrusion detection system for IoT using EXPSO-STFA feature selection for LAANN to detect attacks. Cluster Computing, 26(1), 559-574
[25] Liu, X., & Du, Y. (2023). Towards Effective Feature Selection for IoT Botnet Attack Detection Using a Genetic Algorithm. Electronics, 12(5), 1260.
[26] Mushtaq, E., Zameer, A., & Nasir, R. (2023). Knacks of a hybrid anomaly detection model using deep auto-encoder driven gated recurrent unit. Computer Networks, 226, 109681.
[27] Kaur, B., Dadkhah, S., Shoeleh, F., Neto, E. C. P., Xiong, P., Iqbal, S., ... & Ghorbani, A. A. (2023). Internet of Things (IoT) security dataset evolution: Challenges and future directions. Internet of Things, 100780.
[28] Kumar, Y., & Subba, B. (2023). Stacking ensemble-based HIDS framework for detecting anomalous system processes in Windows-based operating systems using multiple word embedding. Computers & Security, 125, 102961.
[29] Bhayo, J., Shah, S. A., Hameed, S., Ahmed, A., Nasir, J., & Draheim, D. (2023). Towards a machine learning-based framework for DDOS attack detection in software-defined IoT (SD-IoT) networks. Engineering Applications of Artificial Intelligence, 123, 106432.
[30] Sumaiya Thaseen, I., Saira Banu, J., Lavanya, K., Rukunuddin Ghalib, M., & Abhishek, K. (2021). An integrated intrusion detection system using correlation-based attribute selection and artificial neural network. Transactions on Emerging Telecommunications Technologies, 32(2), e4014.
[31] Alqahtani, M., Mathkour, H., & Ben Ismail, M. M. (2020). IoT Botnet Attack Detection Based on Optimized Extreme Gradient Boosting and Feature Selection. Sensors, 20(21), 6336.
[32] Panigrahi, R., Borah, S., Pramanik, M., Bhoi, A. K., Barsocchi, P., Nayak, S. R., & Alnumay, W. (2022). Intrusion detection in cyber–physical environment using hybrid Naïve Bayes—Decision table and multi-objective evolutionary feature selection. Computer Communications, 188, 133-144.
[33] Chu, W. L., Lin, C. J., & Chang, K. N. (2019). Detection and classification of advanced persistent threats and attacks using the support vector machine. Applied Sciences, 9(21), 4579.
[34] Soleymanzadeh, R., Aljasim, M., Qadeer, M. W., & Kashef, R. (2022). Cyberattack and fraud detection using ensemble stacking. AI, 3(1), 22-36.
[35] Carrera, F.; Dentamaro, V.; Galantucci, S.; Iannacone, A.; Impedovo, D.; Pirlo, G. Combining Unsupervised Approaches for Near Real-Time Network Traffic Anomaly Detection. Appl. Sci. 2022, 12, 1759.
[36] Cao, B., Li, C., Song, Y., Qin, Y., & Chen, C. (2022). Network intrusion detection model based on CNN and GRU. Applied Sciences, 12(9), 4184.
[37] Fu, Y., Du, Y., Cao, Z., Li, Q., & Xiang, W. (2022). A deep learning model for network intrusion detection with imbalanced data. Electronics, 11(6), 898.
[38] Imrana, Y., Xiang, Y., Ali, L., Abdul-Rauf, Z., Hu, Y. C., Kadry, S., & Lim, S. (2022). χ 2-bidlstm: a feature-driven intrusion detection system based on χ 2 statistical model and bidirectional lstm. Sensors, 22(5), 2018.
[39] Albulayhi, K., Abu Al-Haija, Q., Alsuhibany, S. A., Jillepalli, A. A., Ashrafuzzaman, M., & Sheldon, F. T. (2022). IoT intrusion detection using machine learning with a novel high-performing feature selection method. Applied Sciences, 2022,12(10), 5015.

[40] De Souza, C. A., Westphall, C. B., & Machado, R. B. (2022). Two-step ensemble approach for intrusion detection and identification in IoT and fog computing environments. Computers & Electrical Engineering, 98, 107694.
[41] Tubishat, M., Rawshdeh, Z., Jarrah, H., Elgamal, Z. M., Elnagar, A., & Alrashdan, M. T. (2022). Dynamic generalized normal distribution optimization for feature selection. Neural Computing and Applications, 34(20), 17355-17370.
[42] Qaddoura, R., M. Al-Zoubi, A., Faris, H., & Almomani, I. (2021). A multi-layer classification approach for intrusion detection in IoT networks based on deep learning. Sensors, 21(9), 2987.
[43] Wardhani, R. W., Putranto, D. S. C., Jo, U., & Kim, H. (2023). Toward Enhanced Attack Detection and Explanation in Intrusion Detection System-Based IoT Environment Data. IEEE Access, 11, 131661-131676.
[44] Karamollaoğlu, H., DOĞRU, İ., & Yücedağ, İ. (2024). An Efficient Deep Learning-based Intrusion Detection System for Internet of Things Networks with Hybrid Feature Reduction and Data Balancing Techniques. Information Technology and Control, 53(1).
[45] Qaddos, A., Yaseen, M. U., Al-Shamayleh, A. S., Imran, M., Akhunzada, A., & Alharthi, S. Z. (2024). A novel intrusion detection framework for optimizing IoT security. Scientific Reports, 14(1), 21789.
[46] Alrayes, F. S., Zakariah, M., Amin, S. U., Khan, Z. I., & Alqurni, J. S. (2024). CNN Channel Attention Intrusion Detection System Using NSL-KDD Dataset. Computers, Materials & Continua, 79(3).
[47] Vibhute, A. D., Patil, C. H., Mane, A. V., & Kale, K. V. (2024). Towards detection of network anomalies using machine learning algorithms on the NSL-KDD benchmark datasets. Procedia Computer Science, 233, 960-969.
[48] Saheed, Y. K., Abdulganiyu, O. H., & Tchakoucht, T. A. (2024). Modified genetic algorithm and fine-tuned long short-term memory network for intrusion detection in the internet of things networks with edge capabilities. Applied Soft Computing, 155, 111434.
[49] Zhou, W., Xia, C., Wang, T., Liang, X., Lin, W., Li, X., & Zhang, S. (2025). HIDIM: A novel framework of network intrusion detection for hierarchical dependency and class imbalance. Computers & Security, 148, 104155.
[50] Li, C., He, A., Liu, G., Wen, Y., Chronopoulos, A. T., & Giannakos, A. (2024). RFL-APIA: a Comprehensive Framework for mitigating poisoning attacks and promoting model aggregation in IIoT Federated Learning. IEEE Transactions on Industrial Informatics.
[51] Kleiner, J., & Tull, S. (2021). The mathematical structure of integrated information theory. Frontiers in Applied Mathematics and Statistics, 6, 74.
[52] Ullah, I., & Mahmoud, Q. H. (2020, May). A scheme for generating a dataset for anomalous activity detection in IoT networks. In Canadian conference on artificial intelligence (pp. 508-520). Cham: Springer International Publishing.
[53] Kang, H.; Ahn, D.H.; Lee, G.M.; Yoo, J.D.; Park, K.H.; Kim, H.K.(2019) IOT Network Intrusion Dataset.Available online: https://ieee-dataport.org/open-access/iot-network-intrusion-dataset (accessed on 2 February 2023).
[54] Zhou, W., Xia, C., Wang, T., Liang, X., Lin, W., Li, X.,... Zhang, S. (2025). HIDIM: A novel framework of network intrusion detection for hierarchical dependency and class imbalance. Computers & Security, 148, 104155. doi: https://doi.org/10.1016/j.cose.2024.104155
[55] Gong, Y., Yao, H., Xiong, Z., Chen, C. L. P., & Niyato, D. (2025). Blockchain-Aided Digital Twin Offloading Mechanism in Space-Air-Ground Networks. IEEE Transactions on Mobile Computing, 24(1), 183-197. doi: 10.1109/TMC.2024.3455417
[56] Gong, Y., Yao, H., Liu, X., Bennis, M., Nallanathan, A.,... Han, Z. (2024). Computation and Privacy Protection for Satellite-Ground Digital Twin Networks. IEEE Transactions on Communications, 72(9), 5532-5546. doi: 10.1109/TCOMM.2024.3392795
[57] Xiao, J., Ren, Y., Du, J., Zhao, Y., Kumari, S., Alenazi, M. J. F.,... Yu, H. (2025). CALRA: Practical Conditional Anonymous and Leakage-Resilient Authentication Scheme for Vehicular Crowdsensing Communication. IEEE Transactions on Intelligent Transportation Systems, 26(1), 1273-1285. doi: 10.1109/TITS.2024.3488741
[58] Gong, Y., Yao, H., Liu, X., Bennis, M., Nallanathan, A.,... Han, Z. (2024). Computation and Privacy Protection for Satellite-Ground Digital Twin Networks. IEEE Transactions on Communications, 72(9), 5532-5546. doi: 10.1109/TCOMM.2024.3392795
[59] Xu, G., Kong, D., Zhang, K., Xu, S., Cao, Y., Mao, Y.,... Chen, X. (2024). A Model Value Transfer Incentive Mechanism for Federated Learning With Smart Contracts in AIoT. IEEE Internet of Things Journal. doi: 10.1109/JIOT.2024.3468443

[60] Wang, J., Bai, L., Fang, Z., Han, R., Wang, J.,... Choi, J. (2024). Age of Information Based URLLC Transmission for UAVs on Pylon Turn. IEEE Transactions on Vehicular Technology, 73(6), 8797-8809. doi: 10.1109/TVT.2024.3358844

[61] Zhang, X., Hou, D., Xiong, Z., Liu, Y., Wang, S.,... Li, Y. (2024). EALLR: Energy-Aware Low-Latency Routing Data Driven Model in Mobile Edge Computing. IEEE Transactions on Consumer Electronics. doi: 10.1109/TCE.2024.3507158.

[62] Naif Alsharabi, A. Bhardwaj, Abdulaziz Ayaba, and Amr Jadi, "Threat Hunting for Adversary Impact Inhibiting System Recovery," Computers & Security, pp. 104464–104464, Mar. 2025, doi: https://doi.org/10.1016/j.cose.2025.104464.

[63] IJCNC JOURNAL, "IJCNC 01," International Journal of Computer Networks & Communications (IJCNC), Aug. 17, 2024. https://ijcnc.com/2024/08/17/ijcnc-01-39/ (accessed Mar. 21, 2026).

[64] N. Alsharabi, A. Bhardwaj, T. Alshammari, S. alotaibi, D. Alshammari, and A. Jadi, "IAPN: Framework to secure IoT-based infrastructures using Private APN," Egyptian Informatics Journal, vol. 30, p. 100671, Mar. 2025, doi: https://doi.org/10.1016/j.eij.2025.100671.

[65] IJCNC JOURNAL, "IJCNC 07," International Journal of Computer Networks & Communications (IJCNC), Aug. 23, 2025. https://ijcnc.com/2025/08/23/ijcnc-07-42/ (accessed Mar. 21, 2026).