SvedbergOpen
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

Explainable Analysis of ANN-Based Intrusion Detection Classifiers: A Comparative Evaluation on ToN-IoT and RT-IoT 2022 Datasets

Samir Kumar Patro[1], Chinmaya Kumar Nayak[2], Ashalata Panigrahi[3], Binod Kumar Pattanayak[4]

[1]Faculty of Engineering and Technology, Sri Sri University, Cuttack, India. E-mail: samir.p2021 22ds@srisriuniversity.edu.in
[2]Faculty of Engineering and Technology, Sri Sri University, Cuttack, India. E-mail:chinmaya.n@srisriuniversity.edu.in,
[3]NIST University, Berhampur, India. E-mail: ashalata.panigrahi@nist.edu
[4]Department of Computer Science & Engineering. ITER, SOA Deemed to be University, Bhubaneswar, India.
E-mail: binodpattanayak@soa.ac.in

Abstract
Internet of Things (IoT) has opened up many innovative applications but the unique characteristics of IoT devices have made it vulnerable to cyber-attacks. To deal with these threats, robust mechanisms are to be built to ensure safe and reliable IoT ecosystems. In this work, an Artificial Neural Network (ANN) based approach has been investigated to build intrusion detection models that are capable of analyzing anomalous behavior of network users and flag deviations from normal network activities. Four ANN based classification techniques, viz., Radial Basis Function Network (RBFN), Self-Organizing Map (SOM), Deep Neural Network (DNN), and Multi-pass Learning Vector Quantization (MLVQ) have been used to classify communication data in an IoT network. In order to enhance the model performance, feature engineering has been applied using five feature selection techniques namely, information gain, gain ratio, symmetrical uncertainty, Relief-F, and chi-square. Results show that the DNN classifier consistently achieved the highest detection accuracy across all feature-selection subsets (up to 99.78%), while Relief-F produced the strongest overall feature subset across classifiers in ToN_IoT dataset. DNN classifier consistently achieved the highest detection accuracy across all feature-selection subsets (up to 99.52%), while Symmetric Uncertainty produced the strongest overall feature subset across classifiers of RT_IoT2022 dataset.
Keywords: IoT networks, intrusion detection, machine learning, artificial neural network, ToN-IoT data set,RT_IoT2022 data set, classification, performance analysis.

1. Introduction

Internet of Things (IoT) refers to a network of physical devices embedded with sensors and software components. These devices capture data from their environment and exchange over the internet. Because of its ability to gather data from remote locations on a real-time basis, IoT has become a popular technology that has pervaded into many critical applications in smart city, smart home, industrial automation, agriculture, healthcare, etc. The IoT ecosystem is continually evolving, driving new applications on a daily basis. There are several challenges to its wide-spread acceptance such as interoperability, scalability, and data privacy. But, protecting data and IoT devices from cyber-attacks is a major concern which is becoming more and more prevalent with the passage of time. Among others Intrusion Detection Systems (IDS) provide a viable defense mechanism to protect IoT systems from malicious activities. Signature-based IDS is one of the approaches which can detect known intrusion patterns. But, as the attackers maneuver subtle techniques to break into IoT networks, detecting such attacks becomes very difficult. Anomaly-based IDS has gained prominence due to its capability to detect previously unknown threats by identifying deviations from normal behavior. Machine learning algorithms are being explored in anomaly-based IDS to learn normal patterns and detect anomalies in IoT traffic. Though several such algorithms are being tried out, the machine learning models require continuous refinement to make them suitable for a dataset under consideration. Besides this, different data preprocessing technique are also required to achieve higher level of performance.

Contributions of this paper can be summarized as follows:

i. Resolving issues with the ToN-IoT dataset and ART_IoT dataset such as irrelevant features and scaling that impact the IDS's model performance.
ii. Application of feature selection methods namely Relief-F, information gain, gain ratio, symmetric uncertainty, and chi-square to select the most important features.
iii. The existing neural network-based algorithms are evaluated on the ToN-IoT dataset and ART_IoT dataset for binary intrusion detection.

iv. Fifteen evaluation criteria such as accuracy, precision, false positive rate have been used for performance assessment.
v. The efficacy of the proposed model has been compared with the existing works.

Rest of the article is organized as follows: Section 2 provides a detailed review of related works, section 3 presents the proposed model, section 4 deals with the experimental set up, section 5 presents and analyses the results, and section 6 results and discussions ,section 7 explainable AI(XAI) and section 8 highlights the contributions made in the present work with pointers to future research

2. Related Work

With the rapid growth of IoT ecosystem, the need for robust and efficient intrusion detection systems (IDS) has become increasingly crucial. By leveraging ML algorithms, IDSs can analyze network traffic, identify patterns, and learn from experience to detect unknown threats and anomalies, thereby enhancing the overall security posture of IoT. In the study by Jing Li et al. [1], feature extraction outperformed feature selection in IoT intrusion detection, showing better detection performance with fewer features. Imad Tareq et al. [2] demonstrated the superiority of the Inception Time model for cyber-attack detection for the ToN-IoT dataset. Guo et al. [3] proposed an ML-based IDS framework using the TON-IoT dataset, with the stacking-ensemble model achieving high performance in binary and multiclass classification. ToN-IoT dataset comprises data from all layers of the IoT system, including cloud, fog, and edge. Various ML methods were evaluated for both binary and multi-class classification problems across different nodes of the proposed distributed detection system [4]. The Synthetic Minority Oversampling Technique (SMOTE) was used for class balancing and comparative studies have shown that XGBoost consistently outperformed other ML algorithms in detecting intrusions across different scenarios achieving high accuracy in both binary and multi-class classifications [5]. Awotunde et al. [6] proposed an ensemble model using Chi-Square for selection of important features and various classifiers, with XGBoost showing superior performance. Injadat [7] combined Bayesian Optimization-Gaussian Process with ensemble tree-based learning, demonstrating improved detection accuracy. Mohy-Eddine et al. [8] designed an intrusion detection model using Isolation Forest and Pearson's Correlation Coefficient for feature engineering, combined with Random Forest classification. These approaches have shown promising results in detecting and classifying IIoT attacks, with high accuracy rates ranging from 99.18% to 99.99%. The studies highlight the effectiveness of ensemble methods and feature selection techniques in improving intrusion detection performance for IIoT security. O. Rabie et al. [9] proposed a framework combining Decisive Red Fox optimization and Descriptive Back Propagated Radial Basis Function classification, achieving improved detection accuracy and training speed. In this research focused on using artificial neural networks (ANNs) and deep learning for malware detection and classification in IoT networks. Alhanaya & Al-Shqeerat [10] analyzed various machine learning models and feature reduction techniques across multiple IoT datasets, highlighting the impact of dataset selection on performance. Elijah M. Maseno, & Zenghui Wang [11] proposed stacked ensemble intrusion detection technique which combines multiple base learner classifiers with a meta-learning algorithm to enhance performance of IDS.In [12] the authors have combined Hill-climbing method with random forest algorithm for detection of intrusions in IoT networks using the TON_IoT dataset. While Hill-climbing has been used for selection of relevant features, random forest algorithm has been applied for the separation of intrusions. Experimental results show that the RFHC mechanism achieved more acceptable results with respect to recall, precision, and accuracy compared to a baseline mechanism. Alshathri et al. [13] evaluated various ML algorithms on the TON_IoT dataset, with CART outperforming others in detecting intrusions across multiple IoT datasets.

3. Proposed Model Building Architecture

This section discusses the proposed intrusion detection architectural model for detecting intrusive behaviour of IoT network users. The model architecture contains several clearly defined stages of operations as shown in figure 1.

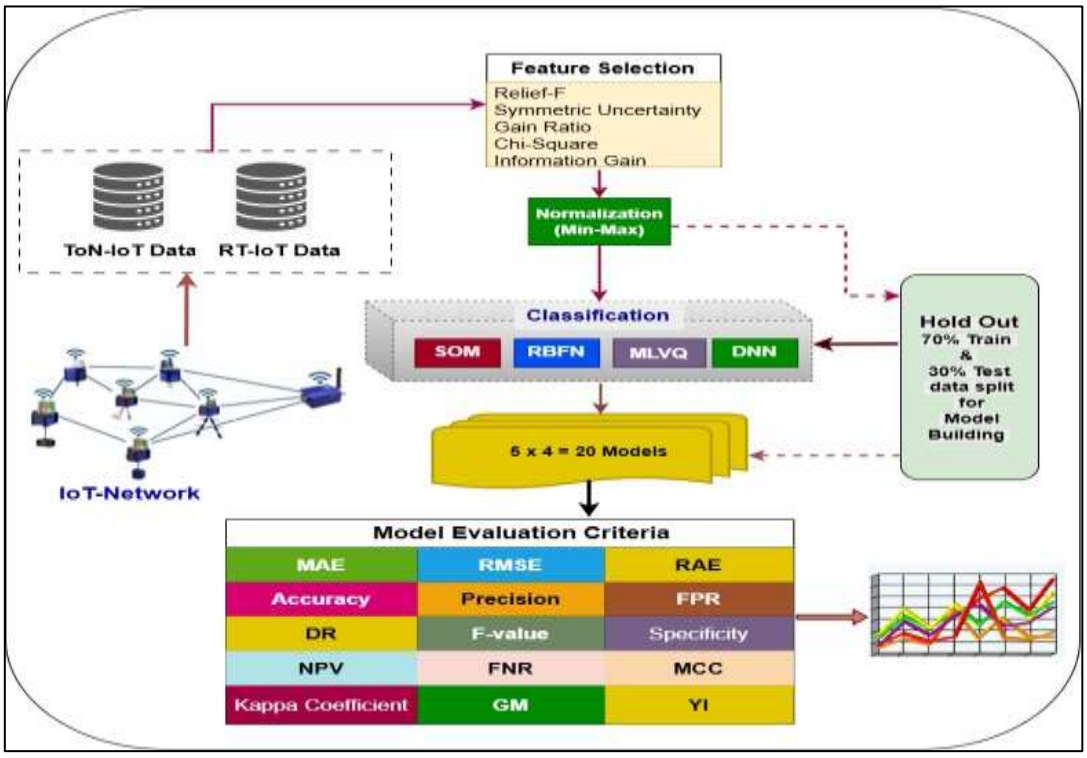

Figure 1: Proposed Workflow for intrusion detection Model in IoT Networks

3.1 Classification Model Building

In order to classify the intrusion data as belonging to normal or attack category, the following four competitive Artificial Neural Network (ANN) based classifiers have been chosen.

Self-Organizing Map (SOM)

SOM [16] is also known as Kohonen map which is essentially an unsupervised learning algorithm normally used for clustering. But it can also be used for classification by first clustering the data and then labelling the clusters based on the majority class.

Radial Basis Function Network (RBFN)

In RBFN [17], when a new input is fed into the network the hidden layer computes the radial basis function for each neuron based on the input. The output layer combines these values using the weights learnt so far to produce the final output, which is typically the class label.

Multi-Pass Learning Vector Quantization (MLVQ)

It is a prototype-based supervised learning algorithm designed for classification tasks. Multi-pass LVQ involves multiple passes through the training data to refine the prototype vectors [18]. Each class is represented by one or more prototype vectors which are essentially representative examples of each class. In LVQ, prototypes are initialized randomly or by selecting examples from the training data. Then the algorithm is allowed to iteratively adjust the prototypes based on the training data to better represent their respective classes. MLVQ, starts with a set of prototype vectors for each class which can be initialized by randomly selecting training examples or using some heuristic. For each training example, the nearest prototype vector is found using a distance metric (usually Euclidean distance). The entire dataset is passed through multiple times (thus "multi-pass"), each time refining the prototypes further. The number of passes and learning rate schedule are typically hyperparameters that need to be set. By making multiple passes through the data and fine-tuning the prototypes, the algorithm can effectively capture the underlying structure of the data and provide accurate classification results.

Deep Neural Network (DNN)

A Deep Neural Network (DNN) [19] consists of multiple hidden layers between the input and output layers. In a DNN, each layer learns a different set of features from the output of the previous layer, enabling the network to model complex nonlinear relationships effectively. Due to its deep architecture and powerful learning capability, a DNN can identify and classify new or previously unseen attacks in IoT networks with high accuracy[25][26].

Table 1: Hyper- parameters of the Machine Learning Model

Machine Learning Models	Hyper Parameters

SOM	Learning rate: 0.3, Neighbourhood size: 8, Neighbourhood Function: Gaussian, Topology: Hexagonal, Training iterations: 100
RBFN	Batch size 100, minStdDev 0.1, ridge 1.0E-8, Number of basis functions in hidden layer: 2
MLVQ	Learning Rate: 0.05, window Size: 0.3 , Codebook vectors: 20
DNN	Activation Function: Relu, Dropout: 0.3, Learning rate: 0.001

3.2 Model Evaluation

In this work, each of the 4 classifiers are combined with the 5 feature selection techniques and two validation methods. Thus, there are 20 different combinations of feature selection with classifiers leading to 20 possible models. The efficacy of each of the model is tested using 15 standard performance measurements, namely, Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Relative Absolute Error (RAE), Accuracy, Precision, False Positive Rate (FPR), Detection Rate, F-Value, Specificity, NPV, FNR, MCC, Kappa Coefficient, YI and Geometric Mean (GM).

3.2.1 Confusion Matrix

Confusion matrix is a M × M table which represents four values, namely, ĂTN (True Negative), ČFP (False Positive), ĎFN (False Negative), and ĚTP (True Positive). The matrix elements help one to calculate various performance measurements indicating the quality of the mode.

		Predicted	
		Normal	Attack
Actual	Normal	**$\breve{A}_{TN}$**	**$\check{C}_{FP}$**
	Attack	**$\check{D}_{FN}$**	**$\check{E}_{TP}$**

3.2.2 Model Evaluation Metrics

MAE is the average of all absolute errors which is calculated as:

$$MAE = (1 / t) \sum_{i=1}^{t} |z_i - z| \quad (1)$$

where t = the number of errors, and |zi – z| = the absolute error

RMSE is calculated as:

$$RMSE = \text{Sqrt}((1 / t) \sum_{i=1}^{t} (yp_i - ya_i)) \quad (2)$$

where ypi = predicted output, yai = actual output.

Absolute Relative Error (ARE) = |(true value - estimated value) / true value|

$$\text{Accuracy} = (\check{E}_{TP} + \breve{A}_{TN}) / (\check{E}_{TP} + \breve{A}_{TN} + \check{C}_{FP} + \check{D}_{FN}) \quad (3)$$

$$\text{Sensitivity or TPR or Recall} = (\check{E}_{TP}) / (\check{E}_{TP} + \check{D}_{FN}) \quad (4)$$

$$\text{Precision (PPV)} = (\check{E}_{TP}) / (\check{E}_{TP} + \check{C}_{FP}) \quad (5)$$

$$\text{False Positive Rate (FPR)} = (\check{C}_{FP}) / (\breve{A}_{TN} + \check{C}_{FP}) \quad (6)$$

$$\text{F-Score} = 2 \times PPV \times TPR / (PPV + TPR) \quad (7)$$

$$\text{Specificity (or TNR)} = (\breve{A}_{TN}) / (\breve{A}_{TN} + \check{C}_{FP}) \quad (8)$$

$$NPV = (\breve{A}_{TN}) / (\breve{A}_{TN} + \check{D}_{FN}) \quad (9)$$

$$\text{False Negative Rate (FNR)} = (\check{D}_{FN}) / (\check{D}_{FN} + \check{E}_{TP}) \quad (10)$$

Matthews Correlation Coefficient (MCC) =

$$[(\check{E}_{TP} \times \breve{A}_{TN}) - (\check{C}_{FP} \times \check{D}_{FN})] / \text{Sqrt}[(\check{E}_{TP} + \check{C}_{FP})(\check{E}_{TP} + \check{D}_{FN})(\breve{A}_{TN} + \check{C}_{FP})(\breve{A}_{TN} + \check{D}_{FN})] \quad (11)$$

$$\text{Error Rate} = (\check{C}_{FP} + \check{D}_{FN}) / (\check{E}_{TP} + \breve{A}_{TN} + \check{C}_{FP} + \check{D}_{FN}) \quad (12)$$

Cohen's Kappa Coefficient, (KC) =

[Total Accuracy ($\check{T}_A$) – Random Accuracy ($\check{R}_A$)] / [1 – Random Accuracy ($\check{R}_A$)]

where

$\check{T}_A = (\check{E}_{TP} + \breve{A}_{TN}) / (\check{E}_{TP} + \breve{A}_{TN} + \check{C}_{FP} + \check{D}_{FN})$

$\check{R}_A = [(\breve{A}_{TN} + \check{C}_{FP})(\breve{A}_{TN} + \check{D}_{FN}) + (\check{E}_{TP} + \check{D}_{FN})(\check{E}_{TP} + \check{C}_{FP})] / (\check{E}_{TP} + \breve{A}_{TN} + \check{C}_{FP} + \check{D}_{FN})^2$

Geometric Mean (GM) = Sqrt(Sensitivity × specitivity)

Youden's Index (YI) = Sensitivity + Specificity – 1

4 Experimental Setup

4.1. Dataset

The ToN-IoT (Telemetry and Operational Network Internet of Things) dataset is a comprehensive and diverse collection of data used for building and testing machine learning models for intrusion detection, anomaly detection, and other cybersecurity applications. The dataset is labelled, indicating whether the data is associated with normal or malicious activities, thus helps in supervised learning for building IDS systems. The ToN-IoT dataset consists of a wide range of attributes that capture various aspects of network

traffic, telemetry data, and system logs. The network ToN-IoT dataset consists of 44 attributes and an attack-type labeled as normal or attack. Thus, the ToN-IoT dataset serves as a valuable resource to analyze these attributes and identify patterns associated with normal and malicious activities.
The train-test ToN-IoT dataset consists of 461043 records out of which 300000 records indicate normal and 161043 records indicate attack instances. The attack instances are further classified into nine attack categories, namely, scanning, DoS (Denial of Services), injection, DDoS (Distributed Denial of Services), password, XSS (Cross-Site Scripting), ransomware, backdoor, and MITM (Man-In-The-Middle). For a balanced sampling of data, only 50291 number of normal instances are taken randomly for analysis. The distribution of records are provided in Table 2.

Table 2: Normal and Attack Instances in the ToN-IoT Dataset

Class	Number of Records	% of Occurrence
Normal	50291	23.8
Attack	161043	76.2

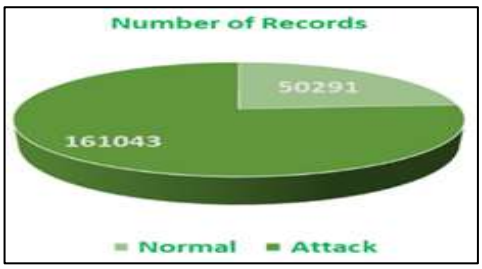

Figure 2: Distribution of Records

5 Results and Discussion

The four neural network-based classifiers (RBFN, SOM, DNN, MLVQ) are combined with five feature selection methods (information gain, gain ratio, symmetrical uncertainty, Relief F and chi-squared) to build 20 different models. Each of the model is evaluated with respect to 15 different metrics. For better analysis the evaluation results are split into three tables.
Table 3 provides performance evaluation of the 20 models with respect to the metrics MAE, RMSE, RAE, accuracy, and precision. Looking across the whole table, the DNN classifier consistently gives the best results no matter which feature selection method is paired with it, and among these the Symmetric Uncertainty + DNN combination comes out on top with the lowest error values (MAE 0.0022, RMSE 0.0465, RAE 0.93) and the highest accuracy (99.78), while Relief F + DNN is a very close second, actually edging it out slightly on precision (0.9990 vs 0.9978); on the other end, MLVQ paired with Symmetric Uncertainty gives the weakest performance overall, with the highest MAE (0.0906), highest RMSE (0.3009), highest RAE (38.95), and the lowest accuracy (90.94) and precision (0.8975) in the entire table, and SOM and RBFN classifiers generally sit in the middle, performing noticeably better than MLVQ but far behind DNN across every feature selection method. Overall, the pattern shows that error metrics (MAE, RMSE, RAE) drop sharply and accuracy/precision rise sharply whenever DNN is used, meaning the choice of classifier has a much bigger impact on performance than the choice of feature selection technique, though Symmetric Uncertainty and Relief F do give DNN a slight extra edge compared to Info gain, Gain Ratio, or Chi square. We can refer the figure 4 for the comparison.

Table 4 provides performance evaluation of the 20 models with respect to specificity, FNR, NPV, FAR, and recall. Across specificity, FNR, NPV, FPR, and recall, DNN again outperforms every other classifier under all feature selection methods. The best overall result comes from Relief F + DNN, with the highest specificity (0.9967) and lowest FPR (0.0033), while Symmetric Uncertainty + DNN records the lowest FNR (0.0006) and the highest NPV (0.9981) and Recall (0.9994) in the entire table. Gain Ratio + DNN is also strong, with specificity 0.9912 and FPR 0.0088. On the weaker side, Gain Ratio + MLVQ has the lowest specificity (0.6318), and Symmetric Uncertainty + SOM has the highest FPR (0.3405). Chi square + SOM shows a high FNR (0.0256), and Relief F + MLVQ shows moderate weakness with specificity 0.7140 and FPR 0.2860. SOM and MLVQ consistently show lower specificity and higher FPR/FNR than RBFN and DNN. RBFN performs in the mid-range across all methods (specificity around 0.72–0.78, FPR around 0.21–0.28). Overall, DNN combinations achieve near-perfect specificity, NPV, and recall with minimal FNR and FPR, confirming that classifier selection has a stronger impact on performance than the feature selection method, with

Symmetric Uncertainty and Relief F giving DNN the best results. We can refer the figure 5 for the comparison.

Table 3: MAE, RMSE, RAE, Accuracy, and Precision for the 20 Models

Feature Selection Method	Classifier Techniques	Evaluation Metric				
		MAE	RMSE	RAE	Accuracy	Precision
Info gain	SOM	0.0765	0.2766	32.90	92.35	0.9150
	RBFN	0.0597	0.2443	25.66	94.03	0.9327
	MLVQ	0.0785	0.2802	33.76	92.15	0.9130
	DNN	0.0091	0.0955	3.92	99.09	0.9919
Gain Ratio	SOM	0.0857	0.2927	36.85	91.43	0.9059
	RBFN	0.0841	0.2900	36.18	91.59	0.9201
	MLVQ	0.0879	0.2965	37.80	91.21	0.8994
	DNN	0.0032	0.0563	1.36	99.68	0.9973
Chi square	SOM	0.0902	0.3003	38.78	90.98	0.9138
	RBFN	0.0690	0.2627	29.68	93.10	0.9261
	MLVQ	0.0768	0.2772	33.05	92.32	0.9187
	DNN	0.0029	0.0543	1.27	99.71	0.9983
Symmetric Uncertainty	SOM	0.0858	0.2930	36.92	91.42	0.9058
	RBFN	0.0823	0.2869	35.41	91.77	0.9200
	MLVQ	0.0906	0.3009	38.95	90.94	0.8975
	DNN	0.0022	0.0465	0.93	99.78	0.9978
Relief F	SOM	0.0851	0.2916	36.58	91.49	0.9009
	RBFN	0.0546	0.2337	23.49	94.54	0.9385
	MLVQ	0.0747	0.2733	32.11	92.53	0.9195
	DNN	0.0023	0.0483	1.00	99.77	0.9990

Table 4: Specificity, FNR, NPV, FAR, and Recall for the 20 Models

Feature Selection Method	**Classifier Techniques**	**Evaluation Metric**				
		Specificity	**FNR**	**NPV**	**FPR**	**Detection Rate/ Recall**
Info gain	SOM	0.6956	0.0075	0.9658	0.3044	0.9925
	RBFN	0.7634	0.0061	0.9744	0.2366	0.9939
	MLVQ	0.6877	0.0077	0.9645	0.3123	0.9923
	DNN	0.9730	0.0037	0.9876	0.0270	0.9963
Gain Ratio	SOM	0.6602	0.0087	0.9584	0.3398	0.9913
	RBFN	0.7204	0.0249	0.8976	0.2796	0.9751
	MLVQ	0.6318	0.0030	0.9847	0.3682	0.9970

	DNN	0.9912	0.0014	0.9952	0.0088	0.9986
Chi square	SOM	0.6968	0.0256	0.8917	0.3032	0.9744
	RBFN	0.7395	0.0110	0.9532	0.2605	0.9890
	MLVQ	0.7116	0.0127	0.9442	0.2884	0.9873
	DNN	0.9945	0.0022	0.9928	0.0055	0.9978
Symmetric Uncertainty	SOM	0.6595	0.0087	0.9583	0.3405	0.9913
	RBFN	0.7192	0.0222	0.9074	0.2808	0.9778
	MLVQ	0.6246	0.0043	0.9779	0.3754	0.9957
	DNN	0.9926	0.0006	0.9981	0.0074	0.9994
Relief F	SOM	0.6373	0.0009	0.9951	0.3627	0.9991
	RBFN	0.7849	0.0060	0.9754	0.2151	0.9940
	MLVQ	0.7140	0.0106	0.9531	0.2860	0.9894
	DNN	0.9967	0.0020	0.9933	0.0033	0.9980

Table 5 provides performance evaluation of the 20 models with respect to the metrics GM, YI, F-Value, KC, and MCC which is also depicted in Figure 6. Looking at GM, YI, F-Value, KC and MCC together, DNN again gives the strongest results with every feature selection method. The best overall result is Relief F with DNN, which has the highest GM (0.9973), YI (0.9946), F-Value (0.9985), KC (0.9935) and MCC (0.9935), all together in one row. Gain Ratio with DNN is also very strong, with GM 0.9948 and YI 0.9897. On the low end, Gain Ratio with MLVQ gives the weakest results, with the lowest GM (0.7937), YI (0.6288) and F-Value (0.9457). Symmetric Uncertainty with MLVQ is also weak, with a low KC (0.7097) and MCC (0.7369). Info gain with MLVQ shows similar low performance too. Across the table, SOM and MLVQ generally give lower GM, YI, KC and MCC values, while RBFN performs moderately, sitting between SOM/MLVQ and DNN. So overall, DNN is the clear best classifier for all five metrics, and Relief F is the best feature selection method to pair it with, while MLVQ combined with Gain Ratio or Symmetric Uncertainty gives the weakest performance.

Table 5: GM, YI, F-Value, KC, and MCC for the 20 models

Feature Selection Method	**Classifier Techniques**	**Evaluation Metric**				
		GM	**YI**	**F-Value**	**KC**	**MCC**
Info gain	SOM	0.8309	0.6881	0.9522	0.7625	0.7785
	RBFN	0.8711	0.7574	0.9624	0.8192	0.8289
	MLVQ	0.8261	0.6800	0.9510	0.7556	0.7724
	DNN	0.9846	0.9693	0.9941	0.9743	0.9744
Gain Ratio	SOM	0.8090	0.6515	0.9467	0.7308	0.7504
	RBFN	0.8381	0.6955	0.9468	0.7469	0.7541
	MLVQ	0.7937	0.6288	0.9457	0.7186	0.7456
	DNN	0.9948	0.9897	0.9979	0.9911	0.9911
Chi square	SOM	0.8240	0.6711	0.9431	0.7265	0.7353

	RBFN	0.8552	0.7285	0.9565	0.7903	0.8004
	MLVQ	0.8382	0.6988	0.9517	0.7645	0.7766
	DNN	0.9962	0.9923	0.9981	0.9918	0.9918
Symmetric Uncertainty	SOM	0.8086	0.6508	0.9466	0.7302	0.7499
	RBFN	0.8386	0.6970	0.9480	0.7513	0.7594
	MLVQ	0.7886	0.6203	0.9441	0.7097	0.7369
	DNN	0.9960	0.9920	0.9986	0.9939	0.9939
Relief F	SOM	0.7979	0.6363	0.9475	0.7275	0.7551
	RBFN	0.8833	0.7789	0.9654	0.8358	0.8437
	MLVQ	0.8405	0.7033	0.9531	0.7707	0.7834
	DNN	0.9973	0.9946	0.9985	0.9935	0.9935

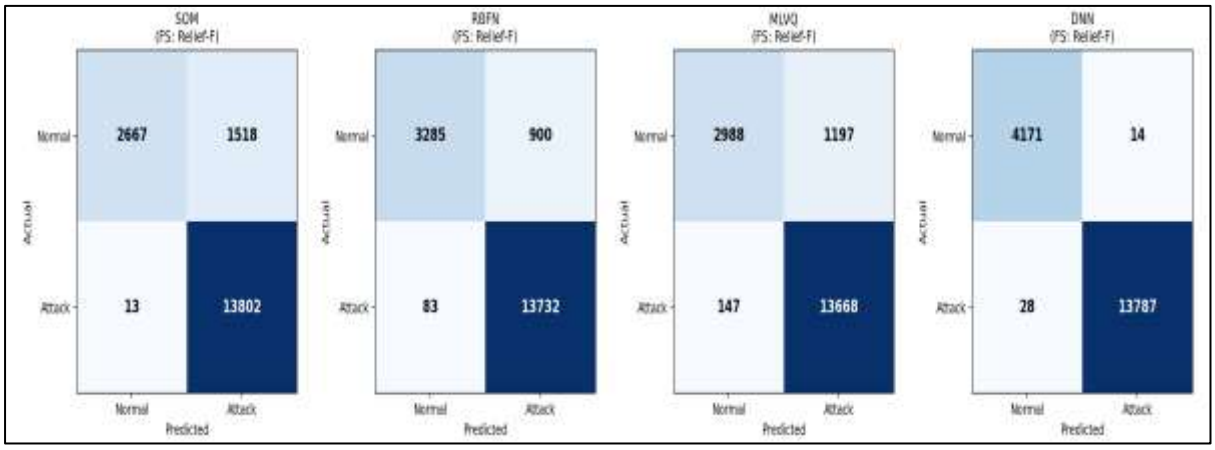

Figure 3. Confusion matrix heatmaps for SOM, RBFN, MLVQ, and DNN (Feature Selection: Relief-F).

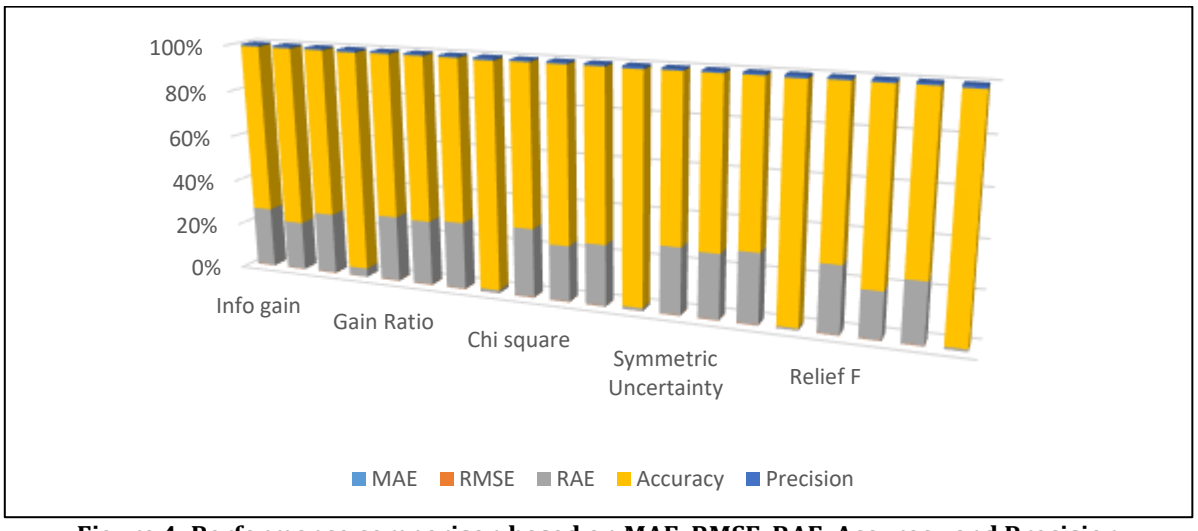

Figure 4: Performance comparison based on MAE, RMSE, RAE, Accuracy and Precision

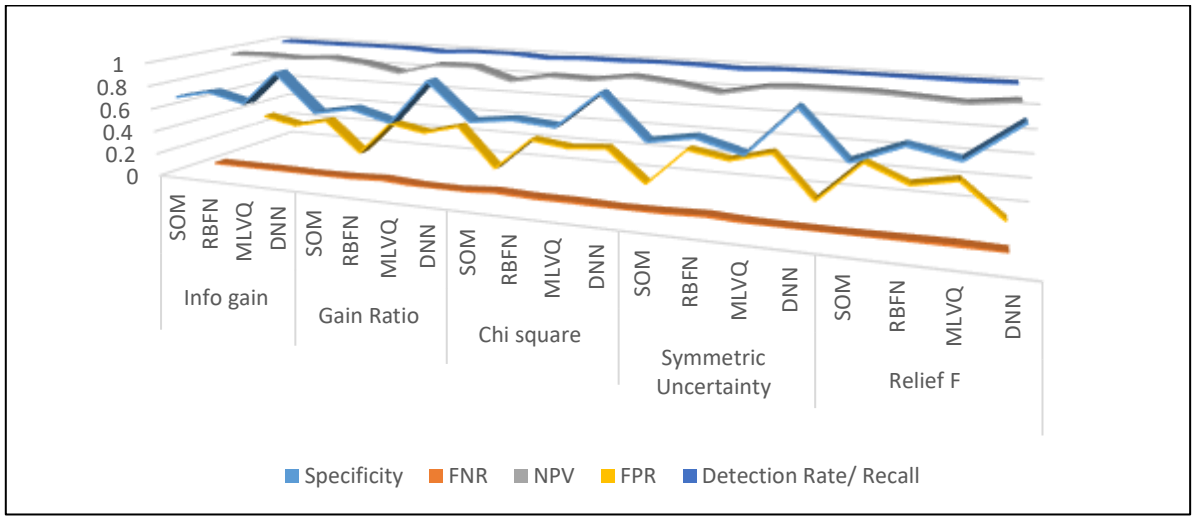

Figure 5: Performance comparison based on Recall, FAR, NPV, FNR, Specificity

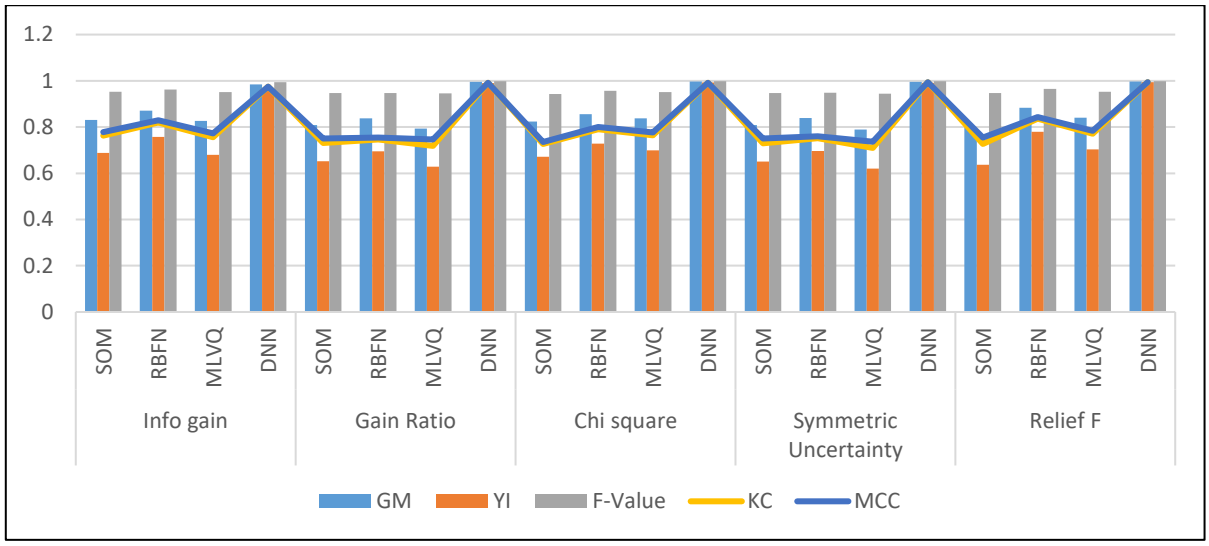

Figure 6: Performance comparison based on GM, YI, F-Value, KC, MCC

The comparison of classification techniques—SOM, RBFN, MLVQ, and DNN—shows a clear performance hierarchy across all evaluation metrics. RBFN emerges as the best-performing model, achieving the highest values in accuracy of 0.9998, precision of 0.9999, recall of 0.9998, and other metrics, while maintaining the lowest error rates such as MAE

It is evident from the above comparison, RBFN consistently achieves maximum values across almost all performance metrics, while SOM shows the minimum values, making them the best and worst performers respectively. Although feature selection methods such as Information Gain, Symmetrical Uncertainty, and Chi-Square are applied, they have minimal impact on the overall trend, as the relative performance of the classifiers remains unchanged. Overall, the models can be ranked from highest to lowest performance as RBFN, followed by DNN, MLVQ, and SOM.

6. Dataset

The **RT-IoT2022** dataset used in this study contains 123,117 labeled network-flow records with 85 attributes describing bidirectional flow-level statistics — packet counts, header sizes, TCP flag counts, payload statistics, inter-arrival times, subflow and bulk-transfer measures, and window sizes — together with a categorical Attack_type field mapped to a binary Normal/Attack label [7], [8]. Categorical attributes were label-encoded and all numerical features were normalized using min-max scaling to the range [0, 1], consistent with standard IoT-IDS preprocessing practice [5]. A stratified subsample of 60,000 records was drawn to preserve the original class distribution (89.8% attack, 10.2% normal) while keeping the twenty feature-selection × classifier permutations computationally tractable, and each permutation was trained and tested using a 70:30 stratified split, matching the protocol requested for this study

6.1 Results and Discussion

In Table 6 Comparing MAE, RMSE, RAE, accuracy and precision together, DNN gives the best results across all feature selection methods. The strongest combination overall is Gain Ratio with DNN, which has the lowest MAE (0.0048), lowest RMSE (0.0695), lowest RAE (4.76), and the highest accuracy (99.52). Chi square with MLVQ has the highest precision in the table (0.9980), very close to Gain Ratio with DNN (0.9978). On the weaker side, Info gain with MLVQ performs the worst, with the highest MAE (0.0584), highest RAE (57.46), and the lowest accuracy (94.16). Chi square with MLVQ also shows a high RMSE (0.2350), and Gain Ratio with SOM has the lowest precision (0.9518). Overall, SOM and MLVQ consistently show higher error values and lower accuracy compared to RBFN and DNN, while DNN stays close to near-perfect accuracy and very low error across every feature selection method. This confirms that DNN is the best-performing classifier here, with Gain Ratio and Chi square giving it the strongest support, while MLVQ combined with Info gain gives the weakest overall performance. We can refer the figure 8 for the comparison.

Table 6: MAE, RMSE, RAE, Accuracy, and Precision for the 20 Models

Feature Selection Method	Classifier Techniques	Evaluation Metric				
		MAE	RMSE	RAE	Accuracy	Precision
Info gain	SOM	0.0457	0.2138	45.00	95.43	0.9860
	RBFN	0.0352	0.1877	34.66	96.48	0.9948
	MLVQ	0.0584	0.2416	57.46	94.16	0.9619
	DNN	0.0056	0.0745	5.47	99.44	0.9966
Gain Ratio	SOM	0.0517	0.2273	50.85	94.83	0.9518
	RBFN	0.0352	0.1875	34.61	96.48	0.9889
	MLVQ	0.0547	0.2338	53.80	94.53	0.9621
	DNN	0.0048	0.0695	4.76	99.52	0.9978
Chi square	SOM	0.0457	0.2138	45.00	95.43	0.9860
	RBFN	0.0337	0.1835	33.13	96.63	0.9942
	MLVQ	0.0552	0.2350	54.35	94.48	0.9980
	DNN	0.0074	0.0860	7.27	99.26	0.9946
Symmetric Uncertainty	SOM	0.0433	0.2080	42.59	95.67	0.9744
	RBFN	0.0339	0.1841	33.35	96.61	0.9944
	MLVQ	0.0550	0.2345	54.13	94.50	0.9964
	DNN	0.0053	0.0730	5.25	99.47	0.9961

Relief F	SOM	0.0497	0.2230	48.93	95.03	0.9859
	RBFN	0.0348	0.1865	34.23	96.52	0.9900
	MLVQ	0.0492	0.2217	48.39	95.08	0.9967
	DNN	0.0074	0.0863	7.33	99.26	0.9952

In Table 7 specificity, FNR, NPV, FPR and recall together, the results are a bit mixed compared to earlier tables. The best specificity comes from Chi square with MLVQ (0.9836), followed closely by Gain Ratio with DNN (0.9803). The lowest FNR and highest NPV both come from Symmetric Uncertainty with DNN (FNR 0.0020, NPV 0.9822), which also gives the highest recall (0.9980). The lowest FPR is from Chi square with MLVQ (0.0164). On the weak side, Gain Ratio with SOM performs worst overall, with the lowest specificity (0.5555) and the highest FPR (0.4445). Chi square with MLVQ, despite its strong specificity and FPR, shows the highest FNR (0.0596) and the lowest NPV (0.6511) and lowest recall (0.9404) in the table, showing an unusual trade-off. Overall, DNN remains the most balanced and reliable classifier, performing consistently well across specificity, FNR, NPV and recall in almost every feature selection method, while SOM and MLVQ show inconsistent results, sometimes strong in one metric but weak in others, especially Gain Ratio with SOM which is the weakest combination overall. We can refer the figure 9 for the comparison.

Table 7: Specificity, FNR, NPV, FAR, and Recall for the 20 Models (Boldfaced Values Represent the Best Values when Compared with other Values)

Feature Selection Method	**Classifier Techniques**	**Evaluation Metric**				
		Specificity	**FNR**	**NPV**	**FPR**	**Detection Rate/ Recall**
Info gain	SOM	0.8792	0.0372	0.7276	0.1208	0.9628
	RBFN	0.9552	0.0341	0.7599	0.0448	0.9659
	MLVQ	0.6594	0.0265	0.7381	0.3406	0.9735
	DNN	0.9699	0.0028	0.9753	0.0301	0.9972
Gain Ratio	SOM	0.5555	0.0072	0.8967	0.4445	0.9928
	RBFN	0.9038	0.0283	0.7834	0.0962	0.9717
	MLVQ	0.6599	0.0224	0.7693	0.3401	0.9776
	DNN	0.9803	0.0032	0.9723	0.0197	0.9968
Chi square	SOM	0.8792	0.0372	0.7276	0.1208	0.9628
	RBFN	0.9502	0.0318	0.7714	0.0498	0.9682
	MLVQ	0.9836	0.0596	0.6511	0.0164	0.9404
	DNN	0.9524	0.0028	0.9743	0.0476	0.9972
Symmetric Uncertainty	SOM	0.7726	0.0224	0.7956	0.2274	0.9776

	RBFN	0.9519	0.0323	0.7693	0.0481	0.9677
	MLVQ	0.9699	0.0578	0.6549	0.0301	0.9422
	DNN	0.9650	0.0020	0.9822	0.0350	0.9980
Relief F	SOM	0.8792	0.0417	0.7046	0.1208	0.9583
	RBFN	0.9136	0.0289	0.7812	0.0864	0.9711
	MLVQ	0.9721	0.0516	0.6807	0.0279	0.9484
	DNN	0.9574	0.0035	0.9690	0.0426	0.9965

In Table 8 GM, YI, F-Value, KC and MCC together, the results here are fairly close across DNN and RBFN, unlike the clear DNN dominance seen in earlier tables. The best overall combination is Gain Ratio with DNN, which has the highest GM (0.9885), highest YI (0.9772), highest F-Value (0.9973), and highest KC and MCC (0.9736 each). Symmetric Uncertainty with DNN is very close behind, with GM 0.9814 and KC/MCC 0.9706. Interestingly, RBFN performs strongly too, with Symmetric Uncertainty + RBFN giving a high KC and MCC (0.8321/0.8381), better than several DNN rows in the FPR-related tables. On the weak side, Gain Ratio with SOM is the worst overall, with the lowest GM (0.7426), lowest YI (0.5483), and lowest KC/MCC (0.6596/0.6821). Info gain with MLVQ is also weak, with GM 0.8012 and KC/MCC around 0.66. Overall, DNN still gives the most consistent high performance across all five metrics, but RBFN comes close in several rows, while SOM and MLVQ show the weakest and most inconsistent results, especially Gain Ratio with SOM. We can refer the figure 10 for the comparison.

Table 8: GM, YI, F-Value, KC, and MCC for the 12 models

Feature Selection Method	**Classifier Techniques**	**Evaluation Metric**				
		GM	**YI**	**F-Value**	**KC**	**MCC**
Info gain	SOM	0.9200	0.8419	0.9742	0.7707	0.7751
	RBFN	0.9605	0.9210	0.9801	0.8268	0.8337
	MLVQ	0.8012	0.6329	0.9677	0.6643	0.6656
	DNN	0.9835	0.9671	0.9969	0.9695	0.9695
Gain Ratio	SOM	0.7426	0.5483	0.9719	0.6596	0.6821
	RBFN	0.9371	0.8755	0.9803	0.8197	0.8223
	MLVQ	0.8032	0.6375	0.9698	0.6804	0.6829
	DNN	0.9885	0.9772	0.9973	0.9736	0.9736
Chi square	SOM	0.9200	0.8419	0.9742	0.7707	0.7751
	RBFN	0.9592	0.9184	0.9810	0.8328	0.8385
	MLVQ	0.9617	0.9240	0.9684	0.7534	0.7745
	DNN	0.9745	0.9496	0.9959	0.9591	0.9592
Symmetric Uncertainty	SOM	0.8690	0.7501	0.9760	0.7599	0.7600
	RBFN	0.9598	0.9196	0.9809	0.8321	0.8381

	MLVQ	0.9560	0.9121	0.9685	0.7517	0.7707
	DNN	0.9814	0.9630	0.9970	0.9706	0.9706
Relief F	SOM	0.9179	0.8375	0.9719	0.7546	0.7605
	RBFN	0.9419	0.8847	0.9805	0.8228	0.8260
	MLVQ	0.9602	0.9205	0.9720	0.7737	0.7897
	DNN	0.9767	0.9539	0.9959	0.9590	0.9590

Confusion Matrix Heatmaps

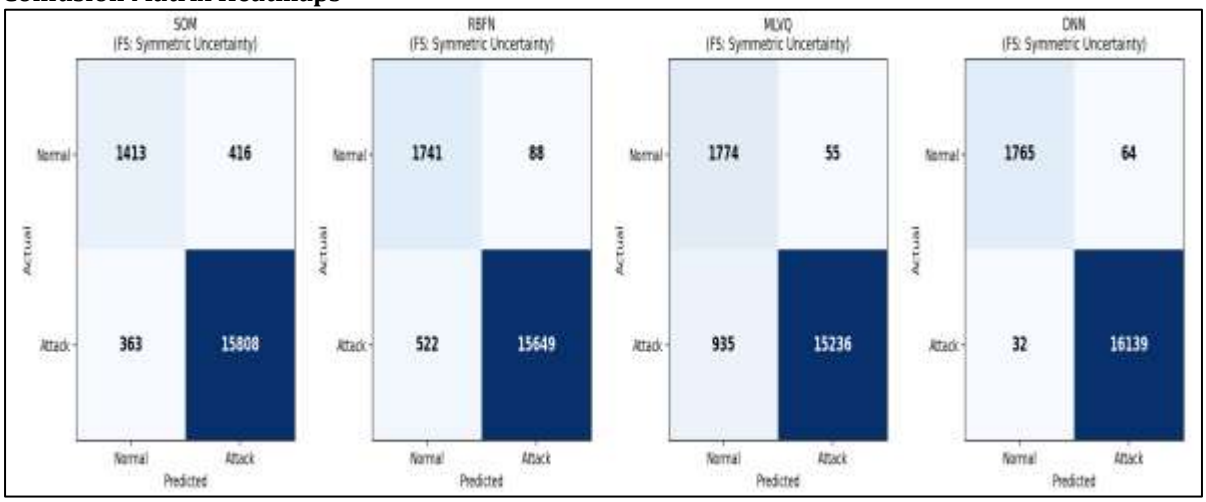

Figure. 7. Confusion matrix heatmaps for SOM, RBFN, MLVQ, and DNN (Feature Selection: Symmetric Uncertainty).

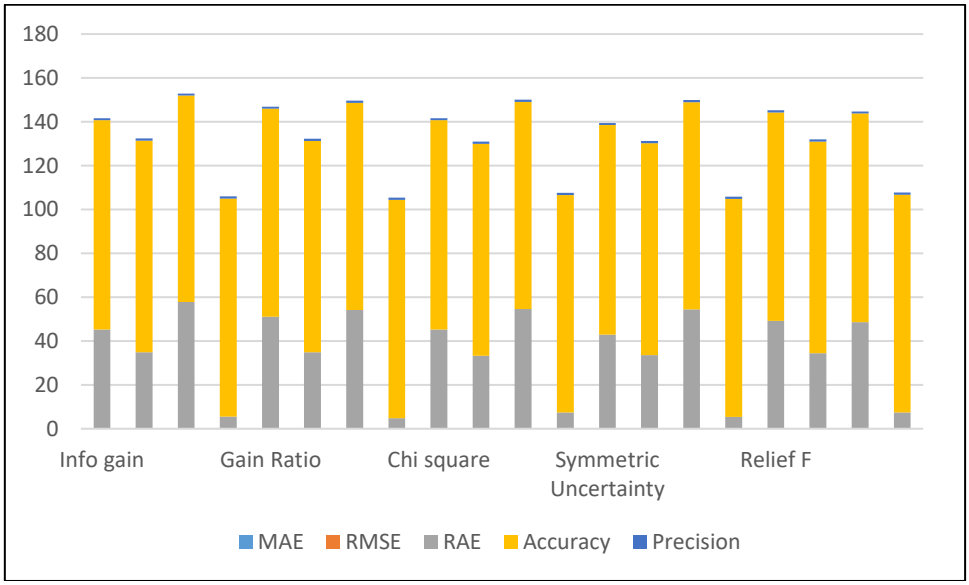

Figure 8. Confusion matrix heatmaps for SOM, RBFN, MLVQ, and DNN (Feature Selection: Relief-F).

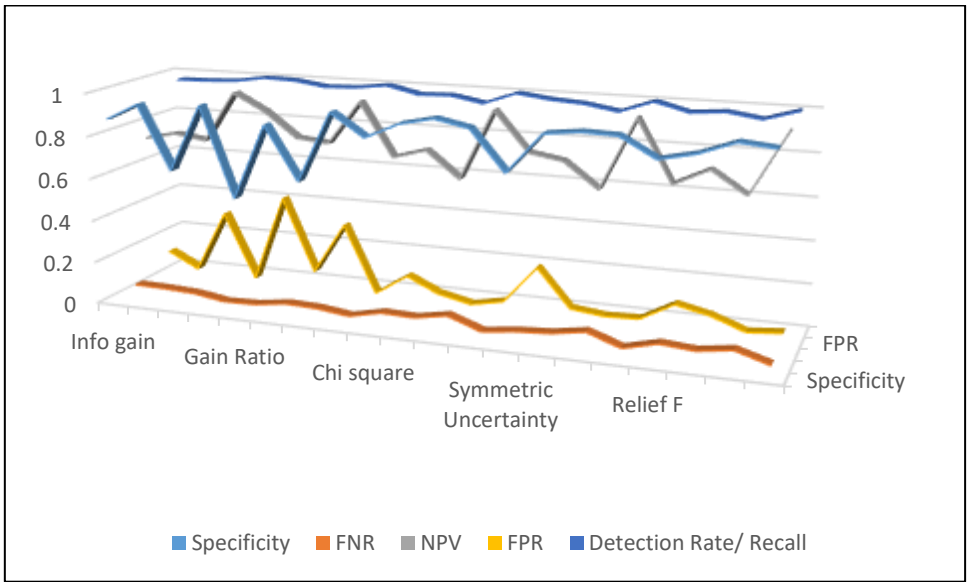

Figure 9: Performance comparison based on Recall, FAR, NPV, FNR, Specificity

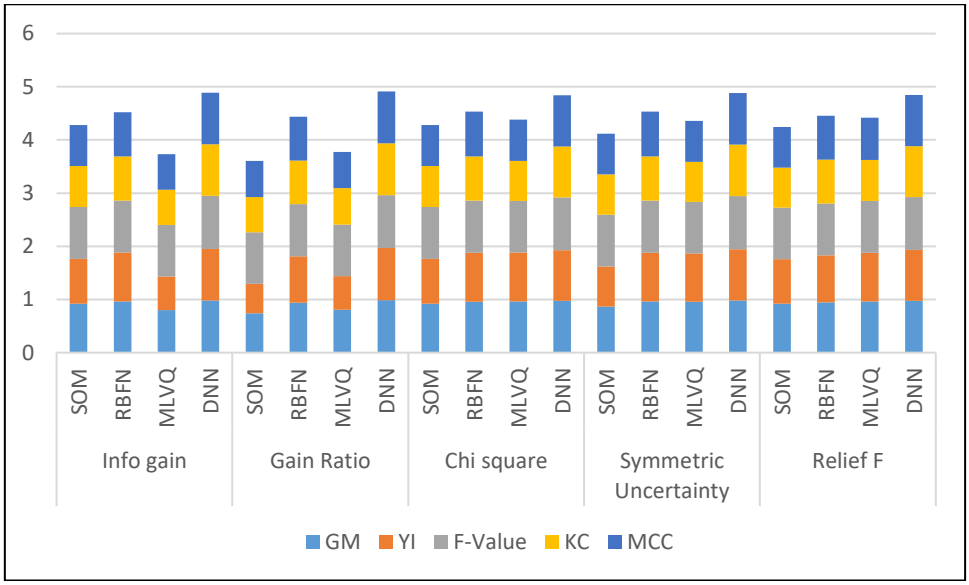

Figure 10: Performance comparison based on GM, YI, F-Value, KC, MCC

7. Explainable AI (XAI) Analysis of ANN-Based Intrusion Detection

Model transparency is increasingly required for intrusion detection systems deployed in security-critical IoT environments, since operators need to understand *why* a flow was flagged as an attack before acting on it [27]. Because the four classifiers evaluated in this study (SOM, RBFN, MLVQ, DNN) differ fundamentally in their internal structure — competitive clustering, kernel-based discrimination, prototype-based vector quantization, and a differentiable deep network respectively — a single post-hoc, model-agnostic explanation technique was applied uniformly across all four: permutation feature importance [28]. For each feature, values were randomly shuffled across the test set and the resulting drop in accuracy (averaged over 5 repeats) was recorded as that feature's importance; a larger drop indicates the classifier relies more heavily on that feature to make correct decisions. This analysis was performed on the Symmetric Uncertainty feature-selection subset (18 features), which produced the best mean accuracy across classifiers in the main study.

Because DNN is a genuine differentiable neural network, it was additionally explained using SHAP (SHapley Additive exPlanations) with a Kernel Explainer [29], which decomposes each individual prediction into per-feature contributions and reveals both the magnitude and the direction (increases vs. decreases attack probability) of each feature's effect — a complementary view that permutation importance alone cannot provide.

7.1 Permutation Feature Importance — All Four Classifiers

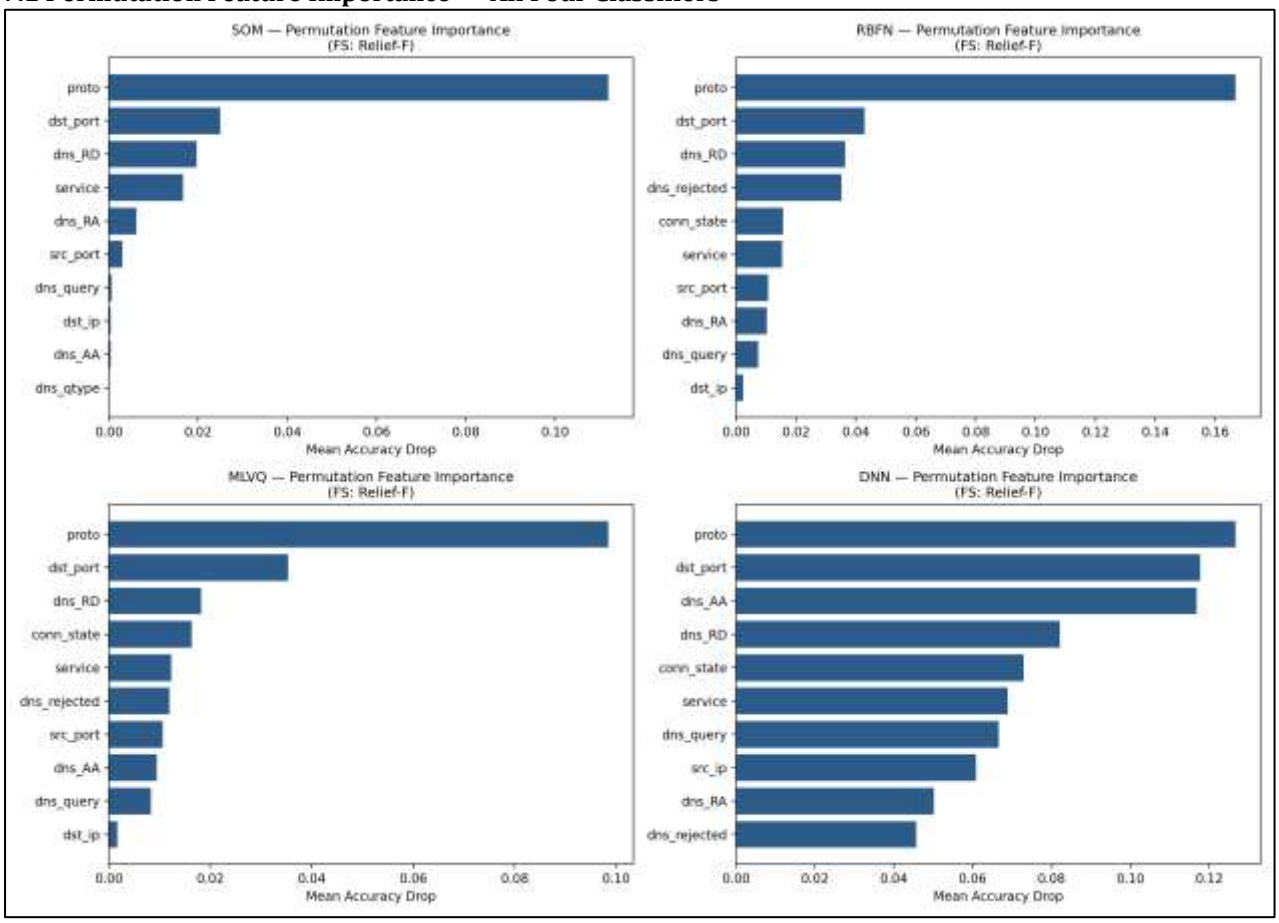

Fig. 11. Top-10 permutation feature importances for SOM, RBFN, MLVQ, and DNN (Feature Selection: Relief-F).

Across all four classifiers, the `proto` attribute (transport-layer protocol) is by far the most influential feature, followed consistently by `dst_port` and a cluster of DNS-response flags (`dns_RD`, `dns_AA`, `dns_rejected`) together with `conn_state`, suggesting that protocol/port identity and connection/DNS-response behaviour are the dominant discriminators between normal and attack flows in the ToN-IoT dataset. DNN shows the largest overall accuracy drops under permutation, consistent with it placing more concentrated decision weight on a small number of highly informative features rather than distributing importance thinly, which is typical of high-capacity neural classifiers [28].

7.2 Top-5 Most Influential Features per Classifier

Table 9. Top-5 Permutation-Important Features per Classifier

Classifier	Baseline Accuracy	Top-5 Features (mean accuracy drop)
SOM	91.49%	proto (0.1120); dst_port (0.0250); dns_RD (0.0197); service (0.0166); dns_RA (0.0062)
RBFN	94.54%	proto (0.1668); dst_port (0.0430); dns_RD (0.0365); dns_rejected (0.0352); conn_state (0.0158)
MLVQ	92.45%	proto (0.0984); dst_port (0.0354); dns_RD (0.0182); conn_state (0.0164); service (0.0123)
DNN	99.77%	proto (0.1267); dst_port (0.1177); dns_AA (0.1168); dns_RD (0.0821); conn_state (0.0729)

7.3 SHAP Analysis — DNN

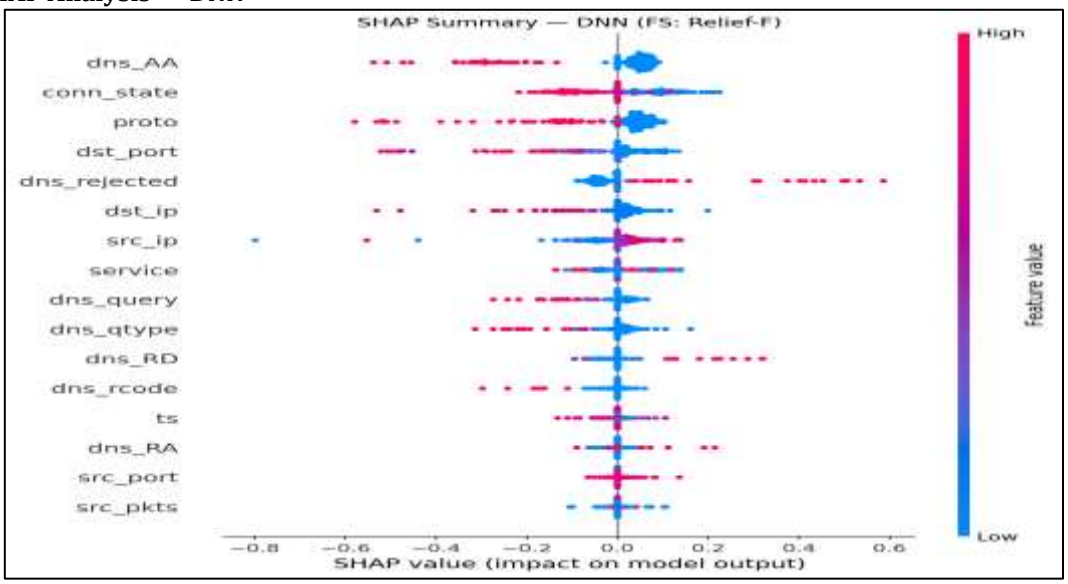

Fig. 12. SHAP summary (beeswarm) plot for DNN, showing per-sample feature contribution direction and magnitude (Feature Selection: Relief-F).

The SHAP beeswarm plot confirms the permutation-importance ranking while adding directionality: distinct values of `proto` and `dst_port` separate cleanly into regions that push predictions toward the majority Attack class versus toward Normal, indicating the model has learned a clear, structured decision rule around these attributes rather than a purely arbitrary split. The DNS-response flags (`dns_AA`, `dns_RD`) similarly show consistent separation between high and low SHAP contributions, reinforcing that anomalous DNS-response behaviour is a genuine attack signature in this dataset rather than a spurious correlation picked up during training [29].

7.4 Permutation Feature Importance — All Four Classifiers

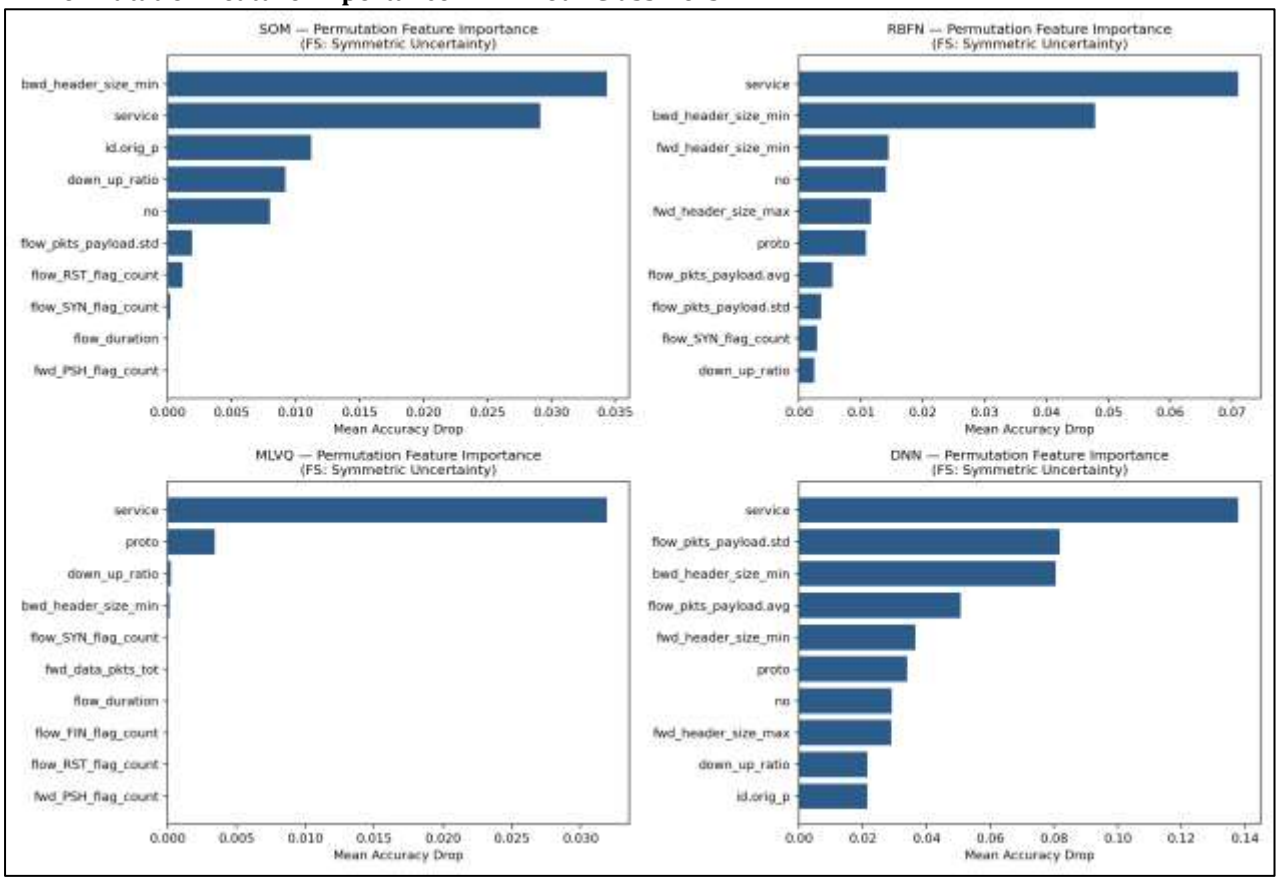

Figure. 13. Top-10 permutation feature importances for SOM, RBFN, MLVQ, and DNN (Feature Selection: Symmetric Uncertainty).

Across all four classifiers, the `service` attribute (application-layer protocol/service label) and header-size related features (`bwd_header_size_min`, `fwd_header_size_min/max`) emerge as consistently influential, suggesting that protocol identity and packet-header structure are the dominant discriminators between normal and attack flows in the ART-IoT2022 dataset. DNN shows the largest overall accuracy drops under permutation, consistent with it placing more concentrated decision weight on a small number of highly informative features rather than distributing importance thinly, which is typical of high-capacity neural classifiers [28].

7.5. Top-5 Most Influential Features per Classifier

Table 10. Top-5 Permutation-Important Features per Classifier

Classifier	Baseline Accuracy	Top-5 Features (mean accuracy drop)
SOM	95.67%	bwd_header_size_min (0.0344); service (0.0291); id.orig_p (0.0112); down_up_ratio (0.0092); no (0.0080)
RBFN	96.61%	service (0.0711); bwd_header_size_min (0.0479); fwd_header_size_min (0.0145); no (0.0141); fwd_header_size_max (0.0116)
MLVQ	93.84%	service (0.0319); proto (0.0034); down_up_ratio (0.0002); bwd_header_size_min (0.0001); flow_SYN_flag_count (0.0000)
DNN	99.47%	service (0.1379); flow_pkts_payload.std (0.0819); bwd_header_size_min (0.0806); flow_pkts_payload.avg (0.0507); fwd_header_size_min (0.0364)

7.6 SHAP Analysis — DNN

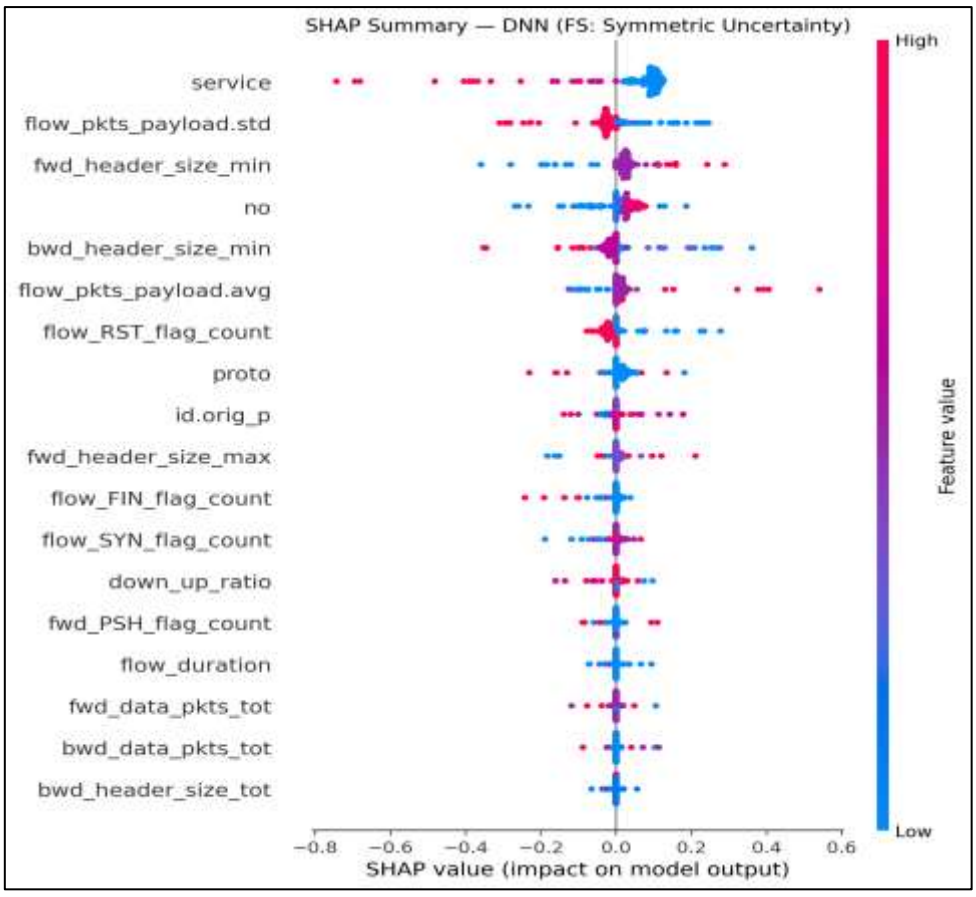

Figure. 14. SHAP summary (beeswarm) plot for DNN, showing per-sample feature contribution direction and magnitude (Feature Selection: Symmetric Uncertainty).

The SHAP beeswarm plot confirms the permutation-importance ranking while adding directionality: high values of `service` (red points) push predictions toward the majority Attack class, whereas low values (blue) push toward Normal, indicating the model has learned a clear, monotonic-like decision rule around this attribute rather than a purely arbitrary split. Payload-statistics features (`flow_pkts_payload.std/avg`) similarly show consistent separation between high and low SHAP contributions, reinforcing that traffic-volume irregularity is a genuine attack signature rather than a spurious correlation picked up during training [29].

8. Conclusions

In case of ToN-IoT dataset all five feature selection methods, DNN comes out clearly on top every single time, with accuracy always above 99%. The best combination overall is Symmetric Uncertainty + DNN at 99.78% accuracy, closely followed by Relief F + DNN (99.77%) and Chi Square + DNN (99.71%). The other three classifiers — SOM, RBFN, MLVQ — stay in the 90–94% range no matter which feature selection method is paired with them, so there's a big gap between DNN and the rest.

The weakest combination in ToN-IoT is Symmetric Uncertainty + MLVQ at 90.94% accuracy, very close to Chi Square + SOM at 90.98%. So MLVQ and SOM are consistently the weaker classifiers here, and Symmetric Uncertainty as a feature selector doesn't help them much.

In case RT-IoT2022 dataset Same pattern holds — DNN again gives the best results across the board, all above 99%. The single best combination is Gain Ratio + DNN at 99.52% accuracy, with Chi Square + DNN and Relief F + DNN both close behind at 99.26%. RBFN also performs quite well here (96–97% range), better than it did on ToN-IoT.

The lowest accuracy in this dataset is Info Gain + MLVQ at 94.16%, with Gain Ratio + MLVQ close behind at 94.53%..Based on the accuracy comparison across both datasets, DNN emerges as the most reliable and consistently high-performing classifier, achieving over 99% accuracy regardless of which feature selection method is applied, in both ToN-IoT and RT-IoT2022. Among the feature selection methods, Symmetric Uncertainty and Relief F paired with DNN give the strongest results for ToN-IoT, while Gain Ratio paired with DNN performs best for RT-IoT2022. On the other end, MLVQ and SOM consistently produce the weakest results across both datasets, indicating these classifiers are less suited to this type of intrusion detection data regardless of the feature selection technique used. Overall, the combination of DNN with information-theoretic feature selection methods (particularly Symmetric Uncertainty, Relief F, and Gain Ratio) offers the most accurate and dependable intrusion detection performance across both datasets.

References

[1] Li, Jing, Mohd Shahizan Othman, Hewan Chen, and Lizawati Mi Yusuf. "Optimizing IoT Intrusion Detection System: Feature Selection versus Feature Extraction in Machine Learning." *Journal of Big Data* 11 (2024): 36. https://doi.org/10.1186/s40537-024-00892-y.

[2] Tareq, Islam, B. M. Elbagoury, S. El-Regaily, and E. S. M. El-Horbaty. "Analysis of ToN-IoT, UNSW-NB15, and Edge-IIoT Datasets Using Deep Learning in Cybersecurity for IoT." *Applied Sciences* 12, no. 19 (2022): 9572. https://doi.org/10.3390/app12199572.

[3] Guo, G., X. Pan, H. Liu, F. Li, L. Pei, and K. Hu. "An IoT Intrusion Detection System Based on TON-IoT Network Dataset." In *Proceedings of the 2023 IEEE 13th Annual Computing and Communication Workshop and Conference (CCWC)*, 333–338. Piscataway, NJ: IEEE, 2023. https://doi.org/10.1109/CCWC57344.2023.10099144.

[4] Gad, A. R., M. Haggag, A. A. Nashat, and T. M. Barakat. "A Distributed Intrusion Detection System Using Machine Learning for IoT Based on ToN-IoT Dataset." *International Journal of Advanced Computer Science and Applications* 13, no. 6 (2022). https://doi.org/10.14569/IJACSA.2022.0130667.

[5] Guo, G. "An Intrusion Detection System for the Internet of Things Using Machine Learning Models." In *Proceedings of the 2022 3rd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)*, 332–335. Piscataway, NJ: IEEE, 2022. https://doi.org/10.1109/ICBAIE56435.2022.9985800.

[6] Awotunde, J. B., S. O. Folorunso, A. L. Imoize, J. O. Odunuga, C. C. Lee, C. T. Li, and D. T. Do. "An Ensemble Tree-Based Model for Intrusion Detection in Industrial Internet of Things Networks." *Applied Sciences* 13, no. 4 (2023): 2479. https://doi.org/10.3390/app13042479.

[7] Injadat, M. "Optimized Ensemble Model towards Secured Industrial IoT Devices." In *Proceedings of the 2023 24th International Arab Conference on Information Technology (ACIT)*, 1–5. Piscataway, NJ: IEEE, 2023. https://doi.org/10.1109/ACIT58888.2023.10453914.

[8] Mohy-Eddine, M., A. Guezzaz, S. Benkirane, M. Azrour, and Y. Farhaoui. "An Ensemble Learning Based Intrusion Detection Model for Industrial IoT Security." *Big Data Mining and Analytics* 6, no. 3 (2023): 273–287. https://doi.org/10.26599/BDMA.2022.9020032.

[9] Rabie, O. B. J., S. Selvarajan, T. Hasanin, A. M. Alshareef, C. K. Yogesh, and M. Uddin. "A Novel IoT Intrusion Detection Framework Using Decisive Red Fox Optimization and Descriptive Back Propagated Radial Basis Function Models." *Scientific Reports* 14 (2024): 386. https://doi.org/10.1038/s41598-024-51154-z.

[10] Alhanaya, Moody, and Khalil Hamdi Ateyeh Al-Shqeerat. "Performance Analysis of Intrusion Detection System in the IoT Environment Using Feature Selection Technique." *Intelligent Automation & Soft Computing* 36, no. 3 (2023): 3709–3724. https://doi.org/10.32604/iasc.2023.036856.

[11] Maseno, Elijah M., and Zenghui Wang. "Intrusion Detection in IoT Using Ensemble Approach." In *Proceedings of the 5th International Symposium on Advanced Technologies and Applications in the Internet of Things (ATAIT 2023)*. Kusatsu, Japan, August 28–29, 2023. *CEUR Workshop Proceedings*. https://ceur-ws.org/Vol-3459/invited2.pdf.

[12] Sekar, S. K., P. R. Parvathy, L. Pinjarkar, R. Latha, M. Sathish, M. K. Reddy, and S. Murugan. "Random Forest Algorithm with Hill Climbing Algorithm to Improve Intrusion Detection at Endpoint and Network." *Indonesian Journal of Electrical Engineering and Computer Science* 37, no. 1 (2025): 134–142. https://doi.org/10.11591/ijeecs.v37.i1.pp134-142.

[13] Alshathri, S., A. El-Sayed, W. El Shafai, and E. E. D. Hemdan. "An Efficient Intrusion Detection Framework for Industrial Internet of Things Security." *Computer Systems Science and Engineering* 46, no. 1 (2023): 819–834. https://doi.org/10.32604/csse.2023.034095.

[14] UNSW Canberra Cyber. "ToN-IoT Datasets." Accessed July 31, 2026. https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-ton-iot-Datasets/.

[15] Han, Jiawei, and Micheline Kamber. *Data Mining: Concepts and Techniques*. 2nd ed. San Francisco: Morgan Kaufmann, 2006.

[16] Kohonen, Teuvo. "The Self-Organizing Map." *Proceedings of the IEEE* 78, no. 9 (1990): 1464–1480. https://doi.org/10.1109/5.58325.

[17] Chakravarthy, S. V., and Joydeep Ghosh. "Scale-Based Clustering Using the Radial Basis Function Network." *IEEE Transactions on Neural Networks* 7, no. 5 (1996): 1250–1261. https://doi.org/10.1109/72.536318.

[18] Nova, D., and P. A. Estévez. "A Review of Learning Vector Quantization Classifiers." *Neural Computing and Applications* 25, no. 3 (2014): 511–524. https://doi.org/10.1007/s00521-013-1535-3.

[19] Liu, Wei, Zidong Wang, Xiaohui Liu, Nianyin Zeng, Yuriy Liu, and Fuad E. Alsaadi. "A Survey of Deep Neural Network Architectures and Their Applications." *Neurocomputing* 234 (2017): 11–26. https://doi.org/10.1016/j.neucom.2016.12.038.

[20] Dharini, N., V. S. Janani, and Jeevaa Katiravan. "Efficient Detection of Intrusions in TON-IoT Dataset Using Hybrid Feature Selection Approach." *Scientific Reports* 16 (2026): 7763. https://doi.org/10.1038/s41598-026-37834-y.

[21] Logeswari, G., J. D. Roselind, K. Tamilarasi, and V. Nivethitha. "A Comprehensive Approach to Intrusion Detection in IoT Environments Using Hybrid Feature Selection and Multi-Stage Classification Techniques." *IEEE Access* 13 (2025): 24970–24987. https://doi.org/10.1109/ACCESS.2025.3532895.

[22] Nazir, A., Z. Memon, T. Sadiq, H. Rahman, and I. U. Khan. "A Novel Feature-Selection Algorithm in IoT Networks for Intrusion Detection." *Sensors* 23, no. 19 (2023): 8153. https://doi.org/10.3390/s23198153.

[23] Dey, A. K., G. P. Gupta, and S. P. Sahu. "Hybrid Meta-Heuristic Based Feature Selection Mechanism for Cyber-Attack Detection in IoT-Enabled Networks." *Procedia Computer Science* 218 (2023): 318–327. https://doi.org/10.1016/j.procs.2023.01.014.

[24] Pati, Abhilash, Parhi, Manoranjan, Pattanayak, Binod Kumar, Singh, Debabrata, Samanta, Debebrata, Banerjee, Amit, Biring, Sajal and Dalapati, Goutam Kumar, Diagnose, Diabetic Mellitus Illness Based on IoT Smart Architecture, Wireless Communications and Mobile Computing, Vol. 2022, No. 1, pp.7268571, 2022. https://doi.org/10.1155/2022/7268571.

[25] Kasongo, Sydney Mambwe. "A Deep Learning Technique for Intrusion Detection System Using a Recurrent Neural Networks Based Framework." *Computer Communications* 199 (2023): 113–125. https://doi.org/10.1016/j.comcom.2022.12.010.

[26] Desai, Rohan, and Venkatesh Tiruchirai Gopalakrishnan. "Network Intrusion Detection through Machine Learning with Efficient Feature Selection." In *2023 15th International Conference on COMmunication Systems & NETworkS (COMSNETS)*, 797–801. Piscataway, NJ: IEEE, 2023. https://doi.org/10.1109/COMSNETS56262.2023.10041315.

[27] Pan, Zhixin, and Prabhat Mishra. *Explainable AI for Cybersecurity*. Cham: Springer, 2023. https://doi.org/10.1007/978-3-031-46479-9.

[28] Mahapatra, Ajit Kumar, Panda, Nibedan and Pattanayak, Binod Kumar, Hybrid PSO (SGPSO) with the Incorporation of Discretization Operator for Training RBF Neural Network and Optimal Feature

Selection, Arabian Journal of Science and Engineering, Vol.48, No.8, pp. 9991-10019, 2023. https://doi.org10.1007/s13369-022-07408-x.

[29] Lundberg, Scott M., and Su-In Lee. "A Unified Approach to Interpreting Model Predictions." In *Advances in Neural Information Processing Systems 30*, 4765–4774. Red Hook, NY: Curran Associates, 2017. https://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.