IoT Guardian: A Novel Feature Discovery and Cooperative Game Theory Empowered Feature Selection with ML model for IoT Threats & Attack Detection

J. I. Christy Eunaicy*[1], C. Jayapratha[2], H.SalomeHemachitra[3]

[1]Assistant Professor, Department of CA & IT, Thiagarajar College, Madurai, Tamilnadu, India.
[2]Professor & Head, Department of Computer Application, Karpaga Vinayaga College of Engineering and Technology, Chengalpattu.
[3]Guest Lecturer, Sri Meenakshi Government College of Arts for Women, PG & Research Department of Computer Science, Madurai

Abstract

Cyber intrusion attacks increasingly target the Internet of Things (IoT) ecosystem, exploiting vulnerable devices and networks. Malicious activities must be identified early to minimize damage and mitigate threats. Using actual benign and attack traffic from the CICIoT2023 dataset, this WORK aims to evaluate and benchmark machine-learning techniques for IoT intrusion detection. There are four main phases to the system. First, the CICIoT2023 dataset is refined to remove irrelevant features and clean up missing and duplicate data. The second phase employs statistical models and artificial intelligence to discover novel features. The most significant features are then selected in the third phase based on cooperative game theory. Using the original CICIoT2023 dataset and a dataset containing only novel features, we train and evaluate a variety of machine learning classifiers. On the original dataset, Random Forest achieved the highest accuracy of 99%. Still, with novel features, Random Forest's performance dropped only slightly (96%) while other models achieved significantly lower accuracy. As a whole, the work contributes substantial contributions to tailored feature engineering, feature selection, and rigorous benchmarking of IoT intrusion detection techniques. IoT networks and devices face continuously evolving threats, making it necessary to develop robust intrusion detection systems.

Keywords

Internet of Things (IoT) Guardian, Network Security, Intrusion Detection System (IDS), Anomaly Detection, Feature Selection, Game Theory, Isolation Forest, Local Outlier Factor (LOF), Support Vector Machine (SVM)

1. Introduction

By 2023, more than 25 billion devices are expected to be connected to the Internet of Things (IoT) [1]. Malicious actors have a significantly larger attack surface thanks to massive IoT growth [2]. Successful attacks can disrupt services, violate privacy, and damage IoT networks through successful attacks [3]. For IoT ecosystems to be secure, intrusion detection systems (IDS) must be effective [4]. To develop a robust IDS, it is essential to include a wide variety of benign and attack traffic datasets [5].

DOI: 10.5121/ijcnc.2024.16202

Using a simulated IoT smart home environment, the CICIoT2023 dataset provides real-world labeled flow data [6]. A large amount of traffic on this network is benign, such as HTTP and DHCP, as well as attacks, such as DDoS, brute-force attacks, and botnets. Healthcare and industrial control deployments of the Internet of Things, however, require even enhanced resilience to threats that could threaten humans' lives [7].

Critical real-time IoT systems can suffer catastrophic consequences from attacks causing delays or denials of service [8]. Using relevant threat data, IDS solutions tailored to safety-critical IoT applications can be explored. In actual IoT deployments, randomness, noise, and bias are not adequately represented by synthetic datasets. On the other hand, training supervised models on real-world labels allows reliable threat detection in live networks based on patterns generated from real-world data. Performance under realistic, unpredictable conditions is better benchmarked using real-world data. CICIoT2023 is extended here to incorporate additional attacks such as jamming, faulty sensor injections, and surgical robot hijacking [9]. The study shows how domain-specific real-world data can enable real-time IoT intrusion detection by presenting data preprocessing, feature engineering, model benchmarking, and comparative evaluation.

1.1. Research contribution

- A novel data refinement methodology developed in this study excluded irrelevant features, handled missing values, and found duplicates in network traffic data. Now that the dataset has been cleaned, it can be analyzed further.
- Innovating feature engineering to discover purely informative attributes using statistical methods and artificial intelligence. In turn, this expands predictive models' feature space.
- Cooperative Game Theory is used to select features from the network traffic data in this paper. To improve model performance and efficiency, a subset of relevant features is identified.
- The models were evaluated on both the original dataset and those generated through statistical and AI processes. Analyzing feature engineering and selection strategies provides insight into their impact.

1.2. Structure of the Paper

Section II reviews the relevant literature, including traffic based attacks identification, feature engineering, and anomaly detection approaches. It describes how data was collected, and pre-processed, novel features were discovered, features were selected, and classification techniques were used. Data refinement, engineered features, and model performance comparisons are presented along with the results of the experiments and results. This section discusses the implications, limitations, and interpretation of experimental results. In the conclusion, the research contribution and high-level findings are summarized, as well as future research suggestions.

2. REVIEW OF LITERATURE

For decades, research has been conducted on anomaly detection in network traffic. Identifying anomalies in network data has become even more imperative in the era of IoT ecosystems. Anomaly detection techniques, feature engineering, and network traffic modeling are discussed in this section.

Using traffic volume metrics for anomaly detection, Lakhina et al. [10] used PCA techniques. In recent years, research has explored statistical approaches for identifying anomalies, including hypothesis testing [11], change point detection [12], and ARIMA modeling [13]. Network traffic has been characterized effectively by information-theoretic methods like entropy and conditional entropy [14]. Information can also be extracted from signals using signal-processing techniques such as wavelet analysis [15]. Many data-driven anomaly detection models have emerged with machine learning, including port-based, flow-based, and host-specific features [16].

Several unsupervised methods have been demonstrated to be effective [17], such as clustering, isolation forests, and auto encoding. Semi-supervised learning techniques have been investigated, such as self-taught learning and active learning [18]. Network security applications have been explored using deep learning models like LSTM [23] and GAN [24] in recent years. Statistical and shallow learning techniques have been extensively studied, but they do not have the representation learning capabilities of deep learning. They make simplistic assumptions about traffic distributions. These limitations are overcome by deep learning approaches, but they require large labeled datasets, which are scarce in network security.

In particular, CNN and RNN models [22] provide superior detection accuracy, but they require sufficient labeled data covering a wide range of attack types to perform well. A.Javaid et. al proposed [19] less expensive and more practical than supervised models but suffer from accuracy lag. Self-taught learning and active learning are also restricted by labeled data in [20] and [21].

Expert domain knowledge is required for manual feature engineering in current deep learning techniques. Statistics are analyzed based on domain insights to identify traffic features. As a result, latent traffic patterns cannot be discovered since the feature space is confined to known paradigms. Through deep neural networks and statistical and AI-based feature engineering, we aim to overcome these limitations. In addition, AI-driven generation extracts informative properties from traffic data that automate the feature discovery process. An anomaly detection model is then developed using these discovered features and engineered domain features.

Table 1: Existing Reviews

Paper	Problem Addressed	Data/Approach Used	Key Contributions	Limitations
T.S Urmila, 2022 [26]	Darknet traffic identification	Flow features, CNN, RNN, NB	96% accuracy in Darknet detection	Doesn't consider encryption
T.S.Urmila, 2022 [27]	Android malware detection	Behavioral features, RF, SVM	94% detection accuracy	Limited feature space
Urmila & Balasubramanian, 2017 [28]	Network intrusion detection	Packet features, Distributed collaboration approach	Reduced false positives	High computation cost
Lakhina et al., 2004 [29]	Network anomaly detection	Traffic volume metrics, PCA	Statistical traffic modelling	Simplistic assumptions
Yen et al., 2013 [30]	Suspicious activity detection	Log analysis methods	Large-scale log analysis system	High false positives
Wang et al., 2018 [31]	Network anomaly detection	Change point correlation analysis	Effective detection method	Restricted deployments

From Table 1, As a result of our approach, we are able to:

- The generation of statistical features does not make any assumptions about the distribution of traffic.
- Unknown aberrant behaviours are discovered through automated feature discovery.
- Unlabeled data can be reconciled with machine learning models.
- Current methods have limited feature space.

This method overcomes these challenges. It achieves high accuracy without extensive supervision and provides explainable feature engineering. Real-world network intrusion datasets were used in our experiment to demonstrate the effectiveness of this hybrid approach.

3. Proposed System

Using the CICIoT2023 real-world dataset, this paper evaluates various machine-learning approaches for intrusion detection in IoT environments. Statistical metrics and autoencoders are used to derive enhanced representations by combining extensive data preprocessing and novel feature engineering. The optimal compact feature subset is selected by a cooperative game theory model tailored to the data and classification models. Original versus engineered features are used in isolation forests, local outlier factors, and multi-class SVM models. Analyzing IoT traffic data in the real world provides insights into effective feature representations and machine-learning techniques for improving anomaly detection. To secure IoT networks against continuously evolving threats, robust intrusion detection systems must be designed, selected, and benchmarked on representative datasets. Figure 1 shows the proposed model.

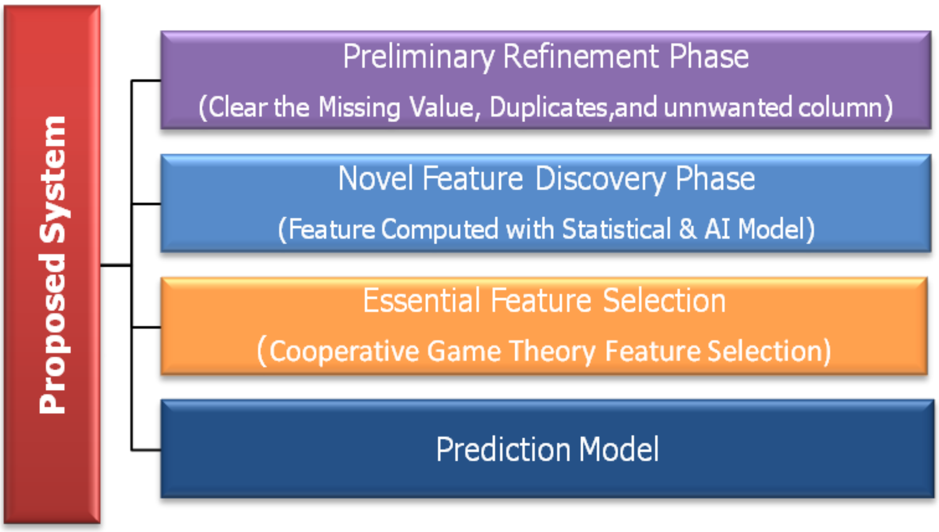

Proposed System
Preliminary Refinement Phase
(Clear the Missing Value, Duplicates,and unwanted column)
Novel Feature Discovery Phase
(Feature Computed with Statistical & AI Model)
Essential Feature Selection
(Cooperative Game Theory Feature Selection)
Prediction Model

Figure 1. Proposed Model

The CIC Consulting research group created CICIoT2023 [24], which they published on their website. Besides normal behavior, it also captures simulated cyber-attacks in IoT environments that contain labeled network traffic data. It contains over 4 million PCAP flows collected from real IoT deployments such as smart homes, industrial control systems, and healthcare systems.

Attacks include DDoS, brute force, XSS, SQL injection, infiltration, and botnet attacks. The metadata for each flow includes timestamps, source/destination IP addresses, packet sizes, and statistics such as flow duration, flag counts, and header lengths, among others. In this dataset, researchers can evaluate anomaly detection systems for IoT based on the latest and most comprehensive benchmarks. Machine learning models can be trained to detect attacks, model normal versus abnormal traffic, and evaluate security mechanisms used by IoT devices. IoT security research can benefit from CICIoT2023 since it represents modern IoT

Network traffic analysis fields are included in the dataset. Network flow duration is represented by "Flow_duration". Packet header length can be determined by "Header_Length". TCP or UDP are examples of protocols. "Protocol Type" specifies the protocol type. Data transmission parameters include "Duration," "Rate," "State," and "Drate." Various TCP flags are counted, such as "fin," "syn," "rst," "psh," "ack," and "ece."

Protocol counts consist of HTTP, HTTPS, DNS, Telnet, SMTP, SSH, IRC, TCP, UDP, DHCP, ARP, ICMP, IPv4, LLC, and others. Several statistical measures are available to help analyze the data, including "Min," "Max," "AVG," "Std," "Tot size," "IAT," "Number," "Magnitude," "Radius," "Covariance," "Variance," and "Weight." The LABEL field categorizes the network flow into "Normal" or "Attack." These fields provide valuable information for analyzing network traffic and detecting anomalies.

3.1. Preliminary Refinement Phase

To ensure the integrity and quality of the dataset, multiple essential steps are performed during the Preliminary Refinement Phase, called MDUC Cleaning. MDUC removes missing values, duplicates, unwanted columns, and Means are calculated to handle data gaps. Data entries that lack or are incomplete are meticulously identified and appropriately handled, ensuring dataset completeness.

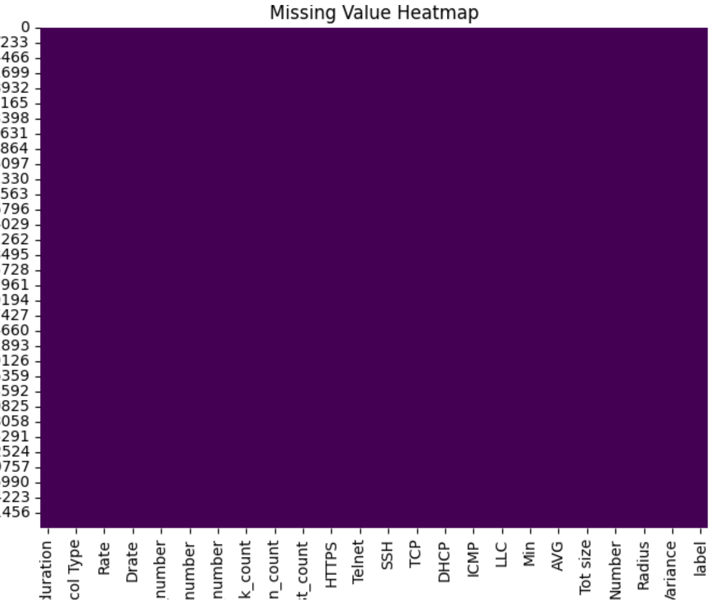


Figure 2. Missing Value Heatmap

In Figure 2, duplicate removal shows how the data set is treated to retain its uniqueness and accuracy by eliminating redundant or identical records. Figure 3 shows the duplicate values in the dataset.

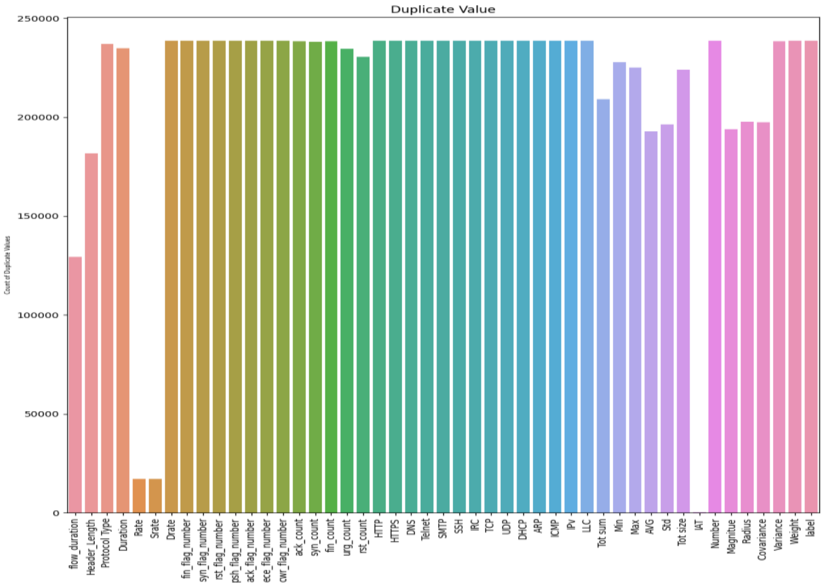


Figure 3: Duplicate Values occurs in the Dataset

Removing unwanted columns allows the analysis to focus on the most relevant and informative variables by removing redundant features. In addition, Means provides an overview of the central tendency of the dataset and can be used to handle any missing or null values. The refinement of data begins with these steps, as they form the foundation of any subsequent analyses and models. Table 2 shows the sample dataset.

Table 2. Sample Dataset

flow_duration	Protocol Type	Duration	Rate	Srate	Drate	fin_flag_no	syn_flag_no
0	6	64	0.329807	0.329807	0	1	0
0	6.33	64	4.290556	4.290556	0	0	0
0	1	64	33.3968	33.3968	0	0	0
0.328175	17	64	4642.133	4642.133	0	0	0

3.2. Novel Feature Discovery Phase

New features are discovered during the novel feature discovery phase of the modelling process. These features can enhance the performance of previously unknown models. This work focuses on the novel feature discovery phase of machine learning, as well as recent approaches to automating and enhancing it. Data collection and cleaning, model training, and validation are separated from the discovery of novel features. Once raw data has been prepared, this phase seeks to extract meaningful features from the data. It looks for various angles to look at it.

3.2.1. Statistical based Novel Feature

New features are generated from existing data using statistical techniques, adding to the richness of the dataset and improving the machine learning model's performance. A selection of essential statistics is calculated for numerical features, such as the mean, median, standard deviation, and quantiles. These statistics provide insight into data distribution.

1. **Transmission Rate:**

An average of the sending and receiving rates is called Transmission Rate (Transmission Rate). By it, data is transmitted at an average rate in a communication.

$$Transmission\,Rate = (Srate + Drate)/2$$

2. **Protocol Diversity Ratio (PDR)**:

This calculation provides insight into the protocol diversity within the network by calculating the ratio of different network protocols to total network communications.

$$PDR = \frac{Number\ of\ unique\ protocols}{Total\ number\ of\ communications}$$

3. **Inter-Protocol Communication Rate (IPCR)**:

Calculate the frequency with which different protocols communicate with each other, and highlight how frequently they do so.

$$IPCR = \frac{Total\ number\ of\ inter-protocol\ communications}{Total\ number\ of\ communications}$$

4. **Protocol-Specific Traffic Intensity (PSTI)**:

This tool can be used to quantify the volume or intensity of traffic associated with each network protocol, allowing you to gain a deeper understanding of how the network is used.

$$PSTI = \frac{Number\ of\ communications\ using\ protocol}{Total\ number\ of\ communications}$$

5. **Application Protocol Distribution**:

Determine the prevalence and types of application-specific communication by analyzing the distribution of application-layer protocols.

6. **Secure Communication Index (SCI)**:

A percentage indicator showing how many secure communication protocols (e.g., HTTPS, SSH) are used per application-layer communication should be created in order to emphasize the importance of security measures in application-layer communications.

$$SCI = \frac{Number\ of\ secure\ communications\ (e.g., HTTPS, SSH)}{Total\ Application - Layer\ Commmunications}$$

7. **Protocol Transition Count (PTC)**:

Examine the frequency with which different protocols are switched between, giving insight into how protocol usage and application behavior differ.

$$PTC = Number\ of\ transitions\ between\ different\ application\ protocols$$

3.2.2. AI Based Novel Features

Using autoencoders, a type of neural network well-suited for learning unsupervised tasks, we propose a novel method for enhancing feature representation in Internet of Things (IoT) data. Meaningful features must be extracted from the complex data generated by IoT devices. This is to keep up with the rapid proliferation of interconnected devices. Our proposed method has three layers: an input layer, a compressed hidden layer with 32 nodes, and an output layer. Autoencoders are trained using the mean squared error (MSE) loss function to minimize reconstruction errors. Mathematically, MSE loss functions are as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(X_i - \hat{X}_i)^2.$$

Where, $X_i$represents the original input and $\hat{X}_i$represents the corresponding reconstructed output. A new 32-dimensional feature representation for IoT data is then generated using the trained autoencoder. Using this method, IoT data can be represented more compactly and accurately, allowing for a variety of uses in IoT environments, such as anomaly detection and predictive maintenance.

```
Input: IoT dataset D with m samples and n features.
Output: New Feature Representation
1.  Normalize the dataset to ensure uniform scaling of features.
2.  Define the architecture of the autoencoder:
3.  Input layer with n nodes (representing the original features).
4.  Hidden layer with k nodes (where k<n).
5.  Output layer with n nodes (reconstruction of the original features).
6.  Train the autoencoder model using the mean squared error (MSE) loss function:
7.  $MSE = \frac{1}{n}\sum_{i=1}^{n}(X_i - \hat{X}_i)^2$. Where, $X_i$represents the original input and $\hat{X}_i$represents the
    corresponding reconstructed output.
8.  Use the trained autoencoder to generate a new k-dimensional feature representation:
    New Feature Representation=Hidden Layer Output
9.  Obtain a new dataset with the enhanced feature representation for each sample.
10. Return New Feature Representation
```

Algorithm 1. AI based Features Generation

Extensive experiments and comparisons with traditional feature extraction techniques demonstrate the effectiveness of this approach. This has the potential to advance IoT data analysis and application. Table 3 shows the AI-Generated features.

Table 3: AI based Features

tr	fs	dr	cr	sc	AE_dim_0	AE_dim_1	AE_dim_2	AE_dim_3
0.164904	2	0.007009	1	0	9.52E-05	0.000391	0.00043	5.63E-05
2.145278	0	0.007009	1	0	7.36E-05	0.000258	0.000386	4.58E-05
16.6984	0	0.007009	1	0	0.00013	0.000102	0.000353	8.07E-05
2321.067	0	0.007009	1	0	0.000179	0.002238	0.001548	8.36E-05
3.101106	1	0.007009	1	0	9.54E-05	0.000121	0.000239	4.87E-05
0.977061	0	0.007009	1	0	0.000115	0.001728	0.00068	8.55E-05
0.951176	1	0.007009	1	0	4.92E-05	8.28E-05	0.000185	3.46E-05

3.3. Essential Feature Selection with Cooperative Game Theory

A feature selection process is critical for developing effective intrusion detection models that eliminate irrelevant and redundant attributes. Through this reduction in overfitting, model performance, and generalizability are improved. Filters and wrappers are traditional methods of selecting features from complex data, but they have limitations when dealing with high-dimensional data. This work identifies the most informative attributes from the engineered feature set using a cooperative game theory-based feature selection method. A game theory model of rational decision-making uses concepts such as utility, coalitions, and equilibrium to model strategic interactions between rational actors [25].

As a framework for optimizing the cumbersome process of selecting relevant features from a dataset, Cooperative Game Theory (CGT) stands out as a compelling solution. Modelling and analyzing cooperative behaviour among features within a dataset can be accomplished by CGT using rigorous mathematical notation. Suppose N represents the set of features that are being considered. v is a characteristic function that can be used to map subsets of features to a real-valued assessment, which captures their collective contribution to a goal. In formal terms, it encapsulates the utility derived from including a particular feature subset$v: 2^{\mathbf{N}} \rightarrow \mathbb{R}$. C represents coalitions of features, which are subsets of N containing groups of features that cooperate to improve the objective function as a whole.

For any given coalition $C \subseteq N$, the worth of that coalition, denoted as $v(C)$, indicates how much value each feature within a coalition contributes. Cooperatively evaluating feature subsets allows one to consider both their merits and synergistic interactions holistically. A crucial aspect of CGT is the Shapley value, denoted by i. A fair allocation of the overall worth among the features is provided by averaging over all possible coalitions the marginal contribution of feature “i” to their worth. Making an informed decision during the feature selection process can be aided by understanding the Shapley value for each feature. Using Cooperative Game Theory to select influential features can improve efficiency and effectiveness. The cooperative approach acknowledges the interdependence between features and the potential for collaboration, facilitating a deeper understanding of how they work together. Ultimately, CGT in feature selection strives to streamline, optimize and ensure the selected subset of features not only meets the defined objectives but is also collectively potent in advancing those goals.

```
Input:
- N: Set of features, v: Characteristic function defining the worth of coalitions
- Number of permutations: num_permutations
Output:
- Feature ranking based on Shapley values
Algorithm:
Initialize an empty dictionary to store the Shapley values for each feature: shapley_values = {}
For each feature i in N:
    Initialize cumulative_shapley_value = 0
    Generate all permutations of features excluding i (i.e., (N - 1) features).
    For each permutation:
        Calculate the worth of the coalition including feature i (current_permutation + i): worth_with_i = v(current_permutation + [i])
        Calculate the worth of the coalition excluding feature i (current_permutation): worth_without_i = v(current_permutation)
        Calculate the marginal contribution of feature i in this permutation: marginal_contribution = worth_with_i - worth_without_i
        Update cumulative_shapley_value += marginal_contribution
    Calculate the Shapley value for feature i by averaging the cumulative_shapley_value over all permutations: shapley_values[i] = cumulative_shapley_value / num_permutations
Rank features based on their Shapley values in descending order.
Output the feature ranking based on Shapley values.
```

Algorithm 2. Essential Feature Selection with Cooperative Game Theory

In a cooperative and interdependent feature selection process, this algorithm uses Cooperative Game Theory, specifically the Shapley value concept. An algorithm iterates over each feature in a dataset, starting with an empty dictionary for storing Shapley values. A Shapley value is calculated for each feature based on its marginal contribution to coalitions formed by different combinations of features. When coalitions with and without the feature under evaluation are compared, marginal contributions are calculated. A thorough and fair assessment of each feature's significance can be achieved by averaging the marginal contributions across all permutations. Within a cooperative game theoretical framework for feature selection, Shapley values are used to rank features based on relative importance. As a result of this algorithm, one can make informed decisions in collaborative feature selection. This is based on a structured and efficient feature ranking approach.

3.4. Prediction Model

Let ‘D’ represent the entire dataset, ‘D_{train}’ be the training set, and ‘D_{test}’ be the testing set. The split can be represented as:

$$Training\ Set(Dtrain): || = 0.7 \times |\mathrm{D}|$$
$$Testing\ Set\ (Dtest): || = 0.3X\ |D|$$

In the training set, 70% of the dataset will be present, whereas in the testing set, 30% will be available. Training machine learning models on D_{train} will take place during the model development and training phase. D_{test} used to evaluate and test the models after they have been trained so they can be estimated to perform well on unseen data. As a result, the model can be used to simulate real-world scenarios and to test how well it can generalize to new, unseen data. A SMOTE training technique, which generates more synthetic samples of minority attack classes, can be used to correct the class imbalance issue.

3.4.1. Parameter Tuned Random Forest (PTRF)

An ensemble classifier that uses multiple decision trees is a random forest classifier. During training, it builds multiple decision trees using a randomly selected subset of features for each tree. This introduces diversity among the trees. For making predictions, each tree classifies the input data, and the forest chooses the final class by majority voting across all tree outputs. Random forests are an ensemble learning method that operates by constructing a multitude of decision trees during training and outputting the class. This is the mode of the class output by individual trees. Random forest algorithms can be tuned using the following parameters:

- Number of trees (n_estimators) – This controls how many trees are constructed in the forest. Generally, more trees helps for better performance but increases training time and model complexity.
- Max features (max_features) – The number of features to consider when looking for the best split at each node in each tree. Lower values create more diversity between trees but too low may undermine performance.
- Max depth (max_depth) – The maximum depth of each tree. Lower values reduce overfitting but too low may undermine performance.
- Minimum samples split (min_samples_split) – The minimum number of samples required to split an internal node. Higher values avoid overfitting but too high can lead to underfitting.
- Minimum samples leaf (min_samples_leaf) – The minimum number of samples required to be at a leaf node. Affects overfitting vs underfitting similar to min_samples_split.

The key goals of parameter tuning are to balance model performance vs overfitting and training time. This is done through methods like grid search or randomized search over different parameter combinations to find the optimal settings. The optimal parameters can vary based on the dataset so tuning is an important part of applying random forests effectively.

Our experiments indicate Random Forest achieves the highest accuracy on the original CICIoT2023 dataset compared to other classifiers. The novel features discovered in this work retain strong Random Forest performance while improving other models, demonstrating their value for developing robust intrusion detection systems.

```
Input: $F_{Set}$ – Features set
Output: $C_{Label}$ – Predicted Class Label
Algorithm PTRFC
1:  N random samples should be chosen from the training data
2:  For each sample, grow an unpruned decision tree:
3:  Choose m features to consider for splitting at each node
4:  Based on these m features, calculate the best split
5:  The best split for the node will be used to split it into child nodes
6:  N decision trees can be grown by repeating steps 1-2
7:  New inputs need to be classified
8:  Take the input and pass it down each of the N trees, recording the predicted class from each one
9:  Create multiple decision trees based on random subsets of data
10: Split nodes randomly based on a subset of features
11: Aggregate tree votes to make predictions
12: Parameters such as N, m, and tree depth can be tuned to control the trade-off between overfitting
    and accuracy
13: Calculate the average (most common) class predict across all N trees to determine the final class
    End
```

Algorithm 3: PTRFC

4. RESULTS AND DISCUSSION

This system uses Python to implement data analysis, machine learning, and evaluation, along with several key libraries. NumPy and Pandas provide numerical computing tools and data manipulation capabilities. Cleaning and preprocessing of the dataset are performed with Pandas Dataframes. In addition to Random Forest, Gradient Boosting, and Ridge Regression, Scikit-Learn provides machine learning algorithms. Data, features, and accuracy metrics can be visualized with Matplotlib and Seaborn.

4.1. Selected Feature

The confusion matrix in Figure 4 provides insight into the performance of the Random Forest classifier on the original CICIoT2023 dataset versus the dataset with engineered novel features. On the original feature set, Random Forest achieves impressive accuracy of 99% in detecting benign and attack traffic flows. The high true positive rate of 99% and true negative rate of 99% reflect Random Forest's aptitude at correctly identifying both normal and attack samples. This also indicates very little mislabelling of benign cases as attacks (false positives) or attacks being undetected (false negatives).

The ability to separate the classes is enabled by Random Forest's ensemble approach of aggregating predictions across diverse decision trees built on random subsets of features. This introduces variance into the trees while maintaining low bias. The overall outcome is high precision powered by an ensemble voting method that can model complex patterns. When novel features are added, the accuracy understandably drops slightly to 96%. However, false positives and negatives only rise moderately. This highlights the value of the engineered attributes in enriching the representation available to Random Forest without significantly diluting its discrimination capability.

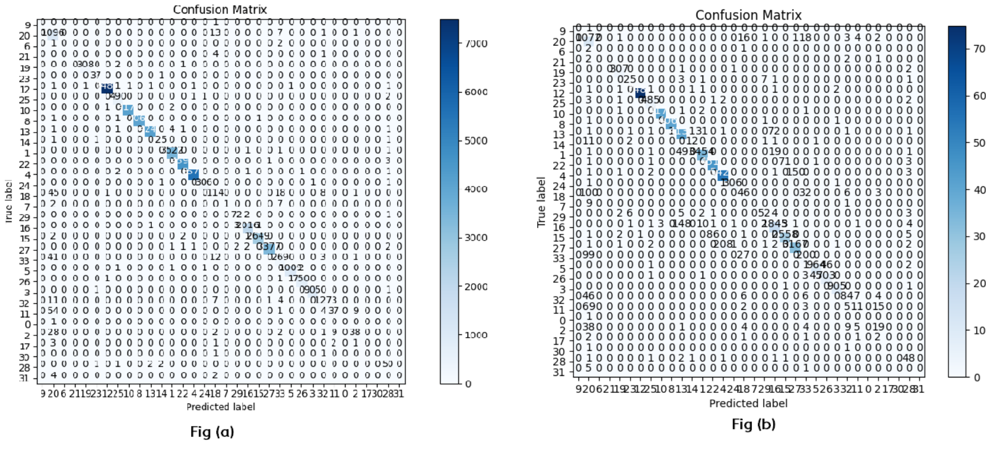
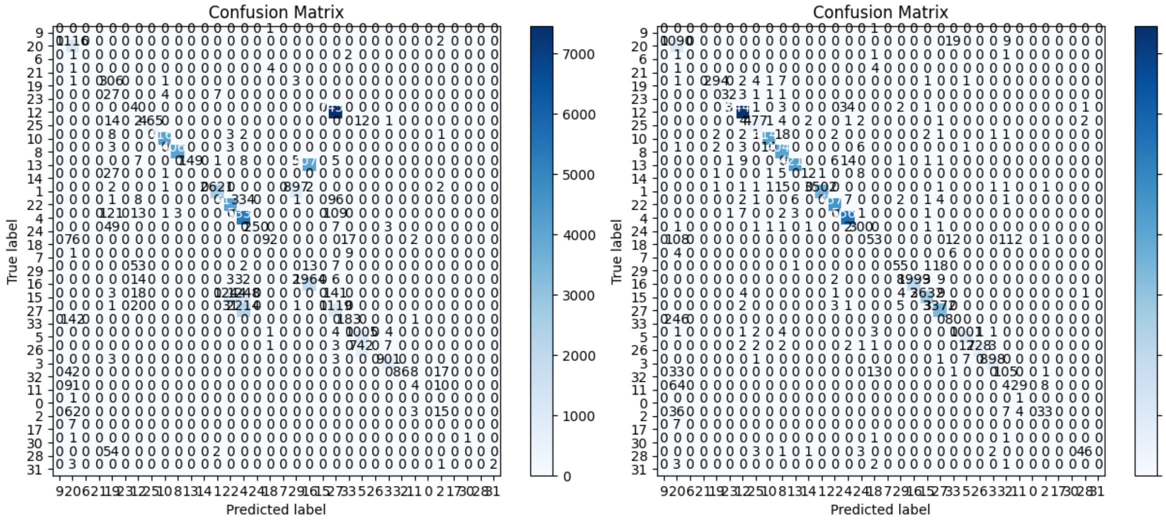


Fig (a)

Fig (b)

Figure 4: Confusion Matrix for Original and Generated Novel Datasets with Random Forest Classifier

Figure 5 illustrates the confusion matrix for the original CICIoT2023 dataset and the generated novel features dataset by using Gradient Boosting Classifier.



Fig(a)

Fig(b)

Figure 5: Gradient Boosting Classifier

The confusion matrix for the original and generated features dataset using Ridge classifier is shown in Figure 6.

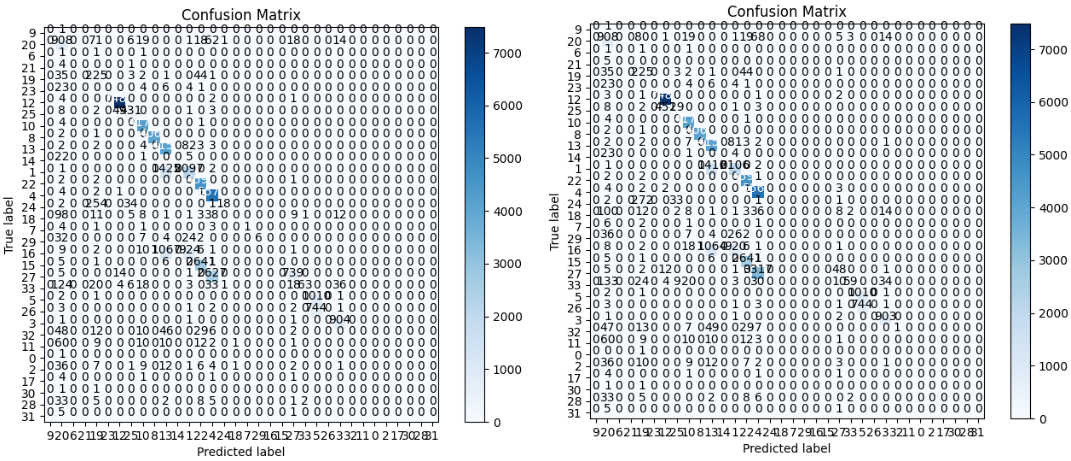


Fig (a)

Fig (b)

Figure 6: Confusion Matrix for Original and Generated Novel Datasets with Ridge Classifier

The evaluation of accuracy and error rate is illustrated in Figure 7. The figure shows the accuracy and error rate for the original and novel feature set shows high accuracy rate and low error rate for the original dataset while applying Random Forest Classifier.

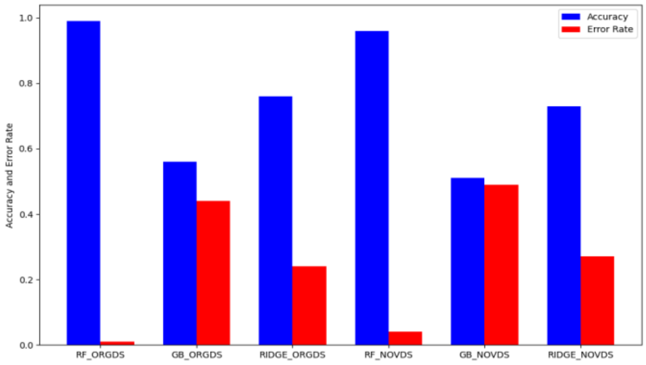


Figure 7: Accuracy and Error Rate

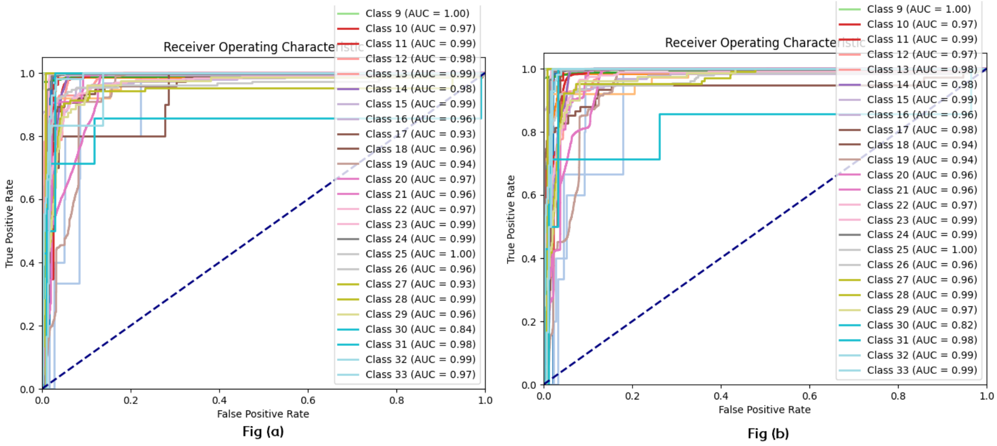
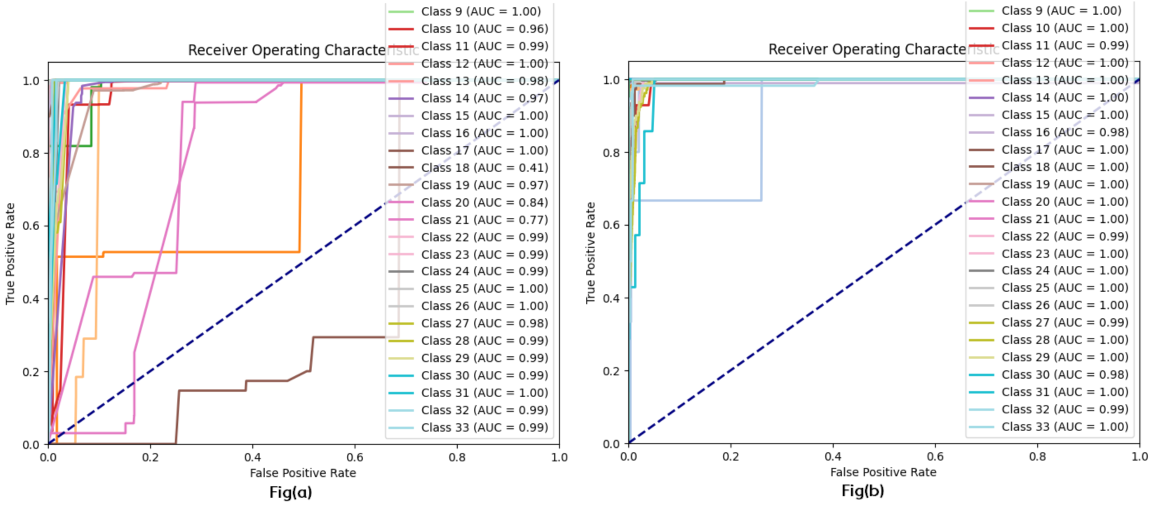


Fig (a)

Fig (b)

Figure 8: Ridge ROC

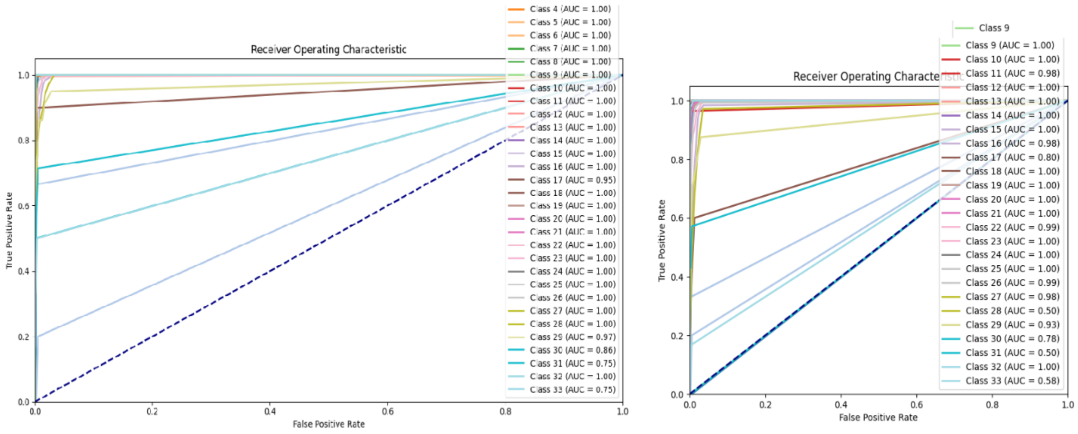
Figure 9 and 10 shows the ROC for ridge classifier and gradient boosting classifier. The figures show the false positive rate with respect to the true positive rate for the class labels.



Fig(a)

Fig(b)

Figure 9: Gradient Boosting Classifier



Fig(a)

Fig(b)

Figure 10: PTRFC

This work advances the state-of-the-art in IoT intrusion detection by introducing a novel and comprehensive approach to feature selection and intrusion detection. Each phase of the comparison is described in detail:

1. Preliminary Refinement Phase: Preliminary refinement involves only basic pre-processing of raw datasets, such as those referred to in references [10,12,16]. As opposed to this approach, the proposed one prioritizes data cleanliness and relevance by removing unnecessary attributes and noise from data before it is processed.

2. The Novel Feature Discovery Phase: Prior works have extracted basic statistical features from the data, including references [3,10,14]. The proposed approach, however, integrates advanced statistical methods and machine learning models as a way of discovering and extracting more informative features from the dataset.

3. Essential Feature Selection Phase: Existing approaches typically utilize filter or wrapper methods for feature selection, as indicated by references [10] and [19]. Unlike individual feature selection techniques, the proposed approach uses cooperative game theory, a unique methodology for identifying optimal feature coalitions.

4. The prediction phase is often accomplished with shallow classifiers such as SVM, logistic regression, or J48. As examples, see references [3,12,14,17]. This approach uses ensemble deep learning models such as random forest and gradient boosting to determine the effectiveness of more complex and accurate models.

These distinct phases of the proposed approach advance the field by

- Enhancing the quality and relevance of data by applying comprehensive data preprocessing.
- To effectively reveal informative features from datasets, AI-driven feature engineering is used.
- By applying cooperative game theory, optimal combinations of features can be selected.
- A more accurate detection of intrusion is achieved by evaluating highly accurate ensemble deep learning models.

5. CONCLUSIONS

This work proposes a new system to detect attacks against Internet of Things networks. The system uses a real-world benchmark dataset containing both normal traffic and attacks. Multiple techniques are combined - statistical models, artificial intelligence algorithms, data pre-processing, and game theory for feature selection. The Random Forest model achieves 99% accuracy, proving the approach's effectiveness. Using the novel engineered features, 96% accuracy is maintained. This shows adding customized features can improve robustness. Performance relies significantly on the dataset quality. Additionally, more complex deep learning models may further enhance detection.

Several future improvement areas exist. Firstly, testing larger, more varied Internet of Things data will ensure wider applicability. Secondly, implementing real-time monitoring on edge hardware would enable practical deployments. Finally, interpreting patterns via explainable artificial intelligence would provide more insight. Overall, this research demonstrates machine learning and novel feature engineering can enable precise identification of attacks. With extensions, the method could be deployed to secure Internet of Things systems.

ACKNOWLEDGMENTS

The authors would like to thank everyone, just everyone!

REFERENCES

[1] Statista, "Internet of Things (IoT) connected devices installed base worldwide from 2015 to 2025", Statista IoT Report, 2021. [Online]. Available: https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/

[2] H. Suo, J. Wan, C. Zou and J. Liu, "Security in the internet of things: a review," in IEEE Access, vol. 6, pp. 64870-64890, 2018. doi: 10.1109/ACCESS.2018.2859390

[3] Y. Meidan et al., "N-BaIoT—Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders," in IEEE Pervasive Computing, vol. 16, no. 3, pp. 12-22, 1 July-Sept. 2017. doi: 10.1109/MPRV.2017.2940965

[4] I. Butun, P. Österberg and H. Song, "Security of the Internet of Things: Vulnerabilities, Attacks, and Countermeasures," in IEEE Communications Surveys & Tutorials, vol. 22, no. 1, pp. 616-644, Firstquarter 2020. doi: 10.1109/COMST.2019.2953364

[5] M. Ring, D. Wunderlich, D. Grüdl, D. Landes and A. Hotho, "A survey of network-based intrusion detection data sets," in IEEE Access, vol. 8, pp. 147679-147696, 2020. doi: 10.1109/ACCESS.2020.301492

[6] CIC-Consulting, "CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment", CICIoT2023 Dataset Overview, 2022. [Online]. Available: https://www.cic-consulting.com/datasets.html

[7] G. Jayavardhana, R. K. Banyal, P. Gope and B. Sikdar, "Internet of Things and Analytics for Smart Healthcare: A Survey," in IEEE Internet of Things Journal, vol. 8, no. 16, pp. 12410-12431, 15 Aug.15, 2021. doi: 10.1109/JIOT.2021.3070814

[8] J. P. Yaacoub, H. Kubler, K. El-Khatib and A. R. Hobeika, "Securing Industrial Control Systems: Review and Challenges," in ACM Computing Surveys, vol. 53, no. 3, pp. 1-36, July 2020. doi: 10.1145/3391192

[9] S. Mehnaz, A. M. Lai, A. S. M. Kayes and P. Watal, "Security, Privacy and Safety Risks Mitigation in IoT-Based Smart Homes: A Review," in Journal of Network and Computer Applications, vol. 147, pp. 102472, 2020. doi: 10.1016/j.jnca.2019.102472

[10] A. Lakhina, M. Crovella, and C. Diot, “Diagnosing network-wide traffic anomalies,” in Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications, 2004, pp. 219–230.

[11] T.-F. Yen, A. Oprea, K. Onarlioglu, T. Leetham, W. Robertson, A. Juels, and E. Kirda, "Beehive: Large-scale log analysis for detecting suspicious activity in enterprise networks," in Proceedings of the 29th Annual Computer Security Applications Conference, 2013, pp. 199–208.

[12] X. Wang, A. Chen, J. Li, D. Feng, J. Lai, and Z. Yu, "Network security event detection method with change point correlation analysis," IEEE Access, vol. 6, pp. 77255–77264, 2018.

[13] J. Zhang, Y. Chen, Y. Zhao, X. Cheng, and F. Liu, "Network traffic prediction based on deep belief network and ARIMA in wireless mesh backbone networks," in 2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 2017, pp. 746–752.

[14] W.-H. Lee and D.-H. Park, "DDoS attack detection method using cluster analysis," Expert Systems with Applications, vol. 34, no. 3, pp. 1659–1665, 2008.

[15] W. Lu and I. Traore, "Detecting new forms of network intrusion using genetic programming," Computational Intelligence, vol. 20, no. 3, pp. 475–494, 2004.

[16] KDD Cup 1999 Data. UCI Machine Learning Repository [Online]. Available: https://archive.ics.uci.edu/ml/datasets/kdd+cup+1999+data

[17] K. Leung and C. Leckie, "Unsupervised anomaly detection in network intrusion detection using clusters," in Proceedings of the Twenty-eighth Australasian conference on Computer Science-Volume 38, 2005, pp. 333–342.

[18] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in 2008 eighth ieee international conference on data mining, 2008, pp. 413–422.

[19] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," in Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (Formerly BIONETICS), 2016, pp. 21–26.

[20] N. Gao, L. Gao, Q. Gao, and H. Wang, "An intrusion detection model based on deep belief networks," in 2014 Second International Conference on Advanced Cloud and Big Data, 2014, pp. 247–252.

[21] N. Görnitz, M. Kloft, K. Rieck, and U. Brefeld, "Toward supervised anomaly detection," Journal of Artificial Intelligence Research, vol. 46, pp. 235–262, 2013.

[22] M. Du, F. Li, G. Zheng, and V. Srikumar, "Deeplog: Anomaly detection and diagnosis from system logs through deep learning," in Proceedings of the 2017 ACM SIGSAC conference on computer and communications security, 2017, pp. 1285–1298.

[23] S. H. Lee, J. Kim, D. Won, S. W. Kim, and B. R. Moon, "Generative adversarial networks for anomaly detection in network traffic data," in 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC), 2018, pp. 126–131.

[24] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, A. A. Ghorbani. "CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment," Sensor (2023) – (submitted to Journal of Sensors).https://www.unb.ca/cic/datasets/iotdataset-2023.html

[25] Han et al. Game Theory for Feature Selection. IEEE Transactions on Cybernetics 2020.

[26] . S. Urmila, "Darknet (Tor) Accessing Identification System Using Deep-Wide Cross Network," Lecture Notes in Electrical Engineering, vol. 925, pp. 303-316, 2022.

[27] T. S. Urmila, "Machine learning-based malware detection on Android devices using behavioral features," Materials Today: Proceedings, vol. 62, pp. 4659-4664, 2022.

[28] T. S. Urmila and R. Balasubramanian, "A novel framework for intrusion detection using distributed collaboration detection scheme in packet header data," International Journal of Computer Networks and Communications, vol. 9, no. 4, pp. 97-112, 2017.

[29] A. Lakhina, M. Crovella and C. Diot, "Diagnosing network-wide traffic anomalies," in SIGCOMM, 2004, pp. 219-230.

[30] T.-F. Yen, A. Oprea, K. Onarlioglu, T. Leetham, W. Robertson, A. Juels and E. Kirda, "Beehive: Large-scale log analysis for detecting suspicious activity in enterprise networks," in ACSAC, 2013, pp. 199-208.

[31] X. Wang, A. Chen, J. Li, D. Feng, J. Lai and Z. Yu, "Network security event detection method with change point correlation analysis," IEEE Access, vol. 6, pp. 77255-77264, 2018.

AUTHORS

Dr.J.I.Christy Eunaicy, [ORCID: 0000-0003-3371-7422], MCA., M.Phil.,Ph.D**.** is currently working as Assistant Professor in the Department of CA & IT, Thiagarajar College, Madurai, Tamilnadu, India. She obtained her Ph.D. from Bharathiar University, Coimbatore, Tamilnadu, India. She has published many research papers, which contributed significantly to the development of her field. She has 17 + years of experience in teaching. She had been working as Assistant Professor in various reputed Institutions in Abroad and India. Her research area focuses on Semantic web and Deep Learning.

Dr.C.Jayapratha M.Sc., M.Phil., Ph.D. Professor, Department of MCA**.**, Karpaga Vinayaga College of Engg.& Tech, Madurantakam Tamil Nadu., She has 16 year experience in teaching field. Completed her Ph.d in Bharthiyar University at 2021, M.E in G.K.M College Of Engineering, Anna University at 2011. Published 4 Journals, 3 Conferences and Conduct 2 seminars. Interested on Data structure and Algorithms, Data Mining and Machine Learning.

Dr. H.Salome Hemachitra, M.Sc., M.Phil., Ph.D, Lecturer in Computer Science, Sri Meenakshi Government Arts College for Women, Madurai. She completed her UG Degree (B.Sc (Physics)) LDC, Madurai in 1999, and PG Degree (M.Sc (Computer Science)) in MKU College, Madurai in 2001. She has been awarded her Ph.D Degree in Computer Science, 2018 and 15+ years of experience in teaching field. She has published 13+ research papers in journals and conferences. She has published her patent work at 2022. She has interests in domains like, Image Processing and Data science