ISSN: 2182-2069 / E-ISSN: 2182-2077

IoT-Traffic Networks Effective Features Based on NSGA-II Technique

Sahar Jasim Mohammed[1], Mohammed Sami Mohammed[2*], and Nihad Mahmoud Nasir[3]

[1]Computer Department, College of Education for Pure Science, University of Diyala, Baquba, Iraq. m.sc.sahar.jasim.m@uodiyala.edu.iq, https://orcid.org/0000-0001-8946-1070

[2*]Computer Department, College of Education for Pure Science, University of Diyala, Baquba, Iraq. dr.mohammed.sami@uodiyala.edu.iq, https://orcid.org/0000-0001-5446-2620

[3]Computer Department, College of Education for Pure Science, University of Diyala, Baquba, Iraq. Nihadalnasir1@yahoo.com, https://orcid.org/0009-0006-7640-1726

Received: January 11, 2024; Revised: March 02, 2024; Accepted: April 04, 2024; Published: May 30, 2024

Abstract

Applying approaches such as crowding distance between samples that belong to the Non/Dominated Sorting version of the Genetic Algorithm type II can gradually improve the feature subset by selecting the most relevant based on their distance values. The number of subsets has been decreased for the applied dataset from (41) features to (29) features with accuracy maintained after applying machine learning techniques. K-NN, SVM, and DT provided an improvement in predicted accuracy from (80.81, 76.6, and 86.7) to (81.9, 81.6, and 87.6) with features minimizations based on the proposed model. In addition, a study was done on the applied dataset for level and basic types of IoT regarding its affection on prediction accuracy with specifying a value of these features. This study was also done after applying the crowding distance with info gain for the first 10 features of 29 in total plus a comparison with four feature selection methods. The packet as a level type as well as the basic type was the most dominant feature of all other IoT features on the prediction accuracy. The accuracy of overall utilized Machine learning was maintained or even increased regarding these types only, especially for SVM. Connection level features and traffic types (such as the same service or same host) were less effective features on the overall machine learning prediction accuracy. This study was done for the NSL-KDD dataset to record reasonable numbers with more variety to protect machine learning methods from frequent records affection. In addition to accuracy, other evaluated parameters were done such as Precision, Recall, ROC, and PRC area to demonstrate the privileges of features minimization. This study was compared to relative studies based on feature specification or not with accuracy evaluations based on state-of-the-art methods. Packet level had a 3.8% effective ratio on the SVM method, while Basic type had a 4.5% effective ratio on the same overall SVM accuracy. Comparisons have been provided to demonstrate the validity of these results with previously utilized methods.

Keywords: Genetic Algorithm, Crowding Distance, K-Nearest Neighbor, Internet of Things.

Journal of Internet Services and Information Security (JISIS), volume: 14, number: 2 (May), pp. 247-267.
DOI: *10.58346/JISIS.2024.I2.016*

*Corresponding author: Computer Department, College of Education for Pure Science, University of Diyala, Baquba, Iraq.

1 Introduction

Devices of IoT have sensors and actuators which are built in as well as connectivity features which collect and share data to enable various applications. The superior objective of utilizing IoT is to create a smarter world by enabling seamless communication with better automation between devices. Ensuring secure communication, data encryption and authentication are essential objectives for protecting IoT systems. In (Huang, 2022; Vinh et al., 2023) and (Javed et al., 2022; Abdullah, 2020) authors presented a study of attack prediction systems to detect suspicious coming connections or undesired requests using SVM, DT and other popular Machine Learning (ML). IoT has diverse applications across various industries, including smart homes, healthcare, agriculture, manufacturing, and transportation in addition to more services. Examples include smart cities (Li et al., 2022; Liu & Ke, 2023; Bhoi et al., 2022), connected cars, industrial automation, remote healthcare monitoring as well as in smart greenhouses (Bersani et al., 2022). Achieving compatibility between different devices and platforms is a challenge in IoT especially for the newly presented approaches as well. Standardization efforts aim to establish protocols to enable smooth transmission between different devices related to IoT (Priyanka et al., 2023). It has the potential to alter the living/working way through improving efficiency, reducing costs and enabling innovative applications (Wang et al., 2022; Bhoi et al., 2022). However, it also provides challenges related to privacy, security and the ethical use of data, which means more careful consideration as the IoT system continues to evolve (Mohaisen et al., 2023). In the context of IoT, the packet flow and connection level can vary based on the communication protocols in addition to the utilized network architectures. However, a general overview of how packets, flow and connections might work in typical IoT scenarios which is presented in (Huang, 2022). These devices collect data from sensors or other sources in the form of sensor readings, status updates or other relevant information. They may perform some initial processing on the data before transmission like data aggregation, filtering or transformation. The processed data is transmitted over the network, which could be a local network depending on application fields. In some IoT architectures, there may be an edge device that performs additional processing on data which is known as edge computing. After this processing step, data is sent to a cloud or central server for further processing and then stored in databases where analytics algorithms may be applied to derive insights actions based on the received dataset. IoT devices may communicate directly with each other in a peer-to-peer fashion for local network types especially. This is common in scenarios where devices need to exchange information without relying on a central server while in other architectures, IoT devices communicate with a gateway device. The mixture of IoT and ML can enhance various industries by enabling intelligent decision-making and predictive analytics providing a high performance for IDS (Albulayhi et al., 2022). Raw data from IoT devices often requires preprocessing to handle not only missing values but also outliers and data noise. It may need specific features extracted from raw data to improve performance and then trained using historical IoT data to learn patterns. Trained ML provides a detection process depending on real-time or historical IoT data as in (Awotunde & Misra, 2022; Trivedi et al., 2023) when authors worked for two types of real datasets. It can identify abnormal patterns in the data, signalling potential issues as well as security threats. These models can predict when IoT devices are likely to fail and enable early maintenance. It can analyze energy consumption patterns from IoT-connected devices and optimize usage for efficiency. IoT devices in healthcare can collect patient data with ML models can provide insights for personalized treatment plans. It is also can analyze data from IoT-connected sensors in the supply chain to optimize inventory management. IoT devices in urban environments for example can collect data on traffic, waste management and public services, while on the other hand ML models can optimize city activities. It also analyzes patterns in IoT data to detect security breaches/fraudulent activities (Musleh et al., 2023). Handling sensitive data from IoT devices requires robust security measures as was presented and studied for Aposematic IoT-23 and other IDS datasets with a reduction time of 70% (Alani & Miri, 2022). The scalability of ML models and infrastructure becomes crucial when built-in devices are increased. Ensuring compatibility and smooth communication between different IoT devices provides a new challenge for researchers. The convergence of IoT and ML opens up new possibilities for innovation across various domains, but it also brings challenges to researchers and needs validation to realize its full relations.

2 Crowding Distance

Crowding distance is a parameter often associated with multi-objective optimization algorithms, particularly in the evaluation of evolutionary algorithms such as Genetic Algorithms (GA) as explained in (Raquel & Naval, 2005; Sheikholeslami & Navimipour, 2017). Multi-objective evaluations (MOEs) involve optimizing multiple conflicting objectives while crowding distance used to maintain diversity in the population of candidate solutions and helping the algorithm explore a diverse set of solutions as applied in an improved version of it (Yue et al., 2021). Crowding distance is more related to optimization problems not directly to ML, where there are multiple inconsistent objectives to be optimized simultaneously. Therefore; it will be a unique usage of this feature to be applied to calculating information gain and any other attribute evaluations. The goal of MOEs is to find solutions on the Pareto front, which represents a set of non-dominated solutions. They are considered non-dominated types when no solution is better in all objectives as more explained by utilizing NSGA such as (Zheng & Doerr, 2022; Mohammed & Vural, 2019; Fortin & Parizeau, 2013). Crowding distance means how crowded a value is in the objective space and utilized to maintain diversity in the newly generated population. Higher crowding distance values are preferred due to their maximization of the solutions area which belongs to the objective space. In (Liu & Chen, 2019), the low values of crowding distance were omitted by using the CDE version with ZDT three types to be tested for. By favouring solutions with higher crowding distance during the selection process, the algorithm ensures that the population explores a diverse set of solutions. Such calculations are used to solve different issues as one MinMax and engineer designing issues when presented by authors in (Zheng & Doerr, 2022; Zhao et al., 2022). This helps prevent premature convergence to a specific region of the required issues and promotes a more thorough exploration of the Pareto front.

3 Feature Selection Methods

Recently information is more available to authors than before, especially with network-invented modern technologies based on IoT available datasets. Each group of data comes with risk and the detection of attackers to these data is not adequate but the need to specify which type of levels lead to the most effective prediction system. Articles and a variety of author works presented in recent years to classify these types of intrusion (Sreenivasu et al., 2022; Liloja, 2023). A deep Network was applied as an example as explained in different works to identify several types of datasets with high accuracy of detection process. Utilizing a hybrid model for three different types of datasets is presented for IoT detection systems in (Abu Alghanam et al., 2023; Kareem et al., 2022). Feature selection was taken apart as well as mixed with some classification techniques to detect any strange activities as explained in (Moustafa et al., 2019; Guo, 2021; Alhakami et al., 2019). Where a hybrid model between filtering techniques and Decision Tree (DT) to gain this goal was applied. In this paper, the authors achieved a high accuracy of about 95% compared to previous works which mentioned in (Ahmad et al., 2021). A review study with its characteristics and field applications was presented in many articles such in (Thakkar & Lohiya, 2020). The reviewed articles deal with three similar datasets based on different utilized approaches. Depending on ML and DL in addition to optimization techniques, the author provided the previously studied results. While authors (Alani & Miri, 2022), focused on models that utilized the feature selection methods as an assistance model to enhance the prediction accuracy. In the same field, feature extraction is utilized with DL also for 4 types of datasets (Fatani et al., 2022) by combining it with Aquila Optimization. In (Lin et al., 2023), the authors focused only on ML that used feature selection as a key model for enhancing the prediction model. In this paper, after specifying the outcome of previous studies that related to ML and feature selection, the authors provided advice on the beneficial methods that might fit the prediction to be applied as a feature selection. With a high dimension of the dataset, ML has a shortage of performances and for that reason, authors (Kasongo & Sun, 2019) used feature selection to reduce features to gain a low prediction error. In addition, noising records provided a high FPR without specifying the most effective features. In this paper, DT with the XGBoost model provided 90% accuracy compared to the previous result with 88% accuracy for the same model. Some features have more impact on prediction, while others have no significant effect on the prediction which leads to be omitted so the time of prediction will be better or even higher than before. In (Balaji & Narayanan, 2023), a hybrid model was introduced between a fuzzy depended search algorithm and Neural Network (NN) to minimize dataset with related attributes. This paper provided an NSL-KDD dataset with a good result with a shorter processing time. The fast speed of cyber security to identify any potential unsecured reaches depends on the amount received and the most specific dataset and features. In (Bakhshad et al., 2022), two feature selections were applied to reduce the network collected data size and provide a high accuracy. While Random Forest Tree (RFT) was applied to demonstrate these results. Variable types of feature methods are also presented in articles such as PSO type which is followed by the Look Ahead type of NN (Jeyaselvi et al., 2023). Based on the parameter reduction attempts process due to IoT resources detrimental, authors presented short and deep extraction methods of features as explained in (Basati & Faghih, 2022). Authors in this work divided input data into 3 groups and used fewer layer numbers compared to traditional 2D or 3D filters. Different approaches of feature selection are also introduced to provide a new design of IDS related to the dataset such as developing the feature extraction, optimising features before applying on a prediction system and feature standardization as in (Gopalakrishnan & Purusothaman, 2022; Wu et al., 2024; Booij et al., 2022). GAs and ML are both fields within the domain of Artificial Intelligence (AI) and there is a relationship between them as explained in more detail when applied for 3 GAs types mixed with 3 types of ML (Saif et al., 2022). GAs explore different combinations of hyperparameters to find the set that yields the best performance on a required task. Also, GAs can be applied to feature selection problems, where the most affected subset of features is the required objective. They can assist in selecting the subset in automatic approaches of features that most affect the model's performance as presented in (Kavitha & Elango, 2022). On the other hand, many ML problems involve optimization, such as finding the optimal set of parameters for a model. Also, ML models often have hyper-parameters that need to be mutual for optimal performance. GAs or other evolutionary techniques are used in ML for optimization, search and adaptation. Some ML algorithms use a population-based approach for learning policies which need to balance exploration (trying new approaches) and exploitation (leveraging known good strategies). GAs can be a valuable tool within the wide field of applications, particularly for solving complex optimization problems. Not only ML utilized for this purpose, Deep Learning with different enhancement and features selection combinations are also introduced in (Zhao et al., 2022; Derhab et al., 2020; Parimala & Kayalvizhi, 2021; Sharma et al., 2022; Ren et al., 2022; Bakhshad et al., 2022).

4 Methodology

Three applied techniques for this work which are (KNN, SVM and DT). KNN was selected for their simplicity and widely used algorithm in ML for classification or different tasks. KNN is called lazy learning where the algorithm does not have a special function obtained from data but can memorize it. It is often used as a basic model for classification tasks (Huang, 2022). It is suitable for small to medium-sized datasets where the computational cost is not limited such as in (Mohy-Eddie et al., 2023) when authors used it to build an IDS network to enhance detection accuracy. Additionally, it can be effective when the decision boundary is complex with non-linearity privilege. A proposed algorithm is also presented by (Saif et al., 2022) for NSL-KDD applied by combining K-NN in six hybrid models. Where SVM primary objective is to find a hyperchange in an N-dimensional space that leads to the right classes as employed for IDS for this purpose (Rahman et al., 2021; Ahmad et al., 2021). SVM is widely used in image classification, text categorization as well as in bioinformatics. It's particularly powerful when the data has complex relationships and when there is a need for high accuracy as presented in (Nivaashini & Thangaraj, 2019; Nugroho et al., 2020; Mishra & Paliwal, 2023). On the other hand, DT works by recursively dividing samples into smaller groups depending on attribute conditions at each node which is widely for both classification/regression tasks as in (Sarhan et al., 2022). Each internal node of DT represents a decision based on a feature and each branch represents a result of that decision. DT is valuable when explainable, ease of use and understanding the decision overall process are important as the review study introduced by authors (Nugroho et al., 2020). The selection among these algorithms was based on differences between each one of these, for example, KNN is utilized where the algorithm depends on the nearest values and its related classes to the required testing point. SVM is a discriminative model that is used to find the most significant features line border that best separates the data into different classes. Also, DT is a conditional method and is used for splitting data based on the most significant attribute. Also, some differences in solutions limitations for KNN when the decision boundary depends on the distribution of the data. In SVM, the decision boundary is determined by the most affected area that maximally separates the classes accurately.

A combination of two techniques in some articles provided a good design to build a framework of IDS such as in (Nivaashini & Thangaraj, 2019) when authors used a hybrid system between SVM and K-NN. A review study also is given for different ML in this field with future issues suggested to be worked with. Other versions of decision tree types additionally provide a good system for IDS prediction such as RFT (Ahmad et al., 2021; Siddharthan et al., 2022).

5 Proposed Algorithm

In the information gain process, some of the properties should be explained and specified in more detail to create an evaluation process of both information gain and the crowding distance process (Nimbalkar & Kshirsagar, 2021). Information gain is used to evaluate the quality of how attributes are important or the degree of affection for the desired class. If all samples belong to the same class, it's called pure node which researchers always want to aim and achieve. Table 1 shows an example of a Pure Node where all samples belong to the same class.

Table 1: Pure Nodes to One Predicted Class

Duration	Packet Size	Class
3.2	10	Introduces
2.8	51	Introduces
8.5	84	Introduces

In the same form, when having multiple types of class then it will be named as an impure node as shown in Table 2.

Table 2: Impure Nodes with Multiple Classes

Duration	Packet Size	Class
3.2	10	Introduces
2.8	51	Warms
8.5	84	Dos

Also, the entropy value will be zero for a single class while increasing when the number of classes is between 0 and 1. Moving forward with class type makes the dataset impure. Entropy for a double type of classes will be in range of [0, 1], while multiple classes highest value is $\log_2 n$. As much as classes increased such in NSL-KDD means that there are attributes definition requirements.

Calculations of entropy were given and shown in Table 3 and equation (1) below, where C refers to the classes number in S entropy, P is the class possibility occurred for the given dataset.

$$Entropy\ (S) = \sum_{i=1}^{C} - P_i(\log_2(P_i)) \tag{1}$$

Table 3: Entropy Calculations of the Impure Nodes Cases

Duration	Packet Size	Class
T_1	146	Normal (N)
T_2	151	Worms (W)
T_3	138	Normal (N)
T_4	190	Worms (W)
T_5	120	Normal (N)

The calculations for Table 3 are evaluated for two classes N=3 and W=2 as evaluated in equation (2).

$$Entropy\ (N = 3, W = 2) = 0.958 \tag{2}$$

The information gain provides data reduction in entropy for each node after the whole process, for this reason, the crowding distance is useful in this work. For the previous calculations the gain is given according to equation (3). Where S is the dataset and A is the attributes.

$$Gain\ (S, A) = Entropy(S) - \sum_{Attributes} \frac{|S_v|}{|S|} * Entropy\ |S_v| \tag{3}$$

In this equation, if information gain is taken for packet size with the condition of packet size ≥ 149, then before splitting dataset has the following information and graph. (E_Dataset) =0.958, $E\ (Packet\ Size\ \geq 149) = Zero$, $E\ (Packet\ Size < 149) = Zero$ and $E\ (Packet\ Size(\text{based on 149 condition})) = Zero$. For this reason, the entropy of packet size is also zero. So the information gain is equal to 0.958 which means a good division in dataset is available to deal with ML when the condition is taken ≥ 145 then $E\ (Packet\ Size\ \geq 145) = \frac{-2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3} = 0.9116$, $E\ (Packet\ Size < 145) = Zero$ and $E\ (Packet\ Size(\text{based on 145 conditions})) = \frac{3}{5} * 0.9116 + \frac{2}{5} * Zero = 0.547$.

The information gain for this division is 0.411 means that this splitting type is not enough and needs more specifications. Then crowding distance calculations are evaluated when combined with information gain values. This process was applied in this work to obtain the best line division between normal and up-normal classes. Considering the classes to be a cost function and providing ranks according to repeated occurred type of classes. For example, having 8 classes in the dataset if it is given as Table 4.

Table 4: Samples Regarding the Repeating Classes

Classes name	Number of Samples
Class_1	300
Class_2	200
Class_3	50
Class_4	70
Class_5	110
Class_6	210
Class_7	276
Class_8	91

Then these classes would be arranged with related attributes in descending order to find the crowding distance values of each column to select the main node at each step for impurity reduction. In addition, the increment of information gain searching values leads to specifying risk factors and deleting the least effect features for better solutions. Suppose that classes have this in an ascending order such in classes (1, 7, 6, 2, 5, 4, 3) with this rank for each class respectively according to its repeated and importance (from 7 to 1) as shown in Table 5.

Table 5: Samples Conversion for Rank and Crowding Distance Calculations

Duration	f_1	f_1 as a rank	Descending order of f_1
T_1	Class_2	4	T_2
T_2	Class_1	7	T_4
T_3	Class_3	1	T_5
T_4	Class_7	6	T_1
T_5	Class_6	5	T_7
T_6	Class_4	2	T_6
T_7	Class_5	3	T_3

Based on NSGA-II crowding distance the first state (T_2) and last state (T_3) are infinite values according to $d_{I1}^{m} = d_{IL}^{m}$. Where 1 and L refer to the first and last state values. The overall crowding distance is calculated according to equation (4). As an example, was taken for both T_2 and T_2 in equation (5) and (6).

$$d_{Ij}^{m} = d_{Ij}^{m} + \frac{f_m^{I(j+1)} - f_m^{I(j-1)}}{f_m^{max} - f_m^{min}} \tag{4}$$

$$d(T_4) = d(T_4) + \frac{f_{T4}^{T5} - f_{T4}^{T2}}{f_{T4}^{T2} - f_{T4}^{T3}} = -0.1666 \tag{5}$$

$$d(T_5) = d(T_5) + \frac{f_{T5}^{T1} - f_{T4}^{T4}}{f_{T4}^{T2} - f_{T5}^{T3}} = -0.333 = d(T_1) = d(T_7) = d(T_6) \tag{6}$$

Giving rank to class would be an issue to add crowding distance to information gain, as shown in the calculation previously should be given a rank value as an objective function value according to repeated samples. Then applying the equation of crowding distance as in Table 6.

Table 6: Samples Arrangement for Rank and Crowding Distance

Duration	Classes	Number of Samples	Value	Descending Order
T_2	Class_1	300	30	T_2
T_1	Class_2	200	20	T_4
T_3	Class_3	50	5	T_5
T_6	Class_4	70	7	T_1
T_7	Class_5	110	11	T_7
T_5	Class_6	210	21	T_6
T_4	Class_7	276	28	T_3

As calculated before the crowding distance of first and last state are zero. And other values are shown through equation (7) to equation (9): -

$$d'(T_4) = d(T_4) + \frac{f_{T_4}^{T_5} - f_{T_4}^{T_2}}{f_{T_4}^{T_2} - f_{T_4}^{T_3}} = -0.28 \tag{7}$$

$$d'(T_5) = d(T_5) + \frac{f_{T_5}^{T_1} - f_{T_5}^{T_4}}{f_{T_5}^{T_2} - f_{T_5}^{T_3}} = -0.32 \tag{8}$$

$$d'(T_1) = -0.4,\ d'(T_7) = -0.52 \text{ and } d'(T_6) = -0.24 \tag{9}$$

Taking absolute value and the highest distance between samples according to given rank and repeated classes related to attributes will be selected to be calculated with information gain to select the most effective ones.

The flow chart of calculating the information gain after crowding distance calculations was based on Figure 1.

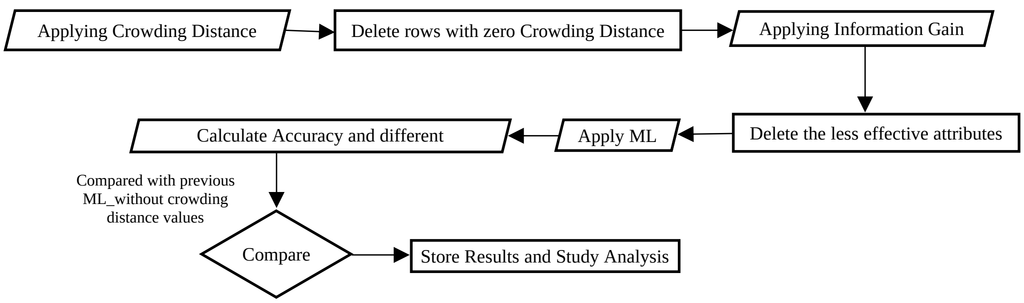

Figure 1: Information Gain Based on Crowding Distance

In addition, all of the entropy calculations after applying crowding distance will be evaluated based on Equation 10 through equation (12).

$$Entropy\ (S^*) = \sum_{i^*=1}^{C} - P_i^{\,*}(\log_2(P_i^{\,*})) \tag{10}$$

Where * refers to attributes related to crowding distance calculations.

$$Gain\ (S^*,\ A^*)|(node\ condition) = Entropy(S^*) - \sum_{Attributes^*} \frac{|S_v^{\,*}|}{|S^*|} * Entropy\ |S_v^{\,*}| \tag{11}$$

When $S_v^{\,*} = S_v - S_v^{\,0}$ and $S_v^{\,0}$ refers to the deleted attributes for min. crowding distance calculations.

$$S_v^{\,0} = S|\ d_j^m\ |_{condition=Zero}$$

$$S_v^{\,0} = S|\ dI_j^m + \frac{f_m^{I_{j+1}} - f_m^{I_{j-1}}}{f_m^{Max} - f_m^{Min}}\,|_{condition=Zero} \tag{12}$$

Where $dI_j^m = \infty$ for first and last rows value and $S_v^{\,0}$ = zero where $f_m^{I_{j+1}} - f_m^{I_{j-1}}$= zero for the same (repeated rows). This means providing a well-spread dataset and having a good distribution. At the same time, where f_{max}-f_{min} is large then the information gain is low as well, which means there is a high crowding distribution between range and other outliers value exceeding 1.5 * highest or 1.5*lowest value. Therefore; crowding distance calculations lead to minimising the number of rows to be well distributed. In the same way, it lead to a good information gain characteristic which omit the less effective attributes that can harm the prediction accuracy instead of assistance privileges. NSGA-II is a popular EAs designed for solving multi-objective optimization problems. It is known for its efficiency and ability to handle multiple related objectives as introduced in (Yusoff et al., 2011). One of the privileges of NSGA-II is the crowding distance approach which is used to maintain diversity in the regenerated population. Solutions within each line are sorted based on their calculated crowding distances, when solutions have larger crowding distances provide solutions which are real and will impact the desired function along the Pareto line. In multi-objective problems, the goal is to find solutions that form a line of possible solutions which is called as the Pareto front that has been introduced by many works such as (Raquel & Naval, 2005). The crowding distance provides good distribution among solutions in regenerated ones. The flow chart of the proposed model is based on three ML techniques combined with NSGA-II as crowding distance calculations shown in Figure 2.

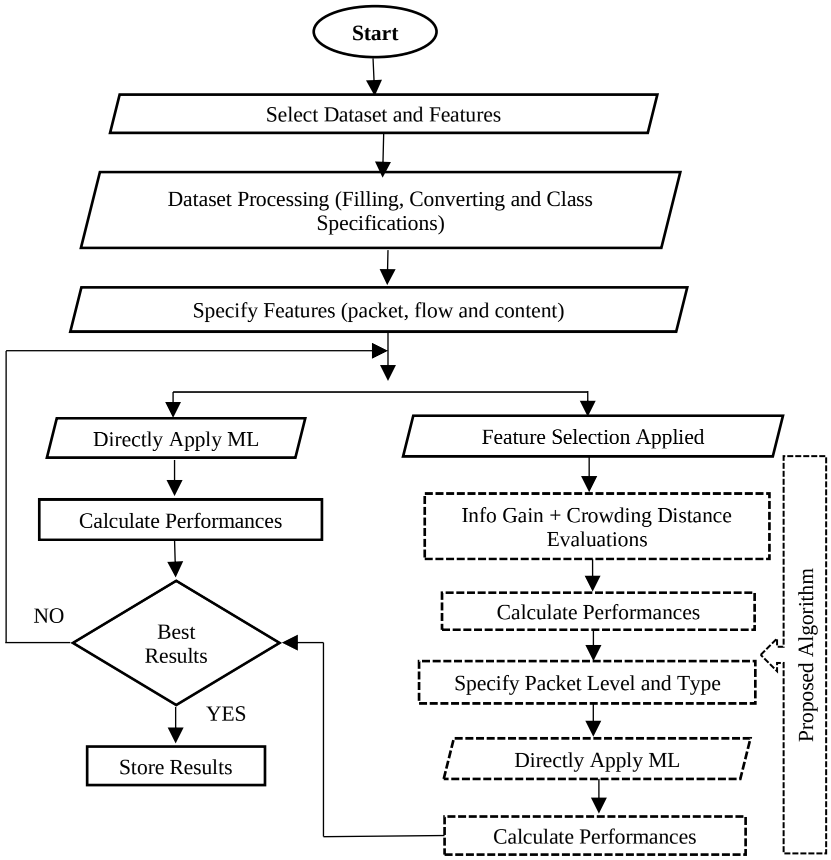

Figure 2: The Flow Chart of the Proposed Model based on ML + Crowding Distance

The effective Value of different types of features can be calculated based on equation (13).

$$\text{Effec\_Value} = \frac{N_i}{N_n} * |[Acc_{m1} - Acc_{m0}]| \quad (13)$$

Where i, is the rest features, n is the total features, m in Acc_{m0} is referred to m for type of applied ML such as KNN, SVM and DT. While 0 in Acc_{m0} referred to the previous state of calculations and 1 in Acc_{m1} referred to the next state of calculations. NSL-KDD dataset has a good distribution with 41 attributes, which is already utilized in several previous articles such in (Javed et al., 2022; Nimbalkar & Kshirsagar, 2021; D. Li et al., 2019; Abu Alghanam et al., 2023; Nugroho et al., 2020).

6 Results and Discussion

Before Applying the Proposed Algorithm

It was applied for 50441 rows and 41 attributes, which is shown in Table 7 with parameters evaluations and metrics related to the applied machine learning techniques.

Table 7: IoT Prediction Values Regardless Crowding Distance for 3 ML Techniques

Parameters	KNN	SVM	DT
TP+TN	40800	38.651	43764
FP+FN	9641	11790	6677
Accuracy (%)	80.81	76.6	86.7
Kappa Static	0.742	0.697	0.821
MAE	0.0387	0.0474	0.0311
No. of Classes	10	10	10
Precision (%)	81.2	85	86.8
Recall (%)	80.9	76.6	85.9
ROC Area (%)	93.2	96.1	97
PRC Area (%)	78.4	83.7	86.7
Time Taken (Sec.)	0.1	161.18	6.04

After Applying the Crowding Distance with Info Gain

The selection was calculated according to the first 10 selections of 41 to specify which part of IoT levels are most important, in addition for the same selection according to basic and traffic division. The numbers shown in Table 8 explain number of attributes related to this type of column.

Table 8: IoT Prediction Values Regarding Crowding Distance for 3 ML Techniques

Methods Name	According to Levels			According to IoT Basic and Traffic Types			
	Packet	Flow	Connection	Basic	Content	Same Service	Same Host
Cfs Subset Eval.	6	4	0	4	3	1	2
Classifier Attributes Eval.	7	3	0	6	4	0	0
Correlation Attributes	8	2	0	4	2	2	2
Gain Ratio Attributes	7	3	0	5	3	1	1
Info Gain + Crowding Distance	5	4	1	6	2	1	1

Figure 3 shows the attributes selectors according to levels of IoT type which specify each of affected levels and the related type.

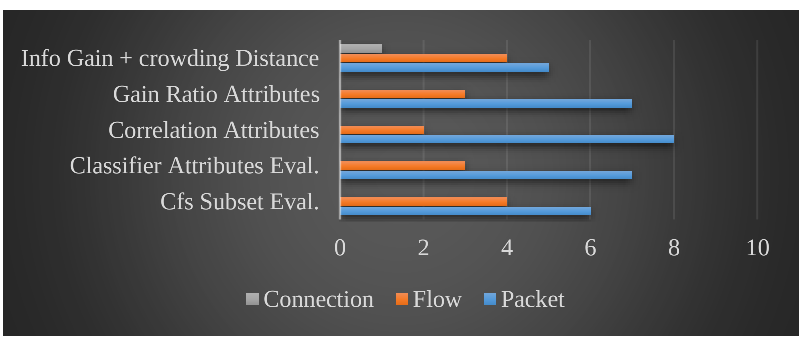

Figure 3: Attributes Selectors According to Levels of IoT Type

While in Figure 4 shows the attribute selector methods based on basic types of IoT type which specify each of the affected levels and the related type. Based on NSL-KDD dataset the number of related samples to each attribute is given in Table 9.

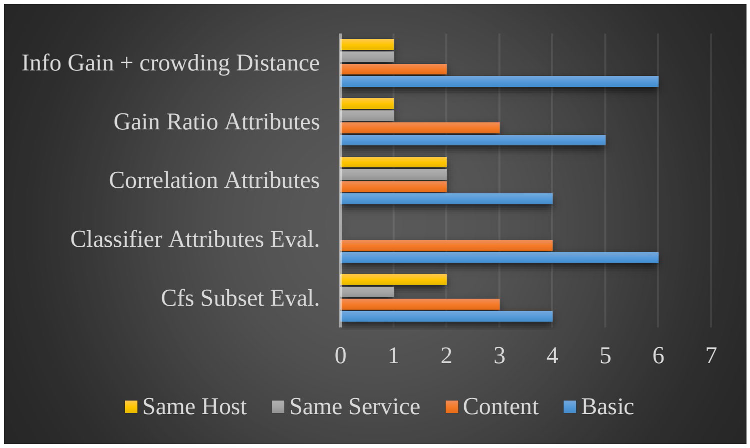

Figure 4: Attributes Selectors Methods Based on Basics Types of IoT

Table 9: IoT Referring Samples Based on Repeated Values

Class Types	Number of Samples	Referred Samples
Analysis	267	A
Backdoor	192	B
Dos	3372	D
Exploits	8033	E
Fuzzers	3223	F
Generic	12022	G
Normal	20519	N
Reconnaissance	2521	R
Shellcode	261	S
Worms	30	W

In Figure 5, the types and number of classes are shown in detail to show the different spread of classes with high and low numbers. These values refer to the complexity of the requirements system that wants to be designed for IoT prediction.

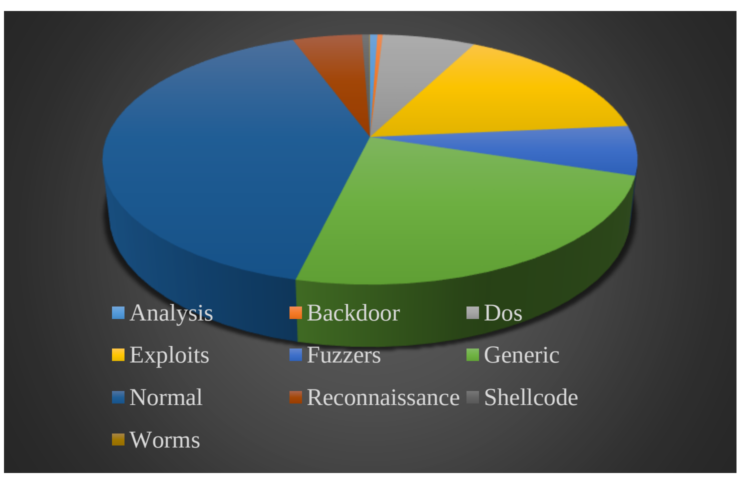

Figure 5: Introduces Class Numbers for the Selected Dataset

The overall calculated information gain combined with the crowding distance for the selected dataset of NSL-KDD is given in Table 10.

Table 10: The Overall Calculated Information Gain Based on NSL-KDD and Proposed Algorithm

InfoGain	Attribute Number in Dataset	attributes	Info gain based on crowding distance calculations
0.62854	25	ackdat	0.374863
0.69022	32	ct_dst_ltm	0.411649
0.75966	34	ct_dst_sport_ltm	0.453063
0.71594	35	ct_dst_src_ltm	0.426988
0.05925	38	ct_flw_http_mthd	0.035337
0.01098	37	ct_ftp_cmd	0.006548
0.69696	33	ct_src_dport_ltm	0.415669
0.61183	39	ct_src_ltm	0.364897
0.78145	40	ct_srv_dst	0.466059
0.74015	30	ct_srv_src	0.441427
0.88304	31	ct_state_ttl	0.526647
1.01138	7	dbytes	0.603189
0.80482	16	dinpkt	0.479996
0.55333	18	djit	0.330007
0.79817	12	dload	0.47603
0.52493	14	dloss	0.313069
0.90466	27	dmean	0.539541
0.78358	5	dpkts	0.467329
0.30131	21	dtcpb	0.179702
0.86324	10	dttl	0.514838
0.82234	1	dur	0.490445
0.30109	22	dwin	0.179571
0.01097	36	is_ftp_login	0.006543
0.00773	41	is_sm_ips_ports	0.00461
0.8644	8	rate	0.51553
0.1062	29	response_body_len	0.063338
1.58643	6	sbytes	0.94615
0.62005	2	service	0.369799
0.72605	15	sinpkt	0.433018
0.57334	17	sjit	0.341941
1.24768	11	sload	0.744119
0.46028	13	sloss	0.274512
1.32438	26	smean	0.789863
0.57826	4	spkts	0.344876
0.56146	3	state	0.334856
0.30144	20	stcpb	0.179779
0.80579	9	sttl	0.480575
0.30503	19	swin	0.181921
0.64967	24	synack	0.387465
0.64843	23	tcprtt	0.386725
0.05829	28	trans_depth	0.034764

After applying this minimization, ML was applied to IoT information as shown in Table 11 below.

Table 11: Prediction Performances Based on the Proposed Algorithm

Parameters	KNN	SVM	DT
TP+TN	41361	41183	44191
FP+FN	9080	9258	6250
Accuracy (%)	81.9	81.6	87.6
Kappa Static	0.75	0.75	0.83
MAE	0.0365	0.161	0.0286
No. of Classes	10	10	10
Precision (%)	82.3	81	97.9
Recall (%)	82	81	98.2
ROC Area (%)	93.6	93.3	97.3
PRC Area (%)	79.4	77.9	87.8
Time Taken (Sec.)	0.1	123	3.51

The related parameters of total detected true and negative in positive and negative sides for both before and after applying the proposed algorithm are shown in Figure 6.

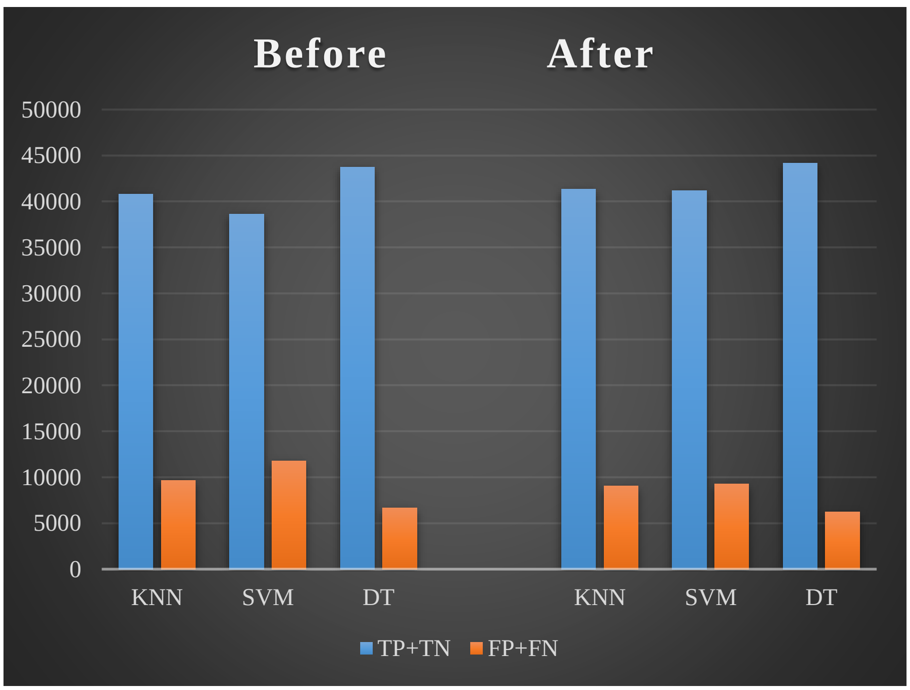

Figure 6: Predicted True and Negative Values Before and After Applying the Proposed Algorithm

Other metrics like accuracy and precision are also shown in Figure 7 to demonstrate that working on the proposed algorithm by combining the crowding distance and information gain provide a better result with dataset minimization. This figure shows the better performances for each selected ML methods and for all parameters. Kappa static calculations and time taken for all required techniques without combination and with combination are shown also in Figure 8. These values are shown in percentage to specify the differences between parameters clearly.

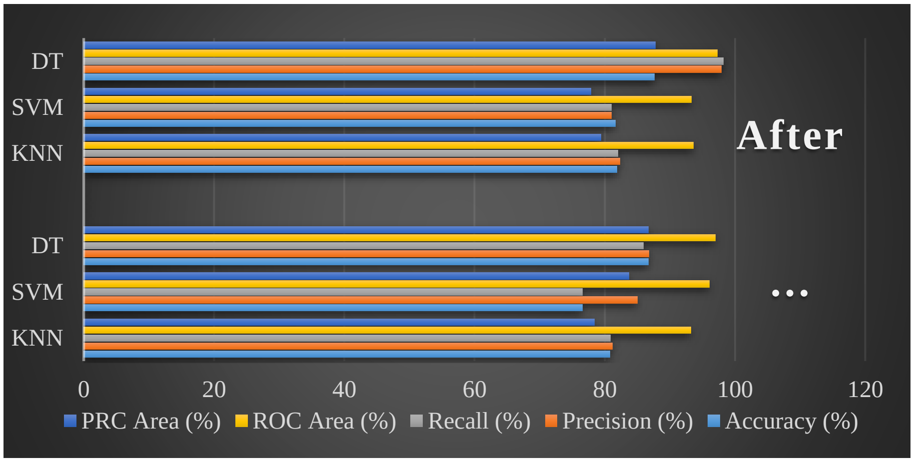

Figure 7: Overall Metrics Before and After Applying the Proposed Algorithm

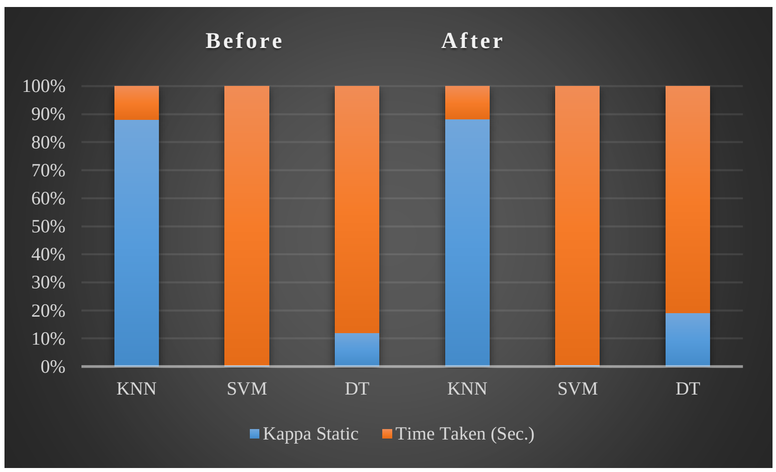

Figure 8: Kappa Static Calculations and Time Taken Before and After Applying the Proposed Algorithm

A minimization of attributes is specified according to Info Gain plus the Crowding Distance, then ML is applied for this calculation. the minimization occurred for 41 attributes to be 29 attributes. Table 12 shows the calculations according to levels, while Table 13, shows the calculations according to basic and host type information.

Table 12: Prediction Performances Effective Ratio According to Dataset Level's Information

Levels	KNN	SVM	DT
Packet	0.85%	3.8%	0.7%
Flow	0.73%	3.3%	0.6%
Connection	0.55%	2.5%	0.45%

Table 13: Prediction Performances Effective Ratio According to Dataset Basic's Information

Type	KNN	SVM	DT
Basic	0.99%	4.5%	0.81%
Content	0.825%	3.75%	0.675%
Same Service	0.48%	2.22%	0.4%
Same Host	0.44%	2%	0.36%

Table 14 shows the effectiveness of features (29) features based on information gain combined with crowding distance. It was applied to remove the ineffective features. The remaining attributes from each type of IoT information were explained in this table to provide a good study for the most effective features of IoT collected information.

Table 14: The Effectiveness of 29-Features Based on Information Gain Combined with Crowding Distance

Type of IoT	All before minimization	Updated (removed the umrequired features)
Packet Level	18	14
Flow Level	21	14
Connection Level	2	1
Basic Type	10	9
Content Type	12	9
Traffic Same Service	9	5
Traffic Same Host	10	6

Figure 9 shows the minimization in features from 41 to 29 features with more specifications in levels and types.

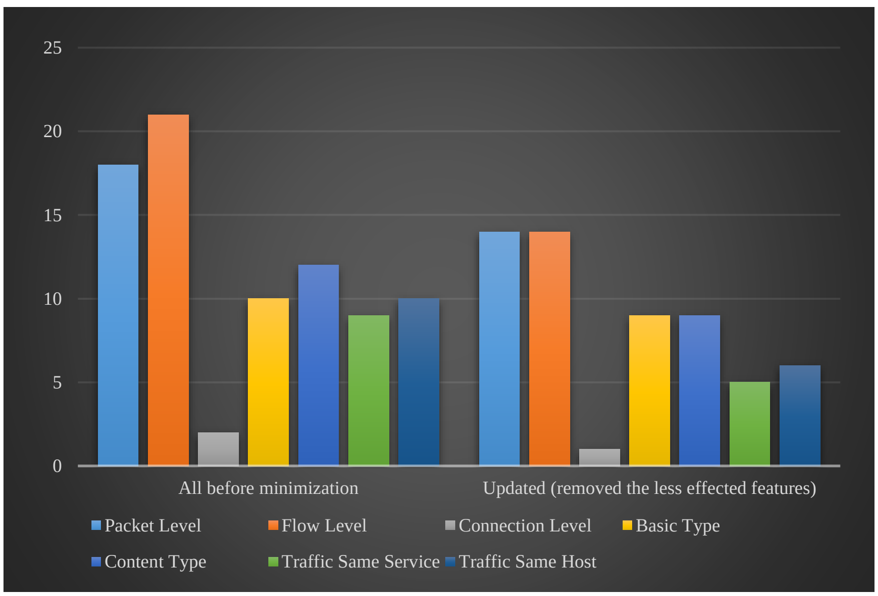

Figure 9: The Minimization in Features from 41 To 29 For Both Levels and Basics

These calculations for K-NN are evaluated based on equation (1) as below based on levels for more details:

$$\text{Packet (\%)} = \frac{14}{18} * |[\,81.9 - 80.8]| = 0.85$$

$$\text{Flow (\%)} = \frac{14}{21} * |[\,81.9 - 80.8]| = 0.73$$

$$\text{Connection (\%)} = \frac{1}{2} * |81.9 - 80.8| = 0.55$$

While based on the types of IoT information for K-NN these calculations are done according to the same equation (1) but for the basic and other types of values as below: -

$$\text{Basic (\%)} = \frac{9}{10} * |[\,1.1]| = 0.99$$

$$\text{Content (\%)} = \frac{9}{12} * |[\ 1.1]| = 0.825$$

$$\text{Same Service (\%)} = \frac{4}{92} * |1.1| = 0.48$$

$$\text{Same Host (\%)} = \frac{4}{10} * |1.1| = 0.44$$

To demonstrate these calculations, providing a comparison between these results of using crowding distance plus information gain technique and other applied techniques is shown in Table 15.

Table 15: Comparison Between Proposed Method and Related Works

Reference Number	Feature Method	Number of Applied Features	Accuracy	Feature Type Specification
[2]	Restricted Boltzmann	/	87%	No
[4]	Novel Feature Method	39	/	No
[5]	Infinite Bounded	41	83.49%	No
[5]	Finite Gaussian Mixture	41	82.52%	No
[5]	Finite Generalized Gaussian Mixture	41	82.77%	No
[5]	Finite Bounded	41	81.34%	No
[9]	Deep Extraction Method	41	/	No
Proposed Model	Crowding Distance + Information Gain Method	29	87.6%	Yes

7 Conclusion

Selection of the most relevant subset features was improved depending on combinations between Crowding Distance among samples and applying the Information Gain method. The evaluation process according to different types of metrics was done before and after applying the proposed model, which provides stable system performances or even better efficiency. Features were minimized from 41 to 29, and then K-NN, SVM, and DT were applied to the reduction dataset to get system accuracy improvement of about 1.1%, 5%, and 0.9% respectively for the applied methods. This study was also done to specify the most affected IoT features on the prediction accuracy depending on the level as well as for the basic types. Packet features among IoT-level parts and the Basic features among the IoT-type parts have been selected as the most effective factors in these techniques' performances. Packet level had a 3.8% effective ratio on the SVM method, while Basic type had a 4.5% effective ratio on the same overall SVM accuracy. These results are demonstrated by calculating the distance between features based on the proposed model as shown in Table (10 –14). The number of features was adequate for all ML techniques especially for SVM due to the variety of dataset samples with noisy reduction ability when applying the proposed model. The less effective features were the Connection from the level parts and the Traffic from the type parts for NSL-KDD. These results have been obtained accurately after applying crowding distance plus information gain due to the reduction of frequent records that prevent ML from these affections. In addition, features specification with its affection ratio on ML performance had been determined in this study which was compared to related previous work for the same purpose.

The feature selection process has a significant impact on any process related to detection and prediction ability due to time, cost, and memory reduction privileges. As shown in this study, even when having a minimum number of features it will be adequate for prediction if these have been well optimized. In addition, the selection of compatible ML has also impact on prediction and feature types as well. It means that applying different ML techniques might be compatible or not with the same selected features. This makes it a key challenge for researchers to study several techniques with another improvement in the selection process to define a group of affected features.

References

[1] Alghanam, O.A., Almobaideen, W., Saadeh, M., & Adwan, O. (2023). An improved PIO feature selection algorithm for IoT network intrusion detection system based on ensemble learning. *Expert Systems with Applications*, *213*, 118745. https://doi.org/10.1016/j.eswa.2022.118745

[2] Ahmad, M., Riaz, Q., Zeeshan, M., Tahir, H., Haider, S.A., & Khan, M S. (2021). Intrusion detection in internet of things using supervised machine learning based on application and transport layer features using UNSW-NB15 data-set. *EURASIP Journal on Wireless Communications and Networking*, *2021*, 1-23. https://doi.org/10.1186/s13638-021-01893-8

[3] Alani, M.M., & Miri, A. (2022). Towards an explainable universal feature set for IoT intrusion detection. *Sensors*, *22*(15), 5690. https://doi.org/10.3390/s22155690

[4] Albulayhi, K., Abu Al-Haija, Q., Alsuhibany, S.A., Jillepalli, A.A., Ashrafuzzaman, M., & Sheldon, F.T. (2022). IoT intrusion detection using machine learning with a novel high performing feature selection method. *Applied Sciences*, *12*(10), 5015. https://doi.org/10.3390/app12105015

[5] Abdullah, D. (2020). A Linear Antenna Array for Wireless Communications. *National Journal of Antennas and Propagation (NJAP), 2*(1), 19-24.

[6] Alhakami, W., ALharbi, A., Bourouis, S., Alroobaea, R., & Bouguila, N. (2019). Network anomaly intrusion detection using a nonparametric Bayesian approach and feature selection. *IEEE access*, *7,* 52181-52190. https://doi.org/10.1109/ACCESS.2019.2912115

[7] Awotunde, J.B., & Misra, S. (2022). Feature extraction and artificial intelligence-based intrusion detection model for a secure internet of things networks. *In Illumination of artificial intelligence in cybersecurity and forensics,* 21-44. Cham: Springer International Publishing.

[8] Bakhshad, S., Ponnusamy, V., Annur, R., Waqasyz, M., Alasmary, H., & Tux, S. (2022). Deep reinforcement learning based intrusion detection system with feature selections method and optimal hyper-parameter in IoT environment. *In IEEE International Conference on Computer, Information and Telecommunication Systems (CITS),* 1-7. https://doi.org/10.1109/CITS55221.2022.9832976

[9] Vinh, D.T., Phi, T.L., & Viet, C.T. (2023). A Secure Proxy Re-Signature Scheme for IoT. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 14*(1), 174-188.

[10] Balaji, S., & Narayanan, S.S. (2023). Dynamic distributed generative adversarial network for intrusion detection system over internet of things. *Wireless Networks, 29*(5), 1949-1967. https://doi.org/10.1007/s11276-022-03182-8

[11] Basati, A., & Faghih, M.M. (2022). DFE: efficient IoT network intrusion detection using deep feature extraction. *Neural Computing and Applications*, *34*(18), 15175-15195. https://doi.org/10.1007/s00521-021-06826-6

[12] Trivedi, J., Devi, M.S., & Solanki, B. (2023). Step Towards Intelligent Transportation System with Vehicle Classification and Recognition Using Speeded-up Robust Features. *Arhiv za tehničke nauke, 1*(28), 39-56.

[13] Bersani, C., Ruggiero, C., Sacile, R., Soussi, A., & Zero, E. (2022). Internet of things approaches for monitoring and control of smart greenhouses in industry 4.0. *Energies*, *15*(10), 3834. https://doi.org/10.3390/en15103834

[14] Bhoi, S. K., Panda, S. K., Jena, K.K., Sahoo, K.S., Jhanjhi, N.Z., Masud, M., & Aljahdali, S. (2022). IoT-EMS: An internet of things-based environment monitoring system in volunteer

computing environment. *Intell. Autom. Soft Comput*, *32*(3), 1493-1507. https://doi.org/10.32604/IASC.2022.022833

[15] Booij, T.M., Chiscop, I., Meeuwissen, E., Moustafa, N., & Den Hartog, F.T. (2021). ToN_IoT: The role of heterogeneity and the need for standardization of features and attack types in IoT network intrusion data sets. *IEEE Internet of Things Journal*, *9*(1), 485-496. https://doi.org/10.1109/JIOT.2021.3085194

[16] Derhab, A., Aldweesh, A., Emam, A.Z., & Khan, F.A. (2020). Intrusion detection system for internet of things based on temporal convolution neural network and efficient feature engineering. *Wireless Communications and Mobile Computing*, *2020*, 1-16. https://doi.org/10.1155/2020/6689134

[17] Sreenivasu, M., Kumar, U.V., & Dhulipudi, R. (2022). Design and Development of Intrusion Detection System for Wireless Sensor Network. *Journal of VLSI Circuits and Systems*, *4*(2), 1-4.

[18] Fatani, A., Dahou, A., Al-Qaness, M.A., Lu, S., & Elaziz, M.A. (2021). Advanced feature extraction and selection approach using deep learning and Aquila optimizer for IoT intrusion detection system. *Sensors*, *22*(1), 140. https://doi.org/10.3390/s22010140

[19] Fortin, F.A., & Parizeau, M. (2013). Revisiting the NSGA-II crowding-distance computation. In *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, 623-630. https://doi.org/10.1145/2463372.2463456

[20] Gopalakrishnan, B., & Purusothaman, P. (2022). A new design of intrusion detection in IoT sector using optimal feature selection and high ranking-based ensemble learning model. *Peer-to-Peer Networking and Applications*, *15*(5), 2199-2226. https://doi.org/10.1007/s12083-022-01336-1

[21] Guo, G. (2021). A machine learning framework for intrusion detection system in iot networks using an ensemble feature selection method. *In IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, 0593-0599. https://doi.org/10.1109/IEMCON53756.2021.9623082

[22] Priyanka, J., Ramya, M., & Alagappan, M. (2023). IoT Integrated Accelerometer Design and Simulation for Smart Helmets. *Indian Journal of Information Sources and Services*, *13*(2), 64–67.

[23] Huang, L. (2022). Design of an IoT DDoS attack prediction system based on data mining technology. *The Journal of Supercomputing*, *78*(4), 4601-4623. https://doi.org/10.1007/s11227-021-04055-1

[24] Javed, S.H., Ahmad, M.B., Asif, M., Almotiri, S.H., Masood, K., & Ghamdi, M.A.A. (2022). An intelligent system to detect advanced persistent threats in industrial internet of things (I-IoT). *Electronics*, *11*(5), 742. https://doi.org/10.3390/electronics11050742

[25] Jeyaselvi, M., Dhanaraj, R.K., Sathya, M., Memon, F.H., Krishnasamy, L., Dev, K., & Qureshi, N.M.F. (2023). A highly secured intrusion detection system for IoT using EXPSO-STFA feature selection for LAANN to detect attacks. *Cluster Computing*, *26*(1), 559-574. https://doi.org/10.1007/s10586-022-03607-1

[26] Mohaisen, M., Saif, W., & Alkhater, N. (2023). Novel IoT Device for Enhanced Security. *International Journal of Communication and Computer Technologies (IJCCTS)*, *11*(2), 68-79.

[27] Kareem, S.S., Mostafa, R.R., Hashim, F.A., & El-Bakry, H.M. (2022). An effective feature selection model using hybrid metaheuristic algorithms for iot intrusion detection. *Sensors*, *22*(4), 1396. https://doi.org/10.3390/s22041396

[28] Kasongo, S.M., & Sun, Y. (2019). A deep learning method with filter-based feature engineering for wireless intrusion detection system. *IEEE Access*, 7, 38597–38607. https://doi.org/10.1109/ACCESS.2019.2905633

[29] Kavitha, G., & Elango, N.M. (2022). Genetic algorithm-conditional mutual information maximization based feature selection for bot attack classification in IoT devices. *Journal of Mobile Multimedia*, 119-134. https://doi.org/10.13052/jmm1550-4646.1816

[30] Li, D., Deng, L., Lee, M., & Wang, H. (2019). IoT data feature extraction and intrusion detection system for smart cities based on deep migration learning. *International journal of information management*, *49*, 533-545. https://doi.org/10.1016/j.ijinfomgt.2019.04.006

[31] Liloja and Ranjana, P. (2023). An Intrusion Detection System Using a Machine Learning Approach in IOT-based Smart Cities. *Journal of Internet Services and Information Security, 13*(1), 11-21

[32] Li, X., Liu, H., Wang, W., Zheng, Y., Lv, H., & Lv, Z. (2022). Big data analysis of the internet of things in the digital twins of smart city based on deep learning. *Future Generation Computer Systems*, *128*, 167-177. https://doi.org/10.1016/j.future.2021.10.006

[33] Lin, H., Xue, Q., Feng, J., & Bai, D. (2023). Internet of things intrusion detection model and algorithm based on cloud computing and multi-feature extraction extreme learning machine. *Digital Communications and Networks*, *9*(1), 111-124. https://doi.org/10.1016/j.dcan.2022.09.021

[34] Liu, C., & Ke, L. (2023). Cloud assisted Internet of things intelligent transportation system and the traffic control system in the smart city. *Journal of Control and Decision*, *10*(2), 174-187. https://doi.org/10.1080/23307706.2021.2024460

[35] Liu, J., & Chen, X. (2019). An improved NSGA-II algorithm based on crowding distance elimination strategy. *International Journal of Computational Intelligence Systems*, *12*(2), 513-518. https://doi.org/10.2991/ijcis.d.190328.001

[36] Mishra, A.K., & Paliwal, S. (2023). Mitigating cyber threats through integration of feature selection and stacking ensemble learning: the LGBM and random forest intrusion detection perspective. *Cluster Computing*, *26*(4), 2339-2350. https://doi.org/10.1007/s10586-022-03735-8

[37] Mohammed, M.S., & Vural, R.A. (2018). NSGA-II+ FEM based loss optimization of three-phase transformer. *IEEE Transactions on Industrial Electronics*, *66*(9), 7417-7425. https://doi.org/10.1109/TIE.2018.2881935

[38] Mohy-Eddine, M., Guezzaz, A., Benkirane, S., & Azrour, M. (2023). An efficient network intrusion detection model for IoT security using K-NN classifier and feature selection. *Multimedia Tools and Applications*, *82*(15), 23615-23633. https://doi.org/10.1007/s11042-023-14795-2

[39] Moustafa, N., Turnbull, B., & Choo, K.K.R. (2018). An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things. *IEEE Internet of Things Journal*, *6*(3), 4815-4830. https://doi.org/10.1109/JIOT.2018.2871719

[40] Musleh, D., Alotaibi, M., Alhaidari, F., Rahman, A., & Mohammad, R.M. (2023). Intrusion detection system using feature extraction with machine learning algorithms in IoT. *Journal of Sensor and Actuator Networks*, *12*(2), 29. https://doi.org/10.3390/jsan12020029

[41] Nimbalkar, P., & Kshirsagar, D. (2021). Feature selection for intrusion detection system in Internet-of-Things (IoT). *ICT Express*, *7*(2), 177-181.https://doi.org/10.1016/j.icte.2021.04.012

[42] Nivaashini, M., & Thangaraj, P. (2018). A framework of novel feature set extraction based intrusion detection system for internet of things using hybrid machine learning algorithms. *In IEEE International conference on computing, power and communication technologies (GUCON)*, 44-49. https://doi.org/10.1109/GUCON.2018.8674952

[43] Nugroho, E.P., Djatna, T., Sitanggang, I.S., Buono, A., & Hermadi, I. (2020). A review of intrusion detection system in IoT with machine learning approach: current and future research. *In IEEE 6th international conference on science in information technology (ICSITech)*, 138-143. https://doi.org/10.1109/ICSITech49800.2020.9392075

[44] Parimala, G., & Kayalvizhi, R. (2021). An effective intrusion detection system for securing IoT using feature selection and deep learning. *In IEEE international conference on computer communication and informatics (ICCCI)*, 1-4. https://doi.org/10.1109/ICCCI50826.2021.9402562

[45] Rahman, M.A., Asyhari, A.T., Wen, O.W., Ajra, H., Ahmed, Y., & Anwar, F. (2021). Effective combining of feature selection techniques for machine learning-enabled IoT intrusion detection. *Multimedia Tools and Applications*, *80*(20), 31381-31399. https://doi.org/10.1007/s11042-021-10567-y
[46] Raquel, C.R., & Naval Jr, P.C. (2005). An effective use of crowding distance in multiobjective particle swarm optimization. *In Proceedings of the 7th Annual conference on Genetic and Evolutionary Computation*, 257-264. https://doi.org/10.1145/1068009.1068047
[47] Ren, K., Zeng, Y., Cao, Z., & Zhang, Y. (2022). ID-RDRL: a deep reinforcement learning-based feature selection intrusion detection model. *Scientific reports*, *12*(1), 15370. https://doi.org/10.1038/s41598-022-19366-3
[48] Saif, S., Das, P., Biswas, S., Khari, M., & Shanmuganathan, V. (2022). HIIDS: Hybrid intelligent intrusion detection system empowered with machine learning and metaheuristic algorithms for application in IoT based healthcare. *Microprocessors and Microsystems*, 104622. https://doi.org/10.1016/j.micpro.2022.104622
[49] Sarhan, M., Layeghy, S., Moustafa, N., Gallagher, M., & Portmann, M. (2022). Feature extraction for machine learning-based intrusion detection in IoT networks. *Digital Communications and Networks*. https://doi.org/10.1016/j.dcan.2022.08.012
[50] Sharma, B., Sharma, L., & Lal, C. (2022). Feature selection and deep learning technique for intrusion detection system in IoT. In *Proceedings of International Conference on Computational Intelligence: ICCI 2020*, 253-261. Springer Singapore. https://doi.org/10.1007/978-981-16-3802-2_21
[51] Sheikholeslami, F., & Navimipour, N. J. (2017). Service allocation in the cloud environments using multi-objective particle swarm optimization algorithm based on crowding distance. *Swarm and Evolutionary Computation*, *35*, 53-64. https://doi.org/10.1016/j.swevo.2017.02.007
[52] Siddharthan, H., Deepa, T., & Chandhar, P. (2022). SENMQTT-SET: An intelligent intrusion detection in IoT-MQTT networks using ensemble multi cascade features. *IEEE Access*, *10*, 33095-33110. https://doi.org/10.1109/ACCESS.2022.3161566
[53] Thakkar, A., & Lohiya, R. (2020). A review of the advancement in intrusion detection datasets. *Procedia Computer Science*, *167*, 636-645. https://doi.org/10.1016/j.procs.2020.03.330
[54] Wang, S., Nie, L., Li, G., Wu, Y., & Ning, Z. (2022). A multitask learning-based network traffic prediction approach for SDN-enabled industrial internet of things. *IEEE Transactions on Industrial Informatics*, *18*(11), 7475-7483. https://doi.org/10.1109/TII.2022.3141743
[55] Wu, J., Haider, S.A., Yu, H., Irshad, M., Soni, M., Bhadla, M.K., & Zikria, Y.B. (2024). An intelligent IoT intrusion detection system using HeInit-WGAN and SSO-BNMCNN based multivariate feature analysis. *Engineering Applications of Artificial Intelligence*, *127*, 107132. https://doi.org/10.1016/j.engappai.2023.107132
[56] Yue, C., Suganthan, P.N., Liang, J., Qu, B., Yu, K., Zhu, Y., & Yan, L. (2021). Differential evolution using improved crowding distance for multimodal multiobjective optimization. *Swarm and Evolutionary Computation*, *62*, 100849. https://doi.org/10.1016/j.swevo.2021.100849
[57] Yusoff, Y., Ngadiman, M.S., & Zain, A.M. (2011). Overview of NSGA-II for optimizing machining process parameters. *Procedia Engineering*, *15*, 3978-3983. https://doi.org/10.1016/j.proeng.2011.08.745
[58] Zhao, R., Gui, G., Xue, Z., Yin, J., Ohtsuki, T., Adebisi, B., & Gacanin, H. (2021). A novel intrusion detection method based on lightweight neural network for internet of things. *IEEE Internet of Things Journal*, *9*(12), 9960-9972. https://doi.org/10.1109/JIOT.2021.3119055
[59] Zhao, W., Zhang, Z., Mirjalili, S., Wang, L., Khodadadi, N., & Mirjalili, S.M. (2022). An effective multi-objective artificial hummingbird algorithm with dynamic elimination-based

crowding distance for solving engineering design problems. *Computer Methods in Applied Mechanics and Engineering*, *398*, 115223. https://doi.org/10.1016/j.cma.2022.115223

[60] Zheng, W., & Doerr, B. (2022). Better approximation guarantees for the NSGA-II by using the current crowding distance. In *Proceedings of the Genetic and Evolutionary Computation Conference*, 611-619. https://doi.org/10.1145/3512290.3528847

Authors Biography

Sahar Jasim Mohammed was born in Diyala/Iraq in 1990. She received the Bachelor Degree in 2010 from University of Diyala/College of Education for Pure Science-Computer Department/Iraq. Master Degree in 2019 from Altinbas University/ Electrical and Computer Engineering-Graduate School of Science and Engineering, Istanbul/Turkey. She is currently works as a lecturer at University of Diyala/College of Education for Pure Science-Computer Department. She was also Responsible of Rehabilitation, Employment and Follow-up Unit and works as an employee for Ensuring quality and university performance at the same university. Here current research interests are Datamining, Deep Learning, Artificial Intelligent, Multi-Objective Evolutionary Algorithm, Prediction Systems and Internet of Things.

Mohammed Sami Mohammed was born in Diyala/Iraq in 1980. He received the B.Sc. degree from the College of Engineering, University of Diyala, Diyala/Iraq in 2002, and the M.Sc. degree from the University of Technology, Baghdad/Iraq in 2009, both in electronic engineering. Got a Ph.D. degree from the Department of Electronics and Communication Engineering, Yildiz Technical University, Istanbul/Turkey. He is currently a Lecturer with the Department of Computer Science, College of Education for Pure Science, University of Diyala. According to Scopus site his H-index is 4 and has more than 50 citations. His current research interests are electronic circuit design optimization, DC-DC converter design, Artificial Intelligent system and Internet of Things.

Nihad Mahmoud Nasir was born in Diyala/Iraq in 1970. He received the B.Sc. degree from University of Technology, Baghdad/Iraq in 1993, and the M.Sc. degree from Iraqi Commissions for Computers and Informatics, Baghdad/Iraq in 1999. Got a Ph.D. degree from the Voronezh State University. He is currently the head of Department of Computer Science, College of Education for Pure Science, University of Diyala/Iraq. He got his professor title in 2020. His current research interests are Artificial Intelligent System, Internet of Things, Software Engineering, optimization techniques and mathematical models.