applied sciences

MDPI

Article

IoT Intrusion Detection Using Machine Learning with a Novel High Performing Feature Selection Method

Khalid Albulayhi [1], Qasem Abu Al-Haija [2], Suliman A. Alsuhibany [3,*], Ananth A. Jillepalli [4], Mohammad Ashrafuzzaman [5] and Frederick T. Sheldon [1]

[1] Computer Science Department, University of Idaho, Moscow, ID 83844, USA; albu3647@vandals.uidaho.edu (K.A.); Sheldon@uidaho.edu (F.T.S.)
[2] Department of Computer Science/Cybersecurity, Princess Sumaya University for Technology (PSUT), Amman 11941, Jordan; q.abualhaija@psut.edu.jo
[3] Department of Computer Science, College of Computer, Qassim University, Buraydah 51452, Saudi Arabia
[4] School of Electrical Engineering and Computer Science, Washington State University, Pullman, WA 99164, USA; ajillepalli@ieee.org
[5] Department of Mathematics and Computer Science, Ashland University, Ashland, OH 44805, USA; mashrafu@ashland.edu
* Correspondence: salsuhibany@qu.edu.sa

Abstract: The Internet of Things (IoT) ecosystem has experienced significant growth in data traffic and consequently high dimensionality. Intrusion Detection Systems (IDSs) are essential self-protective tools against various cyber-attacks. However, IoT IDS systems face significant challenges due to functional and physical diversity. These IoT characteristics make exploiting all features and attributes for IDS self-protection difficult and unrealistic. This paper proposes and implements a novel feature selection and extraction approach (i.e., our method) for anomaly-based IDS. The approach begins with using two entropy-based approaches (i.e., information gain (IG) and gain ratio (GR)) to select and extract relevant features in various ratios. Then, mathematical set theory (union and intersection) is used to extract the best features. The model framework is trained and tested on the IoT intrusion dataset 2020 (IoTID20) and NSL-KDD dataset using four machine learning algorithms: Bagging, Multilayer Perception, J48, and IBk. Our approach has resulted in 11 and 28 relevant features (out of 86) using the intersection and union, respectively, on IoTID20 and resulted 15 and 25 relevant features (out of 41) using the intersection and union, respectively, on NSL-KDD. We have further compared our approach with other state-of-the-art studies. The comparison reveals that our model is superior and competent, scoring a very high 99.98% classification accuracy.

Keywords: cybersecurity; anomaly detection accuracy; feature selection; Internet of Things (IoT); intrusion detection system; and machine learning

Citation: Albulayhi, K.; Abu Al-Haija, Q.; Alsuhibany, S.A.; Jillepalli, A.A.; Ashrafuzzaman, M.; Sheldon, F.T. IoT Intrusion Detection Using Machine Learning with a Novel High Performing Feature Selection Method. *Appl. Sci.* **2022**, *12*, 5015. https://doi.org/10.3390/app12105015

Academic Editors: Changho Seo, Seongsoo Cho and Bhanu Shrestha

Received: 6 April 2022
Accepted: 12 May 2022
Published: 16 May 2022

1. Introduction

New cybersecurity risks have emerged owing to organizations deploying Internet of things (IoT) devices in IT (information technology) and OT (operational technology) environments. Such new risks threaten to undermine structural tenets such as safety, mobility, efficiency, and security of operational ecosystems. New threat vectors not only affect technological aspects of our lives but also pose a risk towards financial and physical wellbeing [1]. The threat of attack has brought several insecurities to online privacy, social networks, business, and critical infrastructure [2,3]. Therefore, the development of resilient strategies has become an essential part of dynamical environments such as the IoT ecosystem. IoT is a constantly evolving emerging technology set [4,5] that changes the security and risk schematics of automated networked systems [6,7]. IoT has spread into a wide range of human systems to shape the core of our industrial society to become

the man–machine interface of life. By 2024, IoT is expected to reach 83 billion devices operationally [8]. IoT applications include smart cities, smart homes, and intelligent transportation. These applications deploy IoT devices to increase productivity and reduce costs by using 'plug-n-play' kits that do not require extensive prior device knowledge. Such a 'plug-n-play' configuration increases the risk of cyber-misbehavior. Compounding factors include the typical mixture of multitudes of wired or wireless communications that employ cloud-connected embedded systems used by consumers to interconnect with each other [9,10].

Intrusion detection systems (IDS) are widely used to improve security posture in an IT infrastructure. An IDS is considered a suitable and practical approach to detect attacks and assure network security by safeguarding against intrusive hackers [3]. Anomaly-based IDS approaches can efficiently detect zero-day (unknown) attacks [11,12]. An intrusion can be defined as a sequence of unexpected activities locally or globally, harming network confidentiality, integrity, and/or availability (i.e., the CIA triad) [13,14]. The network traffic consists of packets associated with packet header fields. Features related to those instances are important to define the purpose of detecting anomalies. The purpose of an IDS is to detect and/or prevent abnormal misbehavior (i.e., unauthorized use), both passive and active network intruder activities, and thus improve CIA.

In recent times, machine learning (ML)-based approaches have been employed for intrusion detection in IoTs IDSs [3,15–21]. Existing IDSs assume that the IoT devices have the same feature pattern and packet types. However, IoT devices vary in some respects, such as hardware characteristics and functionality, computational capability, and different abilities for generating various features [11,22]. The features become sparse when nodes are aggregated to create data, and the irrelevant features (attributes) are set to either nulls or zeros. Data sparsity is one of the disadvantages that affect the accuracy and efficiency of data modeling. Feature selection, an important part of a machine learning-based solution, plays an important role in increasing detection accuracy and speed of the training phase. Several feature selection techniques have been proposed to improve detection of anomalous behavior variants such as Flexible Mutual Information-based Feature Selection (FMIFS) [23], Modified Mutual Information-Based Feature Selection (MMIFS) with Support Vector Machine (SVM) [24], and SVM with Neural Networks (NN) [25]. Those approaches/models and other recent state-of-the-art studies have been presented in the related work section. *Detection accuracy of anomaly-based IDSs is considered the main challenge in the IoT ecosystem due to the constantly evolving nature of the IoT environment* [26,27]. This paper proposes a novel feature selection approach for machine learning-based IDS towards obtaining a resilient performance within the diverse IoT ecosystem.

1.1. Feature Selection

IoT datasets are of intrinsically high dimensionality represented by n instances and m columns (features) [11]. The data matrix is $X \in \mathbb{R}^{N \times M,}$ and the Y is the target variable(s) (class(es)). A target instance (class) may be either discrete or continuous, and the model can also be dynamic or static. A feature selection (FS) enhances model performance by reducing dimensionality. FS can be defined as a subset of $P \ll M$ features, i.e., $X_{FS} \in \mathbb{R}^{N \times P}$, where p are relevant features of the target class. In this research, we endeavored to find an optimal method to detect security violations in the IoT ecosystem; efficient, accurate, and general. What follows provides the rationale we used to find what we claim is optimal.

Feature selection endeavors to eliminate irrelevant and redundant features and to choose the most pertinent and important features. Furthermore, the FS process usually improves the general performance and data dimensionality, reducing the cost of classification and prediction by reducing the time complexity for building the model. On the other hand, applying all features in the IDS model includes several drawbacks: (i) the computational overhead is increased, and training and testing time are slower, (ii) storage requirements increase due to the large number of features, (iii) the error rate of the model increases because irrelevant features diminish the discriminating power of the relevant

features as well as reduce accuracy. FS approaches can be characterized into five categories: (i) filter-based, (ii) wrapper-based, (iii) embedded-based, (vi) hybrid-based, and (v) learning-based. The filter method gives weights to each feature (i.e., dimension), sorts them based on these weights, and then uses those subsets of features to train the model for either classification or prediction. Therefore, the process of feature selection is independent of the classification/prediction techniques. Numerous statistical measures are used in filtering methods to obtain feature subsets.

The model, using a particular FS method, initially uses all features but subsequently omits unrelated features to address the *curse of dimensionality* problem. This refining is designed to acquire the best subset of features based on statistical gauges such as information gain (IG) and gain ratio (GR), Pearson's correlation (PC) [28], chi-square (Chi12) [29], and mutual information (MI) [30–33]. The wrapper method is considered a black box technique [34]. Inductive algorithms are used to select feature subsets in the wrapper method, whereas filter methods are independent of the inductive algorithm. In addition, wrapper methods are more complex and expensive computationally than filter methods because they rely on iterating the learning systems (i.e., ML-derived models) several times until a subset of relevant features is reached. Moreover, the wrapper method accounts for the influence of the model performance on the feature subsets and strives to achieve high classification accuracy.

Embedded methods are incorporated with ML algorithms to select a feature subset during the learning process. The blending of feature selection approaches is used during the learning process to achieve advantages by improving classification, accuracy, and computational cost. Embedded methods can avoid retraining the model when the model needs to add a new feature to the subset. Concerning the structure of the embedded approach, the feature selection process is integrated with the classification algorithm and simultaneously performs feature selection such as random forest, LASSO (Least Absolute Shrinkage and Selection Operator), and L1 regularization [35]. Embedded methods are computationally less intensive than wrapper methods. However, they still have high computational complexity. Furthermore, the selected feature subset result depends on the chosen learning algorithm. Thus, embedded methods endeavor to find the best feature subset during model building by selecting each feature individually. Furthermore, they derive significant advantages in terms of model interaction, accuracy, fewer variables, and computational cost than previous approaches.

Information Gain (IG) [36] is one of the most widely used approaches in preparing features from a filter-based approach. That is, IG provides a classification ranking of all attributes (features) related to the target (class). Then a threshold is assigned to select several features according to the order obtained. Accordingly, a feature that strongly correlates with the target is considered a relevant feature and irrelevant (or redundant) otherwise. However, a weakness of the IG criterion is a bias favoring features with more values, especially when they are not more informative. Thus, IG between the feature in X and the variable (target) y is given here in Equation (1):

$$\mathrm{IG} = H(Y) - H(Y|X) = H(X) - H(X|Y) \tag{1}$$

where $H(x)$ is the entropy of x given y. The entropy of y is defined by Equation (2):

$$H(y) = -\sum_{y \in Y} p(y) \log_2(p(y)) \tag{2}$$

where $p(y)$ is the marginal probability of y on all values of Y. Note, Y is a finite set. Moreover, the conditional entropy of Y given the random variable X is shown in Equation (3):

$$H(Y|X) = -\sum_{x \in X} p(x) \sum_{y \in Y} (p(Y|X) \log_2(p(Y|X)) \tag{3}$$

where $p(y \mid x)$ is the conditional probability of y given x.

IG is a symmetrical measure such as IG (x, y) = IG (y, x), as shown in Equation (1).

The information gained about Y after observing X is equal to the information gained about X after observing Y.

Gain Ratio (GR) [37] is the non-symmetrical measure introduced to compensate for the bias of the IG attribute evaluation. The GR formula is given in Equation (4):

$$\mathrm{GR} = \mathrm{IG}/(H(X)) \quad (4)$$

Accordingly, to ensure a high-performing predictive model, we have applied two feature selection methods, i.e., IG and GR, over the collected dataset. The experiment has been conducted three times to extract various sets of features. The outcomes are reported in Table 1. Indeed, feature selection methods are essential for improving model performance by consuming fewer computing resources, accelerating the training process, and overcoming the overfitting/underfitting issues.

1.2. Our Contributions

The contributions of this paper include that we:

- Present a filter-based method to optimize the FS process using the IG and GR methods that use various techniques to obtain only the most essential features.
- Employ the concept of mathematical sets (intersection and union theory) to generate a hybrid feature selection approach (i.e., called *hybrid* here since we have combined two filter-based feature ranking approaches, IG and GR; to extract the minimum and maximum of the best relevant features). The proposed process consists of two feature selection modules. The first module uses the intersection rule to select the most relevant features from the former phase. The second module plays the same role as the first but instead uses the union rule. The result of these modules is to have the best relevant features selected, which are then fed to ML classifiers in the next phase for the ensemble and singular classifiers. In this way, our hybrid introduces a simple, practical in the context of IoT, and efficient yet effective methodology that requires less training time still better performance compared to other techniques.
- Employ diverse ML algorithms and ensemble ML algorithms with a majority voting to create an *intelligent* IDS scoring a maximum detection accuracy of 99.98% for our ML ensemble-based hybrid feature selection that employs (i) IMF: Intersection Mathematical set theory FS inspired by the intersection theory concept, and (ii) UMF: Union Mathematical set theory FS is inspired by the union theory concept. The method works in a systematic way that has not been published elsewhere in the literature to the best of our knowledge.
- Providing extensive experimental results to gain insights into the proposed approach as an effective and general IoT ecosystem IDS solution methodology.

1.3. Paper Organization

The rest of this paper is organized as follows: Section 2 reviews the recent state-of-the-art research in this subfield poised to secure the IoT ecosystem. Section 3 describes the proposed anomaly-based IDS. Section 4 discusses the experimental analysis and results. Section 5 concludes the paper.

2. Related Work

Significant and fruitful efforts have endeavored to address the security concerns of recent years for the IoT ecosystem. Several new IoT security technologies were established by pairing artificial intelligence techniques and cybersecurity virtues. Several promising state-of-the-art studies have been conducted for IoT security using machine learning (ML) and deep learning (DL) techniques [38–47]. However, only a few were developed by investigating the impact of using different feature selection approaches to improve prediction and classification accuracy. For instance, Albulayhi et al. [11] have proposed and implemented a new minimized redundancy discriminative feature selection (MRD-FS) technique to resolve the issue of redundant features. The discriminating features have been selected based on two criteria, i.e., representativeness and redundancy. Their model was evaluated utilizing the BoT-IoT dataset. Ambusaidi et al. [23] presented a flexible,

mutual information-based feature selection technique (FMIFS) that chooses the best features to enhance the classification algorithm. The proposed model was evaluated using three datasets (NSL-KDD, KDD Cup 99, and Kyoto 2006). The Least Square Support Vector Machine-based IDS (LSSVM-IDS) was used to measure performance. Ambusaidi et al. [23] showed 99.79% accuracy, 99.46% detection rate (DR), and 0.13% FPR over the KDD99 dataset. However, their employed datasets are not up-to-date (date back to 2009, 1999, and 2006 for NSL-KDD, KDD-Cup99, and Kyoto datasets, respectively) and do not fully represent the IoT cyberattacks.

Similarly, Amiri et al. [24] proposed a modified mutual information-based feature selection technique (MMIFS) applied with the SVM to improve the accuracy performance of the classification and to (highly) efficiently detect the various attack types. They demonstrated how high data dimensionality could be enhanced using the feature selection technique. Note, high dimensionality, even if applied to a high-quality ML approach, produces poor detection rate and accuracy performance. MMIFS can reduce features to only eight features (out of 41). For instance, MMIFS with SVM using only eight features, and DR achieved 86.46%. In the first phase, data normalization and reduction are applied by dividing every attribute (feature) value by its maximum value. In the next phase, feature selection is applied based on the imported training data. Further, MMIFS initially takes the feature set as the empty set. In more detail, it calculates the mutual information of the features concerning the class target and then picks the first feature with the maximum mutual information value.

Moreover, Lin et al. [48] proposed an approach integrating k-nearest neighbors with the k-means algorithms (KNN) based on feature extraction to select the best features and classify network attack types. Two-dimensional vectors are created. In the first phase, a clustering algorithm is applied to cluster the training dataset, determining the new feature value based on two distances. The first is between a current feature and its cluster center (centroid), and the second is between the current feature and its nearest neighbor. A new one-dimensional distance based on feature value represents each feature (attribute) in the training dataset. In the next phase, principal component analysis (PCA) is applied to select the relevant features and omit irrelevant ones. In a similar context, Khammassi et al. [49] introduced a wrapper technique for feature selection of their IDS. Their approach uses logistic regression and a genetic algorithm as an exhaustive search strategy for classification methods. Moreover, the decision tree (DT) classifier has been applied in their model, which has enhanced performance: an accuracy of 99.9% and false alarm rate (FAR) of 0.105% on the KDD Cup 99 dataset using only 18 features. They name their approach the genetic algorithm and logistic regression (GA-LR) wrapper approach. However, their utilized dataset is not up-to-date (KDD-Cup99 dataset dates back to 1999), consists of a large number of duplicate samples, and does not fully represent the IoT cyberattacks [40].

Another noticeable work is presented in [50], where the authors propose a feature reduction method using correlation-based methods and Information Gain (IG) to classify the network traffic into normal or attack (abnormal). However, the major disadvantage here is represented in the manual preprocessing performed over the preprocessing phase to fit the information-gain and correlation-based approaches.

In contrast, Sindhu et al. [51] proposed a wrapper approach to select relevant features and remove irrelevant features from the whole feature set to achieve higher detection accuracy using the *neurotree* method. They conducted their model as follows: (1) removing redundant features to make an unbiased detector composed of ML algorithms, (2) employing a wrapper-based feature selection algorithm to identify a suitable subset of features, and (3) combining neurotree with IDS to achieve better detection accuracy. These three phases of wrapper-based features selection have been used to achieve a lightweight IDS system and employ a *neural ensemble decision tree* iterative procedure to select features and optimize performance. A total of six decision tree classifiers for the proposed model are used: random tree, decision stump, naive Bayes' tree, C4.5, random forest, and repre-

sentative tree model have been performed to build the detection model of an anomalous network pattern.

Sung et al. [25] removed one feature at a time to represent their experiments on two selected ML algorithms (SVM and neural network). Then, they applied this process to the intrusion detection dataset of the Defense Advanced Research Projects Agency (DARPA-ID-1998 dataset) to evaluate their proposal. In terms of the five-class classification (target variables) in this dataset, it was found, experimentally, that using only 34 of the "most important" features, rather than the complete 41-feature set, resulted in a statistically insignificant change in performance (i.e., accuracy) for intrusion detection. While in [52], Li et al. presented a wrapper-based feature selection method to build a lightweight IDS (i.e., useful in the IoT ecosystem). They performed two strategies: the first strategy is a modified random mutation hill climbing (RMHC) as the search strategy and the second strategy is a modified linear SVM as an evaluation criterion. The proposal attempts to accelerate feature selection yielding reasonable detection rates which is generally the case.

Additionally, Peng et al. in [32] suggested a minimal-redundancy-maximal-relevance criterion (mRMR) for first-order incremental feature selection. This standard uses a feasible methodology for selecting features at a meager cost. Using three different classifiers, they have used maximal relevance standards to compare with their proposed approach. Their experiments showed that an mRMR feature selection could meaningfully improve the classification accuracy. The technique can be used in both continuous and discrete datasets. Ullah and Mahmoud [53] have presented anomaly detection techniques and characterized different attack categories. Moreover, they have generated a new IoT dataset (IoTID20). In [54], Qaddoura et al. presents an IDS proposal addressing the class imbalance that includes three stages (i.e., integrated clustering, classification, and oversampling techniques). Oversampling is used to tackle the lack of a minority class problem. However, they neglected to choose features carefully. In [55], Yang and Shami have proposed an optimized adaptive sliding windowing (OASW) approach to secure IoT data streams. However, they did not improve their feature selection approaches. Krishna et al. [56] have also discussed various supervised feature selection methods such as filter methods, sequential forward processing using three ML methods namely random forest (RF), SVM, and eXtreme gradient boosting (XGBoost).

Although previous studies [3,11,15–27] revealed that the detection accuracy of anomaly-based IDSs is considered the main challenge in the IoT ecosystem due to the constantly evolving nature of the IoT environment. However, we are convinced that most studies are not conducted to improve the feature selection approaches. Thus, this paper proposes a novel feature selection approach for machine learning-based IDS towards obtaining resilient performance within the diverse IoT ecosystem. In other words, our current paper presents a machine learning-based solution that uses a hybrid feature selection approach to attain a higher detection rate with a low false-positive rate. Table 1 summarizes the salient features of the works surveyed in the literature review.

Identified Literature Gaps and Open Challenges

Even though there are a large number of studies in this field, the reviewed state-of-art models indicate that the redundancy of features and high dimensionality are still open challenges. Herein, therefore, we propose an effective model for optimal feature extraction and reduction of training time complexity. Most studies focus on improving the IDS model to classify the result into the binary classification or the multi classification. In general, the IDS models that fits both attack, non-attack classes, and a high-dimensionality environment are still weak and incomplete. Existing anomaly detection models suffer from a high rate of false alarms. Currently, these studies suggest that any pattern which deviates from the normal pattern is an anomaly, even when this is not the case (i.e., a false indicator). This prediction error is due to the negative correlation in irrelevant features.

Table 1. Summary of different feature selection approaches.

Reference	Feature Selection Approach	Number of Features	Detection Model	Environment	Datasets	Performance Metrics
[23]	FMIFS	Candidate Features Selected	LS-SVM	IoT system	KDD Cup 99 d, NSL-KDD, Kyoto 2006+	DR, FPR, Accuracy
[24]	MMIFS	Candidate Features Selected	SVM	Discrete, Continuous environment	KDD Cup 99	Falsepositve rate (FPR), false detection rate (DR), Accuracy
[25]	-	Candidate Features Selected	SVM, ANN	Continuous environment	DARPA	Accurecy, False positive rate, False negative rate
[32]	mRMR, wrappers, Max, MaxRel	Candidate Features Selected	NB, SVM, LDA	Discrete and continuous environment	HDR, ARR, NCI, LYM	Lowest Error Rate
[48]	PCA	6 and 19	KNN with K-means, CANN	Continuous environment	KDD Cup 99	Accuracy, Detection rate, False alarm
[49]	Wrapper, GA, LR	15, 16, 17, 18, 20, 22	logistic regression, a genetic algorithm, decision tree	IoT system	KDD Cup 99	Accuracy, Recall, DR, FAR
[50]	Correlation (CR), (IG)	25	ANN	IoT system	NSL-KDD	Confusion matrix, Recall, Precision, FPR, TPR, Accuracy
[51]	Wrapper, GA	11, 9, 16, 20, 22, 36, 34, 41	random tree, decision stump, naive Bayes, C4.5, random forest, and representative tree	Continuous environment	KDD Cup 99	TP Rate, FP Rate, Precision, Recall, F-Measure
[52]	Wrapper, RHMC	Candidate Features Selected	SVM	IoT system	KDD Cup 99	False positive rate
[53]	Shapiro-Wilk	Candidate Features Selected	SVM, Gaussian NB, LDA, LR, Descion Tree, Random forest (RF), Ensemble	IoT system	IoTID20	Accurecy, presion, Recall, F score
[54]	-	-	SVM, Gradient Descent (SGD), LR, NB, SLFN, oversampling	IoT system	IoTID20	Accuracy, Precision, Recall, G-mean
[55]	-	-	lightGBM, OASW	IoT system	IoTID20, NSL-KDD	Accuracy, Precision, Recall

Thus, based on our literature survey and the identified gaps, our study here investigates and proposes an (more) intelligent IDS system that treats and removes redundant features from the dataset in an earlier phase of the process. We also study the effects of redundant features on the characterization of the general representation of the data. Accordingly, this study's outcome has both reduced the higher false-alarm rates compared to those found in the literature, as well as reduced training time complexity through dimensionality reduction. Key to this study is the application of a unique feature selection method that combines two entropy-based techniques (IG and GR) to select the best features.

3. Proposed Anomaly-Based IDS for IoT Ecosystem

Identifying and selecting relevant features in the dataset has become crucial to improving ML model performance, especially for anomaly-based IDS. To address the challenge of improving an anomaly-based IDS in the IoT ecosystem, we deal with node data attributes to identify relevant and redundant features. Redundant features affect the models' performance and make those models less reliable [11,28,32]. We assume that the current FS techniques do not always guarantee the best relevant features or eliminate redundant

features. Therefore, we've aimed to find new strategies in dealing with the discovery of useful features and concurrently the removal of unsupportive features. Existing methods may be easily implemented in the practical sense, but at the same time, they can and do consume lots of resources.

To address the selection of relevant features in the IoT ecosystem, this paper defines a new hybrid feature selection mechanism, as described above, utilizing the combination of two entropy-based mechanisms [57] to extract the best features, namely: information gain (IG) and gain ratio (GR). These two techniques are filter-based approaches that use a ranking to score each feature. Given the score, we can select the most relevant features and, conversely, omit irrelevant (i.e., redundant) features from the feature vectors. We have conducted numerous experiments using these approaches by tuning the number of features for each training session.

We applied these two IG/GR approaches independently to extract top-ranked features from the two datasets in phase 2A of Figure 1. The top-ranked features are divided into the following categories: the top 60 ranked features in the first implementation and the top 20 ranked features in the second implementation from the whole IoTID20 dataset. In NSL-KDD, we have extracted the top 20 ranked features and all features. In the first dataset (IoTID20), we have performed the IG and GR approaches four times to extract four groups of relevant features. The first two groups consist of the top 20 ranked features, and the other two groups consist of the top 60 ranked features from the whole dataset separately. Whereas, in the second dataset (i.e., NSL-KDD), we have only one group of top 20 ranked features because we used all features in the second group of the experiment. Each approach (IG, GR) has extracted two groups of best features (60 features and 20 features) from the whole dataset. Each approach was executed two times separately. The two groups of the top 20 ranked features have been sent to Phase 2B of Figure 1 to obtain the final best relevant features. For the other groups (i.e., top 60 ranked features of IoTID20 and all features of NSL-KDD), these are sent to phase 3 directly. The rationale behind using the top 60 ranked versus all the features (in the NSL-KDD case) is for comparison purposes with our proposed hybrid model that we claim extracts the lowest optimum number of features.

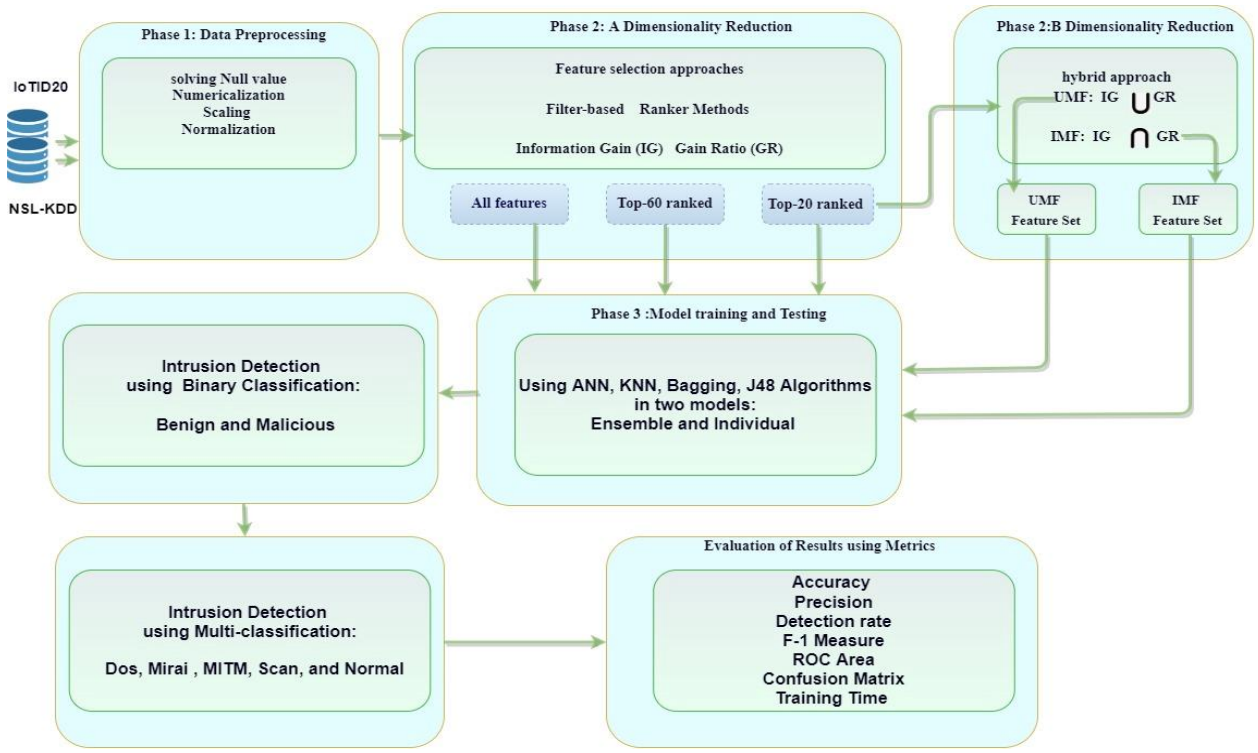

Figure 1. The proposed approach for machine learning-based intrusion detection using hybrid feature selection approach.

Subsequently, the two groups of the top 60 ranked features, hereby referred to as Top60 sets, were used to train ANN, kNN, Bagging, and J48 modules of our proposed hybrid approach. The same two groups were also used to train the ensemble learning technique, which is specified in Phase 3 of Figure 1. We do not pass the Top60 groups on to phase 2B. This reveals the quality of the effect of adding the concept of mathematical set theory on the top 20 ranked features to compare with the top 60 ranked features and the top 20 ranked features. To determine the best and most relevant features to use in practice, the top 20 selected features belonging to IG and the top 20 selected features belonging to GR are combined using the two mathematical set theories (i.e., UMF/IMF) to produce the hybrid feature selection (see Equations (6) and (8)) method. These combined features are then carried over to the model training in Phase 3.

3.1. IoT Network Traffic Apposite

The Internet of Things (IoT) consists of particular limitations such as connectivity limitation, computational capacity, and energy budget [58]. These peculiar characteristics makes it significantly different from other continuous environments. The data are created from a wide range of sources such as different sensors and other types of internet-connected devices. It is usually expressed as multimodal or heterogeneous. IoT data are basically big data because of their volume, velocity, variety, and veracity [59]. Herein, we are obliged to overcome these issues. The basic steps involve three main phases, including (i) data preprocessing, (ii) dimensionality reduction (feature selection) with proposing the hybrid feature selection approach (improving the choice of relevant features), and (iii) ensemble ML algorithms (model training), as shown in Figure 1. Data typically must be refined in the early stage before being used to train the system, *especially when starting with a dataset built from a heterogeneous IoT ecosystem*. Initially, data preprocessing is performed to prepare the data in a proper format for the learning phase by applying several consecutive preprocessing operations over the dataset such as data scaling, data conversion, removing unwanted/invalid data, and fixing missing values. Our model has been embedded into the ensemble framework, as shown in Figure 1, to provide more consistent and reproducible results in such a dynamic environment. The main phases are detailed in the following subsections.

3.2. Data Pre-Processing

According to Figure 1, this phase comprises steps that are needed to improve the classification results. The process proceeds as follows: (i) duplicate instances are removed, (ii) missing values, NaN are replaced by zero, (iii) non-numeric values are converted to numeric values, (iv) the values are scaled between (0, 1). In this way, phase 1 achieves the consistency needed for the feature selection step. Thus, the filtered dataset obtained from the preprocessing data phase 1 is used for the feature selection phase and further classification into normal and attack classes as the final decision.

3.3. Dimensionality Reduction

This step represents phase 2 (A and B) of Figure 1. Our hybrid feature selection approach utilizes a filter-based approach that includes two selection modules. The first uses the intersection rule, while the second uses the union rule. As mentioned above, both the IG and GR attribute selectors are used to reduce feature vectors' high dimensionality and accordingly select the best features (Note, IG is called "info gain attribute evaluator" and GR is called "gain ratio attribute evaluator" in Weka). We choose IG and GR because they are widely used in many different domains. Moreover, they present a similar knowledge bias for selecting features to match our mathematical set theory requirements. Both are considered as one of the best and most popular feature selection approaches. In contrast, different ranking approaches require more memory to load the computational result. Feature selection plays a significate role in improving a classifier/predictive system because it reduces time complexity and enhances model performance measures [11,60]. To design

our hybrid feature selection approach (IMF and UMF as introduced in the definition in Section 1.2 including the details in Section 3.3), we utilize the concept of mathematical set theory to "smooth" the selection of relevant features in a reliable way for building an Anomaly-based IDS model (Note, features with scores close to "1" are deemed to positively affect model performance (more relevant), while "0" indicates a negative impact (meaning irrelevant and/or redundant).

Using IG and GR to produce two groups of relevant features, is the first attempt to extract the best most relevant features in our process per Figure 1. Moreover, in our experiments, the first group consisted of 60 features, and the second consisted of 20 derived from the whole IoTID20 dataset. Furthermore, there is zero correlation between the two groups because all group productions were run independently. In other words, starting with the same full set of features, both IG and GR approaches were used to extract these features one at a time in an iterative fashion resulting in the two groups of 20 and 60. Thus, achieving the first step of phase 2 in Figure 1 (dimensionality reduction) discovering relevant features and eliminating irrelevant features.

3.4. Validation Phase

To validate the steps described above, the top 20 features groups coming from IG and GR are only sent to phase 2B of Figure 1 (subset feature selection, to determine the best features based on IG and GR used together). In contrast, for comparison purposes, we did not pass the top 60 features group to this phase 2B. That is, the top 60 features groups are used to compare with our other results (see Table 2 for the comparison). In phase 2B Figure 1, we establish our hybrid approach utilizing set theory to merge the output of IG with GR and produce a new group of features that represent the final best relevant features. In the end, these features that we obtained in phase 2B, and the other features (the top 60 features and the top 20 features) that are obtained in phase 2A are now used to train the ML models, as shown in Phase 3 of Figure 1.

Table 2. Results from the ranking features process applied to the IoTID20 dataset.

Approach	# Features	Extracted Features
IG	20	7, 1, 5, 3, 8, 82, 25, 22, 73, 71, 12, 69, 11, 60, 15, 80, 13, 59, 23, 44: 20
IG	60	7, 1, 5, 3, 8, 82, 25, 22, 73, 71, 12, 69, 11, 60, 15, 80, 13, 59, 23, 44, 14, 47, 4, 61, 19, 83, 18, 26, 32, 42, 46, 17, 45, 43, 35, 41, 48, 49, 54, 2, 33, 21, 16, 24, 81, 27, 6, 36, 28, 30, 31, 34, 20, 29, 70, 10, 51, 74, 9, 68
GR	20	16, 34, 5, 3, 54, 8, 22, 13, 35, 15, 60, 73, 14, 49, 48, 69, 11, 6, 25, 32
GR	60	16, 34, 5, 3, 54, 8, 22, 13, 35, 15, 60, 73, 14, 49, 48, 69, 11, 6, 25, 32, 82, 1, 83, 12, 71, 7, 26, 23, 4, 80, 42, 19, 61, 59, 47, 33, 18, 27, 44, 46, 81, 28, 24, 17, 41, 36, 30, 45, 43, 20, 51, 29, 31, 76, 79, 78, 2, 21, 77, 10
IG ∩ GR	11	5, 8, 25, 22, 73, 69, 11, 60, 15, 13, 3
IG ∪ GR	28	7, 1, 5, 3, 82, 25, 22, 73, 71, 12, 69, 11, 60, 15, 80, 13, 59, 23, 44, 16, 34, 54, 35, 14, 49, 48, 6, 32

3.5. Hybrid Approach

In phase 2B of Figure 1, the hybrid approach uses the concept of mathematical set theory to select the minimal set of the best features from predetermined selected features based on IG and GR. We can define the four mathematical equations as follows:

$$Intersection\ rule\ (M \rightarrow g1 \cap g2) \tag{5}$$

$$IMFeatues\ (IMF) = \text{IG} \cap \text{GR} \tag{6}$$

$$Union\ rule\ (U \rightarrow g1 \cup g2) \tag{7}$$

$$UMFeatues\ (UMF) = \text{IG} \cup \text{GR} \tag{8}$$

where $g1$, $g2$ are a subset of the whole feature set (F) where $g1$, $g2 \ll F$. For instance, $g1$ holds the relevant features extracted using IG, while $g2$ holds the relevant features extracted using GR. Hence, the union (U) chooses all elements (features) that are located either in $g1$

or $g2$, whereas the intersection (M) selects the elements (features) that are found only in both sets ($g1$ and $g2$).

In phase 2A in Figure 1, we demonstrate feature selection via IG, GR, and filter-based ranking methods. The output of phase 2A is the extraction of two sets of features: top 60 and top 20. These two feature sets are generated independently, i.e., feature ranking from the top 60 is not carried over to the top 20. In phase 2B in Figure 1, we demonstrate the hybrid approach by implementing the intersection and union set theory rules. These rules are labeled IMF and UMF, respectively, for the intersection and union operations. The IMF and UMF rules are described by Equations (5)–(8). The hybrid approach is a manual process to produce IMF and UMF approaches (11 and 28 features from the IoTID20 dataset, and 15 and 25 features from the NSL-KDD dataset) from the top 20 ranked feature set produced by Phase 2A. Phases 2A and 2B together produce six different feature selection sets, which are: top 60 (via IG), top 60 (via GR), top 20 (via IG), top 20 (via GR), 11 IMF and 28 UMF. Due to the use of both traditional and hybrid feature selection approaches, we can, to increase the efficiency of our feature selection process, reduce the computational costs of the learning process (i.e., training and testing workflows). Corresponding features for each of the six feature sets are listed in Table 2. Also, Table 3 shows the results of ranking features from the NSL-KDD dataset.

Table 3. Results of ranking feature NSL-KDD.

Approach	# Features	Extracted Features
-	41	All Feaure
IG	20	1, 3, 4, 5, 6, 12, 23, 25, 26, 29, 30, 31, 33, 34, 35, 36, 37, 38, 39, 40
GR	20	2, 3, 4, 5, 6, 8, 10, 11, 12, 13, 23, 25, 26, 29, 30, 33, 34, 37, 38, 39
IG ∩ GR	15	3, 4, 5, 6, 12, 23, 25, 26, 29, 30, 33, 34, 37, 38, 39

3.6. Model Training

In this phase, our machine learning models are trained. We use two different ML categories to improve the performance of the anomaly-IDS model, namely individual classifiers, and ensemble classifiers with majority voting. The models utilize the proposed hybrid feature selection technique at the training and the testing phases to generate the final set of selected features. The IoTID20 dataset containing only these features is then used to train the ML models. As we have stated above, these subsets of features are: 60% ranked features of IG, 60% ranked features of GR, 20% ranked features of IG, 20% ranked features of GR, IMF (11 features), and UMF (28 features). Specifically, the proposed anomaly-based IDS uses the following ML techniques:

- ANN (*called* MLP in the Weka tool) stands for artificial neural network and is a set of neuron nodes arranged in hidden layers designed to recognize patterns imitating the human brain.
- kNN (*called* IBk or instance-based learner in the Weka Tool) stands for the k-nearest neighbor and is a simplified supervised machine learning algorithm that can be used to address classification or regression tasks. kNN depends on deriving the distance metric between a predicted sample and k-stored samples to provide the classification decision based on the maximum value of the distance of nearest neighbors.
- *C4.5* (*called* J48 in the Weka tool) is an ML algorithm that is used to generate a decision tree. C4.5 is an extension of the ID3 (iterative dichotomies 3) algorithm. It implements the model to consist of a root node, branch nodes, and leaf nodes. C4.5 has some advantages over ID3, including handling both continuous and discrete attributes, handling missing values without human intervention, handling attributes with various costs (i.e., loss function), and providing an optional task to prune the tree after creation.
- Bagging (bootstrap aggregating) fits well for an imbalanced dataset. It reduces variance which helps to avoid overfitting. It can be applied to a different domain. Bagging is a type of ensemble algorithm.

- Ensemble Method: The ensemble method aims to select the final best decision by using majority voting on the outputs from individual classifiers (ANN, kNN, C4.5, and Bagging). With six feature-reduced sets, we established six ensemble methods to construct the final anomaly-based IDS.

4. Experimental Results and Discussion

In this section, we present and discuss our experimental results using different scenarios and evaluation metrics for the various machine learning models and features selection techniques that we choose as most relevant for the IoT ecosystem.

4.1. The Datasets

We have conducted the experiment using the IoTID20 [53] and the NSL-KDD [61] datasets to evaluate the performance of the hybrid feature selection approach. The IoTID20 dataset consists of various types of IoT attacks (i.e., DDoS, DoS, Mirai, ARP Spoofing, etc.) as well as normal (benign traffic). The IoTID20 dataset was collected from the IoT ecosystem of a smart home. The smart home was designed to incorporate multiple interconnected components including AI Speakers (SKTNGU), Wi-Fi cameras (EZVIZ), laptops, smartphones, tablets, a wireless access point (Wi-Fi), and a Wi-Fi router. The cameras and AI speakers were represented as the IoT victim equipment, and the other equipment were represented as the attacking devices. The testbed has been implemented to simulate different actual attacks in the IoT ecosystem using the Network Mapper (Nmap) tool. Figure 2 represents the testbed environment where the IoTID20 dataset [62] was generated and collected. Figure 3 shows a taxonomy of the dataset and the number of instances for each target of the dataset. The IoTID20 dataset has 86 features and contains 625,784 instances.

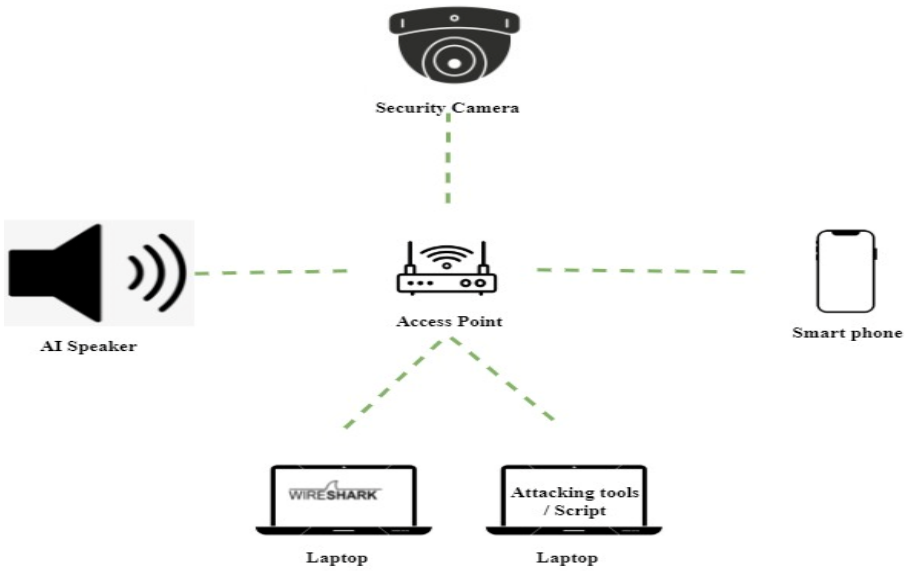

Figure 2. The IoTID20 dataset testbed environment.

NSL-KDD dataset is the second dataset that we have utilized for the validation of this work. It is the improvement of the KDD99 by removing the redundant and duplicate data from the original KDD99 dataset. It contains 41 features (32 continuous and 9 nominal attributes) to describe each activity in an IoT system and the targets (classes), converting five types of Normal, Probe, DoS, R2L, U2R. The advantages of NSL-KDD are focused on minimizing the level of difficulty that exists in the original dataset KDDcup99. However, it has the same problems regarding the real network representation. The NSL-KDD dataset is widely used in realm of intrusion detection, and other related areas. The description of NSL-KDD dataset is shown in Figure 4.

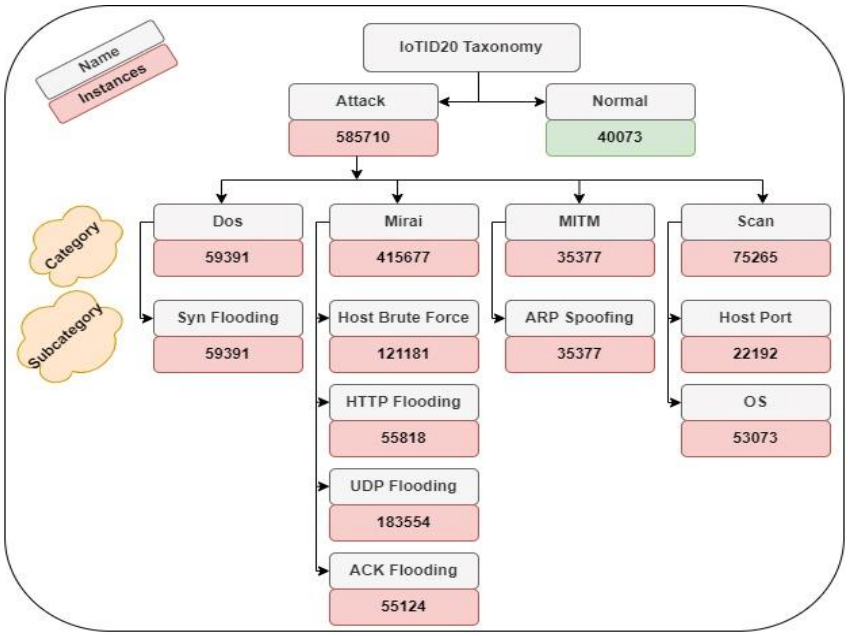

Figure 3. IoTID20 dataset taxonomy.

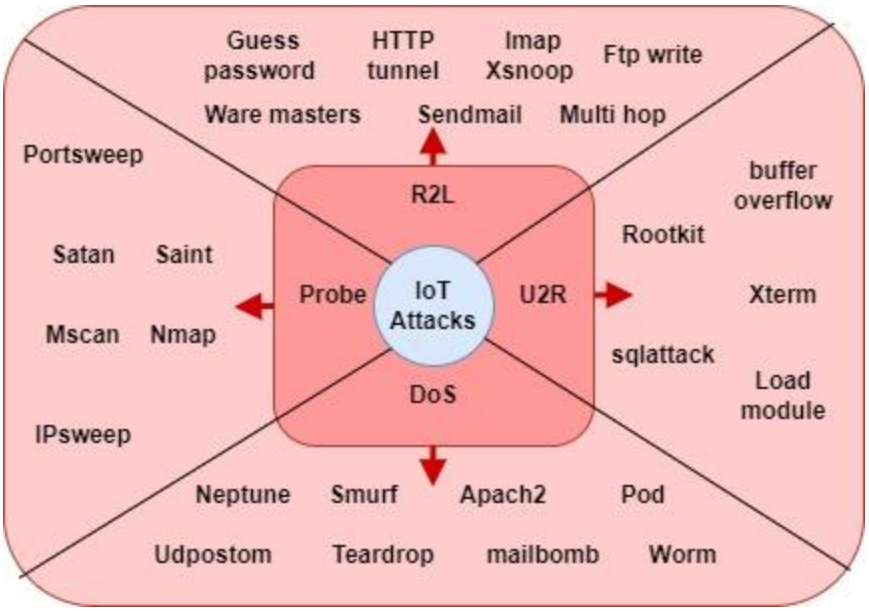

Figure 4. NSL-KDD dataset description.

4.2. Evaluation Metrics

To evaluate the performance of the proposed IDS, we have used eight evaluation metrics including:

- The confusion matrix that is used to report the number of correctly predicted samples (represented in two factors TP and TN) and the number of incorrectly predicted samples (represented as FP and FN). The predicted values can be described as positive and negative values, whereas the actual values can be described as true and false values. The two-class confusion matrix is shown in Figure 5.
- The accuracy indicates the model's power to classify the result of benign instances correctly, as shown in Equation (9):

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (9)$$

		Actual Values	
		Positive (1) (attack)	**Negative (0) (Normal)**
Predicted Values	**Positive (1) (attack)**	**True Positive (TP)**	**False Positive (FP)**
	Negative (0) (Normal)	**False Negative (FN)**	**True Negative (TN)**

Figure 5. Confusion matrix.

- The recall is known as sensitivity or detection rate and indicates the model's power to correctly identify attacks (the actual values) as shown in Equation (10):

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

- The precision indicates the model's power to be correctly predictive, which means how many positive predictions (attacks) are predicted correctly, as shown in Equation (11):

$$Precision = \frac{TP}{TP + FP} \tag{11}$$

- The F1-measure plays a trade-off between recall and precision in all instances to improve contradiction of recall and precision, as shown in Equation (12):

$$F1 - Measure = \frac{2 * Recall * Precision}{Recall + Precision} \tag{12}$$

- The False Positive Rate (FPR) indicates the model's power to calculate the percentage of misclassified attack instances as normal. This is represented as follows in Equation (13):

$$FPR = \frac{FP}{FP + TN} \tag{13}$$

- Receiver operating characteristics area under the curve (ROC AUC) which is used as a measure of the usefulness of a test in general at various threshold settings. The greater area the more useful test (ranges from 0.0% to 100%).
- Training Time is the amount of time duration measured in seconds (s) that the ML model takes to train the model using a specific dataset.

4.3. Experimental Results and Analysis

In this research, to develop and validate the proposed system, distinct computing specifications/tools have been used and configured. At the software level, we have used the Weka Tool [63] and Python running on a 64-bit Microsoft Windows 10 operating system, to implement and evaluate our hybrid feature selection scheme as well as to conduct our experiments with various machine learning methods. We have used 10-fold cross-validation to train/validate each model. At the hardware level, we implemented and evaluated our models using a high-performance computing platform with an Intel ®Xeon® CPU E3-1241 v3 @3.5GHz with 16 GB of memory and a 4 GB GPU (graphical processing unit). Moreover, we have downloaded the IoTID20 dataset from [53]. Accordingly, we have evaluated the proposed model's performance using relevant features obtained through the concept of mathematical set theory. In addition, we made a comparison with various relevant features that are obtained through using existing ranking methods (IG, GR).

Initially, Table 2 provides the experimental evaluation outcomes for the five different machine learning-based IDS models utilizing six feature selection approaches (namely IG_60, GR_60, IG_20, GR_20, IMF_11, and UMF_28). The comparison in the table considers seven performance indicators including detection accuracy, false-positive rate (FPR),

detection precision, detection recall, f1-measure, the area under the curve (AUC), and the training time duration (in seconds). Note that feature selection approaches are abbreviated based on the number of selected features for each approach; for example, IG_60 stands for information Gain with 60 features.

According to the comparison results reported in the table (i.e., Table 4), several performance metrics using the ensemble classifier and individual classifier with a different number of relevant features were selected using IG, GR, IMF, and UMF filter-based feature selection techniques. Consequently, the reported results show that the Bagging classifier has achieved the best performance factors over other ML-based approaches scoring an average performance indication of higher than 99.88% in terms of accuracy, recall, precision, F1-Score, and ROC, for all feature selection strategies. Conversely, the lowest performance results belong to the ANN-based-IDS which requires more features, is affected by irrelevant features, and negatively impacted by the ensemble classifier. Whereas, our ensemble learning-based IDS has achieved better, more promising results when removing the redundant (irrelevant) features.

Table 4. Performance evaluation results for the different feature selection approaches concerning five ML models by using IoTID20.

Metrics	Classifier	IG_60	GR_60	IG_20	GR_20	IMF_11	UMF_28
Accuracy (%)	ANN	99.39	98.83	99.57	99.08	99.07	99.07
	C4.5	99.88	99.85	99.80	99.90	99.80	99.91
	Bagging	99.89	99.88	99.90	99.89	99.90	99.91
	kNN	99.65	99.65	99.64	99.70	99.80	99.66
	Ensemble	99.81	99.74	99.73	99.56	99.98	99.98
False Positive Rate (%)	ANN	0.076	0.113	0.157	0.148	0.134	0.134
	C4.5	0.014	0.016	0.013	0.015	0.013	0.011
	Bagging	0.013	0.015	0.013	0.014	0.013	0.012
	kNN	0.030	0.030	0.024	0.023	0.015	0.027
	Ensemble	0.027	0.038	0.039	0.040	0.008	0.008
Precision (%)	ANN	99.40	98.80	99.60	99.10	99.10	99.10
	C4.5	99.90	99.90	99.90	99.90	99.90	99.90
	Bagging	99.90	99.90	99.90	99.90	99.90	99.90
	kNN	99.70	99.60	99.60	99.70	99.80	99.70
	Ensemble	99.80	99.70	99.70	99.60	99.90	99.90
Recall (%)	ANN	99.40	98.80	99.60	99.10	99.10	99.10
	C4.5	99.90	99.90	99.90	99.90	99.90	99.90
	Bagging	99.90	99.90	99.90	99.90	99.90	99.90
	kNN	99.70	99.70	99.60	99.70	99.80	99.70
	Ensemble	99.80	99.70	99.70	99.60	99.90	99.90
F1-Measure (%)	ANN	99.40	98.80	99.60	99.00	99.00	99.00
	C4.5	99.90	99.90	99.90	99.90	99.90	99.90
	Bagging	99.90	99.90	99.90	99.90	99.90	99.90
	kNN	99.70	99.70	99.60	99.70	99.80	99.70
	Ensemble	99.80	99.70	99.70	99.60	99.90	99.90
AUC (%)	ANN	98.40	98.60	98.20	93.90	90.80	90.80
	C4.5	99.60	99.50	99.60	99.60	99.60	99.80
	Bagging	99.90	99.90	99.90	99.90	99.90	99.90
	kNN	97.10	98.60	98.60	99.20	99.50	99.00
	Ensemble	98.50	98.00	97.90	96.50	99.90	99.90
Training Time (s)	ANN	870.23	920.53	295.16	310.50	42.40	42.40
	C4.5	350.19	360.61	150.15	153.72	44.15	44.18
	Bagging	360.22	370.18	155.28	154.99	50.28	50.28
	kNN	240.02	240.01	80.05	80.03	30.02	30.03
	Ensemble	919.91	940.07	309.86	322.40	56.75	56.75

The majority voting approach of an ensemble classifier takes all individual classifiers into account. Unfortunately, the well-known overhead costs can be observed, namely that model training time increases with the number of features for all ML-based IDSs. Conversely, the ensemble classifier using our proposed feature selection approach (IMF and UMF) has achieved higher performance outcomes than individual classifiers can, for

the same features. The difference is due to the IMF and UMF combination that can actually select the best (or better) features. The result of using our IMF/UMF ensemble learning classifier will be discussed further in the coming paragraphs. We have reproduced our experiments on another benchmark dataset, namely the NSL-KDD, for further validation, of our proposed hybrid method. The comparison from the results of these NSL-KDD experiments are reported in the Table 5. We can observe that the results extracted from the NSL-KDD dataset are similar in behavior to the results we obtained through the IoTID20 dataset. To improve the quality and efficiency of the model performance, we again found it necessary to consider improving the extraction of relevant features and eliminate irrelevant redundant features. These results give an indication that the current feature selection approaches are not able to extract the best relevant features and eliminate irrelevant feature; thus, we come to a trade-off between the two best entropy feature selection approaches to improve the lack in each.

Table 5. Performance evaluation results for the different feature selection approaches with respect to five ML models using NSL-KDD.

Metrics	Classifier	All Features	IG_20	GR_20	IMF_15	UMF_25
Accuracy (%)	ANN	99.37	97.60	96.56	97.69	97.79
	C4.5	99.40	99.59	99.44	99.30	99.70
	Bagging	99.39	99.79	99.83	99.79	99.79
	kNN	99.20	99.39	98.90	99.20	99.68
	Ensemble	99.66	99.56	99.56	99.80	99.84
False Positive Rate (%)	ANN	0.007	0.025	0.038	0.025	0.024
	C4.5	0.006	0.004	0.006	0.007	0.003
	Bagging	0.006	0.002	0.002	0.002	0.002
	kNN	0.008	0.006	0.011	0.008	0.003
	Ensemble	0.004	0.005	0.005	0.002	0.002
Precision (%)	ANN	99.4	97.6	96.7	97.7	97.8
	C4.5	99.4	99.6	99.4	99.2	99.7
	Bagging	99.4	99.8	99.8	99.8	99.8
	kNN	99.2	99.4	98.9	99.3	99.7
	Ensemble	99.7	99.6	99.6	99.8	99.8
Recall (%)	ANN	99.4	97.6	96.6	97.7	97.8
	C4.5	99.4	99.6	99.4	99.3	99.7
	Bagging	99.4	99.7	99.8	99.8	99.8
	kNN	99.2	99.4	98.9	99.2	99.7
	Ensemble	99.7	99.6	99.6	99.8	99.8
F1-Measure (%)	ANN	99.4	97.6	96.6	97.7	97.8
	C4.5	99.4	99.6	98.9	99.3	99.4
	Bagging	98.8	99.8	99.8	99.8	99.7
	kNN	99.2	98.8	98.9	99.2	99.4
	Ensemble	99.7	99.6	99.1	99.8	99.8
AUC (%)	ANN	99.8	99.0	98.8	97.9	98.6
	C4.5	99.6	99.8	99.8	99.8	99.8
	Bagging	99.2	100	100	100	100
	kNN	99.3	99.2	98.9	99.4	99.8
	Ensemble	99.6	99.6	100	1	1.00
Training Time (s)	ANN	814	300	280	90	210
	C4.5	180	200	180	40	110
	Bagging	190	160	160	50	150
	kNN	150	90	90	40	90
	Ensemble	550	350	285	60	200

Additionally, Figure 6, along with Table 6 using the IoTID20 dataset and Figure 7, along with Table 7 using the NLS-KDD dataset, summarizes the empirical outcomes of our model via different features selection methods. As can be clearly noticed, UMF and IMF-based IDSs exhibit better performance for all performance measures (i.e., accuracy, precision, recall, f1-measure, ROC area) over the other selected related studies. This result is attributed to the concept of set theory (intersection and union) that aims to choose the best features. A feature that is present in both approaches (IG and GR) or at least one of

them indicates this feature is more relevant while at the same time, non-redundant. To sum things up, dimensionality reduction approaches improve the performance classification, but we must be assured that only relevant features are selected, and thus we offer our study here as an example of this claim.

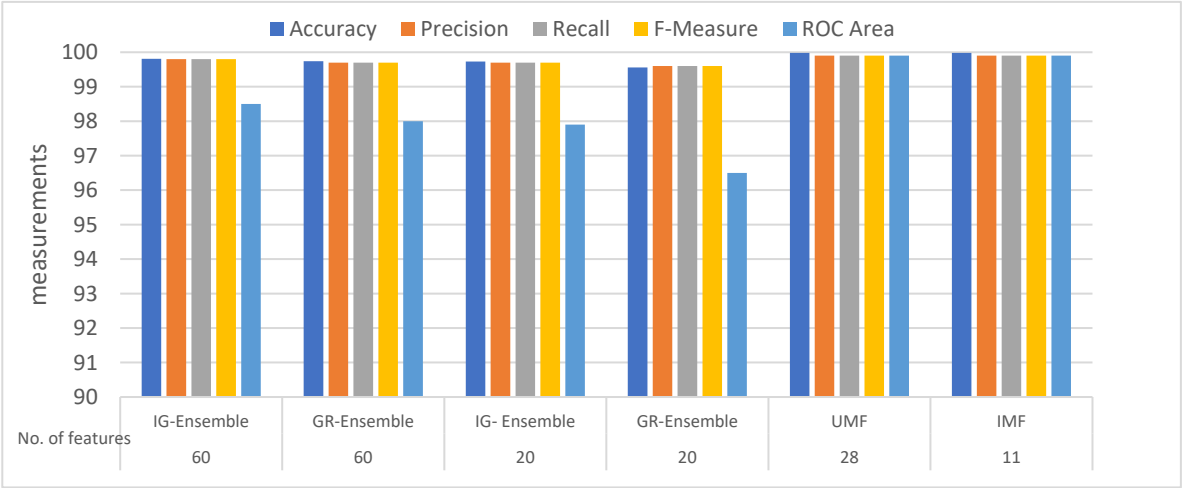

Figure 6. Graphical representation of performance metrics of IoTIDS20 using ensemble with various selected feature sets.

Table 6. Performance metrics of IoTID20 using ensemble with various selected features.

Feature	Approach	Accuracy (%)	Precision (%)	Recall (%)	F1-Measure (%)	ROC Area (%)
60	IG-Ensemble	99.81	99.80	99.80	99.80	98.50
60	GR-Ensemble	99.74	99.70	99.70	99.70	98.00
20	IG-Ensemble	99.73	99.70	99.70	99.70	97.90
20	GR-Ensemble	99.56	99.60	99.60	99.60	96.50
28	UMF	99.98	99.90	99.90	99.90	99.90
11	IMF	99.98	99.90	99.90	99.90	99.90

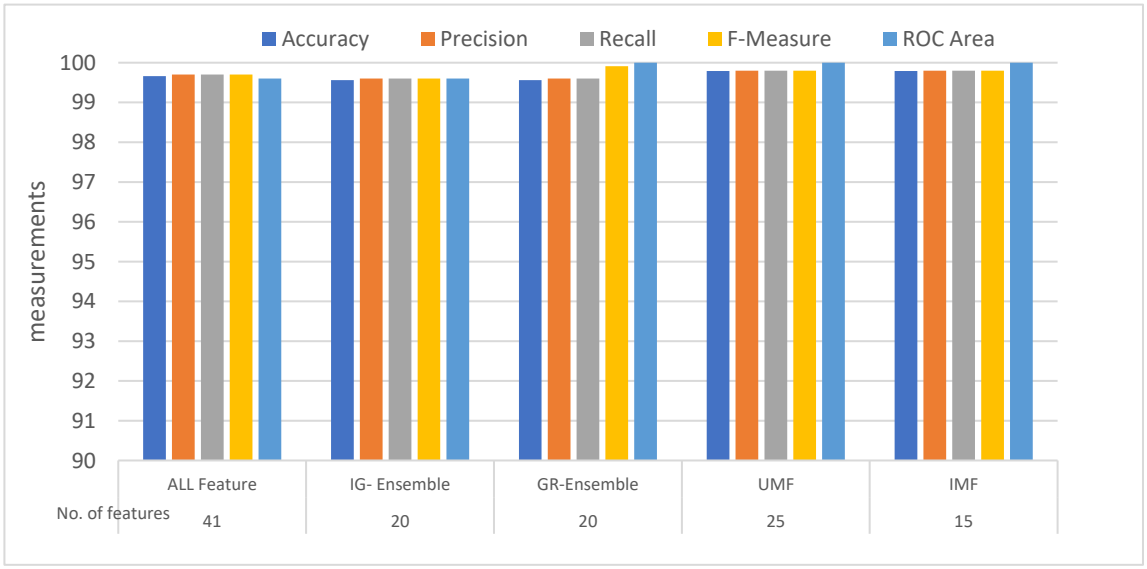

Figure 7. Graphical representation of performance metrics of NSL-KDD using ensemble with various selected features.

Table 7. Performance metrics of NSL-KDD using ensemble with various selected features.

Feature	Approach	Accuracy (%)	Precision (%)	Recall (%)	F1-Measure (%)	ROC Area (%)
41	ALL Feature	99.66	99.7	99.7	99.7	99.6
20	IG-Ensemble	99.56	99.6	99.6	99.6	99.6
20	GR-Ensemble	99.56	99.6	99.6	99.91	100
25	UMF	99.80	99.8	99.8	99.8	100
15	IMF	99.80	99.8	99.8	99.8	100

Figure 8 shows the report of the confusion matrix of our proposed binary classification using the NSL-KDD benchmark dataset after the training / testing phase of five ML classifiers. The diagonal line of each confusion matrix reports the number of correctly classified instances, and reverse diagonal line reports the number of incorrectly classified instances. For instance, Figure 8 (box 1) shows the confusion matrix of the ANN with all features (41) considering attack or non-attack target. The 67,042 instances are classified correctly as an abnormal class whereas the 310 instances are classified incorrectly as an abnormal class. The 482 instances are classified incorrectly as a normal (non-attack) class whereas the 58,145 instances are classified correctly as a normal class. In regards to this particular experiment, using NSL-KDD with five ML classifiers, the confusion matrices demonstrate that our proposed feature selection approach (IMF and UMF) with any ML classifier is the most suitable model.

(1)-All Features ANN			(6)-IG-20 ANN			(11)-GR-20 ANN			(16)-IMF ANN			(21)-UMF ANN		
Abnormal	67042	310	Abnormal	66315	1028	Abnormal	66667	676	Abnormal	66598	745	Abnormal	66631	712
Normal	482	58145	Normal	1985	56645	Normal	3652	54978	Normal	2164	56466	Normal	2062	56568
	Abnormal	Normal		Abnormal	Normal		Abnormal	Normal		Abnormal	Normal		Abnormal	Normal
(2)-All Features C4.5			(7)-IG-20 C4.5			(12)-GR-20 C4.5			(17)-IMF C4.5			(22)-UMF C4.5		
Abnormal	67075	268	Abnormal	67125	218	Abnormal	67023	320	Abnormal	67075	268	Abnormal	67155	188
Normal	479	58151	Normal	293	58337	Normal	561	58069	Normal	479	58151	Normal	188	58442
	Abnormal	Normal		Abnormal	Normal		Abnormal	Normal		Abnormal	Normal		Abnormal	Normal
(3)-All Features Bagging			(8)-IG-20 Bagging			(13)-GR-20 Bagging			(18)-IMF Bagging			(23)-UMF Bagging		
Abnormal	66937	406	Abnormal	67245	98	Abnormal	67239	140	Abnormal	67257	86	Abnormal	67125	218
Normal	367	58263	Normal	163	58467	Normal	140	58490	Normal	183	58447	Normal	293	58337
	Abnormal	Normal		Abnormal	Normal		Abnormal	Normal		Abnormal	Normal		Abnormal	Normal
(4)-All Features kNN			(9)-IG-20 kNN			(14)-GR-20 kNN			(19)-IMF kNN			(24)-UMF kNN		
Abnormal	66852	491	Abnormal	66937	406	Abnormal	66804	539	Abnormal	66852	491	Abnormal	67167	176
Normal	498	58132	Normal	367	58263	Normal	474	58156	Normal	498	58132	Normal	218	58412
	Abnormal	Normal		Abnormal	Normal		Abnormal	Normal		Abnormal	Normal		Abnormal	Normal
(5)-ALL Feature Ensemble			(10)-IG Ensemble			(15)-GR Ensemble			(20)-IMF Ensemble			(25)-UMF Ensemble		
Abnormal	67190	153	Abnormal	67157	186	Abnormal	67155	188	Abnormal	67259	84	Abnormal	67259	84
Normal	279	58351	Normal	364	58266	Normal	356	58274	Normal	118	58512	Normal	118	58512
	Abnormal	Normal		Abnormal	Normal		Abnormal	Normal		Abnormal	Normal		Abnormal	Normal

Figure 8. Confusion matrix yielded by the five various models (2-classes) of NSL-KDD dataset.

Figure 9 shows the confusion matrix of the proposed model for multi-classification challenges using the five various ML classifiers with four FS approaches including our hybrid FS (proposed approach). We used IoTID20 dataset in this experiment. The main diagonal line reports the number of correct instances for each confusion matrix, and other cells represent the incorrect classification instances. For instance, Figure 8 (box 1) shows the classification results of ANN with top 60 ranked feature using IG approach. The 47,907 Mirai instances are classified correctly; 10 Mirai instances are predicted as DoS, 1289 Mirai instances are predicted as Scan, and 663 Mirai instances are predicted as MITM-ARP Spoofing. The 7040 Dos instances are predicted correctly; 52 DoS instances are wrongly classified as Mirai, 1 Dos instance is wrongly classified as Scan, and 7 DoS instances are wrongly classified as MITM-ARP Spoofing. The 8462 Scan instances are predicted correctly;

574 Scan instances are wrongly classified as Mirai, 2 Scan instances are wrongly classified as Dos, 92 Scan instances are wrongly classified as MITM-ARP Spoofing, and Scan instance are wrongly classified as Normal. The 2839 MITM-ARP Spoofing instances are predicted correctly; 1242 MITM-ARP Spoofing instances are wrongly classified as Mirai, 8 MITM-ARP Spoofing instances are wrongly classified as DoS, 103 MITM-ARP Spoofing instances are wrongly classified as Scan, and 4 MITM-ARP Spoofing instance are wrongly classified as Normal. The 4747 normal instances are predicted correctly, and there are no (none) wrongly classified samples. Figure 10 corresponding with Table 8 presenting the accuracy results of the five various ML with different FS approaches for multi-classification using the IoTID20 dataset. In the overall observation, the ML classifiers give better, more suitable results with our proposed feature selection method (IMF, UMF). Figure 10, along with Table 8, shows the accuracy of our study by utilizing the five various ML models with different FS approaches including our hybrid FS approach for multi-classification challenges using the IoTID20 dataset. Figure 10, along with Table 8, shows that our proposed model achieved a higher accuracy of 99.70% with Ensemble using 11 and 28 features to detect the multi-classes compared to other models using different number of features.

Table 8. Accuracy results yielded by the five various ML with different FS approaches for multi-classification using the IoTID20 dataset.

	Classifiers	**Accuracy**	**Classifiers**	**Accuracy**	**Classifiers**	**Accuracy**	**Classifiers**	**Accuracy**
FS Approach Name	**Information Gain**	**#**	**Gain Ratio**	**#**	**Intersection**	**#**	**Union**	**#**
Model	IG-ANN	94.60	GR-ANN	93.86	IMF-ANN	92.80	ANN UMF	94.00
	IG-C.4.5	99.54	GR-C.4.5	99.10	IMF-C.4.5	99.59	C 4.5 UMF	99.60
	IG-Bagging	99.48	GR-Bagging	98.95	IMF-Bagging	99.58	Bagging UMF	99.60
	IG-kNN	98.48	GR-kNN	98.43	IMF-kNN	98.76	kNN UMK	99.10
	IG-Ensemble	99.60	GR-Ensemble	99.14	IMF-Ensemble	99.70	Ensemble UMF	99.70

Tables 9–12 summarize values for different statistical parameters used in the multi-classification problem. Five various ML algorithms have been used with four different strategies of FS approaches. We utilized five various ML and four different ways of the FS approach to derive these performance results. The FS approaches are IG and GR and hybrid FS (IMF and UMF). As we mentioned above, these approaches were created to select the top 60 ranked features and the best 11 and 28 features using the IoTID20 dataset. Table 9 represents the results of using IG; Table 10 represents the results of using GR; Table 11 represents the results of using intersection theory; Table 12 represents the results of using union theory. In Table 9, we can observe that values of (i) false positive rate (FP) and (ii) precision, (iii) recall, (iv) F-measure, and (v) ROC for each individual target (Mirai, DoS, Scan, MAS: abbreviation for MITM ARP Spoofing, normal) and the accuracy overall of the model with each different ML algorithm. For instance, in Table 9, Ensemble with IG approach gives a small false-positive rate for each prediction target which is good for the model since a high FP rate will compromise system security by authorizing malicious data to move into the network. Moreover, the FP rate will significantly increase overheads as well as consume system resources and time. Moreover, our feature selection approaches (intersection and union) as shown in Tables 11 and 12 represent the best overall results. For example, in Table 12, Ensemble with union theory, we succeed in reaching the lowest (i.e., very nearly zero) possible value of FP rate for each class (target): 0.007 for Mirai, 0 for DoS, o for Scan, 0.001 MITM ARP Spoofing, and 0 for normal which is excellent for an IDS deployed into an IoT ecosystem. However, it is not possible to reduce all FP rates to zero in the IDS model because there is the well-known trade-off between these parameters.

(1) IG-60 ANN					
Mirai	47907	10	1289	663	0
Dos	52	7040	1	7	0
Scan	574	2	8462	92	1
MAS	1242	8	103	2839	4
Normal	0	0	0	0	4747
	Mirai	Dos	Scan	MAS	Normal

(2) IG-60 C4.5					
Mirai	49752	1	36	80	0
DoS	6	7092	1	1	0
Scan	86	0	9017	28	0
MAS	97	1	12	4086	0
Normal	0	0	0	0	4747
	Mirai	DoS	Scan	MAS	Normal

(3) IG-60 Bagging					
Mirai	49785	0	15	69	0
Dos	7	7091	1	1	0
Scan	118	0	8992	21	0
MAS	142	0	9	4045	0
Normal	4	0	0	0	4743
	Mirai	Dos	Scan	MAS	Normal

(4) IG-60 kNN					
Mirai	49500	12	132	225	0
Dos	33	7052	7	8	0
Scan	140	1	8961	29	0
MAS	248	2	33	3913	0
Normal	0	0	0	0	4747
	Mirai	Dos	Scan	MAS	Normal

(5) IG-60 Ensemble					
Mirai	49802	0	13	54	0
Dos	6	7092	1	1	0
Scan	96	0	9015	20	0
MAS	98	2	10	4086	0
Normal	0	0	0	0	4747
	Mirai	Dos	Scan	MAS	Normal

(6) GR-60 ANN					
Mirai	47825	7	1286	749	2
Dos	58	7037	3	2	0
Scan	884	1	8181	58	7
MAS	1416	5	124	2651	0
Normal	0	2	0	0	4745
	Mirai	Dos	Scan	MAS	Normal

(7) GR-60 C4.5					
Mirai	49655	1	67	146	0
DoS	4	7094	0	2	0
Scan	125	1	8976	29	0
MAS	276	4	48	3868	0
Normal	0	0	0	0	4747
	Mirai	DoS	Scan	MAS	Normal

(8) GR-60 Bagging					
Mirai	49787	0	31	51	0
Dos	10	7089	0	1	0
Scan	76	0	9029	26	0
MAS	100	0	15	4081	0
Normal	6	0	0	0	4741
	Mirai	Dos	Scan	MAS	Normal

(9) GR-60 kNN					
Mirai	49362	7	256	244	0
Dos	20	7073	5	2	0
Scan	232	3	8845	51	0
MAS	294	3	62	3837	0
Normal	0	0	0	0	4747
	Mirai	Dos	Scan	MAS	Normal

(10) GR-60 Ensemble					
Mirai	49729	1	39	100	0
Dos	7	7091	0	2	0
Scan	135	1	8960	35	0
MAS	298	0	30	3868	0
Normal	0	0	0	0	4747
	Mirai	Dos	Scan	MAS	Normal

(11) IMF ANN					
Mirai	47929	9	1519	411	1
DoS	55	7028	4	12	1
Scan	941	0	8161	27	2
MAS	2404	0	24	1768	0
Normal	1	2	0	0	4744
	Mirai	DoS	Scan	MAS	Normal

(12) IMF C 4.5					
Mirai	49766	0	41	62	0
Dos	5	7093	1	1	0
Scan	67	1	9036	27	0
MAS	85	1	21	4089	0
Normal	0	0	0	0	4747
	Mirai	Dos	Scan	MAS	Normal

(13) IMF Bagging					
Mirai	49714	0	43	112	0
Dos	10	7088	0	2	0
Scan	171	0	8913	47	0
MAS	367	0	25	3804	0
Normal	8	0	0	0	4739
	Mirai	Dos	Scan	MAS	Normal

(14) IMF kNN					
Mirai	49474	14	143	238	0
Dos	30	7060	3	7	0
Scan	149	0	8945	37	0
MAS	262	3	41	3890	0
Normal	2	2	0	0	4743
	Mirai	Dos	Scan	MAS	Normal

(15) IMF Ensemble					
Mirai	49823	1	16	29	0
Dos	0	7092	0	0	0
Scan	75	0	9038	18	0
MAS	91	1	13	4091	0
Normal	0	0	0	0	4747
	Mirai	Dos	Scan	MAS	Normal

(16) UMF ANN					
Mirai	47887	11	1241	730	0
DoS	45	7039	1	13	2
Scan	710	3	8324	78	16
MAS	1500	9	130	2556	1
Normal	0	0	1	2	4744
	Mirai	DoS	Scan	MAS	Normal

(17) UMF C 4.5					
Mirai	49758	3	51	57	0
Dos	7	7092	0	1	0
Scan	77	1	9032	21	0
MAS	87	1	16	4092	0
Normal	0	0	0	0	4747
	Mirai	Dos	Scan	MAS	Normal

(18) UMF Bagging					
Mirai	49812	0	17	40	0
Dos	11	7089	0	0	0
Scan	88	0	9025	18	0
MAS	104	0	12	4080	0
Normal	11	0	0	0	4736
	Mirai	Dos	Scan	MAS	Normal

(19) UMF kNN					
Mirai	49596	13	95	165	0
Dos	33	7056	4	7	0
Scan	117	1	8983	30	0
MAS	185	3	36	3972	0
Normal	2	0	0	0	4745
	Mirai	Dos	Scan	MAS	Normal

(20) UMF Ensemble					
Mirai	49823	0	14	32	0
Dos	6	7092	0	2	0
Scan	79	0	9037	15	0
MAS	81	1	16	4098	0
Normal	0	0	0	0	4747
	Mirai	Dos	Scan	MAS	Normal

Figure 9. Confusion matrix yielded by the five various ML with different FS approaches for multi-classification of the IoTId20 dataset.

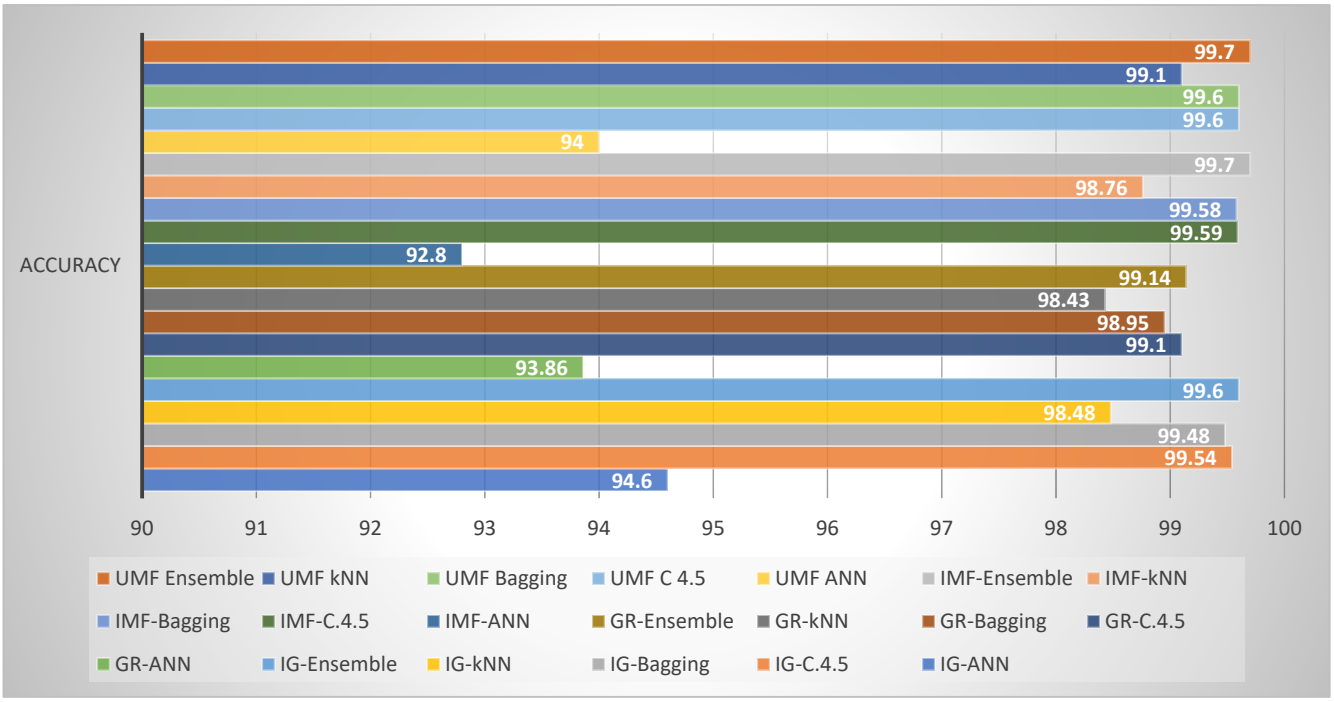

Figure 10. Accuracy results yielded by the five various ML with different FS approaches for multi-classification using the IoTID20 dataset.

4.4. Comparison Analysis of Results

To verify the effectiveness of the proposed solution approaches, we compare our model with several published methods that used the same evaluation datasets in term of binary classification and multi-classification. Figure 11, along with Table 13, compares the performance of our proposed models for binary classification with other state-of-art models using IoTID20. The comparison reveals that our proposed approaches achieve higher performance results (accuracy, precision, recall, f1-measure) than previously published approaches performed with different feature selection strategies. The proposed techniques (IMF, UMF) achieve the highest classification accuracy with 99.98% compared to the published benchmarks. Moreover, the proposed model using both IMF and UMF achieves very high-performance precision, recall, and f1-measure. Moreover, we have verified the quality of the proposed model in comparison with other state-of-art models using the second dataset (i.e., NSL-KDD) in term of binary classification (detection model) as shown in Figure 12, along with Table 14. The proposed techniques (IMF, UMF) also achieved the highest classification accuracy with 99.79%. Hence, the proposed model is more efficient compared to the existing state-of-the-art models presented in previous works [53–56] for the IoTID20 dataset and [42,64–74] for the NSL-KDD dataset. This is attributed to the effectiveness of high dimensionality reduction by removing irrelevant features. Figure 13, along with Table 15, shows a performance comparison of our detection model with other state-of-the-art models using the NSL-KDD dataset. From Figure 13, along with Table 15, we observe that our proposed model has a very good accuracy result. From Figure 13, along with Table 15, we can observe that the Intersection (IMF) theory and union (UMF) theory provide the same detection accuracy 99.79%.

Table 9. Summary of calculated statistical parameters for test IoTID20 dataset using IG with top-ranked 60 features.

(A) IG-ANN 60						(C) IG-Bagging 60						(E) IG-Ensemble 60					
Parameters	**Mirai**	**Dos**	**Scan**	**MAS**	**Normal**	**Parameters**	**Mirai**	**Dos**	**Scan**	**MAS**	**Normal**	**Parameters**	**Mirai**	**Dos**	**Scan**	**MAS**	**Normal**
FP Rate	0.074	0	0.21	0.011	0	**FP Rate**	0.011	0	0	0	0	**FP Rate**	0.008	0	0	0.001	0
Precision	96.2	99.7	85.9	78.8	99.9	**Precision**	99.5	100	99.7	97.8	100	**Precision**	99.6	100	99.7	98.2	100
Recall	96.1	99.2	92.7	67.7	100	**Recall**	99.8	99.9	98.5	96.4	99.9	**Recall**	99.9	99.9	98.7	97.4	100
F-Measure	96.2	99.4	89.1	72.8	99.9	**F-Measure**	99.6	99.9	99.1	97.1	100	**F-Measure**	99.7	99.9	99.2	97.8	100
ROC Area	97.8	99.8	96.9	75	100	**ROC Area**	100	100	99.9	100	100	**ROC Area**	99.5	99.9	99.3	98.6	100
Accuracy	96.2	99.5	89.3	73.3	100	**Accuracy**	99.7	100	99.1	97.1	100	**Accuracy**	99.8	100	99.2	97.8	100
(B) IG-C4.5 60						**(D) IG-kNN 60**											
Parameters	**Mirai**	**Dos**	**Scan**	**MAS**	**Normal**	**Parameters**	**Mirai**	**Dos**	**Scan**	**MAS**	**Normal**						
FP Rate	0.008	0	0.001	0.002	0	**FP Rate**	0.017	0	0.003	0	0						
Precision	99.6	100	99.5	97.4	100	**Precision**	99.2	99.8	98.1	93.7	100						
Recall	99.8	99.9	98.8	97.4	100	**Recall**	99.3	99.3	98.1	93.3	100						
F-Measure	99.7	99.9	99.1	97.4	100	**F-Measure**	99.2	99.6	98.1	93.5	100						
ROC Area	99.7	99.9	99.6	99.3	100	**ROC Area**	98.9	99.7	99.1	96.7	100						
Accuracy	99.7	100	99.2	97.4	100	**Accuracy**	99.3	99.6	98.1	93.5	100						

Table 10. Summary of calculated statistical parameters for test IoTID20 dataset using GR with top-ranked 60 features.

(A) GR-ANN 60						(C) GR-bagging 60						(E) GR-Ensemble 60					
Parameters	**Mirai**	**Dos**	**Scan**	**MAS**	**Normal**	**Parameters**	**Mirai**	**Dos**	**Scan**	**MAS**	**Normal**	**Parameters**	**Mirai**	**Dos**	**Scan**	**MAS**	**Normal**
FP Rate	0.094	0	0.021	0.011	0	**FP Rate**	0.022	0	0.001	0	0	**FP Rate**	0.017	0	0	0.002	0
Precision	95.3	99.8	85.3	76.6	99.8	**Precision**	98.9	100	99.2	95.9	100	**Precision**	99.1	1	99.2	96.6	100
Recall	95.9	99.1	89.6	63.2	100	**Recall**	99.7	99.8	97.6	90.7	99.8	**Recall**	99.7	99.9	98.1	92.2	100
F-Measure	95.6	99.4	87.4	69.3	99.9	**F-Measure**	99.3	99.9	98.4	93.2	99.9	**F-Measure**	99.4	99.9	98.7	94.3	100
ROC Area	98.2	99.4	96.9	95.7	100	**ROC Area**	97.9	100	99.9	99.8	100	**ROC Area**	99	99.9	99	96	100
Accuracy	95.6	99.5	87.5	69.9	99.9	**Accuracy**	99.3	99.9	98.4	93.3	99.9	**Accuracy**	99.4	50.5	98.7	94.4	100
(B) GR-C4.5 60						**(D) GR-kNN 60**											
Parameters	**Mirai**	**Dos**	**Scan**	**MAS**	**Normal**	**Parameters**	**Mirai**	**Dos**	**Scan**	**MAS**	**Normal**						
FP Rate	0.016	0	0.002	0.002	0	**FP Rate**	0.022	0	0.005	0	0						
Precision	99.2	99.9	98.7	95.6	100	**Precision**	98.9	99.8	96.5	92.8	100						
Recall	99.6	99.9	98.3	92.2	100	**Recall**	99	99.6	96.9	91.4	100						
F-Measure	99.4	99.9	98.5	93.9	100	**F-Measure**	98.9	99.7	96.7	92.1	100						
ROC Area	99.6	100	99.6	98.7	100	**ROC Area**	99	99.8	98.8	97.3	100						
Accuracy	99.4	99.9	98.5	93.9	100	**Accuracy**	99	99.7	96.7	92.1	100						

Table 11. Summary of calculated statistical parameters for test IoTID20 dataset using Intersection theory with the best 11 features.

(A) IMF ANN						(C) IMF Bagging						(E) IMF Ensemble					
Parameters	**Mirai**	**Dos**	**Scan**	**MAS**	**Normal**	**Parameters**	**Mirai**	**Dos**	**Scan**	**MAS**	**Normal**	**Parameters**	**Mirai**	**Dos**	**Scan**	**MAS**	**Normal**
FP Rate	0.135	0	0.023	0.006	0	**FP Rate**	0.008	0	0.001	0	0	**FP Rate**	0.007	0	0	0.001	0
Precision	93.4	99.8	84.1	79.7	99.9	**Precision**	99.6	100	99.5	98.1	100	**Precision**	99.7	100	99.7	98.9	100
Recall	96.1	99	89.4	42.1	99.9	**Recall**	99.8	99.8	98.9	97.3	99.9	**Recall**	99.9	99.9	99	97.5	100
F-Measure	94.7	99.4	86.6	55.1	99.9	**F-Measure**	99.7	99.9	99.2	0.98	99.9	**F-Measure**	99.8	99.9	99.3	98.2	100
ROC Area	97.1	99.8	97.1	89.9	100	**ROC Area**	100	100	100	99.9	100	**ROC Area**	100	100	100	99.9	100
Accuracy	99.8	99.9	99.3	97.9	99.9	**Accuracy**	99.4	99.6	98.5	95	100	**Accuracy**	99.8	100	99.4	98.2	100
(B) IMF C4.5						**(D) IMF kNN**											
Parameters	**Mirai**	**Dos**	**Scan**	**MAS**	**Normal**	**Parameters**	**Mirai**	**Dos**	**Scan**	**MAS**	**Normal**						
FP Rate	0.006	0	0.001	0.001	0	**FP Rate**	0.018	0	0.003	0	0						
Precision	99.7	100	99.3	97.8	100	**Precision**	99.1	99.7	98	93.2	100						
Recall	99.8	99.9	99	97.4	100	**Recall**	99.2	99.4	98	92.7	99.9						
F-Measure	99.7	99.9	99.1	97.6	100	**F-Measure**	99.2	99.6	98	93	100						
ROC Area	99.8	100	99.7	99.2	100	**ROC Area**	98.9	99.7	99	96.4	100						
Accuracy	99.8	100	99.2	97.6	100	**Accuracy**	99.2	99.6	98	93	100						

Table 12. Summary of calculated statistical parameters for test IoTID20 dataset using union theory with the best 28 features.

(A) UMF ANN						(C) UMF bagging						(E) UMF Ensemble					
Parameters	**Mirai**	**Dos**	**Scan**	**MAS**	**Normal**	**Parameters**	**Mirai**	**Dos**	**Scan**	**MAS**	**Normal**	**Parameters**	**Mirai**	**Dos**	**Scan**	**MAS**	**Normal**
FP Rate	0.09	0	0.021	0.012	0	**FP Rate**	0.009	0	0	0	0	**FP Rate**	0.007	0	0	0.001	0
Precision	95.5	99.7	85.8	75.6	99.6	**Precision**	99.6	100	99.7	98.6	100	**Precision**	99.7	100	99.7	98.8	100
Recall	96	99.1	91.2	60.9	99.9	**Recall**	99.9	99.8	98.8	97.2	99.8	**Recall**	99.9	99.9	99	97.7	100
F-Measure	95.8	99.4	88.4	67.5	99.8	**F-Measure**	99.7	99.9	99.3	97.9	99.9	**F-Measure**	99.8	99.9	99.3	98.2	100
ROC Area	98.2	99.7	97	95.9	100	**ROC Area**	100	100	100	97.8	100	**ROC Area**	100	100	100	99.9	100
Accuracy	95.8	99.4	88.5	68.3	99.8	**Accuracy**	99.8	99.9	99.3	97.9	99.9	**Accuracy**	99.8	100	99.4	98.3	100
(B) UMF C4.5						**(D) UMF kNN**											
Parameters	**Mirai**	**Dos**	**Scan**	**MAS**	**Normal**	**Parameters**	**Mirai**	**Dos**	**Scan**	**MAS**	**Normal**						
FP Rate	0.009	0	0	0.001	0	**FP Rate**	0.013	0	0.002	0	0						
Precision	99.6	100	99.7	98.6	100	**Precision**	99.3	99.8	98.5	95.2	100						
Recall	99.9	99.8	98.8	97.2	99.8	**Recall**	99.5	99.4	98.4	94.7	100						
F-Measure	99.7	99.9	99.3	97.9	99.9	**F-Measure**	99.4	99.6	98.4	94.9	100						
ROC Area	100	100	100	99.9	100	**ROC Area**	99.2	99.7	99.2	97.4	100						
Accuracy	99.8	99.9	99.3	97.9	99.9	**Accuracy**	99.4	99.6	98.5	95	100						

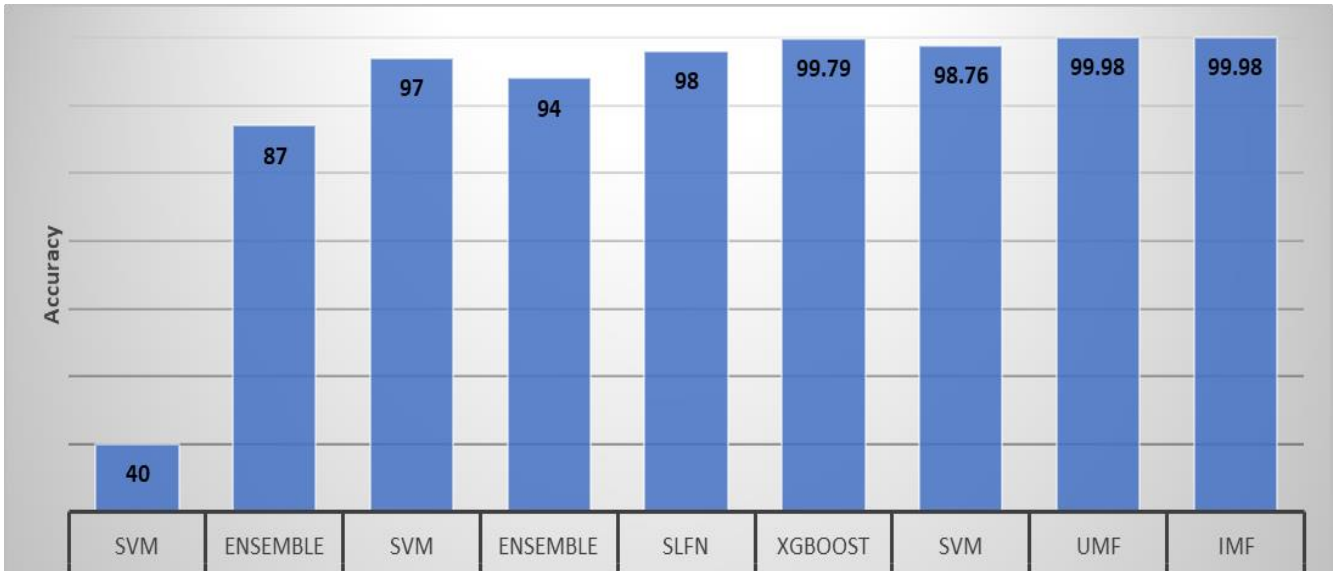

Figure 11. The comparison results of our detection model performance and related works of IoTID20.

Table 13. The comparison results of our detection model performance and related works of IoTID20.

Reference	Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Measure (%)
[53]	SVM	40	55	37	16
[53]	Ensemble	87	87	87	87
[54]	SVM	97	-	-	-
[54]	Ensemble	94	-	-	-
[54]	SLFN	98	-	-	-
[56]	XGBoost	99.79	-	98	1
[56]	SVM	98.76	-	98.00	98.00
Proposed	UMF	99.98	99.90	99.90	99.90
Proposed	IMF	99.98	99.90	99.90	99.90

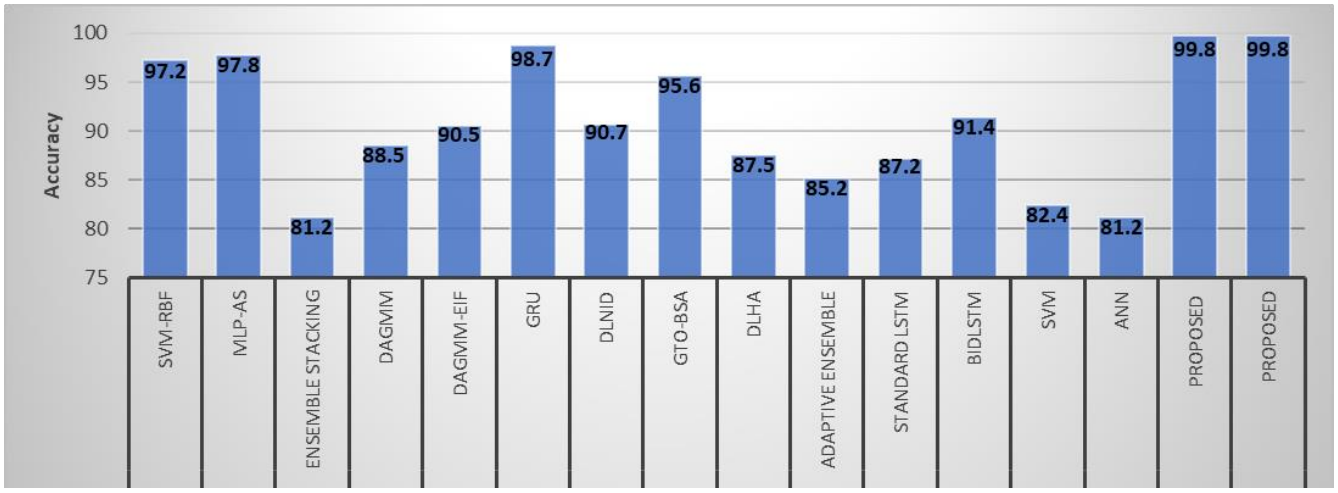

Figure 12. The comparison of accuracy result of our detection model performance and related works of NSL-KDD.

Table 14. The comparison of accuracy result of our detection model performance and related works of NSL-KDD.

Reference	Model	Accuracy (%)
[64]	SVM-RBF	97.2
[64]	MLP-AS	97.8
[65]	Ensemble Stacking	81.2
[66]	DAGMM	88.5
[66]	DAGMM-EIF	90.5
[68]	GRU	98.7
[69]	DLNID	90.7
[70]	GTO-BSA	95.6
[71]	DLHA	87.5
[42]	adaptive Ensemble	85.2
[72]	Standard LSTM	87.2
[72]	BidLSTM	91.4
[73]	SVM	82.4
[74]	ANN	81.2
Proposed	UMF	99.8
Proposed	IMF	99.8

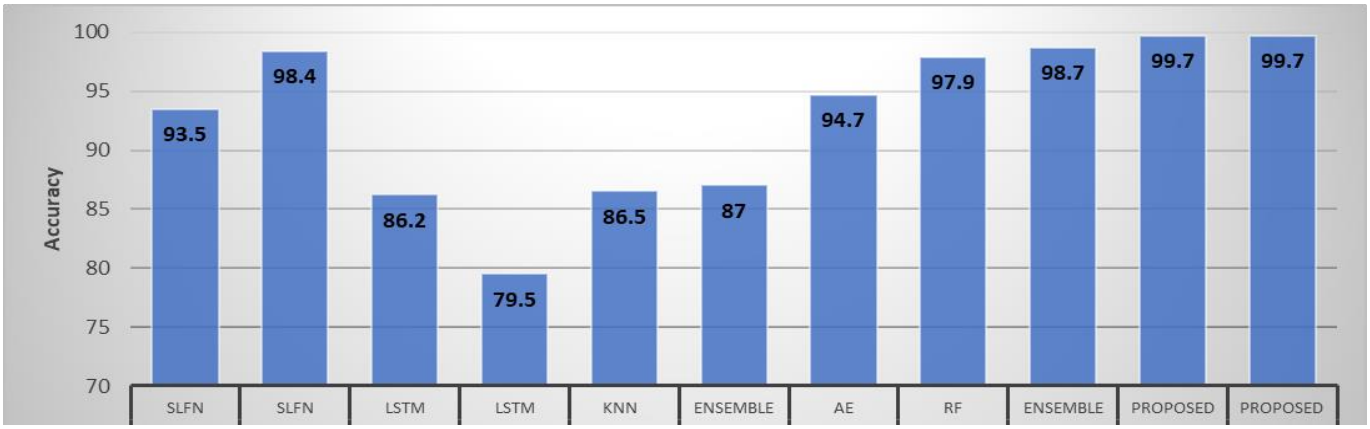

Figure 13. The comparison results of our multi classification model performance and related works of IoTID20 dataset.

Table 15. The comparison results of our multi classification model performance and related works of IoTID20 dataset.

Reference	Model	Accuracy (%)
[54]	SLFN-SVM-SMOTE	93.5
[54]	SLFN	98.4
[75]	SMOTE + LSTM	86.2
[75]	LSTM	79.5
[75]	kNN	86.5
[53]	Ensemble	87
[76]	Autoencoder	94.7
[77]	RF	97.9
[78]	ensemble	98.7
Proposed	UMF	99.7
Proposed	IMF	99.7

Based on the obtained results of these experiments summarized in the tables and figures above, accuracy performance increased via reducing feature dimensionality using IMF-Ensemble and UMF-Ensemble as compared to using only the classification via the IG or GR approach. Furthermore, independent (individual) classifiers achieved bet-

ter classification accuracy using optimum top-ranked features (IMF, UMF) compared to features selection obtained by the IG and GR approaches. For instance, the Bagging classifier achieved the highest accuracy (%99.91) by using the UMF approach (28 optimum top-ranked features) compared to the various numbers of selected features. In the ANN classifier, the accuracy classification fluctuates up and down by applying various feature selection approaches. There is a noticeable increase in the accuracy performance of the ensemble classifier when compared to individual classifiers. In terms of selected features, a small number of relevant features are not required to achieve the higher accuracy whereas many relevant features do not assure that a lower classification accuracy will result.

Based on the obtained results from the IoTID20 dataset, the proposed UMF and IMF results in the improvement of Anomaly-based IDS for accuracy, precision, recall, f1-measure, ROC area of 99.98, 99.90, 99.90, 99.90, 99.90, respectively. The improvement in the performance of IMF-ensemble or UMF-ensemble is because IMF and UMF filter-based approaches use an optimum (min, max) number of top-ranked features relevant for classification. UMF and IMF do not show any fluctuation in the performance since UMF, and IMF incorporates the best features produced by both IG and GR approaches. Furthermore, the model with UMF and IMF is not prone to underfitting when the number of features decrease. According to the results shown in Table 13, the proposed model is superior and competent in comparison to the results stated for models reported in [53–56] using IoTId20 dataset. Furthermore, we used the NSL-KDD dataset for validating our binary classification detection model. Table 14 indicates that the proposed model is superior and competent in comparison to the results stated for the models reported in [42,64–74]. Figure 13, along with Table 15 represent a comparison between our multi classification model performance versus the related works with respect to the IoTID20 dataset. Finally, while our proposed study evaluates the anticipated predictive models using diverse evaluation metrics, some related works failed to evaluate all metrics since they did not consider the choice of different features. Indeed, access to the best features with a minimum of features is extremely important to improve accuracy and conserve resources and reduce training time complexity.

5. Summary and Conclusions

This paper proposes a feature selection approach using the concept of mathematical set theory for machine learning-based IDS to extract efficient subsets of features. The developed machine learning-based IDS scheme has 3-phases: a data preprocessing phase, a dimensionality reduction, and feature selection phase, and a model training and classification phase. The dimensionality reduction phase has two subphases. In phase 2:A, we ranked the features using IG and GR filter-based approaches to produce the top 60 ranked and top 20 ranked subsets of features for the IoTID20 dataset as well as for all features and top 20 ranked for the case of the NSL-KDD dataset. In the second subphase of dimensionality reduction, we developed a hybrid feature selection approach using intersection and union rules. The second subphase produces a features subset that is optimized for performance via the elimination of redundant features. The model training/classification phase applies five ML algorithms, namely bagging, ANN, J48, kNN, Ensemble algorithms to classify the generated subsets of traffic features into normal or intrusion classes as binary classification as well as multi-classes for the purpose of multi-classification. The advantage of the ensemble-based hybrid approach is that the selected feature subsets provide optimum results. Even though dimensionality has been reduced, the resultant features produce an optimum classification. These results give us an indication that the current feature selection approaches are unable to extract the relevant features and eliminate irrelevant features; thus, we come to a trade-off between the two best entropy feature selection approaches to improve the inadequacies of each one.

To conclude, the ensemble method has provided better results in classification performance for both the IoTID20 and NSL-KDD datasets compared to related works and individual ML algorithms. This is the result from eliminating irrelevant features before the

training process. The experiments demonstrated a significant improvement in the realms of accuracy, precision, recall, f1-measure, and the ROC area.

In the future, we seek to deploy our proposed system to be used by an IoT gateway device for the delivery of detection and classification services against various cyber-attacks and intrusions within a network of IoT devices (e.g., a network of Advanced RISC Machine (ARM) or Arduino raspberry Pi nodes). Further investigation regarding resource consumption can be characterized and reported as additional evaluation parameters to enhance our studies of the proposed IoT-IDS testbed. This can include the analysis of energy consumption, inferencing overhead, memory utilization, and processing complexity using resource aware IoT nodes with tiny system elements.

Author Contributions: Conceptualization, K.A.; methodology, K.A.; software, K.A., Q.A.A.-H.; validation, K.A., F.T.S. and Q.A.A.-H.; formal analysis, K.A.; investigation, K.A., F.T.S., Q.A.A.-H., A.A.J. and M.A.; resources, K.A., Q.A.A.-H.; data curation, K.A.; writing—original draft preparation, K.A., Q.A.A.-H.; writing—review and editing, K.A., Q.A.A.-H., F.T.S., S.A.A., A.A.J. and M.A. visualization, K.A.,Q.A.A.-H., A.A.J. and M.A.; supervision, F.T.S. project administration, K.A.; funding acquisition, S.A.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: The researchers would like to thank the Deanship of Scientific Research, Qassim University for funding the publication of this project.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Turton, W.; Mehrotra, K. Hackers breached colonial pipeline using compromised password. Available online: https://www.bloomberg.com/news/articles/2021-06-04/hackers-breached-colonial-pipeline-using-compromised-password (accessed on 1 October 2021).
2. Smadi, A.A.; Ajao, B.T.; Johnson, B.K.; Lei, H.; Chakhchoukh, Y.; Al-Haija, Q.A. A Comprehensive Survey on Cyber-Physical Smart Grid Testbed Architectures: Requirements and Challenges. *Electronics* **2021**, *10*, 1043. [CrossRef]
3. Albulayhi, K.; Smadi, A.A.; Sheldon, F.T.; Abercrombie, R.K. IoT Intrusion Detection Taxonomy, Reference Architecture, and Analyses. *Sensors* **2021**, *21*, 6432. [CrossRef]
4. Farooq, M.U.; Waseem, M.; Mazhar, S.; Khairi, A.; Kamal, T. A review on internet of things (IoT). *Int. J. Comput. Appl.* **2015**, *113*, 1–7.
5. Aborujilah, A.; Nassr, R.M.; Al-Hadhrami, T.; Husen, M.N.; Ali, N.A.; Al-Othmani, A.; Syahela, N.; Ochiai, H. Security Assessment Model to Analysis DOS Attacks in WSN. In *International Conference of Reliable Information and Communication Technology*; Springer: Berlin/Heidelberg, Germany, 2019.
6. Agrawal, K.; Kamboj, N. Smart agriculture using IOT: A futuristic approach. *Int. J. Inf. Dissem.Technol.* **2019**, *9*, 186–190. [CrossRef]
7. Pawar, P.; Trivedi, A. Device-to-device communication based IoT system: Benefits and challenges. *IETE Tech. Rev.* **2019**, *36*, 362–374. [CrossRef]
8. CISOMAG. IoT Connections to Reach 83 Billion by 2024: Report. Available online: https://cisomag.eccouncil.org/iot-connections-to-reach-83-billion-by-2024-report/ (accessed on 12 July 2021).
9. Kumar, S.; Solanki, V.K.; Choudhary, S.K.; Selamat, A.; González Crespo, R. Comparative Study on Ant Colony Optimization (ACO) and K-Means Clustering Approaches for Jobs Scheduling and Energy Optimization Model in Internet of Things (IoT). *Int. J. Interact. Multimed. Artif. Intell.* **2020**, *6*, 107–116. [CrossRef]
10. Nimbalkar, P.; Kshirsagar, D. Feature selection for intrusion detection system in Internet-of-Things (IoT). *ICT Express* **2021**, *7*, 177–181. [CrossRef]
11. Albulayhi, K.; Sheldon, F.T. An Adaptive Deep-Ensemble Anomaly-Based Intrusion Detection System for the Internet of Things. In Proceedings of the 2021 IEEE World AI IoT Congress (AIIoT), Seattle, WA, USA, 10–13 May 2021.
12. Eskandari, M.; Janjua, Z.H.; Vecchio, M.; Antonelli, F. Passban IDS: An Intelligent Anomaly-Based Intrusion Detection System for IoT Edge Devices. *IEEE Internet Things J.* **2020**, *7*, 6882–6897. [CrossRef]
13. Abraham, A.; Grosan, C.; Martin-Vide, C. Evolutionary design of intrusion detection programs. *Int. J. Netw. Secur.* **2007**, *4*, 328–339.
14. Ilgun, K.; Ustat, A. A Real-Time Intrusion Detection System for Unix. Master's Thesis, University of California Santa Barbara, Santa Barbara, CA, USA, 1992.

15. Verma, A.; Ranga, V. Machine learning based intrusion detection systems for IoT applications. *Wirel. Pers. Commun.* **2020**, *111*, 2287–2310. [CrossRef]
16. Sarker, I.H. Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. *SN Comput. Sci.* **2021**, *2*, 1–20. [CrossRef]
17. Siddiqi, M.A.; Pak, W. An Agile Approach to Identify Single and Hybrid Normalization for Enhancing Machine Learning-Based Network Intrusion Detection. *IEEE Access* **2021**, *9*, 137494–137513. [CrossRef]
18. Latif, S.; Zou, Z.; Idrees, Z.; Ahmad, J. A Novel Attack Detection Scheme for the Industrial Internet of Things Using a Lightweight Random Neural Network. *IEEE Access* **2020**, *8*, 89337–89350. [CrossRef]
19. Heigl, M.; Weigelt, E.; Fiala, D.; Schramm, M. Unsupervised Feature Selection for Outlier Detection on Streaming Data to Enhance Network Security. *Appl. Sci.* **2021**, *11*, 12073. [CrossRef]
20. Sarker, I.H. Deep Cybersecurity: A Comprehensive Overview from Neural Network and Deep Learning Perspective. *SN Comput. Sci.* **2021**, *2*, 1–16. [CrossRef]
21. Balogh, S.; Gallo, O.; Ploszek, R.; Špaček, P.; Zajac, P. IoT Security Challenges: Cloud and Blockchain, Postquantum Cryptography, and Evolutionary Techniques. *Electronics* **2021**, *10*, 2647. [CrossRef]
22. Alrubayyi, H.; Goteng, G.; Jaber, M.; Kelly, J. Challenges of Malware Detection in the IoT and a Review of Artificial Immune System Approaches. *J. Sens. Actuator Networks* **2021**, *10*, 61. [CrossRef]
23. Ambusaidi, M.A.; He, X.; Nanda, P.; Tan, Z. Building an Intrusion Detection System Using a Filter-Based Feature Selection Algorithm. *IEEE Trans. Comput.* **2016**, *65*, 2986–2998. [CrossRef]
24. Amiri, F.; Yousefi, M.R.; Lucas, C.; Shakery, A.; Yazdani, N. Mutual information-based feature selection for intrusion detection systems. *J. Netw. Comput. Appl.* **2011**, *34*, 1184–1199. [CrossRef]
25. Sung, A.H.; Mukkamala, S. Identifying important features for intrusion detection using support vector machines and neural networks. In Proceedings of the 2003 Symposium on Applications and the Internet, Orlando, FL, USA, 27–31 January 2003.
26. Jose, S.; Malathi, D.; Reddy, B.; Jayaseeli, D. A Survey on anomaly-based host intrusion detection system. In *Journal of Physics: Conference Series*; IOP Publishing: Bristol, UK, 2018.
27. Khraisat, A.; Gondal, I.; Vamplew, P.; Kamruzzaman, J. Survey of intrusion detection systems: Techniques, datasets and challenges. *Cybersecurity* **2019**, *2*, 1–22. [CrossRef]
28. Biesiada, J.; Duch, W. *Feature Selection for High-Dimensional Data—A Pearson Redundancy Based Filter, in Computer Recognition Systems 2*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 242–249.
29. Jin, X.; Xu, A.; Bie, R.; Guo, P. Machine learning techniques and chi-square feature selection for cancer classification using SAGE gene expression profiles. In *International Workshop on Data Mining for Biomedical Application*; Springer: Berlin/Heidelberg, Germany, 2006.
30. Thang, N.D.; Lee, Y.-K. An improved maximum relevance and minimum redundancy feature selection algorithm based on normalized mutual information. In Proceedings of the 2010 10th IEEE/IPSJ International Symposium on Applications and the Internet, Seoul, Korea, 19–23 July 2010.
31. Estévez, P.A.; Tesmer, M.; Perez, C.A.; Zurada, J.M. Normalized Mutual Information Feature Selection. *IEEE Trans. Neural Networks* **2009**, *20*, 189–201. [CrossRef] [PubMed]
32. Peng, H.; Long, F.; Ding, C. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 1226–1238. [CrossRef] [PubMed]
33. Kwak, N.; Choi, C.-H. Input feature selection by mutual information based on Parzen window. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 1667–1671. [CrossRef]
34. Kohavi, R.; John, G.H. Wrappers for feature subset selection. *Artif. Intell.* **1997**, *97*, 273–324. [CrossRef]
35. Osman, H.; Ghafari, M.; Nierstrasz, O. Automatic feature selection by regularization to improve bug prediction accuracy. In Proceedings of the 2017 IEEE Workshop on Machine Learning Techniques for Software Quality Evaluation (MaLTeSQuE), Klagenfurt, Austria, 21 February 2017.
36. Quinlan, J.R. Induction of decision trees. *Mach. Learn.* **1986**, *1*, 81–106. [CrossRef]
37. Han, J.; Pei, J.; Kamber, M. *Data Mining: Concepts and Techniques*; Elsevier: Amsterdam, The Netherlands, 2011.
38. Abu Al-Haija, Q.; Zein-Sabatto, S. An Efficient Deep-Learning-Based Detection and Classification System for Cyber-Attacks in IoT Communication Networks. *Electronics* **2020**, *9*, 2152. [CrossRef]
39. Bendiab, G.; Shiaeles, S.; Alruban, A.; Kolokotronis, N. IoT Malware Network Traffic Classification using Visual Representation and Deep Learning. In Proceedings of the 2020 6th IEEE Conference on Network Softwarization (NetSoft), Ghent, Belgium, 29–31 July 2020.
40. AAbu Al-Haija, Q.; McCurry, C.D.; Zein-Sabatto, S. Intelligent Self-reliant Cyber-Attacks Detection and Classification System for IoT Communication Using Deep Convolutional Neural Network. In *Selected Papers from the 12th International Networking Conference, Rhodes, Greece, 19–21 September 2020*; Springer: Cham, Switzerland, 2021.
41. Taher, K.A.; Jisan, B.M.Y.; Rahman, M. Network intrusion detection using supervised machine learning technique with feature selection. In Proceedings of the 2019 International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), Dhaka, Bangladesh, 10–12 January 2019.
42. Gao, X.; Shan, C.; Hu, C.; Niu, Z.; Liu, Z. An Adaptive Ensemble Machine Learning Model for Intrusion Detection. *IEEE Access* **2019**, *7*, 82512–82521. [CrossRef]

43. Sapre, S.; Ahmadi, P.; Islam, K. A robust comparison of the KDDCup99 and NSL-KDD IoT network intrusion detection datasets through various machine learning algorithms. *arXiv* **2019**, arXiv:1912.13204.
44. Chowdhury MM, U.; Hammond, F.; Konowicz, G.; Xin, C.; Wu, H.; Li, J. A few-shot deep learning approach for improved intrusion detection. In Proceedings of the 2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON), New York, NY, USA, 19–21 October 2017.
45. Javaid, A.; Niyaz, Q.; Sun, W.; Alam, M. A Deep Learning Approach for Network Intrusion Detection System. In In Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS), New York, NY, USA, 3–5 December 2015.
46. Imamverdiyev, Y.; Sukhostat, L. Anomaly detection in network traffic using extreme learning machine. In Proceedings of the 2016 IEEE 10th International Conference on Application of Information and Communication Technologies (AICT), Baku, Azerbaijan, 12–14 October 2016.
47. Al-Haijaa, Q.A.; Ishtaiwia, A. Machine Learning Based Model to Identify Firewall Decisions to Improve Cyber-Defense. *Int. J. Adv.Sci. Eng. Inf. Technol.* **2021**, *11*, 1688–1695. [CrossRef]
48. Lin, W.-C.; Ke, S.-W.; Tsai, C.-F. CANN: An intrusion detection system based on combining cluster centers and nearest neighbors. *Knowledge-Based Syst.* **2015**, *78*, 13–21. [CrossRef]
49. Khammassi, C.; Krichen, S. A GA-LR wrapper approach for feature selection in network intrusion detection. *Comput. Secur.* **2017**, *70*, 255–277. [CrossRef]
50. Manzoor, I.; Kumar, N. A feature reduced intrusion detection system using ANN classifier. *Expert Syst. Appl.* **2017**, *88*, 249–257.
51. Sindhu, S.S.S.; Geetha, S.; Kannan, A. Decision tree based light weight intrusion detection using a wrapper approach. *Expert Syst. Appl.* **2012**, *39*, 129–141. [CrossRef]
52. Li, Y.; Wang, J.L.; Tian, Z.H.; Lu, T.B.; Young, C. Building lightweight intrusion detection system using wrapper-based feature selection mechanisms. *Comput. Secur.* **2009**, *28*, 466–475. [CrossRef]
53. Ullah, I.; Mahmoud, Q.H. *A Scheme for Generating a Dataset for Anomalous Activity Detection in IoT Networks*; Springer International Publishing: Berlin/Heidelberg, Germany, 2020.
54. Qaddoura, R.; Al-Zoubi, A.M.; Almomani, I.; Faris, H. A Multi-Stage Classification Approach for IoT Intrusion Detection Based on Clustering with Oversampling. *Appl. Sci.* **2021**, *11*, 3022. [CrossRef]
55. Yang, L.; Shami, A. A Lightweight Concept Drift Detection and Adaptation Framework for IoT Data Streams. *IEEE Internet Things Mag.* **2021**, *4*, 96–101. [CrossRef]
56. Krishnan, S.; Neyaz, A.; Liu, Q. IoT Network Attack Detection using Supervised Machine Learning. *Int. J. Artif. Intell. Expert Syst.* **2021**, *10*, 18–32.
57. Abe, N.; Kudo, M. Entropy criterion for classifier-independent feature selection. In *International Conference on Knowledge-Based and Intelligent Information and Engineering System*; Springer: Berlin/Heidelberg, Germany, 2005.
58. Ukil, A.; Sen, J.; Koilakonda, S. Embedded security for Internet of Things. In Proceedings of the 2011 2nd National Conference on Emerging Trends and Applications in Computer Science, Shillong, India, 4–5 March 2011.
59. Soldatos, J. *A 360-Degree View of IoT Technologies*; Artech House: New York, NY, USA, 2020.
60. Siddiqi, M.A.; Pak, W. Optimizing Filter-Based Feature Selection Method Flow for Intrusion Detection System. *Electronics* **2020**, *9*, 2114. [CrossRef]
61. Cybersecurity, C.I.f. Available online: https://www.unb.ca/cic/datasets/nsl.html (accessed on 21 April 2022).
62. Kang, H.; Ahn, D.H.; Lee, G.M.; Yoo, J.D.; Park, K.H.; Kim, H.K. IOT Network Intrusion Dataset. 2019. Available online: https://ieee-dataport.org/open-access/iot-network-intrusion-dataset (accessed on 2 February 2022).
63. Witten, I.H.; Frank, E. Data mining: Practical machine learning tools and techniques with Java implementations. *SIGMOD Rec.* **2002**, *31*, 76–77. [CrossRef]
64. Chu, W.-L.; Lin, C.-J.; Chang, K.-N. Detection and Classification of Advanced Persistent Threats and Attacks Using the Support Vector Machine. *Appl. Sci.* **2019**, *9*, 4579. [CrossRef]
65. Soleymanzadeh, R.; Aljasim, M.; Qadeer, M.W.; Kashef, R. Cyberattack and Fraud Detection Using Ensemble Stacking. *AI* **2022**, *3*, 22–36. [CrossRef]
66. Carrera, F.; Dentamaro, V.; Galantucci, S.; Iannacone, A.; Impedovo, D.; Pirlo, G. Combining Unsupervised Approaches for Near Real-Time Network Traffic Anomaly Detection. *Appl. Sci.* **2022**, *12*, 1759. [CrossRef]
67. Iliyasu, A.S.; Abdurrahman, U.A.; Zheng, L. Few-Shot Network Intrusion Detection Using Discriminative Representation Learning with Supervised Autoencoder. *Appl. Sci.* **2022**, *12*, 2351. [CrossRef]
68. Cao, B.; Li, C.; Song, Y.; Qin, Y.; Chen, C. Network Intrusion Detection Model Based on CNN and GRU. *Appl. Sci.* **2022**, *12*, 4184. [CrossRef]
69. Fu, Y.; Du, Y.; Cao, Z.; Li, Q.; Xiang, W. A Deep Learning Model for Network Intrusion Detection with Imbalanced Data. *Electronics* **2022**, *11*, 898. [CrossRef]
70. Kareem, S.S.; Mostafa, R.R.; Hashim, F.A.; El-Bakry, H.M. An Effective Feature Selection Model Using Hybrid Metaheuristic Algorithms for IoT Intrusion Detection. *Sensors* **2022**, *22*, 1396. [CrossRef]
71. Wisanwanichthan, T.; Thammawichai, M. A Double-Layered Hybrid Approach for Network Intrusion Detection System Using Combined Naive Bayes and SVM. *IEEE Access* **2021**, *9*, 138432–138450. [CrossRef]

72. Imrana, Y.; Xiang, Y.; Ali, L.; Abdul-Rauf, Z.; Hu, Y.C.; Kadry, S.; Lim, S. χ 2-BidLSTM: A Feature Driven Intrusion Detection System Based on χ 2 Statistical Model and Bidirectional LSTM. *Sensors* **2022**, *22*, 2018. [CrossRef]
73. Pervez, M.S.; Farid, D.M. Feature selection and intrusion classification in NSL-KDD cup 99 dataset employing SVMs. In Proceedings of the 8th International Conference on Software, Knowledge, Information Management and Applications (SKIMA 2014), Dhaka, Bangladesh, 18–20 December 2014.
74. Ingre, B.; Yadav, A. Performance analysis of NSL-KDD dataset using ANN. In Proceedings of the 2015 international conference on signal processing and communication engineering systems, Vijayawada, India, 2–3 January 2015.
75. Qaddoura, R.; Al-Zoubi, M.; Faris, H.; Almomani, I. A Multi-Layer Classification Approach for Intrusion Detection in IoT Networks Based on Deep Learning. *Sensors* **2021**, *21*, 2987. [CrossRef]
76. Song, Y.; Hyun, S.; Cheong, Y.-G. Analysis of Autoencoders for Network Intrusion Detection. *Sensors* **2021**, *21*, 4294. [CrossRef]
77. Hussein, A.Y.; Falcarin, P.; Sadiq, A.T. Enhancement performance of random forest algorithm via one hot encoding for IoT IDS. *Period. Eng. Nat. Sci.* **2021**, *9*, 579–591. [CrossRef]
78. De Souza, C.A.; Westphall, C.B.; Machado, R.B. Two-step ensemble approach for intrusion detection and identification in IoT and fog computing environments. *Comput. Electr. Eng.* **2022**, *98*, 107694. [CrossRef]