A multi-measure feature selection algorithm for efficacious intrusion detection

Herrera-Semenets, Vitali; Bustio-Martínez, Lázaro; Hernández-León, Raudel; van den Berg, Jan

DOI
10.1016/j.knosys.2021.107264

Publication date
2021

Document Version
Final published version

Published in
Knowledge-Based Systems

Knowledge-Based Systems 227 (2021) 107264

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

A multi-measure feature selection algorithm for efficacious intrusion detection

Vitali Herrera-Semenets [a,*], Lázaro Bustio-Martínez [b], Raudel Hernández-León [a], Jan van den Berg [c]

[a] Advanced Technologies Application Center (CENATAV), 7a ♯ 21406, Playa, C.P. 12200, Havana, Cuba
[b] National Institute for Astrophysics, Optics and Electronic, Luis Enrique Erro No 1, Sta. Ma. Tonantzintla, C.P. 72840, Puebla, Mexico
[c] Intelligent Systems Department, Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands

ARTICLE INFO

Article history:
Received 7 April 2021
Received in revised form 7 June 2021
Accepted 25 June 2021
Available online 30 June 2021

Keywords:
Feature selection
Data reduction
Intrusion detection algorithms

ABSTRACT

Every day the number of devices interacting through telecommunications networks grows resulting into an increase in the volume of data and information generated. At the same time, a growing number of information security incidents is being observed including the occurrence of unauthorized accesses, also named intrusions. As a consequence of these two developments, Information and Communications services providers require automated processes to detect and solve such intrusions, and this should done quickly in order to keep the related cybersecurity risks at acceptable levels. However, the presence of large volumes of data negatively interferes with the performance of classifiers used in intrusion detection tasks, which limits their applicability in practical cases. The research reported in this paper focuses on proposing a novel feature selection algorithm for intrusion detection scenarios. To this end, an extensive literature review was executed to first discover issues in the feature selection algorithms reported. Based on the insights obtained, the new multi-measure feature selection algorithm was designed that uses qualitative information provided by multiple feature selection measures, and reduces the dimensionality of the training data set. The algorithm proposed was next extensively tested using various data sets. It provides greater efficacy than other feature selection algorithms used for intrusion detection purposes. We finalize by providing some ideas on future research in order to further improve the algorithm.

1. Introduction

The proliferation of smart devices in the last decade brought a massive increase in the data streams generated by all kind of human activities and transmitted over the telecommunications networks [1]. According to a report about Internet of Things from Cisco, almost 7 smart devices are expected to be connected to the Internet by every person in the world during 2020 [2]. This increases the possibilities for committing malicious activities, especially intrusions in the telecommunications networks, which is one of the industries that report most economic losses every year [3].

Recent studies show that global losses caused by cyber-attacks can reach $6 trillion USD annually by 2021 [3]. Besides, companies take approximately 46 days, spending an average of $32,000 USD per day, to correct the consequences of a cyber-attack [4]. In this sense, many companies choose to have intrusion detection systems that guarantee protection against the eventual occurrence of an attack.

From the data classification perspective, the main goal of building an Intrusion Detection System (IDS) is to build a classification model by applying a learning algorithm that learns the model from a given, labeled data set (see Fig. 1) [5]. The model learned can be used to predict, given a new input, the classification class (the output of the model). It is important to notice that the accuracy of the classification model not only depends on the classifier used, but also on the quality of the training data.

In practice, the characteristics of an intrusion detection scenario (such as the presence of redundant information and/or noise, and the enormous amount of data generated) can negatively affect the classifiers performance. This is why a preprocessing stage is often used to obtain a better quality data set in order to improve the performance of the classifier, measured in terms of efficiency (given by spatial or temporal complexity) and efficacy (given by accuracy) [6]. Considering the characteristics of the intrusion detection scenarios, described above, the preprocessing stage in the context of our research mainly focuses

* Corresponding author.
E-mail addresses: vherrera@cenatav.co.cu (V. Herrera-Semenets), lbustio@inaoep.mx (L. Bustio-Martínez), rhernandez@cenatav.co.cu (R. Hernández-León), j.vandenberg@tudelft.nl (J. van den Berg).

https://doi.org/10.1016/j.knosys.2021.107264
0950-7051/

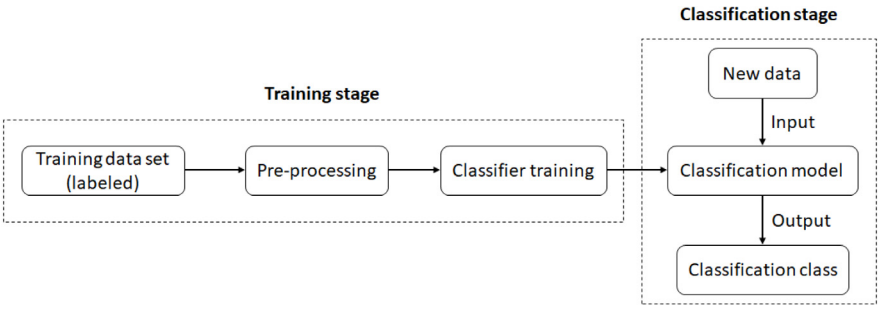

Fig. 1. General scheme for intrusions detection from the data classification perspective.

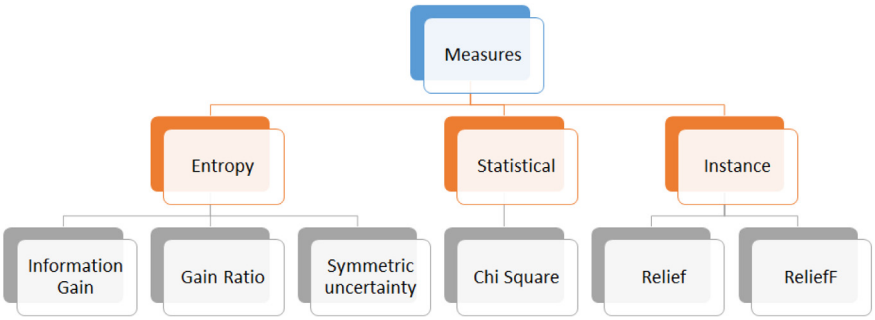

Fig. 2. Taxonomy of the main features selection measures.

on a data reduction process [7]. The main goal of such process is to obtain a reduced representation of the original data while maintaining its statistical characteristics and integrity.

Feature selection is a pre-processing technique frequently used in intrusion detection tasks. However, one specific aspect of these feature selection algorithms usually negatively affects the efficacy of the later classifiers; *i.e.*, the use of a single measure usually leads to a biased result of the classification process [8–10]. In other words, given the qualitative information that a measure estimates, some features may be favored over others. This fact may lead to discarding features that are relevant in the data set and, if included, would improve the performance of the learned classifier.

Based on these observations, we decided to design and implement a new feature selection algorithm that uses the qualitative information provided by different feature selection measures. The proposed algorithm uses a heuristic for selecting the final feature set based on relevant features for each measure.

The remainder of this paper is structured as follows. Related works are described in Section 2. The proposed algorithm is introduced in Section 3. In Section 4, the experimental results using different intrusion detection data sets are discussed. Finally, the obtained conclusions are outlined in Section 5.

2. Related works

The training data sets used for intrusion detection tasks are characterized by having large number of instances and features [11]. Not all of these features contribute equally to explain the information retained in the data. Thus, a process to determine which features contribute substantially to intrusion detection is needed.

Feature selection processes can be categorized in several ways [12], the most popular and used in the revised literature entail two categories [13]: (1) *Filter* and (2) *Wrapper*.

The features selection algorithms that follow the *Filter* approach use a heuristic that assesses the quality or robustness of the feature subset obtained [14–16]. The main advantages of the *Filter* approach are noise removal, data simplification and the improvement in the performance of any learning method used [13]. Besides, *Filter*-based approaches are capable of facing high dimensional input data.

The *Wrapper*-based approaches use a learning method as a black box and a statistical validation method for avoiding overfitting [17–19]. The criteria used for selecting the better features subset is based on the efficacy achieved by the learning method [17].

After reviewing the literature, it was concluded that the *Filter* approach seems most promising. This is because the performance of the *Wrapper*-based algorithms is greatly affected, in a negative sense, when they deal with large data sets [13]. In addition, the measures used by the *Filter*-based approaches for selecting the best subset of features are often less expensive than computing the efficacy of the learning process, which makes the *Filter*-based approaches more efficient than the *Wrapper* ones. The simplicity and low time complexity of the measures used by the *Filter*-based algorithms make them more suitable for processing large data sets [13].

In the revised literature we observed several measures for feature selection. These measures can be categorized into: (1) Entropy-based measures, (2) Statistical measures and (3) Instances-based measures [20]. Fig. 2 shows a taxonomy of the main feature selection measures considering the former categories.

The measures reviewed in this work assign a score to each feature in the data set. Then, the features scores are sorted in descending order, creating a feature ranking. In such ranking, while higher is the score of a feature, more representative it is in the data set. The kind of information estimated by each measure is described below, from a qualitative point of view.

Information Gain (IG) depends on how much information is available before knowing the value of the feature, and how much information is available after knowing the value of the feature [21]. For instance, if the data set is defined by only one class, it is possible to know the class label without seeing any of the feature value and IG $= 0$. On the other hand, if the classes

to be predicted are represented in equal quantities in the data set and the feature separates the data perfectly according to the classes, then the value of IG = 1.

An extension of *Information Gain* is the *Gain Ratio* (GR) measure, which normalizes IG using as intrinsic value the entropy of a given feature *a* [21]). A drawback of the GR measure is that it can select features just because its intrinsic value is very low, favoring features that have few different values in the data set.

Symmetric Uncertainty (SU) is another measure aimed at compensating for the inherent bias of the IG (IG favors features that have many different values in the data set) by dividing it by the sum of the entropies of a given class *C* and feature *a* [21]. The score assigned by the SU measure is normalized at [0, 1]. A score near 1 indicates a greater correlation between *C* and *a*. Similar to GR measure, SU is biased towards the selection of features with few values.

GR and SU measures eliminate a limitation of IG that considers features with many different values, this elimination may be useful in certain scenarios. An example of this is identifying distinct customer profiles, which is often performed for offering personalized attention and improving the new customer's experiences [22]. In this case, the feature "credit card number" will obtain a high score using IG measure since this feature identifies unambiguously each customer. Nevertheless, this feature does not offer any information that contributes to the specific treatment of any customer profile and does not allows deciding what treatment to give to a new customer. If this data set were processed to detect malicious activities or attacks, the feature "credit card number" could indicate the origin of any fraudulent action or the possible target of an attack; so, this indeed would have valuable information that would be essential to take preventive actions. Therefore, in intrusion detection scenarios, the use of the IG measure allows to select features that provide useful information for malicious activities detection.

Measures described so far were focused on estimating the difference between the a priori and the a posteriori expected uncertain using one feature. Nevertheless, there are other measures, such as the Chi-Square statistic, which estimates the value of a feature from the value of another feature [23]. The Chi-Square statistic is a non-parametric statistical measure that computes the correlation between the distribution of a feature and the distribution of the class.

Besides the described measures, there exist others such as *Relief* (Rf) and *ReliefF* (RfF) that estimate how well a feature can differentiate similar instances from different classes [24]. The *ReliefF* (RfF) measure is an extension of the Rf measure, and its goal is to enhance the effectiveness of working on multi-classes systems, which is a drawback of *Relief*, that is mainly oriented to bi-classes problems. Also, RfF incorporates the KNN (*k*-nearest neighbor) algorithm [25] for searching the nearest neighbors to the instances in the training set, whether they are of the same or different classes.

The previously explained measures have been used in the preprocessing stage of intrusion detection systems [26,27]. Also, they have been used as the starting point for several feature selection algorithms reported for intrusion detection. We now continue with describing the applied approaches and, after that, with providing some identified limitations of them, more specifically in terms of reduced efficacy.

In [28], Anand et al. proposed a feature selection algorithm, in which the IG measure is computed for the training set. After creating the ranking, the feature with the higher score is selected. Then the classification process is conducted using a rules-based classifier [29]. Those instances that were miss-classified were used to create a data subset using some clustering criteria. For each subset created, this classifying process is repeated until each instance is correctly classified.

In [30], the IG measure is used for reducing the dimensionality for malware detection in the Android OS. The authors select the top-10 features in the IG-ranking, diminishing the computational complexity for the classifying process. In [31], Sheen et al. also were focused on improving the efficacy for malware detection in Android through features selection. In their paper, the IG, CS and Rf measures were used independently. Similar to [30], the top-10 ranked features for each measure were selected. Experiments conducted demonstrate that for malware detection in Android, the subset of features selected using the Rf measure allows to obtain the best efficacy during the classification process. A recent approach for malware detection is presented by Wang et al. in [32], where the CS measure is used for features reduction. In this work, several features subsets were selected varying the number of features retained in each subset. This fact allowed to analyze how the performance of the classifiers behaves using each subset of selected features.

The algorithm proposed in [9] is also based on the entropy for feature selection in intrusion detection scenarios. In this case, for each feature, its conditional entropy is calculated and then, it is divided by the number of attacks in the data set. The obtained result is considered as a weight that is assigned to each feature and those features exceeding some threshold value are selected.

The preprocessing stage for intrusion detection tasks proposed in [8] uses the IG measure for dimensionality reduction. The features selected at this stage were those whose score exceeded the threshold value of 0.4, defined by the authors.

The use of statistical measures for feature selection has been also used in intrusion detection. The work of Thaseen and Kumar [10] includes a pre-processing stage where its first step is the data normalization, while the second (and last) step is the feature selection. In their work, the CS measure is used for discarding those features with a score value less than the user-defined threshold value. Thaseen and Kumar also worked in another approach [33] where the CS measure is applied without any previously normalization step. In this case, the threshold value is established in the mean value of all features. Features with values higher than this threshold are selected and regarded as the subset of the optimum features.

Algorithms based on the combination of several measures, known as an *ensemble*, has been also used for intrusion detection. An algorithm that follows this idea is presented by Li et al. [14], where the scores obtained using the IG and CS measures are used for selecting the subset of the final features. The idea followed by Li et al. [14] is to intersect the top-6 scored features in each measure.

Another ensemble-based approach is presented by Prati et al. in [15], which uses 5 of the measures described before (IG, GR, SU, CS, and RfF). The presented algorithm, named *Schwartz Sequential Dropping* (SSD) consists of generating a new ranking from those obtained from the used measures. To do this, the position of each feature a_i in the ranking is compared to the position of the other features, creating pairs of features (a_i, a_j). For each pair created, the number of positions of a_i above of a_j and vice-versa is counted. The feature that has the highest number of positions over the other is considered the winner. Using this information, a directed graph is constructed were features are the nodes and edges are weighted considering the number of positions. The direction of the edge depends on the winner feature. If a_i wins to a_j, an edge is added from a_i to a_j. The graph obtained will always have at least one cycle or a single node that is not defeated by others. In this case, the node is positioned as first in the new ranking, remove it from the initial set and the graph is rebuilt for the remaining features. It will be a unique winner in each iteration, and it will be added to the new ranking according to the sequence in it appears. If there are cycles within the graph,

the edges of the cycle whose weight has the lowest value are eliminated.

An ensemble multi-filter feature selection algorithm (EMFFS) is presented in [16]. This algorithm uses the CS, GR, IG and RfF measures. To select the most important features, the EMFFS algorithm selects a subset of features for each measure, which is one-third of the features of each *ranking* generated, including the features with the highest score. Then, a voting process is performed where those features that are in three or more subsets are selected as the most representative.

Some issues were observed in the *Filter*-based algorithms analyzed in this section that could limit the efficacy of the classifiers. This possible limitation in the classifiers will be tested in the experiments described below. The observed issues are:

1. Most of the algorithms for features selection studied uses only one of the measures described before. Additionally, those algorithms that use more than one measure employ them independently, *i.e.*, the results obtained with one measure do not affect those obtained with another measure. In this sense, the advantages that the combination of different feature selection measures can offer are not exploited, since each measure can estimate different qualitative information of features. This issue was observed in the approaches presented in [8–10,26–28,31,33].
2. Most of the algorithms that use several measures do not perform an analysis of the information that each measure provides. In some cases, the authors do not offer any details, information or evidence for choosing the measures that compose their ensemble. This issue can be found in [14–16].
3. Most of the revised algorithms use a predefined threshold value for selecting the final subset of features. This threshold value is established manually by the authors, and no evidence is given about how the threshold is selected. This issue is observed in [15,26,30–32].

Considering issue 1, the combination of several measures allows to select different features, but all of them representative of the data set. Regarding issue 2, an analysis of the information provided by each measure could justify the fact that certain features are favored by some measures and not by others. About issue 3, in real problems, providing to the specialists a subset of features automatically selected, not only makes their work more affordable, but also contributes to build the classification model more efficiently.

3. The new multi-measure feature selection algorithm

After reviewing the state-of-the-art and identifying the issues that the *Filters*-based approaches entail, it is hypothesized that a smart combination of several measures can lead to the selection of the most representative features in the data set. The analysis carried out in the previous section allowed us to define two fundamental aspects that may improve the results of ensemble-based algorithms. The first aspect consists of defining the measures that are used in the feature selection algorithm. For this, it is important to identify and combine measures that estimate different qualitative information in the features. In this way, we can deal with issues 1 and 2, described in the previous section. The second aspect concerns the strategy to select the final feature set. Using a strategy that allows us to automatically select the relevant features from the data set will help us deal with issue 3, described in the previous section.

The taxonomy represented in Fig. 2 groups the most used feature selection measures into three groups: Entropy-based measures, Statistical-based measures, and Instance-based measures. The algorithm proposed in this work uses a representative measure of each group. This is supported by the fact that each group evaluates different information in the features. According to the analysis performed in the previous section, IG is the most widely used entropy-based measure. There are comparative studies where IG and CS are reported as the most effective feature selection measures for classification tasks [34]. From a conceptual point of view, IG measures the amount of information that a feature can provide to the process of determining whether an instance belongs to one class or another.

CS is a non-parametric statistical measure that estimates the correlation between the distribution of a feature and the distribution of the class. Conceptually, it can be said that CS measures the degree of dependence of a feature to its class. Its high efficiency contributes to its practical application in intrusion detection scenarios, being the most used measure of those analyzed in the previous section.

Finally, from the group of Instance-based measures, RfF is better suited than Rf to intrusion detection scenarios. The main cause is due to the fact that the Rf is focused to binary classification problems, while RfF is better suited for multi-class problems, which is very common in intrusion detection scenarios. Conceptually, RfF measures how well a feature can differentiate instances that belongs to different classes, looking for the closest neighbors of the same and different classes.

Based on the conceptual information estimated by the IG, CS and RfF measures, they were used in the Multi Measure Feature Selection Algorithm (MMFSA) proposed in this work, which is described in the Algorithm 1.

Algorithm 1: MMFSA(D)

Input: D: *Data set*
Output: $\check{D}$: *Reduced data set*

$\check{A}, \check{A}_{RfF}, \check{A}_{CS}, \check{A}_{IG} \leftarrow \emptyset$ // *Features set*
processList $\leftarrow$ [] // *List of processes that will be executed in parallel*
processList.Add(FeatureSelector(RfF, D))
processList.Add(FeatureSelector(CS, D))
processList.Add(FeatureSelector(IG, D))
foreach *process* ***in*** *processList* **do**
    process.Start() // *Each feature selection process is started*
end
foreach *process* ***in*** *processList* **do**
    process.Join() // *When each feature selection process finished, the results are joined.*
    if process.Measure() == RfF **then** $\check{A}_{RfF} \leftarrow$ process.Result()
    else if process.Measure() == CS **then** $\check{A}_{CS} \leftarrow$ process.Result()
    else if process.Measure() == IG **then** $\check{A}_{IG} \leftarrow$ process.Result()
end
$\check{A} \leftarrow \check{A}_{RfF} \cup \check{A}_{CS} \cup \check{A}_{IG}$
$\check{D} \leftarrow$ Reduce($D, \check{A}$)
return $\check{D}$

MMFSA receives a data set D as input to reduce its dimensionality. In order to not affect the efficiency of MMFSA, the parallel computation of the three selected measures was performed, applying the principle of task parallelism [35]. This process consists of assigning a task to each processor, *i.e.*, a specific measure of those selected, in such a way that each processor carries out its own sequence of operations. This procedure provides greater

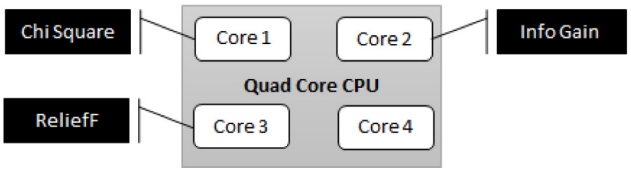

Fig. 3. Task parallelism exploited in MMFSA.

efficiency than running each measure sequentially. Fig. 3 depicts this idea.

The Algorithm 2 performs the features selection process considering a feature selection measure M and a data set D. First, the scores assigned by M to each feature in D are computed. As described in the previous section, a measure M assigns a score to each feature in the data set, therefore, for a measure M a set of scores P_M is obtained (line 1 in Algorithm 2). Next, the mean $\overline{m}_M$ of the scores assigned by M is computed (line 3 in Algorithm 2). Following, the scores values in P_M are traversed selecting those features whose score $m_a > \overline{m}_M$ (lines 4–8 in Algorithm 2). Finally, the set of selected features $\breve{A}_M$ are returned for M (line 9 in Algorithm 2)

Algorithm 2: FeatureSelector(M, D)

Input: M: *Feature selection measure used for determining the score of each feature*, D: *Data set*

Output: $\breve{A}_M$: *Set of features selected by M*

1 $P_M \leftarrow$ Scores(M, D) // *Returns the scores assigned by M to each feature in D.*
2 $\breve{A}_M \leftarrow \emptyset$
3 $\overline{m}_M \leftarrow$ Mean-Score(P_M)
4 **foreach** $m_a \in P_M$ **do**
5 **if** $m_a > \overline{m}_M$ **then**
6 $\breve{A}_M \leftarrow \breve{A}_M \cup \{a\}$
7 **end**
8 **end**
9 **return** $\breve{A}_M$

As it is described in lines 3–5 of Algorithm 1, the *FeatureSelector*(M, D) algorithm is added to the processes list for parallel execution of each measure (lines 6–8 in Algorithm 1). Following, Algorithm 1 waits until each process finished (lines 9–14 in Algorithm 1), where a set of features are selected by each measure (lines 11–13 in Algorithm 1). Next, an union operation is performed among the features set selected by each measure, obtaining the final set $\breve{A}$ (line 15 in Algorithm 1). Finally, the reduced data set $\breve{D}$ is created as a result of representing D with the features in $\breve{A}$ (line 16 in Algorithm 1).

Fig. 4 shows the processing scheme proposed for obtaining the reduced data set $\breve{D}$. First, the three measures are executed in parallel on the data set D. Once the representative features for each measure have been selected, the union of these features is performed, obtaining the final set of features $\breve{A}$. Next, the data set D is reduced using the features selected in $\breve{A}$. As a result of these operations, a reduced data set $\breve{D}$ is obtained.

4. Experiments

To evaluate the proposed algorithm, several experiments were carried out using classifiers from different families, which have been traditionally used in intrusion detection tasks (see Table 1) [36,37].

The experiments were conducted on a PC equipped with a 2.5 GHz Intel Quad-Core processor, 4 GB of RAM memory running Ubuntu 18.04 OS. The data sets used for evaluating MMFSA is described in the next section.

Table 1
Classifiers used for validating the feature selection algorithm (FFMSA) proposed.

Classifier	Family
Classification and Regression Trees (CART) [38] C4.5 [39]	Tree-based
K-Nearest Neighbors (KNN) [25]	Lazy
Support Vector Machine (SVM) [40]	Functions-based
Nearest Neighbor with Generalization (NNge) [41] OneR [42] PART [43] RIpple-DOwn Rule (RIDOR) [44] Decision-Table (DT) [45] Conjunctive Rule (CR) [46]	Rules-based

4.1. Selected data sets

The information processed in the scenarios addressed in this work usually contains sensitive user data. Therefore, working with private information of the users implies that the data has a certain level of confidentiality that does not allow it to be public. In this sense, it is a complex task to obtain data sets from intrusion detection scenarios. However, it has been possible to access some of the most widely used data sets, which are described below.

KDD'99 [47] is considered as a reference data set and has been widely used for intrusion detection tasks [48]. This data set is composed of a wide variety of simulated intrusions in a military network. The training data set is composed of 22 different types of intrusions, plus the "normal" class, while the test data set contains 37 intrusion types, plus the "normal" class. All these intrusions are associated with 4 categories:

- Probe: Surveillance and others probes.
- DoS: Denial of service.
- U2R: Unauthorized access to local superuser privileges (root).
- R2L: Unauthorized access from a remote machine.

Similarly to other reviewed works, all the instances in the KDD'99 data set were classified into two major classes: "attack" and "normal". In consequence, the number of instances that composes the training set is 494,021, while the test set contains 311,029 instances. Each instance is composed of 41 features, of which 9 are discrete and 32 are continuous.

From the statistical analysis of the KDD'99 data set, a new one named NSL-KDD [49] was proposed. The NSL-KDD training set consists of 125,973 instances, while the test set is made up of 22,544 instances. Similarly to KDD'99, the NSL-KDD instances consist of 41 features, 9 discrete and 32 continuous. Each instance can be labeled "anomaly" or "normal".

Another popular data set is the CDMC2012 [50], which was created using several *honeypots* from five different networks. Among the instances of CDMC2012, there are some labeled as "unknown" which were discarded retaining only those ones labeled as "attack" or "normal". Without loss of generality, this data set was divided into two sets, a training set containing 48,357 instances and a test set with 80,000. Each instance is represented by 14 features, including the class.

The CDMC2013 [51] data set was created from a real intrusion detection system. Similarly to CDMC2012 it is necessary to divide the data into 2 sets. Without loss of generality, CDMC2013 was divided into a training set which contains 40,000 instances and

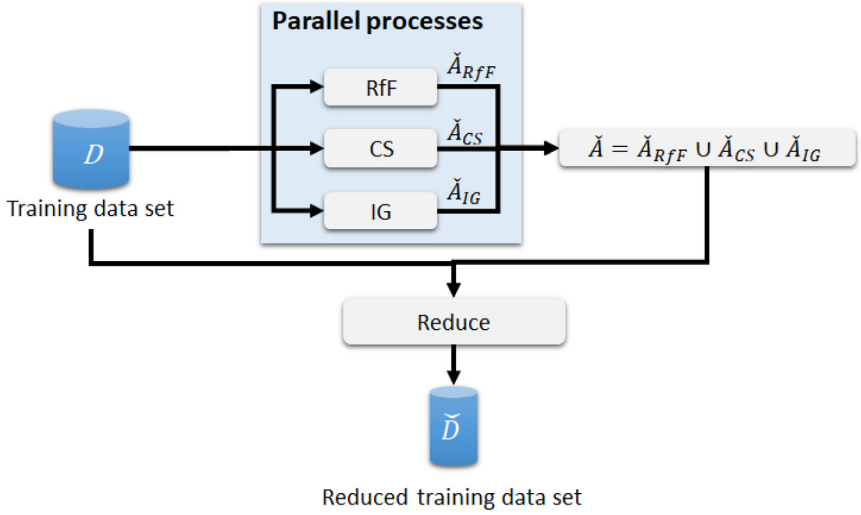

Fig. 4. Processing scheme for feature selection implemented by MMFSA.

the test set, with 37,959 instances. Each instance is composed of 7 numeric features, plus the class.

The presented data sets have been used in international competitions for evaluating the performance of several classifiers in intrusion detection tasks. The experimental results were carried out as follows. For each data set, its training set was reduced using a feature selection algorithm. The reduced data set is then used by a classifier to build its classification model, which is evaluated on the test set. The achieved results on such data sets are presented in the next section.

4.2. Experimental results

To evaluate the quality of the reduced data set obtained by MMFSA, a comparison with algorithms using the feature selection measures individually was performed. Also, the algorithms proposed in [14–16] (which are based on *ensemble* strategies) were used as baseline for comparison purposes. The comparison was performed considering the efficacy achieved by the aforementioned classifiers, employing the reduced data set obtained by each feature selection algorithm for training. For this, the *accuracy* (*Acc*) measure was used, which is computed according to Eq. (1), where T^+, T^-, F^+ and F^- represent true positives, true negatives, false positives, and false negatives respectively.

$$Acc = \frac{T^+ + T^-}{T^+ + T^- + F^+ + F^-} * 100. \quad (1)$$

In addition, an analysis of the effectiveness provided by the feature selection algorithms to the results achieved in each data set is included. The effectiveness is given by the false positive rate (FPR) and the false negative rate (FNR) quality measures, defined in Eq. (2) and Eq. (3) respectively.

$$FPR = \frac{F^+}{T^- + F^+} * 100. \quad (2)$$

$$FNR = \frac{F^-}{T^+ + F^-} * 100. \quad (3)$$

Those algorithms using the selection measures individually as well as the algorithm proposed in [15], need a predefined number of features l. To make a fair comparison, it was established that $l = |\check{A}|$, where $\check{A}$ is the set of automatically selected features whose cardinality provides the best accuracy results to algorithms that require a predefined value of l. For this, an experiment was conducted with the aim of evaluating the efficacy obtained with these algorithms using the values $l = 6$, $l = 13$ and $l = 17$, obtained by [14,16] and MMFSA respectively. Table 2 shows that the efficacy achieved by the classifiers using the reduced data sets with $l = 17$ exceeds the efficacy achieved with $l = 6$ and $l = 13$.

In this sense, for the KDD'99 training set, the value of $l = 17$ (selected by MMFSA) was used to select the most representative features of: (1) each measure used individually and (2) the algorithm proposed in [15]. The algorithm presented in [14] selected 6 features, while the approach reported in [16] kept 13 features. The results achieved by each classifier (concerning the efficacy) using the different feature selection strategies are shown in Table 3. From these results, it is noticed that the efficacy achieved by the classifiers is higher when MMFSA is used as features selector. An exception is the CR classifier, but in this case the efficacy obtained using MMFSA is the same as that achieved using IG and CS. On the other hand, the C4.5 classifier obtains the best result using the approach reported in [15], while the RIDOR classifier obtains its higher efficacy using GR. Nevertheless, the other classifiers achieve their best efficacy using MMFSA. From this experiment it can be concluded that MMFSA allows an improvement in classification accuracy by selecting the relatively best features subset in the KDD'99 data set.

Considering the same analysis and experiment carried out previously, the efficacy achieved by the classifiers using the NSL-KDD reduced data set with $l = 19$ (which is obtained by MMFSA) was higher than the reported with $l = 6$ and $l = 14$, obtained using [14,16] respectively (see Table 4). When the NSL-KDD data set is reduced using the IG and CS, all the evaluated classifiers obtain the same results since the reduced data sets are the same for the l value evaluated. Similar behavior is observed for the CDMC2012 data set (see Table 6).

Table 5 shows the efficacy achieved by classifiers using different features selection algorithms on the NSL-KDD data set. Considering Table 4, the number of 19 features was selected for the algorithms that uses single feature selection measure and for the approach reported in [15]. In NSL-KDD data set, when CART, NNge and OneR classifiers are applied, a tie is reached with the best result using different feature selection algorithms, being MMFSA one of them. Considering the C4.5 classifier, it obtains its best efficacy using the approach proposed in [15], outperforming MMFSA by a slight difference. The same behavior is reported with the CR classifier on the NLS-KDD data set when it is reduced using GR. The remaining 5 classifiers obtains their best efficacy when MMFSA is used as a data reduction strategy. In general, MMFSA

Table 2
Efficacy achieved with algorithms that require a predefined number of features using KDD'99 data set.

Algorithm	CART	C4.5	KNN	NNge	OneR	PART	RIDOR	SVM	DT	CR
CS ($l = 6$)	90.90	90.11	90.58	90.69	88.69	91.16	91.59	89.77	91.87	89.94
CS ($l = 13$)	92.19	91.78	91.92	91.58	90.15	92.88	92.37	90.93	93.09	90.39
CS ($l = 17$)	**93.42**	**92.49**	**92.63**	**92.76**	**90.70**	**93.27**	**93.01**	**91.62**	**93.90**	**91.17**
GR ($l = 6$)	91.29	91.03	90.81	90.84	88.32	90.27	91.50	89.85	91.92	89.42
GR ($l = 13$)	92.82	92.15	92.05	92.74	89.84	92.00	92.83	90.88	93.03	90.22
GR ($l = 17$)	**93.56**	**93.02**	**92.70**	**93.26**	**90.76**	**92.58**	**93.52**	**91.69**	**93.87**	**91.17**
IG ($l = 6$)	90.90	90.11	90.58	90.69	88.69	91.16	91.59	89.77	91.87	89.94
IG ($l = 13$)	92.19	91.78	91.92	91.58	90.15	92.88	92.37	90.93	93.09	90.39
IG ($l = 17$)	**93.36**	**92.43**	**92.59**	**92.68**	**90.79**	**93.30**	**92.77**	**91.73**	**93.87**	**91.03**
RfF ($l = 6$)	90.49	90.36	90.48	90.61	88.81	90.24	90.63	89.06	91.39	88.77
RfF ($l = 13$)	91.34	91.31	91.38	91.43	90.11	91.36	91.09	90.96	92.51	89.91
RfF ($l = 17$)	**92.31**	**92.36**	**92.57**	**93.11**	**90.83**	**92.59**	**92.25**	**91.97**	**93.50**	**90.76**
[15] ($l = 6$)	90.82	90.37	90.52	90.59	88.64	91.42	91.48	89.91	91.44	89.34
[15] ($l = 13$)	92.09	91.98	91.73	91.68	89.92	92.83	92.25	91.12	93.03	89.77
[15] ($l = 17$)	**93.50**	**93.12**	**92.41**	**92.84**	**90.76**	**93.18**	**93.30**	**91.73**	**93.87**	**91.11**

Table 3
Efficacy achieved using the KDD'99 data set.

Algorithm	CART	C4.5	KNN	NNge	OneR	PART	RIDOR	SVM	DT	CR
CS ($l = 17$)	93.42	92.49	92.63	92.76	90.70	93.27	93.01	91.62	93.90	**91.17**
GR ($l = 17$)	93.56	93.02	92.70	93.26	90.76	92.58	**93.52**	91.69	93.87	**91.17**
IG ($l = 17$)	93.36	92.43	92.59	92.68	90.79	93.30	92.77	91.73	93.87	91.03
RfF ($l = 17$)	92.31	92.36	92.57	93.11	90.83	92.59	92.25	91.97	93.50	90.76
[14]	90.90	90.11	90.58	90.69	88.69	91.16	91.59	89.77	91.87	89.94
[15] ($l = 17$)	93.50	**93.12**	92.41	92.84	90.76	93.18	93.30	91.73	93.87	91.11
[16]	93.18	92.41	92.57	92.69	90.70	92.88	93.37	91.61	93.87	**91.17**
MMFSA	**93.88**	92.53	**92.76**	**93.34**	**90.88**	**93.38**	93.39	**91.99**	**93.99**	**91.17**

Table 4
Efficacy achieved with algorithms that require a predefined number of features using NSL-KDD data set.

Algorithm	CART	C4.5	KNN	NNge	OneR	PART	RIDOR	SVM	DT	CR
CS, IG ($l = 6$)	73.28	64.92	63.72	71.48	65.20	63.96	63.23	62.08	60.49	73.53
CS, IG ($l = 14$)	79.19	73.23	72.84	79.37	74.61	73.97	72.89	70.86	69.11	80.59
CS, IG ($l = 19$)	**82.13**	**78.77**	**77.61**	**82.85**	**81.38**	**79.62**	**78.11**	**74.31**	**72.60**	**83.94**
GR ($l = 6$)	71.76	69.82	68.18	69.98	70.12	67.89	70.01	65.23	62.24	73.35
GR ($l = 14$)	78.48	74.29	72.55	75.03	75.47	73.06	75.31	70.95	68.41	79.92
GR ($l = 19$)	**81.11**	**78.87**	**77.01**	**81.10**	**81.38**	**78.13**	**80.07**	**75.10**	**72.60**	**84.04**
RfF ($l = 6$)	71.61	64.49	65.37	66.06	65.13	72.12	66.90	64.89	63.82	57.54
RfF ($l = 14$)	77.54	73.05	73.28	73.74	72.98	77.93	73.72	72.63	71.08	62.44
RfF ($l = 19$)	**79.29**	**76.02**	**77.82**	**78.28**	**75.50**	**80.81**	**76.81**	**75.16**	**73.19**	**66.88**
[15] ($l = 6$)	72.61	66.86	66.22	72.52	72.08	71.82	72.41	66.07	63.39	73.60
[15] ($l = 14$)	78.13	74.75	73.52	77.93	76.89	76.35	77.00	72.26	70.92	80.04
[15] ($l = 19$)	**81.06**	**78.92**	**77.75**	**81.50**	**80.97**	**79.94**	**81.17**	**75.10**	**73.07**	**83.88**

Table 5
Efficacy achieved using the NSL-KDD data set.

Algorithm	CART	C4.5	KNN	NNge	OneR	PART	RIDOR	SVM	DT	CR
CS ($l = 19$)	**82.13**	78.77	77.61	**82.85**	**81.38**	79.62	78.11	74.31	72.60	83.94
GR ($l = 19$)	81.11	78.87	77.01	81.10	**81.38**	78.13	80.07	75.10	72.60	**84.04**
IG ($l = 19$)	**82.13**	78.77	77.61	**82.85**	**81.38**	79.62	78.11	74.31	72.60	83.94
RfF ($l = 19$)	79.29	76.02	77.82	78.28	75.50	80.81	76.81	75.16	73.19	66.88
[14]	73.28	64.92	63.72	71.48	65.20	63.96	63.23	62.08	60.49	73.53
[15] ($l = 19$)	81.06	**78.92**	77.75	81.50	80.97	79.94	81.17	75.10	73.07	83.88
[16]	80.91	78.30	78.13	79.06	**81.38**	73.94	81.40	75.94	73.78	83.94
MMFSA	**82.13**	78.81	**78.21**	**82.85**	**81.38**	**80.92**	**81.51**	**75.95**	**73.88**	83.98

contributes to obtain a better efficacy than the other strategies evaluated on the NSL-KDD data set.

As it is shown in Table 6, the efficacy achieved by the classifiers using the reduced CDMC2012 data set with $l = 9$ (this value was obtained by MMFSA) is higher than the efficacy achieved by the same classifiers when the data set CDMC2012 was reduced with $l = 3$ and $l = 6$, as it is reported by [14,16] respectively. Based on the above, both the individual measures and the algorithm proposed in [15] used the value of $l = 9$ (see Table 7).

Table 7 shows the efficacy achieved using CDMC2012 data set. OneR, CR and PART classifiers achieve their best efficacy with different feature selection strategies, being MMFSA one of them. The KNN classifier obtains its best accuracy result using the CDMC2012 data set reduced by GR. The remaining 6 classifiers report the highest efficacy using MMFSA as a data reduction strategy. In general, the classifiers obtain higher efficacy when they use the training data set reduced by MMFSA.

In the case of the CDMC2013 data set, the same efficacy is reported using CS, IG, GR and the approach reported in [15], both

Table 6
Efficacy achieved with algorithms that require a predefined number of features using CDMC2012 data set.

Algorithm	CART	C4.5	KNN	NNge	OneR	PART	RIDOR	SVM	DT	CR
CS, IG ($l = 3$)	97.37	97.21	96.61	96.14	95.98	97.11	96.84	95.75	95.62	95.98
CS, IG ($l = 6$)	97.92	98.04	97.17	97.54	97.40	98.29	97.95	97.40	97.24	97.08
CS, IG ($l = 9$)	**99.17**	**99.12**	**98.76**	**98.91**	**98.52**	**99.20**	**99.12**	**98.38**	**98.71**	**98.52**
GR ($l = 3$)	96.87	96.42	95.17	94.79	95.08	96.91	96.91	94.58	95.34	95.74
GR ($l = 6$)	98.11	97.99	96.83	97.08	97.61	97.76	97.88	96.73	97.23	96.97
GR ($l = 9$)	**99.04**	**99.08**	**98.94**	**98.89**	**98.52**	**99.06**	**99.08**	**98.07**	**98.64**	**98.52**
RfF ($l = 3$)	94.31	94.88	93.86	94.17	93.29	94.52	94.52	92.85	93.56	93.29
RfF ($l = 6$)	97.24	97.42	96.92	97.01	96.65	97.10	97.10	96.09	96.77	96.65
RfF ($l = 9$)	**99.04**	**99.11**	**98.91**	**99.04**	**98.52**	**99.06**	**99.06**	**98.42**	**98.97**	**98.52**
[15] ($l = 3$)	95.76	96.06	94.37	94.05	93.63	94.96	94.96	93.33	93.57	93.14
[15] ($l = 6$)	97.33	97.54	95.93	95.61	95.12	97.18	97.18	95.22	95.64	95.70
[15] ($l = 9$)	**99.06**	**99.09**	**98.81**	**98.95**	**98.50**	**99.08**	**99.08**	**98.20**	**98.84**	**98.49**

Table 7
Efficacy achieved using CDMC2012 data set.

Algorithm	CART	C4.5	KNN	NNge	OneR	PART	RIDOR	SVM	DT	CR
CS ($l = 9$)	99.17	99.12	98.76	98.91	**98.52**	**99.20**	99.12	98.38	98.71	**98.52**
GR ($l = 9$)	99.04	99.08	**98.94**	98.89	**98.52**	99.06	99.08	98.07	98.64	**98.52**
IG ($l = 9$)	99.17	99.12	98.76	98.91	**98.52**	**99.20**	99.12	98.38	98.71	**98.52**
RfF ($l = 9$)	99.04	99.11	98.91	99.04	**98.52**	99.06	99.06	98.42	98.97	**98.52**
[14]	97.92	98.04	97.17	97.54	97.40	98.29	97.95	97.40	97.24	97.08
[15] ($l = 9$)	99.06	99.09	98.81	98.95	98.50	99.08	99.08	98.20	98.84	98.49
[16]	97.37	97.21	96.61	96.14	95.98	97.11	96.84	95.75	95.62	95.98
MMFSA	**99.20**	**99.13**	98.92	**99.05**	**98.52**	**99.20**	**99.13**	**98.43**	**99.00**	**98.52**

for $l = 4$ and for $l = 6$ (see Table 8). Considering RfF, only the KNN classifier is affected when the data set is reduced with $l = 4$ instead of $l = 6$. In this sense, the result reported by RfF ($l = 6$) was used in the comparison shown in Table 9.

The CDMC2013 data set is composed of 7 features, and it is considered a small data set. Because of this, different features selection algorithms select the same features. In such way, the algorithm proposed in [15], CS, GR, IG and MMFSA selected the same 4 features, whereby, the values shown in Table 9 are the same. The approach proposed in [14] and RfF selected 6 features, while the approach proposed in [16] selected only 3 features. The classifiers reach their best efficacy on CDMC2013 data set using each one of the feature selection algorithms evaluated, except the algorithm proposed in [16].

The Table 10 shows the overall effectiveness achieved with the feature selection algorithms evaluated on different data sets. For this, the overall *FPR* ($OFPR_D$) and the overall *FNR* ($OFNR_D$) provided by each feature selection algorithm for a given data set D are reported. $OFPR_D$ and $OFNR_D$ are defined in Eqs. (4) and (5) respectively, where K is the set of classifiers used, $FPR_{i,D}/FNR_{i,D}$ represents the *FPR*/*FNR* achieved by ith classifier in D.

$$OFPR_D = \frac{\sum_i^{|K|} FPR_{i,D}}{|K|}. \tag{4}$$

$$OFNR_D = \frac{\sum_i^{|K|} FNR_{i,D}}{|K|}. \tag{5}$$

The MMFSA algorithm provided the best results, in terms of effectiveness, in three different data sets (KDD'99, NSL-KDD and CDMC2012). In the case of the CDMC2013 data set, the best false positive rate is reported by RfF and [14], with a minimal difference regarding to that reported by CS, GR, IG, [15] and MMFSA. These latter algorithms reported the best false positive rate in the CDMC2013 data set.

As can be seen in Table 10, the worst effectiveness, specifically in terms of OFPR, is reported in the NSL-KDD data set. Although the objective of this work is not to obtain the best classifier, but rather the best quality reduced data set, we consider that a more in-depth analysis regarding the distribution by classes of interest could lead to better results. The foregoing is raised considering that the imbalanced data sets are common in intrusion detection scenarios [52–54]. The imbalanced data set problem occurs when the size of normal traffic exceeds that of attack traffic. This fact means that the instances belonging to the attack class are often ignored as they are poorly represented in the training set compared to the normal class. In future works, this problem could be addressed from the perspective of the use of cost-sensitive classifiers, which contribute to a better result on imbalanced data sets.

From the experiments conducted it can be noticed that the difference in the efficacy obtained by the classifiers is apparently minimal, however in intrusion detection scenarios, this difference can represent a high number of instances. In the KDD'99 data set a difference of 0.05 represents 156 incorrectly classified instances. Such number of instances, in a real scenario and depending on the type of attack, could be increased [55]. A typical example is the DoS-type attacks using ICMP packets. In this type of attack, a large number of ICMP packets are sent to the victim. Its effect can be multiplied through the use of poorly configured networks on the Internet. This happens when an attacker spoofs the return address of the ICMP packet from the command *ping*,[1] replacing it with the victim's address and sending it to the broadcast address of a network. When the servers on such network respond to the ping request, all responses are directed to the victim, amplifying the attack. In this sense, any improvement achieved in terms of efficacy has a high impact in the detection and prevention of possible attacks.

The results reported by [14] differ strongly from those achieved by the other feature selection algorithms. This may be conditioned by the limitation of its selection strategy, which restricts the maximum number of features to be selected to 6. Such restriction can lead to a drastic reduction in the data set dimensionality, and therefore, the loss of useful information, affecting the classification process.

The approach proposed in [16] selected fewer features than MMFSA on most of data sets. However, none of the classifiers

[1] Ping is a diagnostic tool that allows verifying the connection status of a local host with at least one remote computer on a TCP/IP network.

Table 8
Efficacy achieved with algorithms that require a predefined number of features using CDMC2013 data set.

Algorithm	CART	C4.5	KNN	NNge	OneR	PART	RIDOR	SVM	DT	CR
CS, IG, GR, [15] ($l = 3$)	98.97	99.06	99.83	98.97	91.27	99.83	99.08	98.74	99.08	86.24
CS, IG, GR, [15] ($l = 4$)	**99.86**	**99.83**	**99.83**	**99.86**	**92.15**	**99.83**	**99.86**	**99.74**	**99.86**	**87.57**
CS, IG, GR, [15] ($l = 6$)	**99.86**	**99.83**	**99.83**	**99.86**	**92.15**	**99.83**	**99.86**	**99.74**	**99.86**	**87.57**
RfF ($l = 3$)	98.74	99.02	99.24	98.74	91.96	98.74	98.68	99.05	98.98	86.33
RfF ($l = 4$)	**99.86**	**99.83**	99.77	**99.86**	**92.15**	**99.83**	**99.86**	**99.74**	**99.86**	**87.57**
RfF ($l = 6$)	**99.86**	**99.83**	**99.83**	**99.86**	**92.15**	**99.83**	**99.86**	**99.74**	**99.86**	**87.57**

Table 9
Efficacy achieved using CDMC2013 data set.

Algorithm	CART	C4.5	KNN	NNge	OneR	PART	RIDOR	SVM	DT	CR
CS, IG, GR, [15], MMFSA ($l = 4$) RfF, [14] ($l = 6$)	**99.86**	**99.83**	**99.83**	**99.86**	**92.15**	**99.83**	**99.86**	**99.74**	**99.86**	**87.57**
[16]	98.97	99.06	**99.83**	98.97	91.27	**99.83**	99.08	98.74	99.08	86.24

Table 10
Overall effectiveness achieved using different data sets.

Quality measures	CS	GR	IG	RfF	[14]	[15]	[16]	MMFSA
$OFNR_{KDD'99}$	8.86	8.72	8.91	9.18	11.17	8.76	8.92	**8.58**
$OFPR_{KDD'99}$	7.18	7.07	7.22	7.44	9.06	7.1	7.23	**6.95**
$OFNR_{NSL-KDD}$	14.59	14.73	14.59	16.8	23.64	14.45	14.91	**14.01**
$OFPR_{NSL-KDD}$	25.62	25.83	25.62	29.49	41.51	25.37	26.18	**24.60**
$OFNR_{CDMC2012}$	0.99	1.05	0.99	0.98	2.07	1.02	3.05	**0.93**
$OFPR_{CDMC2012}$	1.35	1.41	1.35	1.32	2.79	1.38	4.10	**1.26**
$OFNR_{CDMC2013}$	**2.14**	**2.14**	**2.14**	2.22	2.22	**2.14**	2.89	**2.14**
$OFPR_{CDMC2013}$	2.17	2.17	2.17	**2.16**	**2.16**	2.17	2.9	2.17

using the approach reported in [16] obtained a higher efficacy than that achieved using MMFSA. An interesting fact is that the approach reported in [16] combines 4 features selection measures (CS, GR, IG and RfF), where 3 of them are included in MMFSA (CS, IG and RfF). The reason why MMFSA obtains better efficacy than the other ensemble-based algorithms, considered in this work, is because MMFSA is oriented to preserve the features that are relevant (those features above the P_M mean) for each of the measures that it combines. To accomplish this, MMFSA is based on the fact that each features selection measure estimates different information in the data, so the degree of relevance of a feature may differ between the measures. For this reason, the final set of selected features may include those that were relevant for only one measure; contrary to the [16] algorithm where the final set is made up of features selected as relevant for at least three measures, without considering what kind of information each measure estimates. This can lead to discard features that provide useful information, but were selected as relevant by a single measure.

In some of the evaluated data sets, the GR measure and the approach reported in [15] were more valuable than MMFSA for certain classifiers. For example, the C4.5 classifier obtain its best efficacy, in the KDD'99 and NSL-KDD data sets, using the algorithm reported in [15]. On the other hand, the GR measure allows to the RIDOR, CR and KNN classifiers to achieve their best efficacy processing the KDD'99, NSL-KDD and CDMC2012 data sets respectively. Nevertheless, in all other cases, the classifiers performed at their best using the MMFSA algorithm. In this sense, the experimental results show that the MMFSA algorithm makes it possible to achieve greater efficacy than the measures it combines, when they are used individually. MMFSA also allows to the classifiers achieve higher accuracy than that achieved with the other ensemble-based algorithms evaluated in this work.

The results obtained by the evaluated algorithms show that the issues observed, and discussed in Section 2, negatively affect the efficacy of the classifiers. The main evidence supporting the above statement is that MMFSA deals with such issues and makes classifiers more efficacious.

5. Conclusions

This work introduced a feature selection algorithm based on the combination of three measures, where each measure estimates different qualitative information in the features. Experimental evidence has been found that the observed issues in the feature selection algorithms entails a limitation in the classifiers, specifically in the efficacy achieved. The experiments conducted show that MMFSA outperforms, in terms of the classifier efficacy, each of the measures that it combines when they are used individually and the rest of the feature selection algorithms compared. This may be explained by the fact that the observed issues are dealt by the proposed algorithm. That is, MMFSA uses several feature selection measures, that, apparently, helps to select the more relevant features. In addition, the proposed algorithm entails a step to select the best features without manually predefining a number x of features, which is not present in most of the algorithms analyzed.

However, such step does not have a theoretical foundation. In this sense, we drag a problem present in many proposals that require the choice of a random parameter value. The optimal parameter value could have been determined by testing several thresholds, either according to the score achieved by each measure or the number of features to select. However, this could lead to overfitting having as consequence a resulting classification model with less predictive power. We consider that remaining limitation could be studied in depth in future work.

Additional future work will be focused on evaluating MMFSA in other application domains such as wireless sensor networks, fraud detection in telephony and banking transactions, to further test its robustness. Furthermore, we intend to compare MMFSA with other feature selection algorithms that follow a different approach from the one addressed in this work and have been applied in these scenarios [56].

CRediT authorship contribution statement

Vitali Herrera-Semenets: Conceptualization, Methodology, Software, Formal analysis, Investigation, Writing - original draft. **Lázaro Bustio-Martínez:** Resources, Validation. **Raudel Hernández-León:** Supervision, Project administration, Formal analysis. **Jan van den Berg:** Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] D.P. Acharjya, N. Syed Siraj Ahmed, Tracing of online assaults in 5G networks using dominance based rough set and formal concept analysis, Peer-To-Peer Network. Appl. 14 (1) (2021) 349–374.

[2] Cisco, The internet of things, 2020, https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/iot-aag.pdf (Accessed 5 September 2020).

[3] Cybersecurity Ventures, Global cybercrime damages predicted to reach $6 trillon annually by 2021, 2020, https://cybersecurityventures.com/cybercrime-damages-6-trillon-by-2021/ (Accessed 5 September 2020).

[4] Gestión, Ciberseguros: Estos son los sectores más expuestos a un ataque de hackers, 2017, https://gestion.pe/tecnologia/ciberseguros-son-sectores-expuestos-ataque-hackers-138595-noticia/ (Accessed 6 September 2020).

[5] Guillermo Francia, Levent Ertaul, Luis Hernandez Encinas, Eman El-Sheikh, Kevin Daimi, Computer and Network Security Essentials, Springer, 2018, p. 618.

[6] Charu C. Aggarwal, Data Mining: The Textbook, Springer, 2015.

[7] Abdulla Amin Aburomman, Mamun Bin Ibne Reaz, Survey of learning methods in intrusion detection systems, in: 2016 International Conference on Advances in Electrical, Electronic and Systems Engineering (ICAEES), IEEE, 2016, pp. 362–365.

[8] Shadi Aljawarneh, Monther Aldwairi, Muneer Bani Yassein, Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model, J. Comput. Sci. 25 (2018) 152–160.

[9] Suleman Khan, Abdullah Gani, Ainuddin Wahid Abdul Wahab, Prem Kumar Singh, Feature selection of denial-of-service attacks using entropy and granular computing, Arab. J. Sci. Eng. 43 (2) (2018) 499–508.

[10] I. Sumaiya Thaseen, Ch Aswani Kumar, Intrusion detection model using chi square feature selection and modified Naïve Bayes classifier, in: Proceedings of the 3rd International Symposium on Big Data and Cloud Computing Challenges (ISBCC–16), Springer, 2016, pp. 81–91.

[11] Vitali Herrera-Semenets, Osvaldo Andrés Pérez-García, Andrés Gago-Alonso, Raudel Hernández-León, Classification rule-based models for malicious activity detection, Intell. Data Anal. 21 (5) (2017) 1141–1154.

[12] S. Krishnaveni, S. Sivamohan, S.S. Sridhar, S. Prabakaran, Efficient feature selection and classification through ensemble method for network intrusion detection on cloud computing, Cluster Comput. (2021) 1–19.

[13] Salvador García, Julián Luengo, Francisco Herrera, Data Preprocessing in Data Mining, Springer, 2016.

[14] Yang Li, Bin-Xing Fang, You Chen, Li Guo, A lightweight intrusion detection model based on feature selection and maximum entropy model, in: Communication Technology, 2006. ICCT'06. International Conference on, Ieee, 2006, pp. 1–4.

[15] Ronaldo C. Prati, Combining feature ranking algorithms through rank aggregation, in: The 2012 International Joint Conference on Neural Networks, IJCNN, Ieee, 2012, pp. 1–8.

[16] Opeyemi Osanaiye, Haibin Cai, Kim-Kwang Raymond Choo, Ali Dehghantanha, Zheng Xu, Mqhele Dlodlo, Ensemble-based multi-filter feature selection method for ddos detection in cloud computing, EURASIP J. Wireless Commun. Networking 2016 (1) (2016) 130.

[17] Sara Mohammadi, Hamid Mirvaziri, Mostafa Ghazizadeh-Ahsaee, Hadis Karimipour, Cyber intrusion detection by combined feature selection algorithm, J. Inform. Secur. Appl. 44 (2019) 80–88.

[18] Omar Almomani, A feature selection model for network intrusion detection system based on PSO, GWO, FFA and GA algorithms, Symmetry 12 (6) (2020) 1046.

[19] Hadeel Alazzam, Ahmad Sharieh, Khair Eddin Sabri, A feature selection algorithm for intrusion detection system based on pigeon inspired optimizer, Expert Syst. Appl. 148 (2020) 113249.

[20] Wangshu Liu, Shulong Liu, Qing Gu, Jiaqiang Chen, Xiang Chen, Daoxu Chen, Empirical studies of a two-stage data preprocessing approach for software fault prediction, IEEE Trans. Reliab. 65 (1) (2015) 38–53.

[21] Mark A. Hall, Lloyd A. Smith, Practical feature subset selection for machine learning, in: Proceedings of the 21st Australian Computer Science Conference, Springer, 1998, pp. 181–191.

[22] Radu Lixandroiu, Catalin Maican, Personalization in E-commerce using profiles similarity, Bull. Transilvania Univ. Brasov. Econom. Sci. Ser. V 8 (1) (2015) 275.

[23] Huan Liu, Rudy Setiono, Chi2: Feature selection and discretization of numeric attributes, in: Proceedings of the 7th International Conference on Tools with Artificial Intelligence, in: Tai '95, IEEE Computer Society, Washington, DC, USA, 1995, pp. 388–391.

[24] Igor Kononenko, Estimating attributes: Analysis and extensions of RELIEF, in: Machine Learning: ECML-94, Springer Berlin Heidelberg, 1994, pp. 171–182.

[25] David W. Aha, Dennis Kibler, Marc K. Albert, Instance-based learning algorithms, Mach. Learn. 6 (1) (1991) 37–66.

[26] Eray Balkanli, A. Nur Zincir-Heywood, Malcolm I. Heywood, Feature selection for robust backscatter ddos detection, in: Local Computer Networks Conference Workshops (LCN Workshops), 2015 IEEE 40th, Ieee, 2015, pp. 611–618.

[27] H.P. Vinutha, B. Poornima, An ensemble classifier approach on different feature selection methods for intrusion detection, in: Information Systems Design and Intelligent Applications, Springer, 2018, pp. 442–451.

[28] K Anand, S Ganapathy, K Kulothungan, P Yogesh, Anand Kannan, A rule based approach for attribute selection and intrusion detection in wireless sensor networks, Procedia Eng. 38 (2012) 1658–1664.

[29] S. Ganapathy, P. Yogesh, A. Kannan, An intelligent intrusion detection system for mobile ad-hoc networks using classification techniques, in: Advances in Power Electronics and Instrumentation Engineering, Springer, 2011, pp. 117–122.

[30] Hyo-Sik Ham, Mi-Jung Choi, Analysis of android malware detection performance using machine learning classifiers, in: ICT Convergence (ICTC), 2013 International Conference on, Ieee, 2013, pp. 490–495.

[31] Shina Sheen, R. Anitha, V. Natarajan, Android based malware detection using a multifeature collaborative decision fusion approach, Neurocomputing 151 (2015) 905–912.

[32] Shanshan Wang, Qiben Yan, Zhenxiang Chen, Bo Yang, Chuan Zhao, Mauro Conti, Detecting android malware leveraging text semantics of network flows, IEEE Trans. Inf. Forensics Secur. 13 (5) (2018) 1096–1109.

[33] Ikram Sumaiya Thaseen, Cherukuri Aswani Kumar, Intrusion detection model using fusion of chi-square feature selection and multi class SVM, J. King Saud Univ.-Comput. Inform. Sci. 29 (4) (2017) 462–472.

[34] George Forman, An extensive empirical study of feature selection metrics for text classification, J. Mach. Learn. Res. 3 (Mar) (2003) 1289–1305.

[35] Suyang Zhu, Sunita Chandrasekaran, Peng Sun, Barbara Chapman, Marcus Winter, Tobias Schuele, Exploring task parallelism for heterogeneous systems using multicore task management API, in: European Conference on Parallel Processing, Springer, 2016, pp. 697–708.

[36] Amira Sayed A. Aziz, E.L. Sanaa, Aboul Ella Hassanien, Comparison of classification techniques applied for network intrusion detection and classification, J. Appl. Log. 24 (2017) 109–118.

[37] Tarfa Hamed, Jason B. Ernst, Stefan C. Kremer, A survey and taxonomy of classifiers of intrusion detection systems, in: Computer and Network Security Essentials, Springer, 2018, pp. 21–39.

[38] Leo Breiman, Classification and Regression Trees, Routledge, 2017, p. 368.

[39] J. Ross Quinlan, C4. 5: Programs for Machine Learning, Elsevier, 2014.

[40] John C. Platt, Fast training of support vector machines using sequential minimal optimization, Adv. Kernel Methods (1999) 185–208.

[41] Roy Sylvain, Nearest neighbor with generalization, 2002, University of Canterbury, Christchurch, New Zealand.

[42] Robert C. Holte, Very simple classification rules perform well on most commonly used data sets, Mach. Learn. 11 (1) (1993) 63–90.

[43] Eibe Frank, Ian H. Witten, Generating Accurate Rule Sets Without Global Optimization, University of Waikato, Department of Computer Science, 1998.

[44] V. Veeralakshmi, D. Ramyachitra, Ripple down rule learner (RIDOR) classifier for IRIS data set, Issues 1 (1) (2015) 79–85.

[45] Ron Kohavi, The power of decision tables, in: European Conference on Machine Learning, Springer, 1995, pp. 174–189.
[46] Mohd Fauzi bin Othman, Thomas Moh Shan Yau, Comparison of different classification techniques using WEKA for breast cancer, in: 3rd Kuala Lumpur International Conference on Biomedical Engineering 2006, Springer, 2007, pp. 520–523.
[47] Atilla Özgür, Hamit Erdem, A review of KDD99 data set usage in intrusion detection and machine learning between 2010 and 2015, PeerJ PrePrints 4 (2016) e1954v1.
[48] Markus Ring, Sarah Wunderlich, Deniz Scheuring, Dieter Landes, Andreas Hotho, A survey of network-based intrusion detection data sets, J. Comput. Secur. (2019) (accepted for publication).
[49] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, Ali A Ghorbani, A detailed analysis of the KDD cup 99 data set, in: Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009. IEEE Symposium on, Ieee, 2009, pp. 1–6.
[50] Cdmc2012, The 3rd cybersecurity data mining competition, 2018, http://www.csmining.org/cdmc2012/ (Accessed September 8, 2020).
[51] J. Song, CDMC2013 intrusion detection data set, Department of Science & Technology Security, Korea Institute of Science and Technology Information (KISTI) (2013).
[52] David A. Cieslak, Nitesh V. Chawla, Aaron Striegel, Combating imbalance in network intrusion data sets, in: GrC, 2006, pp. 732–737.
[53] Sireesha Rodda, Uma Shankar Rao Erothi, Class imbalance problem in the network intrusion detection systems, in: 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), Ieee, 2016, pp. 2685–2688.
[54] Yingying Zhu, Junwei Liang, Jianyong Chen, Zhong Ming, An improved NSGA-III algorithm for feature selection used in intrusion detection, Knowl.-Based Syst. 116 (2017) 74–85.
[55] Daan van der Sanden, Executed from Januari, R. Sadre, Detecting UDP attacks in high speed networks using packet symmetry with only flow data, University of Twente (2008).
[56] Mukaram Safaldin, Mohammed Otair, Laith Abualigah, Improved binary gray wolf optimizer and SVM for intrusion detection system in wireless sensor networks, J. Ambient Intell. Humaniz. Comput. 12 (2) (2021) 1559–1576.