IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 19, 2020, accepted February 25, 2020, date of publication February 28, 2020, date of current version March 11, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2976947

m-OCKRA: An Efficient One-Class Classifier for Personal Risk Detection, Based on Weighted Selection of Attributes

MIRYAM ELIZABETH VILLA-PÉREZ AND LUIS A. TREJO
Tecnologico de Monterrey, Atizapán de Zaragoza 52926, Mexico

Corresponding author: Luis A. Trejo (ltrejo@tec.mx)

The work of Miryam Elizabeth Villa-Pérez was supported in part by CONACYT (Consejo Nacional de Ciencia y Tecnología) of Mexico through the Scholarship under Grant 637273.

ABSTRACT Personal risk detection refers to the timely recognition of situations that may jeopardise a person's physical integrity, for example, during a fall or an accident. During this process, information obtained by monitoring vital signs and human activities is used to develop a mechanism capable of distinguishing between a person's normal behaviour and a risk-prone situation. Such a mechanism is meant to be fully implemented in mobile devices with limited resources in terms of memory, processing power, and battery life. OCKRA (One-Class K-means with Randomly-projected features Algorithm), a one-class classification ensemble specially designed to detect anomalies in a person's behaviour patterns, has been reported in the literature as the best algorithm in terms of accuracy in the context of personal risk detection. Experiments were performed using the publicly available PRIDE (Personal RIsk DEtection) dataset. However, reported training execution times seem prohibitive for mobile implementation. Our contribution is based on two strategies to reduce the execution time during the training phase of a one-class classification algorithm, aiming at its efficient implementation in mobile devices, and at the same time to maintain a good classification performance. The first strategy concerns the PRIDE dataset, for which we applied a filter-based approach to select its most relevant attributes. Eliminating attributes aids to identify sensors that can be turned off to avoid unnecessary data collection, thereby saving hardware resources. In the second strategy, we modified the internal structure of OCKRA based on the analysis of its design. Our proposed algorithm, called m-OCKRA, incorporates weighted attribute projection using filters to create data subsets for each classifier in the ensemble. Also, we reduced its computational complexity from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$. Our results show that m-OCKRA outperforms the original OCKRA version since the gain in execution time for training is almost an order of magnitude, and according to the performed statistical tests, the new algorithm preserves a good and equivalent classification performance.

INDEX TERMS Classifier ensemble, feature selection, one-class classification, personal risk detection, wearable sensors.

I. INTRODUCTION

The rapid growth of emerging technologies such as the Internet of Things (IoT), personalised medicine, virtual reality or artificial intelligence, contribute to the increase in the use of intelligent mobile devices ranging from a phone to all kinds of wearable devices. Consequently, the development of applications that use the information obtained by the sensors embedded in these devices has also increased. Application areas include education, finance, medicine, sports, entertainment, among others.

The healthcare area has benefited from these portable devices [1], [2] and has become a market with large areas of opportunity. Overall, research related to the recognition of activities through the use of integrated sensors has been increasing. In [3], the authors compile all work related to the recognition of activities on mobile devices and provide recommendations so that these systems can be used effectively. Possible applications include elderly care, fall detection and chronic disease monitoring [4]–[7].

The associate editor coordinating the review of this manuscript and approving it for publication was Adnan M. Abu-Mahfouz.

The detection of personal risk takes advantage of the technologies used for the recognition of physical activities; however, its objective is the detection of deviations in physiological patterns and human behaviour [8]. Data from the gyroscope and accelerometer are captured, as well as data related to vital signs such as body temperature and heart rate. In this way, the constructed mechanism is able to differentiate between a person's normal and atypical behaviour.

Just as health monitoring and activity recognition systems aim for early assistance in the event of an incident, so do personal risk detection systems. However, their area of application is more general, because they do not seek to assist in the event of a particular disease or recognise the activities of the user. The idea is to be able to capture data based on ordinary activities of the person's life and detect anomalies in their behaviour patterns, due to situations in which their physical integrity is endangered as in the case of an accident or armed robbery.

When implementing an automatic learning algorithm, some design decisions must be made, involving aspects such as the desired accuracy, type of application, and available resources. In general, ensembles of classifiers are capable of achieving good results and outperforming their individual counterparts; however, the computational cost associated with these types of methods requires finding a balance to increase efficiency without impairing effectiveness.

Among the strategies used to increase the efficiency and improve the effectiveness of a learning algorithm are the selection of attributes or feature selection, the elimination of classifiers with low performance in the ensemble, the adjustment of hyper-parameters, among others.

The PRIDE (Personal RIsk DEtection) dataset [8] is available to the scientific community to provide a baseline for addressing the problem of personal risk detection, making it a reference point for use in this research. From the classifiers reported in the literature to solve the risk detection problem, OCKRA (One-Class K-means with Randomly-projected features Algorithm) [9] and ocSVM (one-class Support Vector Machines) achieved the best performance for the majority of PRIDE users. Given the large amount of data generated and collected to build the PRIDE dataset, it is necessary to implement strategies, either at the data level or at the internal structure of the models, to reduce time in the learning phase.

This work is built upon previous results reported by Barrera-Animas *et al.* [8], in which the authors claimed that it is possible to use PRIDE to develop a personal risk detection mechanism, and showed that abnormal behaviour could be automatically detected by a one-class classifier. In addition, they demonstrated in [9] that OCKRA stood at that time as the state-of-the-art classifier in the context of personal risk detection, followed by ocSVM. However, reported training execution times seem prohibitive for mobile implementation. Reducing the classifier training time can translate into better user experience, thus minimising the chance to stop using the wearable and its application in the short and mid-term.

The main contribution of this research work is the introduction of an algorithm for one-class classification, which corresponds to a modification of the OCKRA classifier presented by Rodríguez *et al.* [9]. The adaptations reduce the execution time in the training phase by approximately one order of magnitude, without significantly affecting the detection results. In this way, it is positioned as a viable candidate to implement in a personal risk detection system in mobile devices.

Besides, a feature selection study is presented for the PRIDE dataset [10], in order to determine the effect of eliminating irrelevant attributes on both execution time and detection performance. Feature selection for one-class problems uses only the instances of the target class; however, after eliminating the least important attributes, at the time of classification, these attributes may be required to discriminate between the target and the atypical class. For this reason, in this study two types of experiments are performed for feature selection, which are described later.

The document is organised as follows: in Section II, we describe work related to our own; in Section III, we give an overview of personal risk detection, we then present the PRIDE dataset used in our experiments, the methodology for feature selection, and the modifications made to OCKRA. Then, in Section IV, the statistical tests and performance metrics used in the experiments are described. Subsequently, in Section V we present the results of our experiments and a discussion about the statistical tests performed on our algorithms. Finally, in Section VI, the main contributions of our research work are given.

II. RELATED WORK

This section describes the classifier ensembles developed for one-class problems, as well as attribute reduction techniques that help accelerate and improve the performance of classification algorithms. Both topics are key components in generating more efficient and effective personal risk detection mechanisms.

A. ENSEMBLE OF ONE-CLASS CLASSIFIERS

Combination of classifier techniques aims to improve the classification performance offered by individual classifiers. They are built from a set of models and then classify new data by a weighted vote of their predictions.

One of the essential elements of classifier ensembles is diversity [11]. Given a dataset, this can be modified so that each ensemble classifier is trained with its own dataset (object selection), either through bootstrap aggregation, bagging, or boosting. At the attribute level, diversity can be created by applying random subspace sampling, causing different subsets of attributes to be used by classifiers (attribute selection). In addition, the classifier ensemble can be trained from multiple instances of the same base classifier or using different models. Also, pruning techniques have been developed to eliminate under-performing classifiers from the ensemble.

There are several examples of object selection for one-class classification. In this sense, Seguí *et al.* [12] introduced a new method for classifier ensembles based on the non-parametric bootstrap aggregation strategy with weights to improve accuracy in the presence of outliers. Another model that uses this strategy is Bagging-RandomMiner created by Camiña *et al.* [13] for the detection of impostors. The algorithm is based on the random selection of objects and chooses as prototypes the objects that best represent the normal class. The idea behind this method is to leave aside objects of minor importance. Both proposals obtained improvements in both the accuracy and robustness of the classification when compared with various one-class and multi-class classifiers reported in the literature.

Feature selection to create classifier ensembles is one of the most common techniques in the literature. In the context of anomaly detection, Perdisci *et al.* [14] use a different attribute space to train each ensemble classifier. In addition to the random selection of attributes, the ensemble is built from multiple instances of ocSVM. Their experiments show that using a set of classifiers trained with different feature spaces significantly reduces the number of false positives and achieves a higher percentage in the detection of anomalies.

Jeong *et al.* [15] propose two support vector data description (SVDD)-based feature selection methods for one-class classification problems. The methods can be used to minimise the size of the boundary of describing normal observations measured through the value of its radius squared. The experimental results show that the proposed techniques perform better than well-known SVM recursive feature elimination (SVM-RFE) method for simulated data and real-life datasets. However, the described approach carries out feature selection as part of the training process and it is dependent on the given classification method. Additionally, Lian [16] shows that dimension reduction with Principal Component Analysis (PCA) works well for one-class SVM. Although the results prove the effectiveness of the method in image recovery tasks, the main disadvantage is that it depends on the use of ocSVM.

Trejo and Barrera-Animas in [10] aimed at reducing training execution time by means of feature selection techniques. Authors succeeded to speed-up the ensemble's training time without sacrificing performance. They run their experiments on the PRIDE dataset after a feature selection procedure, based on a correlation matrix analysis and PCA. However, the main drawback of their results, as they acknowledge, is that they performed feature selection only on the training dataset, which could result in removing attributes that may contribute to the detection of abnormal situations. In contrast, our work performs the feature selection process on the complete dataset, that is, using the training and anomaly datasets. Hence, we reduced the possibility of leaving out an important feature capable of detecting unseen abnormal behaviours.

Regarding the use of different base classifiers in the ensemble, Zainal *et al.* [17] suggest an ensemble of one-class classifiers for network intrusion detection. The authors use three machine learning techniques to construct the ensemble: Linear Genetic Programming (LGP), Adaptive Neural Fuzzy Inference System (ANFIS) and Random Forest (RF). Their work compares the ensemble with the base methods used individually to construct the ensemble. The results show that it is better to combine techniques than to use them separately. Improved intrusion detection accuracy was achieved by assigning appropriate weights to the classifiers in the ensemble.

Krawczyk and Woźniak [18] propose the application of pruning to a set of ensembles built with random subspace sampling attributes and data partitioning. The measures designed by the authors evaluate the differences between the individual models of the set to ensure quality in the diversity of the ensemble and eliminate those that significantly affect the overall performance of the model. The results confirm that the introduction of diversity measures for one-class datasets is a research direction worth exploring. Parhizkar and Abadi [19] propose to apply a binary algorithm of an artificial bee colony, BeeOWA, to eliminate classifiers from the ensemble and find an optimal subset in reasonable computational time. BeeOWA managed to overcome several approaches described in the literature, both in terms of classification performance and in statistical significance.

It is possible to provide diversity to an ensemble of classifiers, either by projecting a subset of features from the dataset to every classifier or by making the individual classifiers different [22]. In this work, we decided to handle diversity based on remarks given by Tax and Duin [20] and Nanni and Lumini [21]. In both studies, they used feature selection techniques and trained every single classifier in an ensemble. They demonstrated that combining features is more effective than combining different classifiers and, in general, produces better results than the individual classifier counterpart.

Table 1 presents a brief comparison of the methods used by the studies reviewed in this paper. In general, the proposed techniques use object selection or attribute selection to create the ensemble; the differences lie in the combination of base classifiers or metrics to evaluate the quality of the ensemble.

B. REDUCTION IN DIMENSIONALITY

Dimensionality reduction consists in downsizing the number of attributes of a dataset. The methods used identify and remove characteristics that are considered irrelevant or redundant, and are divided into *feature selection*, which chooses an important subset of attributes, and *attribute extraction*, which forms new attributes from the originals.

This data processing allows the learning algorithms to work faster and more efficiently. In some cases, it is possible to improve the accuracy of the classifier, as well as to obtain a data model that is more compact and easy to interpret.

1) FEATURE SELECTION

The various approaches proposed for feature selection seek to find the best subset within the original set of attributes. In this way, the best subset contains the fewest number of attributes that contribute most to the prediction [23].

TABLE 1. Comparison of related work.

Authors	Attribute space	Object space	Base classifier	Type of ensemble	Pruning technique	Aggregation technique	Evaluation metrics
Krawczyk and Woźniak [18]	Random	No	ocSVM SVDD	Homogeneous	Adapted measures for ocC	Mean Mean of estimated probabilities	-
Seguí et al. [12]	No	Non parametric bagging with weights	NNDD SVDD MST-CD	Homogeneous	No	-	AUC
Camiña et al. [13]	No	Bagging	RandomMiner	Homogeneous	No	Mean	AUC ZFP
Perdisci et al. [14]	Random	No	ocSVM	Homogeneous	No	Majority	AUC
Parhizkar and Abadi [19]	No	No	-	-	Artificial bee colony	Exponential induced (OWA) ordered weighted averaging	AUC
Zainal et al. [17]	Random	No	LGP RF ANFIS	Homogeneous	No	Majority Mean	Precision
Tax and Duin [20]	Random	No	K-means Autoencoder Parzen	Homogeneous	No	Mean Product of the weighted votes	AUC
Nanni [21]	Random	No	LPD PCAD	Homogeneous	No	Max rule	EER
Barrera-Animas et al. [10]	Random	No	OCKRA	Homogeneous	No	Mean	AUC

- Not specified

To remove an irrelevant characteristic, it is required a selection criterion that can measure the relevance of each characteristic. From a machine learning approach, if a system uses irrelevant variables, then it will use all that information to construct the learning model, resulting in a poor generalisation of the problem to be addressed [24].

Feature selection algorithms for a dataset are classified into filters, wrappers, and embedded. Filter-based methods are independent of the classification algorithm; to select the subset, they use the intrinsic properties of the data, such as information measurements, correlation, and various types of distances [25]–[27]. The result of the selection can be given through a subset of attributes or a ranking of characteristics. Wrappers use the prediction of the classifier to measure the quality of the subset obtained. It is computationally expensive compared to other methods, as they must train the classifier to evaluate the quality of the subset; however, the result of the feature selection occurs automatically after meeting a condition given by the classifier. Finally, in embedded techniques, feature selection is performed during the training process as a stage that is part of the learning algorithm; the two stages cannot be separated. It has a lower computational cost than wrappers methods as they do not require the iterative construction of multiple models. Decision trees are the most common example of this feature selection technique.

2) FEATURE EXTRACTION

Unlike selection, attribute extraction transforms data into a smaller space while retaining as much information as possible [28]. The best-known methods are Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA), which are based on linear projections and are used for unsupervised and supervised learning, respectively [23].

PCA uses the transformation of the initial set of variables into several sets of linear combinations that are known as main components. These components are not correlated, and most of the information in the original dataset is concentrated in the first components. After ranking them by importance, the components that retain the most information are chosen. The disadvantage of this method is that it does not consider the divisibility of classes since the data are not labelled. LDA is very useful when variables are highly correlated or when the number of independent variables is large. It is related to Variance Analysis (ANOVA) or regression, however, during this attribute extraction process the independent variables and the class label are used to find a linear combination of characteristics that separates the object classes.

III. METHODOLOGY

In this section, we present an overview of personal risk detection followed by a detailed description of the dataset used in our experiments. We also present an approach to improve the efficiency of the personal risk detection model by reducing the number of attributes of the dataset. Subsequently, given the performance results obtained by OCKRA [9], the modifications made to the learning model are explained, in order to improve the training phase and its learning rate.

A. PERSONAL RISK DETECTION

Barrera-Animas *et al.* in [8] introduce the concept of personal risk detection. They define it as the timely identification of situations that may put at risk a person's physical integrity, for example, during a health crisis or a car accident. To detect a risk-prone situation, they use an approach built on the rationale that people usually perform based on the same behavioural and physiological patterns or with small variations thereof. The main hypothesis of their work is that a risk-prone situation yields sudden and significant deviations in ordinary physiological and behavioural user patterns; thus, potential threat circumstances can be automatically recognised. A collection of sensors, such as the ones embedded in current wearable technology, can capture these changes. They approached the personal risk detection problem as an anomaly detection one, which can be tackled by a one-class classifier. The classifier is trained only with a user's ordinary conditions data and then, during the classification process, the classifier is capable of detecting anomalies that can be related or not to a risk-prone situation. The authors demonstrated their hypothesis that abnormal behaviour could be automatically detected using a one-class classification approach.

TABLE 2. 26-dimension vector structure from the PRIDE dataset.

Attribute			Number
Accelerometer-Gyroscope	X-axis	$\bar{x}$	1
		s	2
	Y-axis	$\bar{x}$	3
		s	4
	Z-axis	$\bar{x}$	5
		s	6
Angular velocity	X-axis	$\bar{x}$	7
		s	8
	Y-axis	$\bar{x}$	9
		s	10
	Z-axis	$\bar{x}$	11
		s	12
Accelerometer	X-axis	$\bar{x}$	13
		s	14
	Y-axis	$\bar{x}$	15
		s	16
	Z-axis	$\bar{x}$	17
		s	18
Heart rate			19
Skin temperature			20
Steps			21
Velocity			22
UV index			23
Δ Pedometer			24
Δ Distance			25
Δ Calories			26

B. PRIDE DATASET

The PRIDE dataset is the first publicly available dataset for the personal risk detection problem, designed by Barrera-Animas *et al.* [8]. The dataset is built from data captured during daily activities of individuals and complementary, from data captured under stress, atypical or risky conditions. The creators of the dataset established a data collection period of one week, with the help of 23 people of different ages, gender and physical conditions, in order to provide diversity in the data collected. Likewise, stress or simulated risk scenarios were carefully planned in order to exemplify those situations in which people might find themselves.

In order to build the dataset, data were obtained through sensors embedded in the Microsoft Band$^{©}$, and transmitted to a mobile application. The sensors involved are the gyroscope, accelerometer, pedometer, heart rate, distance, skin temperature, UV index, and calories. The capture process experienced an interruption time of approximately 120 minutes daily, since the bracelet takes about 40 minutes to recharge, a process needed about three times a day. Also, the capture process was also interrupted whenever the user had to remove the band for any personal reason.

After the dataset preprocessing, the resulting feature vector has 26 dimensions and is updated every second. Table 2 shows the attributes that are part of the PRIDE structure. Each user has an average of 322,038 instances and a total of 7,406,868 samples for the entire dataset. The highest number of observations is found in user 1 with 466,175 samples, and the lowest number is found in user 17 with 133,795. This difference in the number of samples causes training time to vary between users.

The dataset takes into account information from the person that can be useful to detect physiological or behavioural patterns; in addition, the data gathering from sensors is realised when the user is carrying out their daily life without interfering in their activities.

To build the PRIDE's anomaly conditions dataset, the same 23 users participated in another process to acquire data under specific conditions, for which five scenarios to simulate stressful or abnormal conditions were envisioned. These scenarios consisted of the following activities: running 100 m as fast as possible, climbing stairs as quickly as possible, boxing for a two-minute round, falling back and forth, and holding one's breath for as long as possible. Each activity aimed to simulate a dangerous or abnormal condition in the real world, for example, running away from an unsafe situation, leaving a building due to an evacuation alert, defending against an aggressor during a quarrel, swooning and suffering from breathing problems. The session to perform the five scenarios by each user took about two hours, and it demanded major physical effort.

Personal risk detection is defined as an anomaly detection problem, therefore, it can be tackled through one-class classification algorithms, which build classification models when the negative class does not exist, is poorly sampled or poorly defined [29]. In this way, following the evaluation protocol carried out by Barrera-Animas *et al.* [8], the dataset is divided into five folds to perform a cross-validation (5-FCV). Four groups of the dataset under normal conditions are used for training, and only one group is combined with the anomalies dataset, used for testing purposes. For the latter group, we tagged the last column manually in each row using the labels ‘‘typical’’ or ‘‘atypical’’. The ‘‘typical’’ label indicates that the object represents the normal behaviour, while the ‘‘atypical’’ label indicates that the object represents an abnormal state. This tag is not used to train the classifier, but only during the testing phase. At the end of the procedure, each user has five training sets and five testing sets.

C. FEATURE SELECTION ON PRIDE

To implement feature selection on the dataset, the filter-based technique proposed by Lorena et al. was chosen. Reference [30], which is based on a set of feature importance measures that produce different rankings to select the subset with the highest rated attributes. The chosen method not only allows to select the subset that gathers the most relevant attributes through the intrinsic properties of the data, but also allows to explore multiple views of them. Additionally, the importance of a feature is complemented when the ranks produced by each metric are combined. The difference between this method and feature extraction is that selection through filters maintains a subset of the original features, whereas extraction creates new features that are difficult to interpret.

The list with each characteristic's rank is obtained by means of metrics adapted for the selection of variables of a single class; these assign an order to the attributes of the dataset and are then combined using different aggregation techniques. The feature importance measures implemented in this work are listed below; they are described in detail in [30], [31]:

Information Score (IS): Measures the relevance of each attribute when removed from the dataset. If the value of the entropy decreases when removed, the similarity among the rest of the data is high; therefore, the characteristic can be considered important.

Pearson Correlation (PC): This metric measures the degree of association of each attribute against the others. During this process, the attributes are verified to see if they are linearly dependent. Elevated values indicate a high correlation, hence low values are preferred to maintain unrelated attributes.

Intra-class Distance (ICD): Quantifies the distance of all samples of a class to the centroid of the class. It is expected that this value is not very high. For each attribute, the distance reduction is measured; therefore, in the end, those that are closer to the data are considered better.

Interquartile Range (IQR): The obtained value takes into account the distribution of the characteristic values through their interquartile range. The more dispersed the data, the more likely it is that the interquartile ranges will overlap. An attribute is considered a characteristic of the dataset if its values tend to be more concentrated.

The next step in the feature selection process is to combine the results produced by multiple ranking lists, generating a consensus for the final ranking. The aggregation methods used for feature selection are Mean, Majority and Borda Count. The first method computes the average of the positions of the characteristics in the ranking lists; the second applies a majority voting rule to the positions of each characteristic in the lists; and the third method assigns a score to each attribute according to its position in the ranking. These points increase from the last position to the first. The option with the highest cumulative sum of points ranks best.

Attributes are ordered from 1 to n, where n is the number of attributes of the dataset. The value 1 means the most important and n the least important. For each user in the dataset, the attribute importance ranking was calculated using the three aggregation methods. Subsequently, in order to unify the final results and thus have an order that encompasses all users, the three aggregation methods were reapplied to the rankings obtained by the 23 users.

This generalised approach makes it possible to visualise the importance of the attributes of the entire dataset and in the future, add new users only by calculating the m most important attributes, instead of the original 26.

Once the list of rankings has been obtained for each aggregation method, the selection of attributes consisted of systematically eliminating the two least important attributes from both the training set and the testing set. In this way, only the most important attributes are maintained and the classification algorithm is run in that subset.

In accordance with this procedure, the least important attributes were removed from the dataset in order to verify whether the number of characteristics can be reduced while maintaining the classification performance achieved when all characteristics are used.

D. M-OCKRA

OCKRA, first introduced in [9], is an ensemble of single-class classifiers, which are based on multiple projections of the PRIDE dataset according to random subsets of characteristics. OCKRA is made up of 100 classifiers, each one built upon ten centres computed by k-means++ with Euclidean distance. It works as follows: During the training phase, each individual classifier applies k-means to a random projection of the dataset and stores the centroids of the clusters. During the classification phase, to decide whether a new object is abnormal, each classifier compares it with all of the centroids in order to determine the cluster to which the object belongs. Each classifier returns a similarity value according to the distance of the object relative to its closest cluster centroid. The ensemble returns the average similarity computed by individual classifiers. For a detailed description of the algorithm, refer to [9].

The algorithm calculates the distance between all pairs of objects, which in the worst-case scenario is estimated to require up to $\mathcal{O}(n^2)$ comparisons. As a consequence, for large datasets, the algorithm training is limited. The main drawback of the algorithm is the validation of the OCKRA parameters: Authors fixed to 100 the number of classifiers based on Breiman [32]; also, they use an Euclidian distance function without a strong argument or discussion, and they fixed k, the number of clusters, to 10 from empirical experimentation. The algorithm is meant to run on a mobile device, v.g.

a smartphone, with limited hardware resources such as CPU and memory. OCKRA's designers stated that to determine the optimum value of k and the number of classifiers considering available resources and detection performance, remained an open question.

One of the goals of this work is to reduce the training time taken by the classifier and make it suitable for real-world applications. m-OCKRA accomplishes this by reducing the computational complexity of one of its components, from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$, thus reducing execution time in almost one order of magnitude, as shown later in Section V-B. With this in mind, we have made modifications to OCKRA so that it now requires significantly less time to build a model and produces classification results without a significant statistical difference from the original.

The modified algorithm is called m-OCKRA, where the letter m refers to mobile, since the classifier is adapted for mobile devices, where processing is done with limited resources. The new algorithm incorporates weighted attribute projection (spatial sampling of attributes with probability) to create subsets of data for each ensemble classifier and the new calculation of typical objects using RandomMiner [13]. Table 3 compares OCKRA and the enhanced version m-OCKRA. The following is a formal description of the ensemble training and classification phases.

TABLE 3. Comparing OCKRA against m-OCKRA.

Property	OCKRA	m-OCKRA
Number of Classifiers	100	50
Distance	Euclidian	Chebyshev
Number of clusters	$k = 10$	Percentage of objects in the dataset, called most representative objects (MROs)
Feature Selection	Random selection	Weighted random selection
Core algorithm and Complexity	Clustering Kmeans++ $\mathcal{O}(n^2)$	Bootstrap RandomMiner $\mathcal{O}(n)$
Training execution time	Slow for real-world applications	An order of magnitude faster
Classification accuracy	There is no significant difference	There is no significant difference

1) TRAINING PHASE

m-OCKRA is an ensemble consisting of multiple instances of the same classifier, in this case RandomMiner (Algorithm 1).

Algorithm 1 m-OCKRA Training Phase

Input: T: Training dataset; N: Number of classifiers in the ensemble; F: Fraction of training dataset to bootstrap; $RS_{\%}$: MROs (Most Representative Objects) percentage

Local Variables: W: List of attributes weights; $SelectedAttributes_i$: List of randomly selected attributes; T': Training subset; δ_i: Classifier threshold; X: List of randomly selected objects from T'; $MROs_i$: The Most Representative Objects

Output: P: Classifier parameters

```
1:  $P \leftarrow \{\}$
2:  $W \leftarrow \text{ComputeAttributesWeights}(T)$
3:  for $i = 1..N$ do
4:      $SelectedAttributes_i \leftarrow$
        $\text{RandomWeightedAttributes}(W)$
5:      $T' \leftarrow \text{Project}(T,\ SelectedAttributes_i)$
6:      $\delta_i \leftarrow \text{SumProbabilities}(W, SelectedAttributes_i)$
7:      $X \leftarrow \text{Bootstrap}(F,\ T')$
8:      $MROs_i \leftarrow \text{Sample}(RS_{\%},\ X)$
9:      $P \leftarrow P \bigcup \{\ (SelectedAttributes_i,\ \delta_i,\ MROs_i)\ \}$
10: end for
11: return $P$
```

To train the ensemble, the process begins with an initial training dataset T of size $m \times n$, where m is the number of samples and n is the number of attributes. As a first step, we compute the weights $[w_1, w_2, \ldots, w_n]$ of the attributes (Step 2) using the feature importance measures described in Section III-C. Using the ranking obtained for each characteristic $[r_1, r_2, \ldots, r_n]$, where r_i is a number between 1 and n, each element in the list (w_i) is represented as r_i/n. The estimated weights $[w_1, w_2, \ldots, w_n]$ are used to randomly obtain the attributes (Step 4) that will serve to create the data subset to train the classifier. During this process, n random numbers are generated according to those weights and repeated elements are removed. The result obtained is a list of indices of the attributes to be considered for training. In average, 57% of the information is retained. The selected characteristics are not repeated in the same classifier, but within the ensemble repetition is possible.

The way attributes are selected is based on the fact that there are more relevant attributes for the classification problem than others, so they should appear more frequently in the ensemble data subsets [33]. To achieve this goal, the probabilities of important attributes must be higher than those of lesser importance. Also, the size of the subsets obtained remains smaller than the total number of original attributes, so diversity between the different classifiers is still guaranteed and randomness with probability maintains to some extent the independence between them. As explained in Step 2, the probabilities or weights of the attributes are calculated using importance measures that are based on the intrinsic properties of the data and their ability to describe the original distribution of such data.

The algorithm then projects the dataset T over the selected characteristics (Step 5). The projected dataset T' has a size $m \times n'$, where $n' \leq n$, is the size of the subset of random attributes. To obtain the δ_i threshold of the classifier, the algorithm adds together the probabilities of the attributes that are used to project the T' subset. The reasoning behind this value is that calculations are reduced by eliminating the computation of average distances between all objects and at the same time, the classifier determines the importance of the data subset to be used for training.

OCKRA [9] uses K-means++ as a clustering algorithm, however, among the main drawbacks of K-means++ is

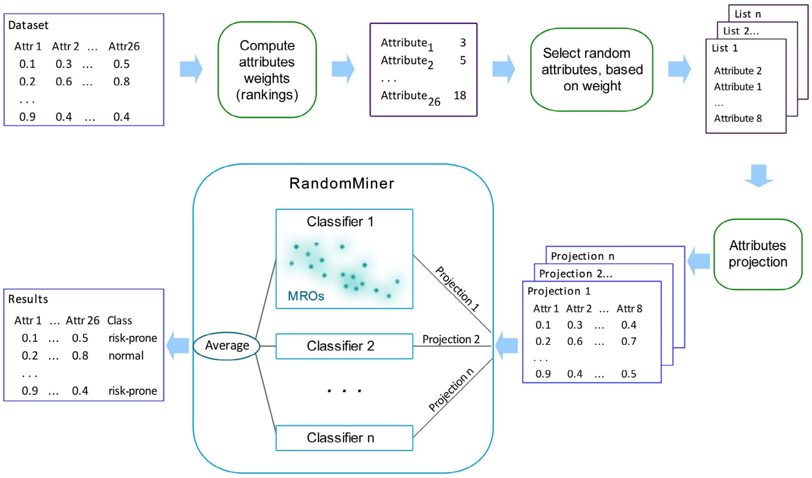

FIGURE 1. m-OCKRA training and classification process.

determining the number k of clusters, since the quality of the obtained partition will depend on this number. Although there is no perfect mathematical criterion, within the literature we find a series of heuristics or validation indices to fine-tune the value of k that obtain promising results. However, the clustering algorithm must be executed repeatedly for different values of k and the partition that seems more significant is selected; hence these options become computationally expensive.

In Section II-A a new algorithm for one-class classification called Bagging-RandomMiner is mentioned, which uses RandomMiner to set the limits of the normal objects of the dataset. The most representative objects (MROs) are selected by random sampling. Its computational complexity is $\mathcal{O}(n)$ in the training phase [13], so in addition to the quality of the results, it has served as an inspiration to replace K-means++ as the basic algorithm of the ensemble of classifiers.

Thus, after projecting the dataset, RandomMiner is applied over T' (Step 7 and 8). First, a random resampling (*bootstrap*) of a fraction of T' objects is done, then a percentage of the samples is taken which are identified as MROs. These data represent typical values of normal user behaviour and replace the centroids calculated by K-means++.

The training phase returns a set of parameters from each classifier in the ensemble. The three parameters consist of the randomly selected attributes, the threshold, and the MROs (Step 9).

2) CLASSIFICATION PHASE

The classification phase receives as input the set of classifiers P generated in the training phase and an object O to classify (Algorithm 2). For each classifier, the object O is projected using the selected attributes, then the minimum distance is calculated between the projected object O' and its closest representative object of the set of MROs (Step 4).

The calculated distance is used to obtain the similarity value (Step 5) of the object, which is between [0,1]. A value of zero indicates risk-prone behaviour, while a value of one indicates that the object resembles the normal behaviour of users. The similarity values of each ensemble classifier are unified by computing their average (Step 7).

Algorithm 2 m-OCKRA Classification Phase

Input: O: Object to classify; P: Ensemble of classifiers as returned by Algorithm 1
Local Variables: O': Projected Object; d_{min}: Minimum distance
Output: s: Risk probability (similarity value)

```
1: $s \leftarrow 0$
2: for each $(Attributes_i,\ \delta_i,\ MROs_i) \in P$ do
3:     $O' \leftarrow \text{Project}(O,\ Attributes_i)$
4:     $d_{min} \leftarrow \min(\text{Distance}(O',\ MROs_i))$
5:     $s \leftarrow s + e^{-0.5(d_{min}\ /\delta_i)^2}$
6: end for
7: $s \leftarrow s/|P|$
8: return $s$
```

To summarise our methodology and for the sake of clarity, Figure 1 shows a block diagram representing the complete process.

IV. EXPERIMENTAL SETUP

This section presents the experimental methodology and evaluation metrics used throughout this work. We carry out two types of experiments. First, we evaluate the feature selection procedure performed over the PRIDE dataset, with the aim of decreasing the execution time of OCKRA, but at the same time obtain classification results that are comparable to the performance achieved when the full set of characteristics is used. Recall from Section III-B that the PRIDE dataset used in our experiments, is a publicly available dataset that provides a baseline for the fair comparison of personal risk detection mechanisms. The dataset structure comprises 26 attributes derived from sensors readouts.

The full dataset contains information from 23 users, with an average of approximately 300 thousands observations each. The second set of experiments aims to compare m-OCKRA against OCKRA, since the latter was reported by Rodriguez *et al.* [9] as the best one-class classifier for the personal risk detection problem.

Section IV-A presents the metrics to evaluate the performance of the tested classifiers. The statistical tests used to compare the classification results are described in Section IV-B.

A. EVALUATION METRIC

To evaluate performance, the Area Under the Curve (AUC) of the True Positive Detection Rate (TPR) versus the False Positive Detection Rate (FPR) was calculated and the average of the 5-FCV results were obtained for each user.

This metric, besides having been used in the context of personal risk detection and one-class classification [8]–[10], [13], [34], [35], gives an idea of the amount of work done by the classifier [36] and is invariant to the distribution of the training set [37].

B. STATISTICAL TESTS

In the first set of experiments, in order to study the differences between the classification performance of the models obtained after feature selection, a comparison of results in terms of AUC is made through the Friedman non-parametric test [38], which provides a ranking and is used to compare more than two models. Afterwards, we applied the Bergmann-Hommel dynamic post-hoc procedure [39], [40] to know which models have statistical differences between them. The results of the post-hoc tests and the order of the classifiers according to the Friedman ranking can be represented visually in a critical difference diagram (*critical difference* or CD) [38]. According to the CD diagram, the best algorithm appears at the right, and statistically similar algorithms are joined by a thick horizontal line.

Additionally, we applied the Wilcoxon signed-rank test [41] to perform a pairwise comparison between statistically similar models according to the CD diagram. In the Wilcoxon test, the sums of the ranks of the two models being compared are presented, and it is determined whether the difference between these measurements is random or not. In the latter case, if the sum of the ranks is higher, then there is a significant statistical difference.

The feature selection evaluation is based on the null hypothesis that using a subset of attributes to train the classifier is the same as working with the original set. To reject the null hypothesis and claim that one model significantly outperforms the other, it is verified that the p value is less than a given level of significance.

For the second set of experiments, we compared the performance obtained by m-OCKRA against OCKRA [9], in AUC terms. To do this, a pairwise comparison is made between the two classifiers, using the Wilcoxon signed-rank test as well.

The statistical tests were performed using the software tool KEEL [42]. The significance level used by KEEL to reject a null hypothesis is $\alpha = 0.05$.

V. RESULTS

A. OCKRA CLASSIFICATION PERFORMANCE WITH FEATURE SELECTION

This section describes the results of applying the feature importance measures described in Section III-C to reduce the number of attributes in the dataset. The goal of this set of experiments is to decrease the execution time of OCKRA, while obtaining classification results comparable to the performance achieved when using the full feature set.

As we are dealing with a one-class problem, if feature selection is made only on the dataset of the normal or positive class, it could leave out characteristics that may contribute to the detection of personal risk [10]; therefore, in order to explore the impact of feature selection and taking this into account, we carried out two types of experiments:

- Feature selection on the normal condition dataset (FS-NCD). The filter method was applied only to the training data, which are the ones containing the normal class.
- Feature selection on the normal condition dataset in combination with the anomalies dataset (FS-NACD). The filter method is applied to the training and testing dataset, where the latter contains the normal and atypical classes.

After applying the feature importance measures, Table 4 shows the *ranking* obtained for each attribute of the PRIDE dataset, using the three aggregation methods, where 1 means the most important attribute and 26 the least important.

Let us recall that once the list of rankings has been obtained for each aggregation method, the selection of attributes consisted of systematically eliminating the two least important attributes from both the training and the testing sets, keeping only the most important ones and evaluating the classifier using that subset.

In Figures 2 and 3 it is possible to observe the results of the Friedman test and Bergmann-Hommel's post-hoc analysis of the two types of experiments, with a significance level of $\alpha = 0.05$, where DS-i refers to the dataset with i attributes, so that $i \in \{12, 14, 16, 18, 20, 22, 24\}$ and *Full* is the dataset with 26 attributes. It is key to remember that a thick horizontal line joins statistically similar models, and the best subset appears at the right.

For the FS-NCD experiment, the smallest subsets of attributes were achieved using the Borda aggregation method. According to the CD diagram, there are no statistical differences between the seven versions of feature subsets and the original version, in addition, the sets with 16 and 18 attributes achieved better average classification than the original one. The Majority method ranks in the first three places the subsets of 24, 20 and 22 attributes, which neither show statistical differences with the original version. With Mean,

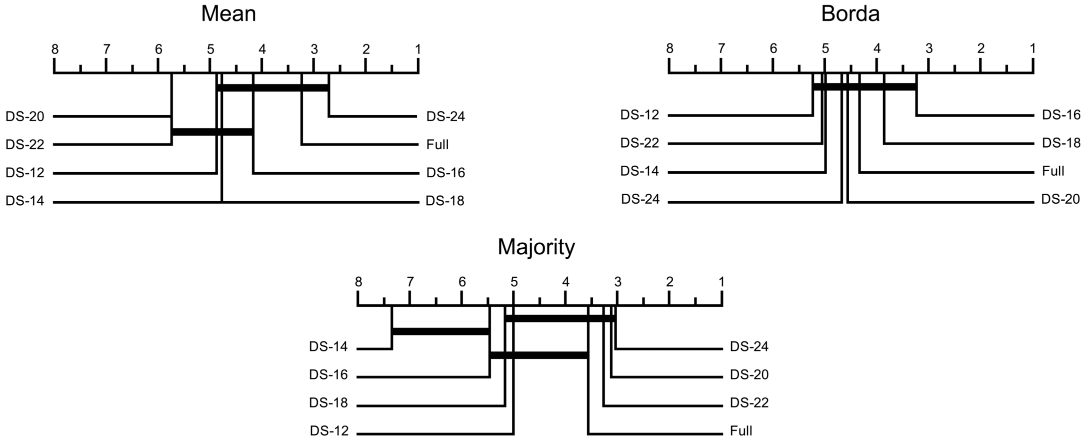

FIGURE 2. CD diagrams with statistical comparisons of subsets obtained from experiments using the FS-NCD dataset with three aggregation methods.

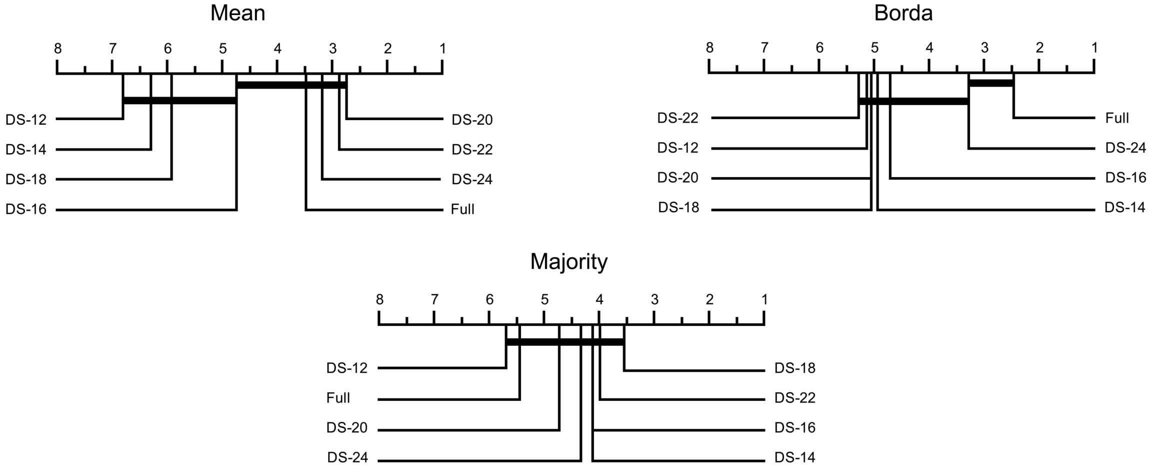

FIGURE 3. CD diagrams with statistical comparisons of subsets obtained from experiments using the FS-NACD dataset with three aggregation methods.

the only subset that outperforms the original is the one with 24 attributes; however, it is possible to reduce the set by using between 12 and 18 attributes without showing statistical differences.

The second type of experiments shows strong contrasts with respect to the first. According to the CD diagram, there are no statistical differences between the seven versions of subsets of characteristics obtained through Majority; however, all subsets outperform the original during classification, except the subset with 12 attributes. Average positions in the first three places the same subsets as Majority with the previous approach. Finally, Borda is the only aggregation method where the original set of characteristics performs like the best. Moreover, only the subset of 24 attributes shows no statistical difference with respect to the full version.

From the experiments carried out, 15 out of 48 cases outperform the complete set of characteristics during classification; thus, to analyse the methods that significantly surpass the original dataset, we run the Wilcoxon signed-rank.

Tables 5 and 6 show the compared datasets (Comparison), the sum of the ranks for cases where the data subset improved to the original (R^+), the sum of the ranks for the opposite (R^-), the result of the null hypothesis (Hypothesis), and the p value calculated by the Wilcoxon signed-rank test. From the results shown in the tables, it is possible to affirm that there are five cases, where the subset of attributes significantly exceeds the classification performance of the original set of 26 attributes.

In the case of FS-NCD, Table 5 shows that using 24 attributes, calculated through Mean and Majority, improves the detection of anomalies for a significance level $\alpha = 0.05$. In the FS-NACD experiment, three cases improve this detection, according to Table 6, with the same value of α.

Exploring different aggregation methods to unify the results of the feature selection process has an impact on the final rankings, and therefore on the detection performance. For both experiments FS-NCD and FS-NACD, Majority as an aggregation method achieves more cases better

TABLE 4. Attribute ranks after feature selection on both datasets: FS-NCD and FS-NACD. The first 12 attributes are highlighted with different colors to distinguish each aggregation method list.

	FS-NCD Normal conditions dataset			FS-NACD Normal and anomaly conditions dataset		
Feature	Mean	Majority	Borda	Mean	Majority	Borda
$\bar{x}$ Gyro Accel X	15	19	21	14	17	21
s Gyro Accel X	26	24	23	21	22	20
$\bar{x}$ Gyro Accel Y	18	1	18	13	16	23
s Gyro Accel Y	20	25	22	25	24	18
$\bar{x}$ Gyro Accel Z	13	2	20	11	1	22
s Gyro Accel Z	21	26	12	22	25	19
$\bar{x}$ GyroAngVel X	6	8	14	4	2	13
s GyroAngVel X	10	9	13	10	3	14
$\bar{x}$ GyroAngVel Y	8	13	11	6	15	8
s GyroAngVel Y	14	14	19	19	10	9
$\bar{x}$ GyroAngVel Z	3	3	15	2	4	15
s GyroAngVel Z	12	11	25	16	11	11
$\bar{x}$ Accel X	17	18	2	18	20	7
s Accel X	24	22	24	23	21	12
$\bar{x}$ Accel Y	19	4	9	17	19	10
s Accel Y	22	23	10	24	23	25
$\bar{x}$ Accel Z	16	10	17	12	13	16
s Accel Z	25	21	8	26	26	17
Heart Rate	9	5	16	15	5	2
Skin Temperature	23	20	7	20	6	24
Pace	5	6	3	7	7	3
Speed	7	12	26	8	12	4
UV	1	15	4	5	14	1
Δ Pedometer	2	17	1	1	18	6
Δ Distance	11	16	6	9	8	5
Δ Calories	4	7	5	3	9	26

TABLE 5. Wilcoxon signed-rank test of the original dataset (full) against data subsets derived from FS-NCD.

	Comparison	R^+	R^-	Hypothesis ($\alpha = 0.05$)	p-value
Mean	DS-24 vs Full	204	72	Rejected	0.04488
Majority	DS-24 vs Full	210	66	Rejected	0.02768

TABLE 6. Wilcoxon signed-rank test of the original dataset (full) against data subsets derived from FS-NACD.

	Comparison	R^+	R^-	Hypothesis ($\alpha = 0.05$)	p-value
Majority	DS-18 vs Full	227	49	Rejected	0.005414
	DS-22 vs Full	219	57	Rejected	0.012294
	DS-24 vs Full	240	36	Rejected	0.0011184

positioned, so it is considered a strong strategy to unify results for the dataset.

Since the objective is to reduce the training time of the classifier without losing precision in its performance, it is necessary to carry out an analysis in this regard. The results of the execution time of two PRIDE users are shown in Table 7 in hh:mm:ss format. As explained in Section III-B, the users with the most and least observations in the training set are user 1 and 17, respectively. The subset with 16 attributes was the smallest and best-positioned subset; hence, our analysis in the execution time is performed decreasing from 24 attributes to that number. The subsets with 12 and 14 attributes were excluded because in no case they performed better than or at least equal to the original dataset during classification.

TABLE 7. Execution time of the training phase of OCKRA using different data subsets. The $\mathcal{G}$ column indicates the percentage gain when using a subset against the full attribute vector.

Dimension	User 1	$\mathcal{G}$	User 17	$\mathcal{G}$
Full	2:26:21		0:51:28	
DS-24	1:53:02	22.8%	0:49:20	4.2%
DS-22	1:47:58	26.2%	0:47:52	7.0%
DS-20	1:48:22	26.0%	0:46:06	10.4%
DS-18	1:45:43	27.8%	0:44:17	13.9%
DS-16	**1:41:13**	**30.8%**	**0:42:04**	**18.3%**

User 17 has the fewest instances in the dataset. By reducing the number of attributes, it reaches an acceleration between 4.2% and 18.3%. User 1, which has the largest number of instances, achieves a higher acceleration after feature selection, between 22.8% and 30.8%. This last value is achieved with the subset of 16 attributes, which according to the CD diagram in Figure 2, appears as best in the ranking and its results show a classification performance comparable to the original model.

For the second type of experiments, where the atypical class is taken into account during the feature selection process, the best model is the subset with 18 attributes. The Wilcoxon signed-rank test confirms that this subset achieves better performance in the context of personal risk detection and achieves a gain of 27.8% during execution time, for the user with the highest number of instances.

From Tables 5, 6, 7, and CD diagrams of Figures 2 and 3, we can confirm that feature selection based on filters maintains or significantly improves the detection performance while using fewer attributes and attains a key gain in execution time of at least 22.8% on average for the user with the largest number of observations.

B. M-OCKRA CLASSIFICATION PERFORMANCE

This section presents the results obtained when comparing OCKRA against its modified version (m-OCKRA). Subsequently, a parameter tuning of the original version of OCKRA was made in order to verify if there is a configuration of parameters that obtains better detection performance or at least similar without increasing training time. OCKRA's performance depends essentially on the value of k (number of centres) and the size of the ensemble [9]. The parameters used by the authors were 100 classifiers and $k = 10$. However, to identify which parameters produce a trade-off between classification performance and the time required for the ensemble construction, we computed the AUC using the values of $k \in \{5, 10, 20, 30, 40, 50\}$ and $N \in \{25, 50, 100\}$. In total, 18 parameter combinations were calculated. According to Friedman's test, there are no significant differences among the models; however, regardless of the number of classifiers, $k = 30$ was better in all cases, with

50 classifiers achieving the best location in the ranking according to their AUC.

Therefore, the following set of experiments comprise a comparison of the three algorithms: The reported version of OCKRA [9] with 100 classifiers and $k = 10$, OCKRA with parameter tuning using 50 classifiers and $k = 30$, and m-OCKRA as described in Section III-D.

1) SETTING M-OCKRA PARAMETERS

This section presents an experimental study to fine-tune the most significant parameters of m-OCKRA (Algorithm 1) described in Section III-D, which are: the number of classifiers of the ensemble (N), the fraction of objects for resampling (F) and the percentage of MROs ($RS_{\%}$).

In order to determine them, we tested 15 combinations of parameters, three for the number of classifiers and five different percentages: $N \in \{25, 50, 100\}$ and $F = RS_{\%} \in \{1, 2, 3, 4, 5\}$; this because beyond 5% there was no improvement in the classification and the size of the ensemble does not affect the result. The experiments showed that regardless of the number of classifiers, using between 3% and 4% for the resampling fraction and MRO objects, it is possible to achieve better detection performance.

Friedman's test shows that there are no significant differences between the models with a value between 3% and 4%; however, we selected $N = 50$, $F = 0.4$ and $RS_{\%} = 0.4$ as base parameters for the ensemble, because it achieved better ranking according to its AUC. However, it is essential to consider that for the other two versions with different ensemble size, the best percentages were $F = 0.3$ and $RS_{\%} = 0.3$.

To compare the closest distance in the classification phase (Algorithm 2), three types of distance were tested: Euclidian, Chebyshev and Manhattan. Experimental tests show that the use of Chebyshev distance improves detection performance in the ensemble of classifiers, followed by Euclidean; therefore, the classifier uses Chebyshev as a comparison metric.

Finally, although the aggregation method is not considered a parameter of the ensemble, it is worth to mention that the Majority method used to estimate the ranking and calculate the probabilities of the attributes subspace sampling, showed comparable or better results than the random method. Neither Mean nor Borda obtained good results, therefore they were discarded as methods for selecting subsets of attributes to construct the classifier ensemble.

2) EXPERIMENTAL RESULTS

Table 8 presents the detection performance results based on the AUC of the original OCKRA classifier, OCKRA with parameter tuning and m-OCKRA. The latter achieved better average AUC among the 23 users and lower standard deviation.

For statistical tests and execution time analysis, we first compared OCKRA against m-OCKRA, then the classifier with the best performance was selected and compared against OCKRA with parameter tuning.

TABLE 8. AUC of OCKRA, OCKRA with parameter tuning, and m-OCKRA.

User	OCKRA	OCKRA with fine-tuned parameters	m-OCKRA
1	**98.60**	98.54	98.19
2	95.91	96.20	**96.66**
3	91.06	91.48	**91.50**
4	88.47	**90.06**	89.32
5	90.17	93.36	**93.73**
6	**97.90**	97.69	97.79
7	**80.08**	79.36	79.02
8	92.35	92.40	**92.57**
9	**92.99**	92.13	92.14
10	**93.00**	91.38	92.65
11	91.06	90.59	**91.85**
12	80.11	**80.27**	79.49
13	80.49	82.40	**83.85**
14	82.39	82.61	**84.72**
15	94.47	**94.79**	94.50
16	87.96	89.42	**89.97**
17	**98.24**	97.89	97.75
18	86.12	87.55	**92.08**
19	88.54	89.08	**90.93**
20	**92.62**	92.57	92.05
21	**97.91**	97.56	97.79
22	78.66	78.36	**80.80**
23	69.76	71.06	**73.49**
Mean	**89.08**	**89.42**	**90.12**
SD	*7.51*	*7.26*	*6.79*

TABLE 9. Wilcoxon signed-rank test of the m-OCKRA average AUC against OCKRA average AUC, using all datasets.

Comparison	R^+	R^-	Hypothesis $(\alpha = 0.05)$	p-value
m-OCKRA vs OCKRA	205	71	Rejected	0.04152

TABLE 10. Wilcoxon signed-rank test of the m-OCKRA average AUC against the average AUC of OCKRA with parameter tuning, using all datasets.

Comparison	R^+	R^-	Hypothesis $(\alpha = 0.05)$	p-value
m-OCKRA vs OCKRA parameter tuning	207	69	Rejected	0.03544

Tables 9 and 10 show the compared classifiers (Comparison), the sum of the ranks where m-OCKRA outperformed the original (R^+), the sum of the ranks for the opposite (R^-), the result of the null hypothesis (Hypothesis), and the p value calculated by the Wilcoxon signed-rank test.

From Table 9, we reject the null hypothesis. Although the significant difference is weak ($p \approx 0.05$), it is important to note the reduction in execution time during the training phase (this analysis will be detailed later) and that the computational complexity of the clustering algorithm used to construct the classifier ensemble is $\mathcal{O}(n)$, as opposed to $\mathcal{O}(n^2)$ reported in [9].

Furthermore, since m-OCKRA achieved the best results, Table 10 shows the comparison of m-OCKRA against OCKRA with parameter tuning. Here again, we reject the null hypothesis.

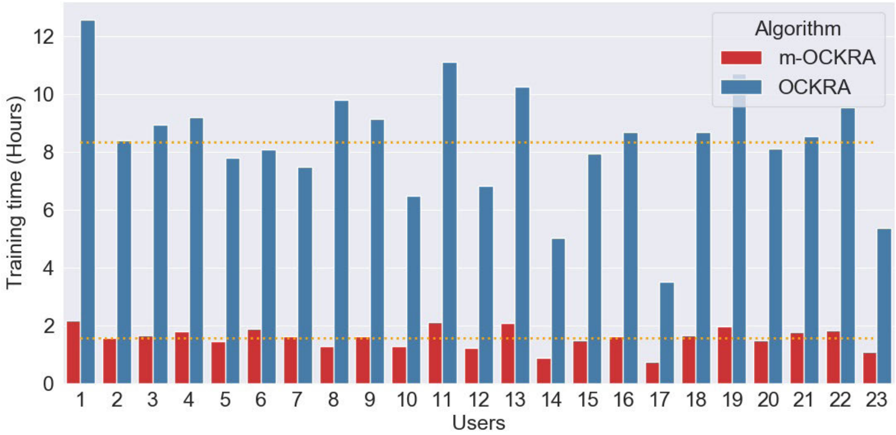

FIGURE 4. Execution time during the training phase of the OCKRA and m-OCKRA algorithms for the 23 PRIDE users.

Given that the goal is to reduce the classifier training time without sacrificing performance, we undertook further analysis. Similarly, as in the case of the feature selection procedure, we selected for this set of experiments, users 1 and 17, which have the highest and lowest number of instances in the training set, respectively. In this case, OCKRA and the modified version of OCKRA (i.e. m-OCKRA), achieved similar classification performance during the training phase; thus, we only focused on these two versions for the following analysis.

As shown in Table 11, m-OCKRA obtained a considerable reduction in the training execution time. The execution time for user 1 decreased from 12 hours to approximately 2 hours, representing a gain of 82.64%; while user 17 reduced its time from 3.5 hours to 44 minutes, that is a 79.01% gain.

TABLE 11. Execution time of the training phase of two versions of OCKRA. The $\mathcal{G}$ column indicates the percentage gain when using OCKRA against m-OCKRA.

Algorithm	User 1	$\mathcal{G}$	User 17	$\mathcal{G}$
OCKRA	12:34:41		3:30:40	
m-OCKRA	2:10:59	82.64%	0:44:12	79.01%

Finally, Figure 4 shows the execution time of OCKRA and m-OCKRA for the 23 PRIDE users. The training time was reduced by 80% on average, since OCKRA in total requires about 8.5 hours on average for the training phase, whereas m-OCKRA only about 1.5 hours.

In real-world applications such as this one, training and classification times are central. Nevertheless, Trejo and Barrera-Animas [10] argued that in the case of OCKRA with the full set of features, the classification time is negligible. The modifications made to the algorithm respected the design principles of OCKRA. This is why we placed more emphasis on the training execution times and not in classification times; the learning phase of the algorithm involves more calculations, and therefore takes much longer.

It is pertinent to recall that m-OCKRA obtained better ranking with the following parameters $N = 50$, $F = 0.4$, and $RS_{\%} = 0.4$; however, there is no statistical difference when using the values of $N = 25$, $F = 0.3$, and $RS_{\%} = 0.3$, therefore the number of classifiers in the ensemble, as well as the execution time, could be further reduced.

Based on the detection performance, m-OCKRA showed a significant difference when compared to the original version. Moreover, Table 11 shows a gain in execution time of 82.64% for the user with the highest number of instances, and 79.01% for the user with the fewest instances of the PRIDE dataset. Additionally, the significant difference remains when comparing m-OCKRA against OCKRA with parameter tuning.

Therefore, we can conclude that the modifications made to the classifier do not affect significantly the detection results in terms of AUC, and it is considerably faster during the training phase.

C. DISCUSSION

Through a set of experiments and statistical tests, we have shown that using a feature selection procedure based on filters to reduce the number of attributes in the dataset, outcomes in a comparable or significantly better classification performance than using the full feature set.

Attribute selection is not limited to the reduction of the number of attributes in the dataset to improve both efficiency and accuracy. In addition, in mobile devices where resources remain limited, eliminating attributes aids to identify sensors that can be turned off to avoid unnecessary data collection, thereby saving battery life and storage space.

In the first set of experiments, using the smallest subset of attributes (16 characteristics), feature selection primarily eliminates the attributes derived from the accelerometer and the gyroscope-accelerometer sensors, retaining only the average or standard deviation of the Z-axis in both cases. Regarding other sensors, skin temperature obtained very low rankings for two of the aggregation methods; hence, it is neither considered among the most important attributes in the PRIDE dataset. The best ranked attributes were those derived from angular velocity on the three axis, heart rate, pedometer, distance, speed, calories, and UV index.

In the second set of experiments, for subsets of the same size, the selected attributes are more diverse depending on the used aggregation method. The Mean and Majority methods show great similarities, both exclude the attributes related to the accelerometer, except for the average on the Z-axis. They also eliminate the standard deviation in the three axis of the gyroscope-accelerometer. The preserved attributes are those derived from angular velocity, heart rate, skin temperature, steps, velocity, and UV index. Borda ranks better the angular velocity in the three axis, the accelerometer in the X-axis and Y-axis; however, the lowest rankings are skin temperature and UV index.

Also, combining anomaly data to select attribute subsets significantly improves classification performance. In this case, it is possible to reduce the number of attributes to 18 and significantly improve the results.

In one-class classification problems, it is important to bear in mind that by reducing the characteristics in the dataset, relevant attributes for anomaly detection could be left out. For example, in the case of personal risk detection, the positive class contains examples of the person's normal behaviour; however, after removing attributes such as heart rate, skin temperature, or those derived from the gyroscope and accelerometer, it could happen that anomalies in the behaviour pattern (such as a fall) could have been detected through these already eliminated values. Combining these data with anomaly data at the time the feature selection process is performed, helps to acquire a more general picture of the most important attributes in problems related to personal risk detection.

Finally, by comparing m-OCKRA against two versions of OCKRA (original and the version with parameter tuning), the former algorithm manages to reduce its execution time during the training phase by almost one order of magnitude, without significantly affecting the detection performance when using the PRIDE dataset.

VI. CONCLUSION

This work describes two strategies to reduce the execution time during the training phase of a one-class classification algorithm, aiming at its efficient implementation in mobile devices, and at the same time maintain a good classification performance. First, from the dataset, we assessed a filter-based attribute selection approach, which uses descriptors or measures extracted from the data to calculate the importance of each of the characteristics of the dataset. We conducted two types of experiments to verify the impact of feature selection on one-class problems. In the second strategy, we modified the internal structure of the classifier based on the analysis of its design, in order to minimise the learning time of the classifier ensemble.

The scope of this research is limited to improving the efficiency of a one-class classification ensemble to detect anomalies in a person's behaviour patterns, without sacrificing classification performance. That is to say, the main objective is to find a balance between efficiency and effectiveness in order to obtain a viable method for its implementation in mobile devices.

In this work, we proposed an enhanced version of OCKRA, which was reported in [9] as the best one-class classifier for the personal risk detection problem, followed by ocSVM. After a filter-based feature selection procedure on the PRIDE dataset, the modified version, called m-OCKRA, achieved to maintain an equivalent performance, without significant statistical difference. Moreover, it attained a speed-up during the training phase of almost one order of magnitude. Yet, we acknowledge that other paths for feature selection are possible, by means of traditional techniques such as forward/backward feature selection, that are worth further exploring.

ACKNOWLEDGMENT

The authors would like to thank the Grupo de Investigación con Enfoque Estratégico-Machine Learning (GIEE-ML) Group at Tecnologico de Monterrey for providing useful observations and advice during the preparation of this article.

REFERENCES

[1] M. Chen, Y. Ma, Y. Li, D. Wu, Y. Zhang, and C.-H. Youn, "Wearable 2.0: Enabling human-cloud integration in next generation healthcare systems," *IEEE Commun. Mag.*, vol. 55, no. 1, pp. 54–61, Jan. 2017.

[2] J. Casselman, N. Onopa, and L. Khansa, "Wearable healthcare: Lessons from the past and a peek into the future," *Telematics Informat.*, vol. 34, no. 7, pp. 1011–1023, Nov. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0736585316307274

[3] M. Shoaib, S. Bosch, O. Incel, H. Scholten, and P. Havinga, "A survey of online activity recognition using mobile phones," *Sensors*, vol. 15, no. 1, pp. 2059–2085, Jan. 2015.

[4] M. Á. Álvarez de la Concepción, L. M. Soria Morillo, J. A. Álvarez García, and L. González-Abril, "Mobile activity recognition and fall detection system for elderly people using ameva algorithm," *Pervas. Mobile Comput.*, vol. 34, pp. 3–13, Jan. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1574119216300505

[5] S. Chernbumroong, S. Cang, A. Atkins, and H. Yu, "Elderly activities recognition and classification for applications in assisted living," *Expert Syst. Appl.*, vol. 40, no. 5, pp. 1662–1674, Apr. 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417412010585

[6] D. Tacconi, O. Mayora, P. Lukowicz, B. Arnrich, C. Setz, G. Troster, and C. Haring, "Activity and emotion recognition to support early diagnosis of psychiatric diseases," in *Proc. 2nd Int. Conf. Pervas. Comput. Technol. for Healthcare*, Jan. 2008, pp. 100–102.

[7] M. V. Albert, S. Toledo, M. Shapiro, and K. Kording, "Using mobile phones for activity recognition in Parkinson's patients," *Frontiers Neurol.*, vol. 3, p. 158, Nov. 2012. [Online]. Available: https://www.ncbi.nlm.nih.gov/pubmed/23162528

[8] A. Y. Barrera-Animas, L. A. Trejo, M. A. Medina-Pérez, R. Monroy, J. B. Camiña, and F. Godínez, "Online personal risk detection based on behavioural and physiological patterns," *Inf. Sci.*, vol. 384, pp. 281–297, Apr. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S002002551630576X

[9] J. Rodríguez, A. Barrera-Animas, L. Trejo, M. Medina-Pérez, and R. Monroy, "Ensemble of one-class classifiers for personal risk detection based on wearable sensor data," *Sensors*, vol. 16, no. 10, p. 1619, Sep. 2016. [Online]. Available: http://www.mdpi.com/1424-8220/16/10/1619

[10] L. Trejo and A. Barrera-Animas, "Towards an efficient one-class classifier for mobile devices and wearable sensors on the context of personal risk detection," *Sensors*, vol. 18, no. 9, p. 2857, Aug. 2018. [Online]. Available: http://www.mdpi.com/1424-8220/18/9/2857

[11] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. New York, NY, USA: Wiley, 2014.

[12] S. Seguí, L. Igual, and J. Vitrià, "Bagged one-class classifiers in the presence of outliers," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 27, no. 5, Sep. 2013, Art. no. 1350014.

[13] J. B. Camiña, M. A. Medina-Pérez, R. Monroy, O. Loyola-González, L. A. P. Villanueva, and L. C. G. Gurrola, "Bagging-RandomMiner: A one-class classifier for file access-based masquerade detection," *Mach. Vis. Appl.*, vol. 30, no. 5, pp. 959–974, Jul. 2018.

[14] R. Perdisci, G. Gu, and W. Lee, "Using an ensemble of one-class SVM classifiers to harden payload-based anomaly detection systems," in *Proc. 6th Int. Conf. Data Mining (ICDM)*, Dec. 2006, pp. 488–498.

[15] Y.-S. Jeong, I.-H. Kang, M.-K. Jeong, and D. Kong, "A new feature selection method for one-class classification problems," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 6, pp. 1500–1509, Nov. 2012.

[16] H. Lian, "On feature selection with principal component analysis for one-class SVM," *Pattern Recognit. Lett.*, vol. 33, no. 9, pp. 1027–1031, Jul. 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S016786551200030X

[17] A. Zainal, M. A. Maarof, S. M. Shamsuddin, and A. Abraham, "Ensemble of one-class classifiers for network intrusion detection system," in *Proc. 4th Int. Conf. Inf. Assurance Secur.*, Sep. 2008, pp. 180–185.

[18] B. Krawczyk and M. Woźniak, "Diversity measures for one-class classifier ensembles," *Neurocomputing*, vol. 126, pp. 36–44, Feb. 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231213006991

[19] E. Parhizkar and M. Abadi, "BeeOWA: A novel approach based on ABC algorithm and induced OWA operators for constructing one-class classifier ensembles," *Neurocomputing*, vol. 166, pp. 367–381, Oct. 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231215003501

[20] D. M. J. Tax and R. P. W. Duin, "Combining one-class classifiers," in *Proc. 2nd Int. Workshop Multiple Classifier Syst. (MCS)*. London, U.K.: Springer-Verlag, 2001, pp. 299–308. [Online]. Available: http://dl.acm.org/citation.cfm?id=648055.744087

[21] L. Nanni and A. Lumini, "An experimental comparison of ensemble of classifiers for bankruptcy prediction and credit scoring," *Expert Syst. Appl.*, vol. 36, no. 2, pp. 3028–3033, Mar. 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417408000249

[22] J. Tian, M. H. Azarian, M. Pecht, G. Niu, and C. Li, "An ensemble learning-based fault diagnosis method for rotating machinery," in *Proc. Prognostics Syst. Health Manage. Conf. (PHM-Harbin)*, Jul. 2017, pp. 1–6.

[23] E. Alpaydin, *Introduction to Machine Learning*. Cambridge, MA, USA: MIT Press, 2010.

[24] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Comput. Electr. Eng.*, vol. 40, no. 1, pp. 16–28, Jan. 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045790613003066

[25] J. Cai, J. Luo, S. Wang, and S. Yang, "Feature selection in machine learning: A new perspective," *Neurocomputing*, vol. 300, pp. 70–79, Jul. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231218302911

[26] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature selection: A data perspective," *ACM Comput. Surv.*, vol. 50, no. 6, Dec. 2017. [Online]. Available: https://0-doi-org.biblioteca-ils.tec.mx/10.1145/3136625

[27] K. Z. Mao, "Orthogonal forward selection and backward elimination algorithms for feature subset selection," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 1, pp. 629–634, Feb. 2004.

[28] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Jan. 2003. [Online]. Available: http://dl.acm.org/citation.cfm?id=944919.944968

[29] S. S. Khan and M. G. Madden, "One-class classification: Taxonomy of study and review of techniques," *Knowl. Eng. Rev.*, vol. 29, no. 3, pp. 345–374, Jun. 2014.

[30] L. H. N. Lorena, A. C. P. L. F. Carvalho, and A. C. Lorena, "Filter feature selection for one-class classification," *J. Intell. Robotic Syst.*, vol. 80, no. S1, pp. 227–243, Sep. 2014, doi: 10.1007/s10846-014-0101-2.

[31] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 4, pp. 491–502, Apr. 2005.

[32] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[33] J. Li, H. Wei, and W. Hao, "Weight-selected attribute bagging for credit scoring," *Math. Problems Eng.*, vol. 2013, pp. 1–13, May 2013.

[34] A. Lopez-Cuevas, M. A. Medina-Perez, R. Monroy, J. E. Ramirez-Marquez, and L. A. Trejo, "FiToViz: A visualisation approach for real-time risk situation awareness," *IEEE Trans. Affect. Comput.*, vol. 9, no. 3, pp. 372–382, Jul. 2018.

[35] M. A. Medina-Pérez, R. Monroy, J. B. Camiña, and M. García-Borroto, "Bagging-TPMiner: A classifier ensemble for masquerader detection based on typical objects," *Soft Comput.*, vol. 21, no. 3, pp. 557–569, Jul. 2016, doi: 10.1007/s00500-016-2278-8.

[36] J. Huang and C. X. Ling, "Using AUC and accuracy in evaluating learning algorithms," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 3, pp. 299–310, Mar. 2005.

[37] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, Jun. 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S016786550500303X

[38] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Jan. 2006. [Online]. Available: http://dl.acm.org/citation.cfm?id=1248547.1248548

[39] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 3–18, Mar. 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2210650211000034

[40] S. García, A. Fernández, J. Luengo, and F. Herrera, "Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power," *Inf. Sci.*, vol. 180, no. 10, pp. 2044–2064, May 2010, doi: 10.1016/j.ins.2009.12.010.

[41] R. F. Woolson, *Wilcoxon Signed-Rank Test*. New York, NY, USA: American Cancer Society, 2008, pp. 1–3. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9780471462422.eoct979

[42] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, and S. García, "KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework," *Multiple-Valued Log. Soft Comput.*, vol. 17, no. 2, pp. 255–287, 2011.

MIRYAM ELIZABETH VILLA-PÉREZ received the degree in computer systems engineering and the M.Sc. degree in computer science from the Tecnologico de Monterrey, in 2016 and 2019, respectively. She is currently a part-time Professor with the Computer Science Department, Tecnologico de Monterrey. Her research interests include pattern recognition, feature selection, one-class classification, and data science.

LUIS A. TREJO received the Ph.D. degree in computer science (parallel processing) from the Université Claude-Bernard de Lyon, France, in 1993. He is currently a full-time Professor with the School of Science and Engineering, Tecnologico de Monterrey. His research interests include internetworking, the Internet of Things, information security, intrusion detection and prevention systems, machine learning, data science, and parallel processing. Since 2015, he has been a member of CONACYT's National Research System, Level 1, and the GIEE-ML (Machine Learning) Research Group, Tecnologico de Monterrey.