Intelligent IoT Traffic Classification Using Novel Search Strategy for Fast Based-Correlation Feature Selection in Industrial Environments

Santiago Egea, Albert Rego, Belén Carro, Antonio Sánchez-Esguevillas, *Senior Member, IEEE and Jaime Lloret*, *Senior Member, IEEE*

Abstract—Internet of Things (IoT) can be combined with Machine Learning in order to provide intelligent applications to the network nodes. Furthermore, IoT expands these advantages and technologies to the industry. In this work, we propose a modification of one of the most popular algorithms for feature selection, Fast Based-Correlation Feature (FCBF). The key idea is to split the feature space in fragments with the same size. By introducing this division, we can improve the correlation and, therefore, the Machine Learning applications that are operating on each node. This kind of IoT applications for industry allows us to separate and prioritize the sensor data from the multimedia-related traffic. With this separation, the sensors are able to detect efficiently emergency situations and avoid both material and human damage. The results show the performance of the three algorithms for different problems and different classifiers, confirming the improvements achieved by our approach in terms of model accuracy and execution time.

Index Terms—Iot, Industry, Multimedia traffic, Emergency detection, Correlation based methods, Feature Selection, Filter Methods, Machine Learning.

1 Introduction

Internet of things (IoT) pretends to extend sensoring, computation and communications to every field and object. One of the most important fields where IoT can be applied is on industry. There are many advantages that industry can obtain from IoT, but also there are many challenges to resolve [1], [2]. However, when these challenges are solved, the ubiquity that industry will obtain from IoT will lead to significant improvements on its procedures. For instance, the increase~~ment~~ of hazard and emergency detection, that currently can save millions of dollars wasted due to the losses produced by those emergencies [3].

One of the techniques that can be applied to IoT is Ma~~r~~chine Learning and artificial intelligence [4]–[6]. Machine Learning has become popular in last decades for many fields, from biology to telecommunications. Machine Learning provides predictive models that are able to predict or detect responses to problems employing knowledge previously collected in a dataset. Nowadays the learning algorithms are more powerful, and our computing tools are more sophisticated. Despite of these facts, the industry poses new and more complex problems each day, with higher accuracy requirements. Applying Machine Learning to IoT introduces new constraints like more energy consumption or computation time. In other words, the complexity of these challenges is increasing constantly. These issues force data scientist to pay attention, not only in learning algorithm designing, but also in efficient information processing. The majority of learning algorithms are able to model problems more accurately when the input of the classifier is optimal [7]. Thereby, to remove useless features is a much recommended practice, and this task is carried out by feature selection methods.

The effectiveness of feature selection has already been proved in numerous works. In fact, these techniques are considered essential in data preprocessing stages [8]. Feature selection consists of selecting the relevant features from the original dataset and remove the rest that could be potentially irrelevant or/and redundant for the problem [7].

The advantages of performing feature selection are well-known [9]: preventing the model from overfitting the training set, thus increasing the accuracy over the test set; reducing both storage and computing resources needed; improving the interpretability of predictive models, since feature selection mitigates the curse of dimensionality; and remaining a suitable tradeoff between number of instances and number of features, as this relationship is crucial for some learning algorithms.

According to the way in which the problem is tackled, feature selection methods are mainly split in three groups [9]–[11]: filter methods, wrappers methods and embedded methods. Filter methods use a relevance measurement in order to classify the features as useful or not, according to

S. E., B. C. and A. S.-E. are with the Universidad de Valladolid, Spain. E-mail: santiago.egea.gomez@gmail.com (S. E.), belcar@tel.uva.es (B. C.), antoniojavier.sanchez@uva.es (A. S.-E.). A. R. and J. L. are with the Universitat Politecnica de Valencia, Spain, E-Mail: alremae@teleco.upv.es (A. R.), jlloret@dcom.upv.es (J. L.)

a threshold [9], [12]. Filter methods are computationally very light and, also, they are scalable and independent of the learning algorithm employed in the problem. However, the subset resulted from filter methods is not the optimal one. Furthermore, a criterion has to be chosen for measuring the feature relevance. Therefore, lots of subgroups are included into this category. The other feature selection technique includes the wrappers methods [9], [10], [12]. Their principles are based on the fact that Machine Learning algorithms are capable of scoring the features during the training process. Once the predictive model is built, we can get a subset for modelling tasks via observing the learning algorithm structure. These methods are slower; since they need to train a classifier and, additionally, the possible subsets have to be validated by cross validation or other validation technique. Furthermore, wrapper methods have difficulties in terms of scalability and have high risk of overfitting the training set. But they usually produce more accurate subsets than filter methods for a specific classifier.

The most modern techniques are the embedded methods. These techniques are implemented inside the learning algorithm and their search strategy is guided by the learning process. As embedded methods are optimized for a specific learning algorithm, they are faster than wrappers methods and achieve the best subsets; however, they are fully dependent on the used learning algorithm.

The feature selection methods are formed by six properties or phases [10]: Initial state of search, creating successors, search strategy, feature evaluation method [13], and stop criterion.

This work is focused on filter methods, and namely, on methods based on correlation measurements. The Fast Correlation Based Filter [14] (FCBF) is the most popular of them. Later, a new strategy approach was introduced in [15], this algorithm is known as FCBF#.

In this paper, we introduce a novel search strategy whose goal is to give a tuning parameter that allows users to control both the algorithm computing time and the intercorrelation among the features contained in the resulting subset. With this proposal, we are able to create an optimal subset of features to classify the traffic propagated through an IoT network implemented in an industrial facility. Therefore, the detection multimedia traffic is improved thanks to this proper selection of the features and can be separated in a better way from the sensor data, increasing the efficiency of the critical and priority use and management of that data. So, the applications using that critical data such emergency detection get better performance. This algorithm is called FCBF in Pieces (FCBFiP).

The paper is organized as follows. First of all, in section 2, a review of the current state of IoT and Machine Learning in the literature is presented. In section 3, we review the prior algorithms and explain our proposal. Next, in section 4, we describe the experiments carried out to validate our proposal. In section 5, we show and discuss the results obtained for our algorithm and the prior ones by using four different datasets. Finally, in section 6, we draw conclusions about the results obtained.

2 RELATED WORK

In this section, some of the works related to IoT for industry and Machine Learning are discussed.

In [16], J. Wan et al. propose and analyze a new entity for production processes in industry called Context-Aware Cloud Robotics (CACR). This new entity does an effective load balancing and provides context-aware services in factories. This CACR improves the material handling. In the paper, the arch~~qu~~itecture of CACR is showed, analyzed and discussed. The results show that CACR, working with decision-making algorithms, works in a more energy-efficient mode and increases the cost-saving during the material handling.

An advantage related to the use of IoT for industry is the reduction of energy-consumption during the production process. These kind of energy-related issues are discussed in [17], where sustainable development and green technologies are the point in order to saving energy and reducing emissions.

Related to environ~~m~~ment, in [18] A. Mehmood et al. propose an artificial neural network in order to save energy and to make the routing scheme more robust. This neural network, called ELDC, has been designed for industry pollution monitoring and increases the lifetime of the nodes by incorporating the features of group based protocols. The nodes are able to put some nodes that increase their energy-consumption pace by sending sleep commands. The results show that the lifetime of the nodes is increased over 40% compared against other algorithms.

There are some published works related to pollution monitoring and saving energy. In [19], the increase of pollution and carbon footprint problems are discussed and a solution given in terms of routing protocol is proposed. This routing protocol, called Secure and Low-energy Zone-based Routing Protocol (SeLeZoR) is designed in order to face two problems: energy consumption and security. Taking some assumptions from the features of Wireless Sensor Networks, the Base Station divides the network into zones and clusters, reducing the number of messages. The results show an increase~~ment~~ around 400% in terms of the time that all nodes are alive. Moreover, the energy wasted is reduced.

Traffic classification and filtering has been deeply ~~appli~~plyed in severals works and fields. A case study is realized by R. Gupta et al. in [20], where the internet traffic survellance and network monitoring in India is studied. Under the context of preventing terrorist attacks, India is working towards development of surveillance systems. One of this kind of systems is NETRA, used by the Indian Government to search suspicious keywords from messages in the network. In [20], NETRA is compared against some other similar systems like Dish Fire, Prism or Echelon. Their work shows how NETRA works and how it filters the messages and traffic. Authors conclude that it shows only a few weakness in spying the content.

This traffic monitoring and processing has been also applied in IoT environments. In [21], J. Zheng et al. introduce a non-intrusive traffic data collection for intelligent transportation systems using wireless sensor networks. Placing magnetic sensor nodes on the road, they are able to collect data from vehicles and obtain the vehicle flow data. This data is sent to a control center using Zigbee protocol, where the final vehicle flow data is calculated by filtering and decision-making algorithms. The architecture is showed and experiments are done in order to demonstrate that the method illustrated is reliable. This process is non-intrusive to the transportation systems.

That traffic monitoring can be used to obtain some flows or patterns like the discussed in the previous reference. However, it can be also used for improving the performance of the network. In [22], M. Avvenuti et al. propose a MAC protocol, an extension from B-MAC+ protocol, which reduces the energy consumption for communication in wireless sensor networks. This protocol is adaptive and asynchronous. It adapts depending on the observed traffic load and changes its operational parameters. The duty cy~~i~~cle is either increase~~a~~d or decreased attending to the incoming packet number variation. The protocol is distributed into the nodes of the network. This protocol is described, an example is given and a performance evaluation is done through two different simulated scenarios. The results show that the adaptive B-MAC+ protocols achieves a network lifetime 1.35 up to 2.8 times longer than the standard B-MAC+ protocol.

Furthermore, the collection and analysis of the data not only is used to reduce energy consumption with MAC-level protocols or to produce new data, but also is used to create a general view of the state of the network. In [23] D.Tang et al. introduce a new congestion-aware routing scheme that is based on the traffic information given from the sensors in a wireless sensor network. Congestion is one of the most important problems in networks and the proposal consists on reducing the delay existent in the network by being aware of the congestion that can be produced. Moreover, the throughput is also increased. The routing scheme described achieves its goals by using a geographic routing scheme. Therefore, the relay node is selected attending to the sensor node location and the current congestion of the area. The traffic sent by that local area is analyzed and due to that traffic information the algorithm selects the next hope node in the path. The simulations presented in the work show that the end-to-end packets transmission delay is reduced by 50% and the throughput of the network is doubled.

Finally, in terms of Machine Learning, that provide us lots of techniques to make this kind of networks intelligent, filter methods are vital to obtain a good performance in decisions. In [14], FCBF is presented, and a new upgrade is described in [15]. This last method is called FCBF#. They are explained in detail in the next section.

Concerning network traffic classification, correlation-based filters have been employed to this modelling task for several years ago. In [24] Williams et al. provided a comparison between learning algorithms, but, additionally, they demonstrated that correlation-based filters are suitable for traffic classification.

Many authors have provided solutions to select the most informative attributes to identify network traffic. In [25] a hybrid feature selection algorithm is presented for high-speed networks. The algorithm consists of two selection phases, the less relevant and most redundant attributes are prefiltered using a new metric called Weighted Symmetrical Uncertainty at the first stage, and later, the final subset is provided training different~~s~~ learning algorithms and evaluating the Area Under Curve performance metric. The authors reported significant improvements in terms of True Positive Rate and False Positive Rate.

More recently, Adil Fahad et al. proposed an novel feature selection scheme to obtain optimal and stable subsets for traffic classification in [26]. They discuss the traffic profiling changes, how they affect the classifier performances and propose new metrics to assess the optimality ~~ans~~ and stability of subsets. In order to avoid performances losses, they present a multi-criterion feature selection method called Global Optimization Approach (GOA). GOA combines well-known feature selection techniques to filter out the irrelevant attributes and the resulting subset is processed to extract the stable features based on information theory measures.

The different works commented in this section search for improving the performance of the sensor networks, either by reducing energy consumption or delay or by increasing throughput and time alive. In order to achieve their goals, the authors proposed new routing schemas, algorithms or data processing.

In this paper, we work on improving the core of the intelligent network decision. A new filter method based on FCBF is presented. That method improves the correlation of the features. Therefore, the algorithms and Machine Learning tools that use it will increase their performance. That makes the classification and detection algorithms better. The method presented is thought to being used for multimedia traffic classification in IoT for industries. Specifically, in facilities where the data sensed is used for emergency detection and is sent through the network beside multimedia traffic. The improvement of detection algorithms and special processing of the sensor data will have repercussions in reducing losses.

3 Fast Correlation Based Feature Selection

Many researchers have approached the feature selection problem from different viewpoints. Filter methods are underpinned by mathematical and statistical concepts as entropy, mutual information [13] or correlation measurements [27]. Relief algorithm [28] measures the feature relevance, but is not capable of removing redundant features. Later, correlation based approaches have been used in order to mitigate features redundancy, like CFS [27]. Afterwards, L. Yu and H. Liu [14] presented the FCBF algorithm, which speeds up the selection process. FCBF algorithm has been tested in many modelling problems, proving its excellent performances. In [15], the search strategy of FCBF was improved and a stop criterion was included. In our proposal, we implement new capabilities for FCBF. The key idea is to split the feature space in pieces with same size, compute the redundancy of each feature with a multivariate evaluation method and rank them. Each piece is processed independently. According to the scores assigned to the features and the number of features selected for the resulting subset, the algorithm drops the worst features and includes the rest into the model. The size of the pieces is a design parameter which allows us to control the tradeoff between execution time of the algorithm and intercorrelation of the resulting subset.

3.1 FCBF Algorithm

Fast Correlation Based Feature selection (FCBF) [14] uses the symmetrical uncertainty as evaluation method. The symmetrical uncertainty takes some advantages against other correlation measures: is normalized between 0 and 1; detects several kind of correlations (not only linear correlation); and compensates for information gain´s bias.

Symmetrical uncertainty uses the concept of entropy to measure the correlation between features. Given a feature X that can take i different values (x_i) with different occurrences, the entropy of X is defined as:

$$H(X) = -\sum_i P(x_i)\log_2(P(x_i)) \tag{1}$$

Where $P(x_i)$ is the probability of X to take x_i. The entropy of X given other feature Y is called conditional entropy of X over Y, and is defined as:

$$H(X \mid Y) = -\sum_j P(x_j)\sum_i P(x_i \mid y_j)\log_2\left(P(x_i \mid y_j)\right) \tag{2}$$

Now, we define the information gain as:

$$IG(X \mid Y) = H(X) - H(X \mid Y) \tag{3}$$

Finally, the symmetrical uncertainty between X and Y is defined as:

$$SU(X,Y) = 2\left[\frac{IG(X \mid Y)}{H(X)+H(Y)}\right] \tag{4}$$

Note that a value $SU(X,Y) = 1$ indicates a completely correlation between X and Y. Meanwhile $SU(X,Y) = 0$ indicates that variables are not correlated.

The search strategy used by FCBF sorts the feature space based on the symmetrical uncertainty between each feature and the class. The overall complexity of FCBF is $O(NlogN)$ [14]. And FCBF does not have stop criterion, so that it finishes the search when the whole feature space has been explored. This fact is a shortcoming, since FCBF removes features without the possibility of choosing the number of features desired for the model. Nevertheless, the FCBF efficiency has already been shown [14].

3.2 FCBF#

FBCF# tries to overcome the above issue, and also modifies the search strategy [15]. A stop criterion has been included in the algorithm by introducing a natural parameter k. When the subset has k features, the algorithm finishes the search and returns the subset. In addition, the search strategy has been changed, so that the process starts removing the irrelevant features during the first iterations. Unlike FCBF, that starts removing the relevant features, in [15], authors have used a stop counter in their FCBF implementation in order to compare models with same number of features. The results prove that the change in the search strategy improves the model accuracy. However, the algorithm is slightly slower than FCBF.

3.2 Our proposal: FCBF in Pieces (FCBFiP)

Our algorithm, FCBFiP, includes two significant modifications respecting to the previous versions: the feature space is divided in P pieces and the criterion to remove the features is based on a scoring step.

Both FCBF and FCBF# consist of two steps. The first one evaluates the relevance of each feature for predicting the target class, and sorts them in descending order (sequence 1). This step remains in our algorithm and the second one is modified to avoid iterations which goes over the whole feature space. At the first step, if there are two or more correlated features, it is expected that they have similar relevance for forecasting the response. Thus, they have to be close in the ordered sequence of features (sequence 1). Then, it is feasible to think that is not necessary to evaluate the redundancy of a variable over the whole feature space but evaluate the redundancy ~~in~~ on its neighboring may be enough. The number of pieces defines the size of the vicinities as:

$$Vsize = \frac{N}{P} \tag{5}$$

Where N is the number of features in the original dataset and P the amount of selected pieces.

In this fashion, we can save up many operations if P is large. On the other hand, the resulting subset could contain redundant features, as the vicinity size is small. In opposite way, if P is lesser, we will spend more time to process each piece and the resulting subset will present lower intercorrelation among the features included in it. To control the degree of redundancy in the resulting subset may be beneficial depending on the nature of the problem we are modeling. Other advantage of splitting the feature space is that modern programming languages offer tools to parallelize the computation, speeding up the algorithm, since each piece can be processed independently. This fact will be considered for future implementations of FCBiP.

As evaluation method for determining the redundancy of each feature, we compute the mean symmetrical uncertainty (6) between a given feature and its neighbors.

$$\overline{SU}\left(X_i,V\right)=\frac{1}{Vsize-1}\sum_{j=V;j\neq i} SU\left(X_i,X_j\right) \tag{6}$$

Where V is the vicinity that contains the feature X_i.

In the scoring step, the aim is to classify the features according to its relevance and redundancy into its piece. After computing the mean symmetrical uncertainty for each feature, they are sorted in ascending order (sequence 2). Next, the score assigned to each feature is the sum of the position they occupy in sequence 1 and sequence 2. Finally, FCBFiP removes the features with greater score until the subset contains k features.

The process to obtain the sequence 2 is described in Fig. 1. Firstly, we split the feature space in P fragments. Next, we compute the $\overline{\text{SU}}$ for each feature into its vicinity. Finally, we order the feature space in ascending $\overline{\text{SU}}$ order to get sequence 2.

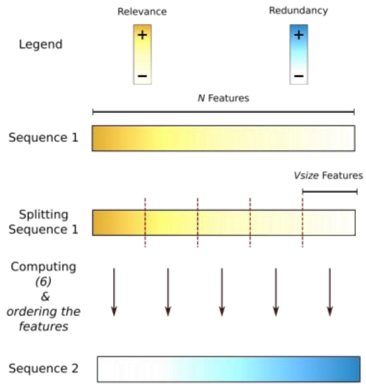

Fig. 1. Description of the process used to obtain the sequence 2.

This approach suffers a crucial limitation. The number of pieces, P, has to be a divisor of N. Thus, when N is a prime number or has few divisors, a feature preselection using FBCF# is the best solution.

4. Methods

In this section we describe the experiments carried out. We have selected four datasets corresponding with different classification problems and related to areas that could extrapolated for IoT. Then, we have preprocessed them in order to suit them to the algorithm inputs. These preprocessing steps differ among them, as the formats of the datasets also differ. The following sections go in depth in the experiment setting.

4.1 Tools

The tools used to perform the experiments were Python libraries. For building the model we used Sklearn [29]. All algorithms were programmed using Numpy [30].

4.2 Datasets

We chose four datasets. To make the results more general, we looked for datasets whose ratio between #Instances-#Features and origin differ. Also the number of classes to forecast differs. Table 1 summarizes the characteristics of each dataset.

The Orange [31] dataset was purposed for the KDD Cup Orange Challenge. Several authors have written about this challenge (e.g. [32], [33]). This dataset is highly complex, therefore we used a small version of the original dataset. Additionally, we simplified the problem to solve only the churn prediction task, therefore, this problem is a binary classification. In the IoT industry, numerous services are rising~~en~~ up and providers of services will compete in an emerging market. Thus, churn prediction also applies to IoT (as a matter of fact many lines affected if a customer changes provider).

The KDD99 [34] dataset consists of about 4.370.000 data flows represented by 41 features. And the aim is to identify whether the each flow corresponds to a computer attack or to a normal behavior. Other works have already been published using this dataset (e.g. [35]). In this experiment, we used a reduced version of this dataset that includes 10% of all samples (437.000). There are 23 different attacks to predict. Although 41 is prime, this is not a limitation, since the algorithm is capable of detecting this situation and dropping the less relevant feature. IoT traffic goes through network infrastructures to implement the communication between devices. Therefore, the IoT sensors are as sensitive to cyber-attacks as other devices, such as personal computers. Thereby, guaranteeing the security of IoT devices is a must to assure the services. Attack detection via Machine Learning could be a promising solution for IoT attacks.

TABLE 1
DATASET INFORMATION

Name	#Features	#Classes	#Instances	Ratio
Small Orange	230	2	50.000	217.39
10% KDD99	41	23	437.000	10658.54
CNAE-9	856	9	1.080	1.26
LSVT voice	309	2	126	0.41

The CNAE-9 [36] dataset is extracted from a text mining problem. The dataset contains 1080 free text business descriptions of Brazilian companies, [37]. The goal is to classify these descriptions in 9 categories. The features are 856 word frequency records. IoT customers are typically enterprises. Therefore, its description is quite useful in order to classify target customers.

The LSVT voice [38] dataset was used to predict Parkinson´s disease evolution, [39]. It is a binary problem, since the authors labelled with "1" patients whose disease evolution is positive, and "0" the opposite case. This dataset has 309 features corresponding to 126 patients. Thus, the ratio between #Instances-#Features is lesser than 1. Digital home virtual assistants is an emerging category of IoT devices. In this context, Machine Learning models could be employed to monitor patients based on their voice inputs.

4.3 Preprocessing

Due to the differences between the datasets used in our experiments, we preprocessed each dataset differently.

The Small Orange Dataset contains artificial variables introduced by the promoters of the challenge. Thus, we have removed the features that only take a value, as they do not give useful information [32]. Also we filled the missing values with the feature mean value in case of the numeric features. This dataset is formed by categorical 40 variables. These variables were encoded with strings to assure the anonymity of the data. Thus, we have mapped these variables with integer values, including the missing values. Finally, the resulting dataset had 212 variables. To suit the KDD99 dataset to our experiments we shuffled randomly the samples several times, since the instances were sorted by the class to predict. Furthermore, this dataset has three categorical features and the class coded as string. All of them were mapped with integer values. The CNAE-9 dataset also had the instances ordered. Thus the samples were shuffled randomly in the same way as the former dataset. Besides, the dataset was normalized between 0 and 1, as the classifier used for this dataset is sensitive to feature ranges. The LSVT voice dataset was also shuffled. Additionally, we normalized the dataset between 0 and 1, as the selected classifier requires. Finally, we have carried out a feature selection step, since the number 309 has only two divisors (3 and 103). To get more divisors, we applied the FCBF# algorithm with $k = 306$.

4.4 Classifers used

For the Orange dataset, we chose a decision tree classifier because this kind of classifier needs less training time than others. To avoid overfitting the training set, the depth of the decision tree was limited to 6, and the minimum samples per leaf was set to 22.

In the case of the KDD99 dataset, we also used a decision tree with the same parameters as above to decrease the computing requirements for the experiments.

For the CNAE-9 dataset, we modeled the problem by using Support Vector Machines (SVM). The regularization parameter, C, was fixed to 40.

For the last dataset (LSVT voice), we observed that logistic linear regression outperforms slightly SVM classifier. Thus, we used logistic regression to tackle this problem. The regularization parameter, C, was set to 1.

For all multiclass problems (KDD99 and CNAE-9), the approach used to assign the final class to the samples was One-vs-the-Rest (OvR) strategy.

4.5 Model Validation

The measurements to assess the model validity were the F1 score for all problems, except for the Orange Dataset. The F1 score was selected due to the fact that it gives information about the model precision and recall [40]. The F1 score is defined as:

$$F1 = 2\frac{precision \cdot recall}{precision + recall} \qquad (7)$$

The AUC-ROC score was used for the Orange Dataset, because it was the score proposed by the promoters of the challenge [32].

As validation algorithm, we chose k-fold cross validation, since it is a low variance method. The folds were fixed to 10 for all datasets; except for KDD99 dataset, we used 5 folds as the dataset contains samples enough. All experiments were repeated 10 times, and we computed the mean of the resulting scores in order to rank the feature selection algorithms. For the multiclass problems, we computed the mean of the score over all possible classes.

5 Results

Figure 2 to Figure 5 present the relevant results obtained from the experiments carried out, both model performance and execution time are shown.

Figure 2 depicts the results obtained for the Orange dataset. FCBFiP did notably speed up the selection process when the feature space was divided in 106 and 53 pieces. However they did not get the highest AUC-ROC score, although, in most cases, their performances are quite close to the other candidates. Even, FCBFiP overcame FCBF# when models with 40 and 60 features were chosen. The FCBF algorithm returned a subset with six features. For this subset size, the best results were achieved by FCBFiP with $P = 4$, but the spent time was significantly greater than the other algorithms. Note also that, for a resulting subset with more than 120 features, it was possible to obtain a model with similar performance that FCBF#, but spending much less time. Finally, the global maximum performance was accomplished by FCBFiP with $P = 2$ for a model that included 180 features. However, the time required was quite greater than the FCBF# algorithm.

In the case of the KDD99 dataset, Figure 3, we note that FCBF# overcame its competitors when 10 features are selected in terms of accuracy. However, the FCBiP algorithm obtained better performances than the other ones for models with more than 10 features. FCBFiP with $P = 10$ achieved the highest score for a model with 20 variables and the same happened with FCBFiP with $P = 8$ for 30 features. These results reveal that the intercorrelation among features in a model may be beneficial in specific cases. However, the time spent in these cases was greater than the time spent by FCBF#. The best results in terms of F1 score were obtained using FCBFiP with $P = 5$ for a model with 12 features, but it lasted more time than FCBF#.

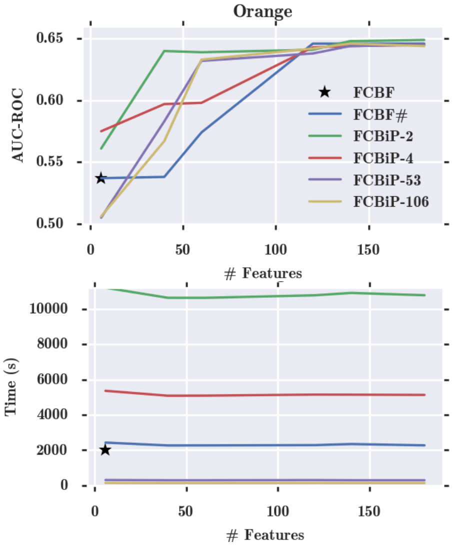

Figure 2. Performances obtained for Orange Dataset

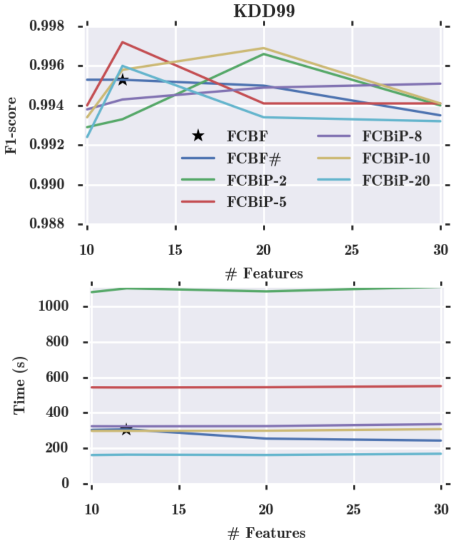

Figure 3. Performances obtained for KDD99 Dataset

Figure 4 shows the results obtained modelling the CNAE-9 problem. Note that FCBF algorithm yielded a model with 47 features. In this case, the FCBF outperformed the other candidates. The FCBF# and FCBFiP performances increased as the number of features included in the model was gradually raised. Also the F1 scores obtained applying FCBFiP differed considerably when the number of pieces varies for models with less than 500 features. In this case, the FCBFiP performances were very poor and were overcame clearly by FCBF and FCBF#. However, the best result was obtained by FCBFiP algorithm with $P = 107$ for a model with 500 features. It achieved higher score than FCBF# taking half of the time. These results show that penalizing the intercorrelation between features may improve the accuracy of the model for specific cases.

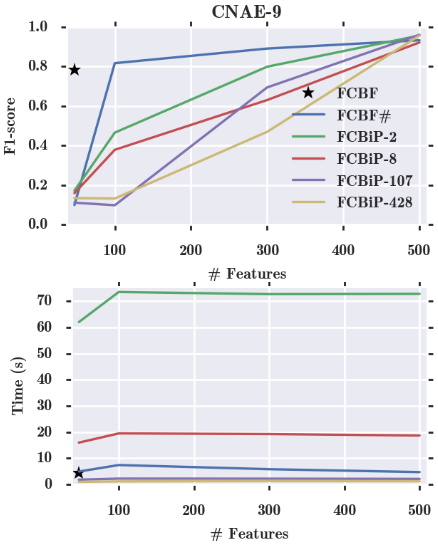

Figure 4. Performances obtained for CNAE-9 Dataset

Figure 5 shows the results obtained by modeling the LSVT voice problem. As FCBFiP-2 and FCBFiP-6 obtained quite high execution times to visualize the plot properly, both temporal curves were excluded from the figure; the execution time was around 145 seconds for FCBFiP-2 and 51 seconds in the instance of FCBFiP-6. This dataset presents a ratio between #Instances-#Features lesser than 0.5. Note that FCBF returned a subset with one feature. In this case, all algorithms converged in the same solution. That fact may be due to the samples scarcity, since it is related to the available information for the selection and modeling processes. Note that there are more cases in which the different algorithms get the same results, for example when a model with 15 feaures are selected. For this experiment, the FCBFiP algorithm did not offer great advantages in terms of execution time. Nonetheless, the most accurate model resulted by using FCBFiP with $P = 6$ and 30 features.

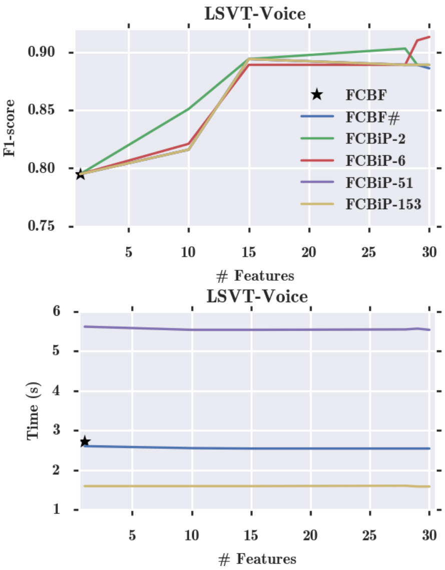

Figure 5. Performances obtained for LSVT-voice

6 Conclusion

In this work we review some feature selection filters based on correlation measurements, and we propose a novel approach for providing new functionalities to the FCBF algorithm in order to improve IoT-based intelligent networks in Industrial facilities. Our proposal consists of a modification of the original FCBF algorithm by changing the evaluation method of the redundancy and including a scoring process for ranking the variables. The new redundancy evaluation was developed in two steps: first, by splitting the feature space in P pieces with the same size; and second, by evaluating the feature redundancy in the piece that contains it with a multivariate correlation measurement. This evaluation method allows us to set the number of pieces to split the whole feature space, being this parameter able to control both the execution time and the redundancy penalty in the selection process. The scoring process is carried out by ordering the sequences of features according to their relevance and redundancy measurements; assigning the scores according to the position each feature occupies in these sequences; and removing the features that obtain the worst scores. We validated our proposal by comparing it with the FCBF and FCBF# algorithms.

The datasets selected for the experiments were very different from each other to make the results more generalizable. Additionally, we modelled the problems by using different learning methods for each dataset, namely: Decision Trees, SVM and Logistic Linear Regression. The global highest performance for each experiment was achieved by our algorithm in terms of accuracy. Note that best accuracy does not always imply less execution time, parameter for which our algorithm offers a clear advantage. It is possible to obtain a subset with similar performances than the obtained by FCBF or FCBF# but spending much less time. Therefore, we have accomplished a more flexible solution by tuning a new design parameter. Furthermore, we can conclude that a lesser redundancy penalty improved the accuracy of the model built for some of the cases under study. We have found that the ratio between #Instances and #Features actually affects the selection process.

Further work can be done opening new lines for upgrading the FCBFiP algorithm: mixing evaluation methods (e.g. including mutual information scores) and parallelizing operations to speed up the algorithm. Besides, performing more experiments using other datasets might complete and expand the conclusions. For this aim, the code of the algorithm has been published in Github [41]. Feedbacks and debug reports are welcome. Moreover, a first implementation can be tested in an IoT environment, using sensor nodes to collect data and FCBFiP algorithm to classify traffic in order to check the increment of performance in the entire IoT system. Nonetheless, this algorithm have already been applied to a network traffic classification task in [42], in that work we employed this algorithm to build consistent subsets to identify Internet traffic in two different contexts.

Another research that can be done from the presented work is to check if this new method of features selection can be used to improve some other ~~tipical~~ typical parameters in IoT networks, like energy consumption o routing decisions.

Acknowledgment

This work was supported in part by the Ministerio de Economía y Competitividad del Gobierno de España and the Fondo de Desarrollo Regional (FEDER) within the project "Inteligencia distribuida para el control y adaptación de redes dinámicas definidas por software, Ref: TIN2014-57991-C3-2-P", and in part by the Ministerio de Economía y Competitividad in the Programa Estatal de Fomento de la Investigación Científica y Técnica de Excelencia, Subprograma Estatal de Generación de Conocimiento within the Project "Distribucion inteligente de servicios multimedia utilizando redes cognitivas adaptativas definidas por software, Ref: TIN2014-57991-C3-1-P". In Addition, this work has been partially supported by the "Ministerio de Educación, Cultura y Deporte", through the "Ayudas para contratos predoctorales de Formación del Profesorado Universitario FPU (Convocatoria 2015)". Grant number FPU15/06837.

References

[1] M. Garcia, D. Bri, S. Sendra, and J. Lloret, "Practical Deployments of Wireless Sensor Networks : a Survey," *Int.*

Con formato: Fuente: Palatino, 9,5 pto, Color de fue
Negro, Espacio ajustado en 8 pto

Con formato: Fuente: Palatino, 9,5 pto, Espacio ajus
8 pto

Con formato: Sangría: Primera línea: 0 cm

J. Adv. Networks Serv., vol. 3, no. 1, pp. 170–185, 2010.

[2] L. Da Xu, W. He, and S. Li, “Internet of Things in Industries: A Survey,” *IEEE Trans. Ind. Informatics*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.

[3] S. Mannan and F. P. Lees, *Lees’ loss prevention in the process industries : hazard identification, assessment, and control.* Elsevier Butterworth-Heinemann, 2005.

[4] D. Ventura, D. Casado-Mansilla, J. López-de-Armentia, P. Garaizar, D. López-de-Ipiña, and V. Catania, “ARIIMA: A Real IoT Implementation of a Machine-Learning Architecture for Reducing Energy Consumption,” Springer, Cham, 2014, pp. 444–451.

[5] Ru Xue, Liang Wang, and Jie Chen, “Using the IOT to construct ubiquitous learning environment,” in *2011 Second International Conference on Mechanic Automation and Control Engineering*, 2011, pp. 7878–7880.

[6] M. A. Alsheikh, S. Lin, D. Niyato, and H.-P. Tan, “Machine Learning in Wireless Sensor Networks: Algorithms, Strategies, and Applications,” *IEEE Commun. Surv. Tutorials*, vol. 16, no. 4, pp. 1996–2018, 2014.

[7] “Feature selection for classification,” *Intell. Data Anal.*, vol. 1, no. 1–4, pp. 131–156, Jan. 1997.

[8] G. H. John, G. H. John, R. Kohavi, and K. Pfleger, “Irrelevant Features and the Subset Selection Problem,” *Mach. Learn. Proc. Elev. Int.*, pp. 121--129, 1994.

[9] I. Guyon, A. Elisseeff, and A. M. De, “An Introduction to Variable and Feature Selection,” *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, 2003.

[10] L. C. Molina, L. Belanche, and A. Nebot, “Feature selection algorithms: a survey and experimental evaluation,” in *2002 IEEE International Conference on Data Mining, 2002. Proceedings.*, pp. 306–313.

[11] Huan Liu and Lei Yu, “Toward integrating feature selection algorithms for classification and clustering,” *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 4, pp. 491–502, Apr. 2005.

[12] “Selection of relevant features and examples in machine learning,” *Artif. Intell.*, vol. 97, no. 1–2, pp. 245–271, Dec. 1997.

[13] Hanchuan Peng, Fuhui Long, and C. Ding, “Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.

[14] L. Yu and H. Liu, “Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution,” *Int. Conf. Mach. Learn.*, pp. 1–8, 2003.

[15] B. Senliol, G. Gulgezen, L. Yu, and Z. Cataltepe, “Fast Correlation Based Filter (FCBF) with a different search strategy,” *2008 23rd Int. Symp. Comput. Inf. Sci. Isc. 2008*, 2008.

[16] J. Wan, S. Tang, Q. Hua, D. Li, C. Liu, and J. Lloret, “Context-Aware Cloud Robotics for Material Handling in Cognitive Industrial Internet of Things,” *IEEE Internet Things J.*, pp. 1–1, 2017.

[17] G. Han, M. Guizani, J. Lloret, H. Wu, S. Chan, and A. Rayes, “Recent advances in green industrial networking [Guest Editorial],” *IEEE Commun. Mag.*, vol. 54, no. 10, pp. 14–15, Oct. 2016.

[18] A. Mehmood, Z. Lv, J. Lloret, and M. M. Umar, “ELDC: An Artificial Neural Network based Energy-Efficient and Robust Routing Scheme for Pollution Monitoring in WSNs,” *IEEE Trans. Emerg. Top. Comput.*, pp. 1–1, 2017.

[19] A. Mehmood, J. Lloret, and S. Sendra, “A secure and low-energy zone-based wireless sensor networks routing protocol for pollution monitoring,” *Wirel. Commun. Mob. Comput.*, vol. 16, no. 17, pp. 2869–2883, Dec. 2016.

[20] R. Gupta and S. K. Muttoo, “Internet Traffic Surveillance &amp; Network Monitoring in India: Case Study of NETRA,” *Netw. Protoc. Algorithms*, vol. 8, no. 4, p. 1, Jan. 2017.

[21] J. Zheng *et al.*, “Non-intrusive Traffic Data Collection with Wireless Sensor Networks for Intelligent Transportation Systems,” *Ad Hoc Sens. Wirel. Networks*, vol. 34, pp. 41–57, 2016.

[22] M. Avvenuti, C. Bernardeschi, L. Cassano, and A. Vecchio, “Adapting the duty cycle to traffic load in a preamble sampling MAC for WSNs: Formal specification and performance evaluation,” vol. 31, pp. 101–129, 2016.

[23] D. Tang, T. Li, and J. Ren, “Congestion-aware routing scheme based on traffic information in sensor networks,” *Ad-Hoc Sens. Wirel. Networks*, vol. 35, pp. 281–300, 2017.

[24] N. Williams, S. Zander, and G. Armitage, “A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 5, p. 5, Oct. 2006.

[25] H. Zhang, G. Lu, M. T. Qassrawi, Y. Zhang, and X. Yu, “Feature selection for optimizing traffic classification,” *Comput. Commun.*, vol. 35, no. 12, pp. 1457–1471, 2012.

[26] A. Fahad, Z. Tari, I. Khalil, A. Almalawi, and A. Y. Zomaya, “An optimal and stable feature selection approach for traffic classification based on multi-criterion fusion,” *Futur. Gener. Comput. Syst.*, vol. 36, pp. 156–169, 2014.

[27] M. A. Hall, “Correlation-based Feature Selection for Machine Learning,” 1999.

[28] K. Kira and L. Rendell, “The feature selection problem: Traditional methods and a new algorithm,” *Aaai*, pp. 129–134, 1992.

[29] F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python,” *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2012.

[30] “NumPy — NumPy.” .

[31] “SIGKDD : KDD Cup 2009 : Customer relationship prediction.” [Online]. Available: http://www.kdd.org/kdd-cup/view/kdd-cup-2009. [Accessed: 29-Nov-2017].

[32] R. Niculescu-mizil *et al.*, “Winning the KDD Cup Orange Challenge with Ensemble Selection.”

[33] U. Yabas and H. C. Cankaya, “Churn prediction in subscriber management for mobile and wireless communications services,” in *2013 IEEE Globecom Workshops (GC Wkshps)*, 2013, pp. 991–995.

[34] “KDD Cup 1999 Data.” [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html. [Accessed: 29-Nov-2017].

[35] M. K. Siddiqui and S. Naahid, “Analysis of KDD CUP 99 Dataset using Clustering based Data Mining,” *Int. J. Database Theory Appl.*, vol. 6, no. 5, pp. 23–34, 2013.

[36] "UCI Machine Learning Repository: CNAE-9 Data Set." [Online]. Available: https://archive.ics.uci.edu/ml/datasets/CNAE-9. [Accessed: 29-Nov-2017].

[37] P. M. Ciarelli and E. Oliveira, "Agglomeration and Elimination of Terms for Dimensionality Reduction," in *2009 Ninth International Conference on Intelligent Systems Design and Applications*, 2009, pp. 547–552.

[38] Athanasios Tsanas, "Athanasios Tsanas Personal Web." [Online]. Available: https://people.maths.ox.ac.uk/tsanas/data.html. [Accessed: 29-Nov-2017].

[39] A. Tsanas, M. A. Little, C. Fox, and L. O. Ramig, "Objective Automatic Assessment of Rehabilitative Speech Treatment in Parkinson's Disease," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 22, no. 1, pp. 181–190, Jan. 2014.

[40] C. Goutte and E. Gaussier, "A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation," Springer, Berlin, Heidelberg, 2005, pp. 345–359.

[41] S. E. Gómez, "FCBF_module," 2016. [Online]. Available: https://github.com/SantiagoEG/FCBF_module.

[42] S. E. Gómez, B. C. Martínez, A. J. Sánchez-Esguevillas, and L. Hernández Callejo, "Ensemble network traffic classification: Algorithm comparison and novel ensemble scheme proposal," *Comput. Networks*, vol. 127, pp. 68–80, Nov. 2017.