AUT Jouranl of Modeling and Simulation

AUT J. Model. Simul., 52(2) (2020) 229-250
DOI: 10.22060/miscj.2020.18038.5202

Customer Churn Prediction in Telecommunication Using Machine Learning: A Comparison Study

Maryam Imani

Faculty of Electrical and Computer Engineering, Tarbiat Modares University, Tehran, Iran

ABSTRACT: Telecommunication operators need to accurately predict the customer churn for surviving in the Telecom market. There is a huge volume of customer records such as calls, SMSs and the use of Internet. This data contains rich and valuable information about costumer behavior and his/her pattern consumption. Machine learning is a powerful tool for extraction of costumer information that can be useful for churn prediction. Although several researchers have studies some types of machine learning methods, but, there is not any work which assess different methods from various point of views. The aim of this work is to assess the performance of a wide range of machine learning methods for churn prediction in the form of a comparison study. In this paper, various machine learning methods consisting of 7 classifiers, 7 target detectors, 10 feature reduction methods containing 4 feature extraction algorithms and 6 feature selection ones are discussed. The performance of these methods are experimented on three Telecom datasets with 6 evaluation measures. The results show that the random forest and feed-forward neural network beside the genetic algorithm outperform other competitors. The superior methods achieve 97%, 62% and 93% prediction accuracy in BigML, kaggle and Telco customer churn datasets, respectively.

Review History:
Received: Mar. 03, 2020
Revised: Jun. 05, 2020
Accepted: Jun. 06, 2020
Available Online: Dec. 01, 2020

Keywords:
machine learning
Telecom
customer churn
classification
feature extraction.

1- INTRODUCTION

Among various communications and telecommunication industries, mobile phone services are one of the most competitive and fastest growing industries. Due to high costs of gathering and analyzing of new customers' behavior and preferences, retaining a customer is more preferable than acquiring a new customer. So, churn prediction, which means to predict a customer whether terminates the contract or not, is a necessity in mobile operator companies [1].

1.1.Problem statement

Churn prediction is usually done by investigating the customer characteristics described by behavioral variables. According to model introduced in [2], certain factors such as customer dissatisfaction, level of service usage and costs can effect on the customer churn. Among various factors, the most important one is customer dissatisfaction that is measured by evaluation of customer behaviors about service characteristics, for example call quality, added value services, traffic level and also the income level of the customer and subscription duration [3]. The factors of service usage such as minutes of monthly use, monthly charge and number of calls can be good signs for identification of churn. Switching costs related to phone number change can cause dispensing with churn. Affective factors in customer churn are assessed in [4]. For identification of customer churn behaviors, a Bayesian belief network based model is constructed in [5].

Churn prediction can be expressed as a two-class classification problem where we want to identify a customer is churn or not. The performance of any classifier is highly dependent on the relevant, useful and informative features. The mobile operators usually collect and save lots of customer consumption behaviors in their databases. Many features such as total day minute, account length and number of SMSs are available. But, all of the existing features may not have useful information about customer churn. Some features contain redundant information or have overlap with other features. So, selection or extraction of the most relevant features can improve the classification accuracy. Moreover, in the case of small data, the high dimensional data leads to curse of dimensionality that reduces the classification accuracy. So, feature reduction is a preprocessing step that may be helpful for data classification. Feature reduction methods are generally divided into two main groups: feature selection and feature extraction. As an instance, the profit-driven feature selection is used beside support vector machine (SVM) for customer churn classification in [6].

Customer churn decision rules are extracted by using an intelligent rule-based decision maker based on rough set theory in [7]. Some rule generation techniques such as covering algorithm, genetic algorithm and exhaustive algorithm have been investigated. The results show that the rough set theory beside the genetic algorithm achieves the

*Corresponding author's email:maryam.imani@modares.ac.ir

best customer churn prediction results.

Logistic regression (LR) and decision tree (DT) are also two popular classification methods in customer churn prediction. Handling interaction effects among variables and handling linear relations among variables are difficulties of LR and DT, respectively. To deal with these difficulties, a hybrid classification method has been introduced in [8]. The basic idea behind the proposed hybrid method is that the construction of different models on the segments of data provides better prediction results than the model construction on entire data.

Classification approaches such as SVM, K-nearest neighbor (KNN), DT and artificial neural network (ANN) have been studied and experimented for customer churn prediction in [9]. The results show that ANN significantly works better than three other methods. ANN is combined with self-organizing map (SOM) for churn prediction in [10]. The results show that the hybrid neural network outperforms the single ANN model. Five classification methods (LR, DT, SVM, ANN and Bayes classifier) have been compared together in [11]. ANN and DT provides the best prediction results.

The performance of ensemble classification methods such as rotation forest and RotBoost have been assessed by using three feature extraction methods in [12]: principal component analysis, independent component analysis (ICA) and sparse random projection. Among different variations, rotation forest classifier beside ICA as feature extractor outperforms other algorithms. The random forest and KNN are used for customer churn prediction in [13] while the particle swarm optimization method is used to deal with imbalance distribution of data.

As said before, churn prediction is a two-class classification problem where it has to identify that a customer decides to churn or not. So, different classification methods can be used for customer churn classification. Generally, due to the small number of churners with respect to non-churners, the customer churn data is imbalance. In other words, the majority of data samples belong to the non-churn class. So, non-churn samples can be seen as background of data while churn samples can be known as targets to be detected. According to this view, customer churn prediction can be modeled as a target detection problem where the aim is to detect churns. However, target detection is itself a two-class classification problem that its aim is to discriminate between target (churn) and non-target (non-churn).

The input of each classification or target detection algorithm is data samples where each sample (each customer here) is characterized with a feature vector. If the feature vector has a high dimensionality, it may lead to curse of dimensionality problem. In addition, presence of non-relevant or non-useful features in the feature vector is possible. So, feature reduction can be used as a preprocessing step before any classification or target detection algorithm. Feature reduction is done in two general ways: feature selection and feature extraction. While in feature selection, some features are selected and the remainder of them are discarded, in feature extraction, a transformation is applied to the original feature vector to extract useful features from the entire original ones.

1.2.Research objectives

Although several researches have stated one or some machine learning methods such as specific classifiers or the use of them beside feature selection approaches, there is not any comprehensive assessment about efficiency of machine learning methods in churn prediction problem. Churn prediction can be automatically done by training a classifier or a target detector. The classifiers and detectors require an appropriate feature vector for discrimination between faithful customers and unfaithful ones. The appropriate feature vector can be selected or extracted from the original data records. This is the task of feature selection or feature extraction methods. According to above statements, we assess performance of various machine learning (ML) methods for customer churn prediction in this work. The assessed ML methods belong to the following four groups:

1-Classification methods
·Decision tree (DT)
·Logistic regression (LR)
·Random forest (RF)
·Feed forward neural network (FFNN)
·Long short term memory (LSTM)
·Support vector machine (SVM)
·Nearest neighbor (NN)

2-Target detection methods
·Matched subspace detector (MSD)
·Adaptive subspace detector (ASD)
·Orthogonal subspace projection (OSP)
·Spectral angle mapper (SAM)
·Kernel SAM (KSAM)
·Constrained energy minimization (CEM)
·Sparsity based target detector (STD)

3-Feature extraction methods
·Principal component analysis (PCA)
·Linear discriminant analysis (LDA)
·Clustering based feature extraction (CBFE)
·Median-mean and feature line embedding (MMFLE)

4-Feature selection methods
·Advanced binary ant colony optimization (ABACO)
·Relief-F
·Feature selection with adaptive structure learning (FSASL)
·Least absolute shrinkage and selection operator (LASSO)
·Genetic algorithm (GA)
·Sequential backward selection (SBS)

All of the above methods have been studied, experimented and compared on three real telecom datasets. The performance of them are assessed from the churn prediction accuracy point of view in terms of various measures. After giving a review on various ML methods in section 2, the performance of them are experimented on three telecom datasets for churn

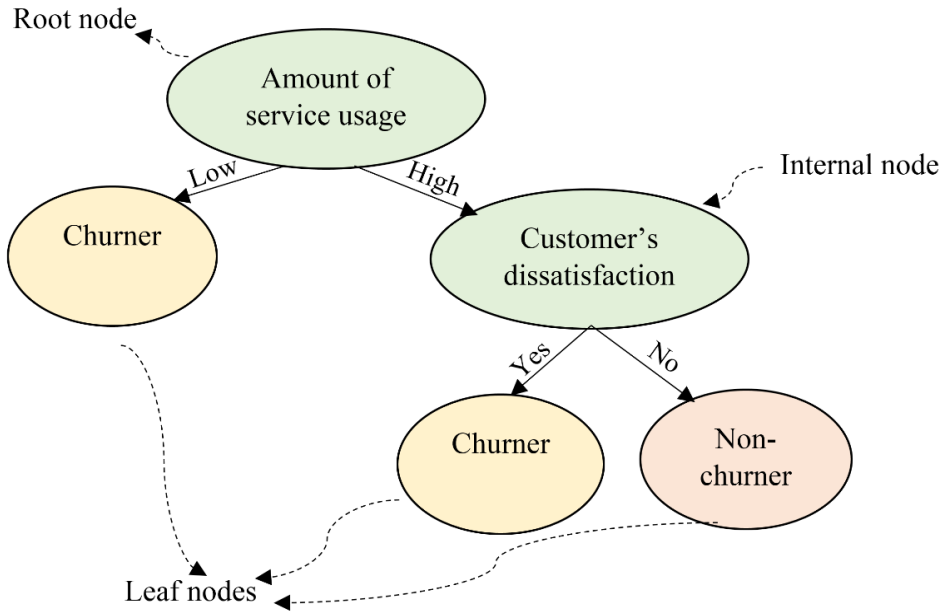

Fig. 1. A simple decision tree for churn classification.

prediction in section 3. Finally, section 4 concludes the work.

2- MACHINE LEARNING METHODS

In this section, various machine learning methods which can be used for customer churn detection are represented. These methods are generally divided into four main groups:

·Classification
·Target detection
·Feature extraction
·Feature selection

In the following some methods from each group are represented:

2.1.Classification

2.1.1.Decision tree (DT)

DT is a simple but a widely used classifier. Let consider a churn detection problem where we would like to discover that a customer is churn or non-churn. A simple approach is to ask a series of queries about the characteristics of a churner. The first general question may be that whether the amount of service usage of the customer is high or low. If it is low, then, it is churn. Otherwise, it may be either a churn or non-churn. In the second case, a follow-up question is asked. For example, is there customer's dissatisfaction or not? If the answer is yes, the customer is churn and otherwise it is known as non-churn. These series of questions and their answers can form a decision tree with a hierarchical structure containing several nodes and

directed edges. Fig. 1 illustrates a simple decision tree for churn classification problem. There are three general nodes in a DT [14]:

-Root node: it has no incoming edge and usually two (or more) outgoing edges.
-Internal (child) node: it has one incoming edge and two (or more) outgoing edges.
-Terminal (leaf) node: it has one incoming edge and zero outgoing edge.

In a DT, a class label is assigned to each leaf or terminal node. The internal nodes separate the records using sample test attributes where different characteristics lead to class discrimination. When a DT is constructed, a test record is straightforward classified with starting from the root node. The test condition is applied to the record and the appropriate branch is followed based on the test's outcome. This may lead the sample test to another internal node for which a new attribute or condition is applied to the sample test or it is terminated with a leaf node.

Many various DTs can be constructed from a given set of features (attributes). Due to grown up size of the search space, making an optimal DT is computationally infeasible. However, the efficient algorithms provide suboptimal but accurate classification results with a reasonable computation time. The sub-optimal algorithms usually use a greedy strategy to grow a DT by generation of sub-optimal decisions about which feature is used for data partitioning. One of the simplest and widely used algorithms is classification and regression tree (CART) [15]. In CART, the test conditions are restricted to only binary splits.

There are different measures to select the best split. Most of them are based on impurity degree of the child nodes. Entropy, Ginin index and classification errors are some instances of the impurity measures. The impurity gain should be maximized to find the best way to split node t. In other words, gain is a criterion for determining the goodness of a split, which is defined by [14]:

$$\Delta = I(parent) - \sum_{j=1}^{k} \frac{N(v_j)}{N} I(v_j) \quad (1)$$

where $I(\cdot)$ denotes the given impurity measure of node $(\cdot)$, k is the number of features (attribute values), N is the total number of samples at the root (parent) node and $N(v_j)$ is the number of samples (records) corresponding to the child node v_j. Gini index as a splitting function of CART is defined as [14]:

$$Gini(t) = 1 - \sum_{i=0}^{c-1} \left[p(i/t) \right]^2 \quad (2)$$

where $p(i/t)$ indicates the fraction of samples belonging to class i at node t.

2.1.2. Logistic regression (LR)

Logistic regression (also known as logistic model or logit model) represents the relationship between multiple independent input variables and a categorical dependent output variable [8]. LR tries to estimate the probability of occurrence of an event through fitting data samples to a logistic curve. Generally, two models of logistic regression have been used: binary LR and multinomial LR. Binary LR is usually used for dichotomous dependent variable while multinomial LR is used for non-dichotomous dependent variable. Odds of an event is defined as the ratio of the probability of occurrence of an event (p) to the probability of non-occurrence $(1-p)$ [50]:

$$odds\ of\ \{Event\} = \frac{p}{1-p} \quad (3)$$

Through LR, the relation between an explanatory variable x with the probability of the interested outcome, i.e., p is modeled by $p = \alpha + \beta x$. The extreme values of x result in values of $\alpha + \beta x$ that do not fall into [0,1]. The LR solution to this problem is to use the natural logarithm:

$$logit(y) = ln(odds) = ln\left(\frac{p}{1-p}\right) = \alpha + \beta x \quad (4)$$

where α and β are the parameters of the LR. By taking the antilog of (4) on both sides, an equation for the prediction of p is given by [50]:Z

$$p = P(y = interested\ outcome / x) = \frac{e^{\alpha+\beta x}}{1+e^{\alpha+\beta x}} \quad (5)$$

The logic of the simple LR can be extended to multiple predictors by constructing a complex logistic regression as:

$$logit(y) = ln\left(\frac{p}{1-p}\right) = \alpha + \beta_1 x_1 + \ldots + \beta_k x_k \quad (6)$$

2.1.3.Random forest (RF)

The classification and regression trees have a main disadvantage: they are instable. In other words, with a little variation in data, the tree picture changes a lot [16]. The performance of classification or regression trees can be improved by using the bagging or bootstrap aggregating. To this end, the bootstrap samples are randomly chosen with replacement from the data. While some observations appear more than one in the bootstrap samples, others are not included (out of bag). After generation of bootstrap samples, the classification or regression models are fitted to the bootstrap samples. Their results are combined through voting (for classification) or averaging (for regression).

Although bagging reduces the base learner's variance, but it has limited impact on the bias. So, the bagging algorithms are efficient for strong base learners with little bias and high variance, who are unstable such as trees. Although, bagging reduces overfitting and breaks the bias variance trade-off [17], but, it has a problem. The constructed trees are similar to each other, and so, there are high correlation among the predictors. To address this problem, RF is used. RF creates low correlated trees. The RF algorithm is implemented as follows [17]:

1)A random sample of size N is taken with replacement from the data. This step is the same as what done in bootstrapping on data in the bagging algorithm.
2)A random sample is taken without replacement of the predictors. This step is bootstrapping on predictors/ features (predictors sampling) and it is without replacement. Note that this step is solution of RF to overcome the highly correlated trees arisen from the bagging algorithm.
3)The first CART partition of the data is constructed. The first bootstrap is partitioned. The Gini index is used for the bootstrapped predictor samples to decide the split.
4)Step 2 is repeated for each subsequent split. This operation is continued until the tree becomes as large as described. No pruning is done.
5)The steps 1-4 are building one tree. Steps 1-4 are repeated a large number of times to build a forest.

2.1.4.Nearest neighbor (NN)

In the NN classifier, a testing sample is assigned to the most common class among the samples similar to it. The NN algorithm is implemented as follows:

1) The distance between the test sample and each of training samples is computed.
2) The distances are sorted.
3) The nearest neighbor is taken. The class label of the nearest neighbor is assigned to the test sample.

Various distances can be used to find the nearest neighbor. The most common distance metric is Euclidean distance that measures the magnitude of two vectors $\boldsymbol{x}=\left(x_1,x_2,\ldots,x_n\right)$ and $\boldsymbol{y}=\left(y_1,y_2,\ldots,y_n\right)$ as follows [51]:

$$d_E\left(\boldsymbol{x},\boldsymbol{y}\right)=\sqrt{\left(x_1-y_1\right)^2+\left(x_2-y_2\right)^2+\ldots+\left(x_n-y_n\right)^2} \quad (7)$$

Some other distances are represented in the following.
Taxican distance [18]:

$$d_1\left(\boldsymbol{x},\boldsymbol{y}\right)=\left\|x-y\right\|=\sum_{i=1}^{n}\left|x_i-y_i\right| \quad (8)$$

Cosine distance [51]:

$$\cos\left(\theta\right)=\frac{\boldsymbol{x}.\boldsymbol{y}}{\left\|x\right\|\left\|y\right\|}=\frac{\sum_{i=1}^{n}x_i.y_i}{\sqrt{\sum_{i=1}^{n}x_i^{\,2}}\sqrt{\sum_{i=1}^{n}y_i^{\,2}}} \quad (9)$$

The cosine distance is a similarity metric ranges from -1 to 1 where -1 means exactly opposite, 1 means exactly the same and 0 means independence. Other in-between metric values indicate intermediate similarity or dissimilarity. In some cases such as information retrieval cases, the cosine similarity of two samples ranges from 0 to 1.

Correlation [51]:

$$\rho=\frac{\sigma_{xy}}{\sigma_x\,\sigma_y} \quad (10)$$

where σ_{xy} is the covariance between two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$; and σ_x and σ_y are the standard deviation of vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively. Although the NN algorithm is a simple and efficient classifier, but, it is inefficient for high dimensional or large scale data. NN is a lazy learning algorithm that means it has not a true learning phase. The result is a high computation time in the testing phase.

2.1.5.Support vector machine (SVM)

The basic SVM classifier is introduced for solving a two-class classification problem that can be extended to a c-class classification problem. Let, a two-class learning problem with n training samples $\left\{\left(\boldsymbol{x}_i,y_i\right)\right\}_{i=1}^{n}$ where $\boldsymbol{x}_i$ is i th input feature vector and y_i is its label, $y_i\in\left\{-1,+1\right\};i=1,\ldots,n$. The aim is to separate two classes with a linear discrimination function [52]:

$$f\left(\boldsymbol{x}\right)=\boldsymbol{w}^T\boldsymbol{x}+b \quad (11)$$

where $\boldsymbol{w}$ is the weight vector. SVM is a maximum margin classifier. In other words, it tries to maximize the geometric margin between two classes, i.e., $\frac{1}{\left\|w\right\|}$, that is equivalent to minimize $\left\|w\right\|^2$ as follows [52]:

$$\min_{\boldsymbol{w},b}\frac{1}{2}.\left\|w\right\|^2$$
$$\text{subject to}\ y_i\left(\boldsymbol{w}^T\boldsymbol{x}_i+b\right)\geq 1,\ i=1,\ldots,n \quad (12)$$

The above formulation is appropriate for linearly separable data. For non-linear cases, a greater margin should be achieved by allowing the classifier to misclassify some samples. To this end, the objective function in (12) is changed to [19]:

$$\min\frac{1}{2}\left\|w\right\|^2+C\sum_{i=1}^{n}\xi_i$$
$$\text{subject to}\ y_i\left(\boldsymbol{w}^T\boldsymbol{x}_i+b\right)\geq 1-\xi_i;\ \ i=1,\ldots,n,\quad \xi_i>0 \quad (13)$$

where $\xi_i>0;i=1,\ldots,n$ denote the slack variables. $0\leq\xi_i\leq 1$ when a sample is in the margin or $\xi_i>1$ when the sample is misclassified. It can be said that $\sum_{i=1}^{n}\xi_i$ is a bound on the number of misclassified samples. To penalize margin errors and misclassification, the term $C\sum_{i=1}^{n}\xi_i$ is minimized where the constant $C>0$ controls the relative importance of minimizing the slack values and maximizing the margin. The above formulation is known as soft margin. In addition to using the slack variables, the SVM classifier uses kernels for dealing with nonlinear separable classes. Suppose the weight vector can be represented as a linear combination of training samples, i.e., $\boldsymbol{w}=\sum_{i=1}^{n}\alpha_i\boldsymbol{x}_i$. So, the discriminant function will be [52]:

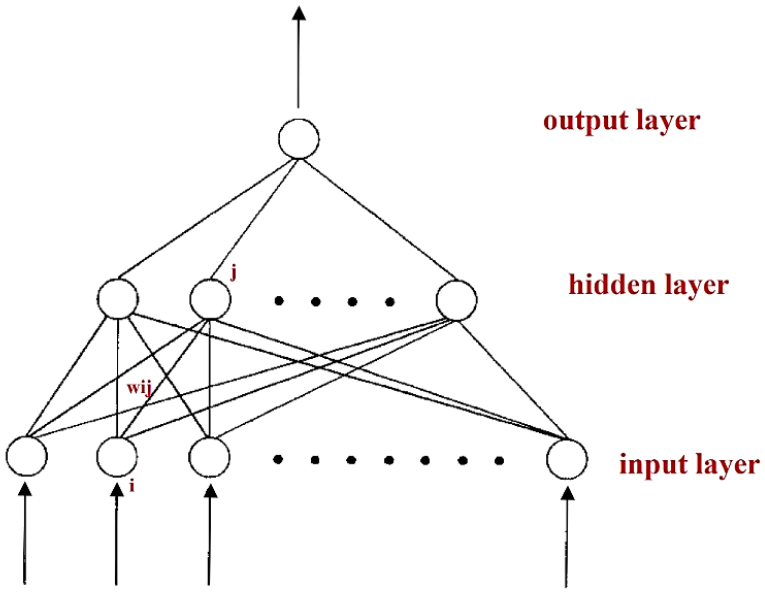

Fig. 2. A typical FFNN composed of three hidden layers and one output.

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i \boldsymbol{x}_i^T \boldsymbol{x} + b \tag{14}$$

The above linear classifier can be changed to a non-linear one by mapping data from the input space χ to a feature space $\mathcal{H}$ through applying a non-linera mapping $\varphi : \chi \to \mathcal{H}$. The result will be:

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i \varphi(\boldsymbol{x}_i)^T \varphi(\boldsymbol{x}) + b \tag{15}$$

By defining a kernel function as $K(\boldsymbol{x}, \boldsymbol{x}') = \varphi(\boldsymbol{x})^T \varphi(\boldsymbol{x}')$, the function (15) can be computed without explicitly computing the mapping function φ. That is known as kernel trick. Polynomial, *tanh* and radial basis function (RBF) are among the widely used kernel functions. Among them, RBF shows more efficiency in various applications that it is defined by:

$$K(x, x') = \exp(-\gamma \|x - x'\|^2) \tag{16}$$

where $\gamma > 0$ is a parameter for controlling the width of the Gaussian function.

2.1.6.Feed-forward neural network (FFNN)

The most popular neural network is multi-layer FFNN consisting of neurons ordered into layers. The first layer is called input layer, the last layer is called output layer and the middle layers are known as hidden layers where their task is usually feature extraction. The operation of a neuron is formally described using a mapping function Γ [20]. For neuron i, $\Gamma(i) \subseteq V$ and $\Gamma^{-1}(i) \subseteq V$ are subsets consisting of all ancestors and predecessors of the given neuron i. Each neuron is fully connected to all neurons of the next layer and the connection between each pair of neurons i and j is characterized by a weigh coefficient w_{ij}. Fig. 2 shows a typical FFNN composed of three hidden layers and one output. The activity of i th neuron denoted as x_i is computed by [20]:

$$x_i = f(\xi_i) \tag{17}$$

where $\xi_i = b_i + \sum_{j \in \Gamma_i^{-1}} w_{ij} x_j$ is the potential of neuron i, b_i is the bias or threshold coefficient of neuron i and $f(\xi_i)$ is the transfer function that can be defined in various manners such as the sigmoid function:

$$f(\xi) = \frac{1}{1 + \exp(-\xi)} \tag{18}$$

The weight coefficient w_{ij} and the threshold coefficient

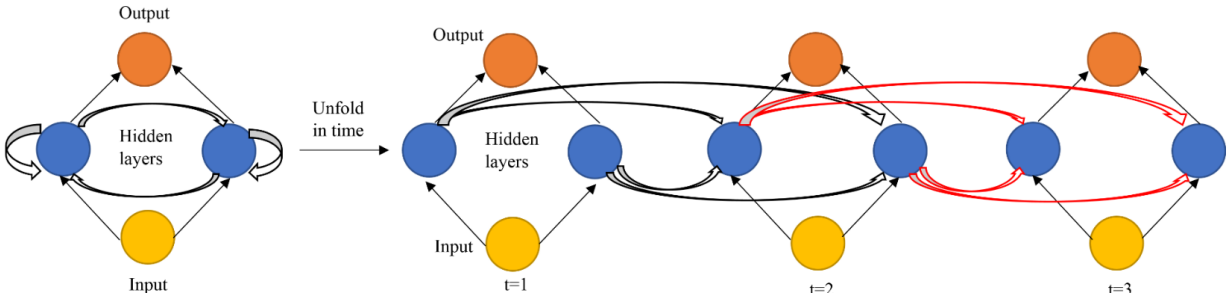

Fig. 3. RNN structure.

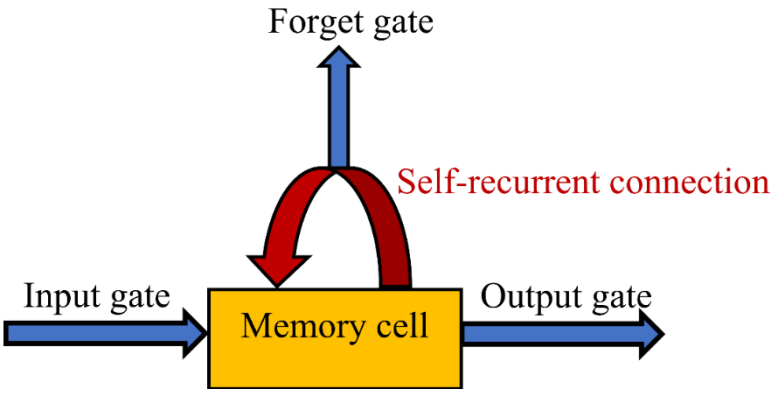

Fig. 4. LSTM structure.

b_i are varied by the supervised adaptive process. In the network training process, the back propagation algorithm is used to minimize the square error between the target output values and the network output ones. By updating the weights through the gradient descent, the partial deviation of the error function with respect to weights is taken to drive the update's rules. Then, the gradient descent is used to adjust weights. The process is iteratively occurred for each layer starting from the last layer working back towards the first (input) layer.

2.1.7. Long short term memory (LSTM)

The recurrent neural network (RNN) is a network with feedback loops [21]. In a RNN, the output at the current time not only depends on the current input but also depends to the previous time steps through the recurrent edges (see Fig. 3) [22]. RNN is trained using the back propagation through time. In other words, the regular back propagation algorithm can be applied to the RNN unfolded in time. But, due to vanishing gradients during back propagation, RNN cannot capture long term dependencies. LSTM as a type of RNN architecture solves this problem. To this end, a LSTM memory cell is designed for maintaining the network state over time. A memory cell consists of an explicit memory and three main gates. The information flow into and out of the memory is regulated by the gates (see Fig. 4) [22].

There are three types of gates: forget gate, input gate and output gate. Some information throw away the memory by controlling the forget gate. New information from the current input is added to the cell state through the input gate while the output gate decides what to go out from the memory.

2.2.Target detection

Target detection is a supervised two-class classification problem where its aim is to separate the targets with known characteristics from the non-target called as background. Usually, in a target detection problem, the number of target samples is much less than the number of background samples. In other words, target detection is usually an imbalanced classification problem. In the following, some popular and widely used target detection methods are represented [23].

2.4.1. Matched subspace detector (MSD)

In the MSD, the test sample is modeled in terms of background subspace and target subspace obtained by the background training samples and target training samples, respectively. The competing hypothesis is represented by [24]:

$$H_0 : \boldsymbol{x} = \boldsymbol{Bb} + \boldsymbol{n} \quad (\text{Target absent})$$
$$H_1 : \boldsymbol{x} = \boldsymbol{St} + \boldsymbol{Bb} + \boldsymbol{n} \quad (\text{Target present}) \tag{19}$$

where $\boldsymbol{B}$ and $\boldsymbol{S}$ matrices denote the background and target subspaces, respectively which their columns are linearly independent. Feature variability of background and target are take into account with the subspace models. The corresponding abundances of matrices $\boldsymbol{B}$ and $\boldsymbol{S}$ are denoted by $\boldsymbol{b}$ and $\boldsymbol{t}$, respectively. $\boldsymbol{n} \sim N\left(0, \sigma_n^2 \boldsymbol{I}\right)$ is the additive white Gaussian noise where σ_n^2 denotes the noise variance. For an input vector $\boldsymbol{x}$, the output of MSD is computed by using the generalized likelihood ratio test (GLRT) as follows:

$$D_{MSD}(\boldsymbol{x}) = \frac{\boldsymbol{x}^T \boldsymbol{P}_{\boldsymbol{B}}^{\perp} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{P}_{\boldsymbol{SB}}^{\perp} \boldsymbol{x}} \tag{20}$$

where $\boldsymbol{P}_{\boldsymbol{B}}^{\perp} = \boldsymbol{I} - \boldsymbol{P}_{\boldsymbol{B}}$ and $\boldsymbol{P}_{\boldsymbol{B}} = \boldsymbol{B}\left(\boldsymbol{B}^T \boldsymbol{B}\right)^{-1} \boldsymbol{B}^T = \boldsymbol{B}\boldsymbol{B}^{\neq}$ is a projection matrix associated with subspace $\boldsymbol{B}$ and $\boldsymbol{SB} = \left[\boldsymbol{S}\ \boldsymbol{B}\right]$ denotes the matrix obtained by combining target $\left(\boldsymbol{S}\right)$ and background $\left(\boldsymbol{B}\right)$ subspaces. The subspaces $\boldsymbol{S}$ and $\boldsymbol{B}$ are generally generated by using the eigenvectors corresponding to the largest eigenvalues of the associated covariance matrices.

2.2.2.Orthogonal subspace projection (OSP)

OSP maximizes the signal to noise ratio (SNR) in the subspace orthogonal to the background subspace. To remove the background (undesired) signature, the background rejection operator is given by$\boldsymbol{P}_{\boldsymbol{B}}^{\perp} = \boldsymbol{I} -$. The operator $\boldsymbol{w}$ is found such a way that maximizes the SNR of the filter output $\boldsymbol{w}\boldsymbol{P}_{\boldsymbol{B}}^{\perp}\boldsymbol{x}$ [25]:

$$SNR(\boldsymbol{w}) = \frac{\left[\boldsymbol{w}^T \boldsymbol{P}_{\boldsymbol{B}}^{\perp} \boldsymbol{d}\right] \alpha_l^2 \left[\boldsymbol{d}^T \boldsymbol{P}_{\boldsymbol{B}}^{\perp} \boldsymbol{w}\right]}{\boldsymbol{w}^T \boldsymbol{P}_{\boldsymbol{B}}^{\perp}\, \mathrm{E}[\boldsymbol{n}\boldsymbol{n}^T] \boldsymbol{P}_{\boldsymbol{B}}^{\perp} \boldsymbol{w}} \tag{21}$$

where $\boldsymbol{d}$ and α_l are the target signature and its abundance, respectively. The operator $\boldsymbol{w}$ is given by the matched filter $\boldsymbol{w} = k\boldsymbol{d}$ where k is a constant and the output of OSP is given by:

$$D_{OSP}(\boldsymbol{x}) = \boldsymbol{d}^T \boldsymbol{P}_{\boldsymbol{B}}^{\perp} \boldsymbol{x} \tag{22}$$

2.2.3.Adaptive subspace detector (ASD)

The competing hypothesis for ASD is expressed as [26]:

$$\begin{aligned} &H_0 : \boldsymbol{x} = \boldsymbol{n} \quad (\text{Target absent}) \\ &H_1 : \boldsymbol{x} = \boldsymbol{S}\boldsymbol{t} + \sigma\boldsymbol{n} \quad (\text{Target present}) \end{aligned} \tag{23}$$

As before $\boldsymbol{S}$ and $\boldsymbol{t}$ are the target subspace and the corresponding abundance, respectively. $\boldsymbol{n} \sim N\left(0, \boldsymbol{C}\right)$ is the Gaussian random noise. Under H_1, $\boldsymbol{x}$ is distributed as $N\left(\boldsymbol{S}\boldsymbol{t}, \sigma^2 \boldsymbol{C}\right)$. The solution given by CLRT is:

$$D_{ASD}\left(\boldsymbol{x}\right) = \frac{\boldsymbol{x}^T \boldsymbol{C}^{-1} \boldsymbol{S}\left(\boldsymbol{S}^T \boldsymbol{C}^{-1} \boldsymbol{S}\right)^{-1} \boldsymbol{S}^T \boldsymbol{C}^{-1} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{C}^{-1} \boldsymbol{x}} \tag{24}$$

and

$$\begin{aligned} &\text{if} \quad D_{ASD}(\boldsymbol{x}) > \eta \quad \text{then} \quad H_1 \\ &\text{if} \quad D_{ASD}(\boldsymbol{x}) < \eta \quad \text{then} \quad H_0 \end{aligned} \tag{25}$$

where η indicates a threshold for determining the targets. ASD is also known as adaptive cosine estimator (ACE) because the above formulation measures the angle between $\boldsymbol{C}^{-\frac{1}{2}}\boldsymbol{x}$ and $\boldsymbol{C}^{-\frac{1}{2}}\boldsymbol{S}$.

2.2.4.Constrained energy minimization (CEM)

CEM builds a linear filter to minimize the total attribute output energy under the constraint that the output of target is a constant [27]. The aim of CEM is to highlight the target's output while suppresses the energy of background for finding a projection vector that well separates target from the undesired background. The average energy of output can be computed by averaging the square of the output. So, the optimization problem of CEM is [27]:

$$\begin{aligned} &\min \mathrm{E}\left(\boldsymbol{x}^2\right) = \boldsymbol{w}^T \boldsymbol{R}\boldsymbol{w} \\ &\text{subjec to } \boldsymbol{w}^T \boldsymbol{d} = 1 \end{aligned} \tag{26}$$

where $\boldsymbol{R}$ is the correlation matrix of data, i.e., $\mathbf{R} = \mathrm{E}\left(\boldsymbol{x}\boldsymbol{x}^T\right)$ and $\boldsymbol{d}$ is the target signature. The result will be

$$\boldsymbol{w} = \frac{\boldsymbol{R}^{-1}\boldsymbol{d}}{\boldsymbol{d}^T \boldsymbol{R}^{-1} \boldsymbol{d}} \tag{27}$$

and the detector output is given by:

$$D_{CEM}\left(\boldsymbol{x}\right) = \boldsymbol{w}^T \boldsymbol{x} \tag{28}$$

2.2.5.Spectral angle mapper (SAM)

SAM is a simple and geometrical interpretable target detector. SAM measures the angle between two vectors

(testing sample $\boldsymbol{x}$ and target signature $\boldsymbol{d}$) [28]:

$$\theta = arc\cos\left(\frac{\boldsymbol{x}\boldsymbol{d}}{\|x\|\|d\|}\right); 0 \le \theta \le \frac{\pi}{2} \tag{29}$$

where $\boldsymbol{x}.\boldsymbol{d}$ is the inner product of $\boldsymbol{x}$ and $\boldsymbol{d}$.

2.2.6. Kernel SAM (KSAM)

The main limitation of SAM is that it only takes into account the second order angle dependencies between the feature vectors. To consider the nonlinear cases, SAM is generalized by means of kernels where it is denoted as KSAM [29]:

$$\theta = arc\cos\left(\frac{K(\boldsymbol{x},\boldsymbol{d})}{\sqrt{K(\boldsymbol{x},\boldsymbol{x})}\sqrt{K(\boldsymbol{d},\boldsymbol{d})}}\right); 0 \le \theta \le \frac{\pi}{2} \tag{30}$$

$K(\boldsymbol{x},\boldsymbol{d})$ is a kernel function such as RBF kernel $K(\boldsymbol{x},\boldsymbol{d}) = \exp\left(-\gamma \boldsymbol{x} - \boldsymbol{d}^2\right)$.

2.2.7. Sparsity based target detector (STD)

STD sparsely represents a test sample by a few training samples including both background and target samples. For implementation of target detection, the reconstruction residuals are directly employed. Assume that the test sample $\boldsymbol{x}$ is sparsely modeled by a union of background and target subspaces as follows [30]:

$$\boldsymbol{x} \approx \boldsymbol{A}^b\boldsymbol{\alpha}^b + \boldsymbol{A}^t\boldsymbol{\alpha}^t = \boldsymbol{A}\boldsymbol{\alpha} \tag{31}$$

where $\boldsymbol{A}^b$ and $\boldsymbol{A}^t$ are the background and target dictionaries consisting of the background and target samples, respectively. $\boldsymbol{\alpha}^b$ and $\boldsymbol{\alpha}^t$ are their corresponding sparse vectors which the concatenation of them, i.e., $\boldsymbol{\alpha}$ is achieved by solving the following optimization problem:

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{A}\boldsymbol{\alpha} - \boldsymbol{x}\|_2^2 \ \ subject\ to\ \|\boldsymbol{\alpha}\|_0 \le L \tag{32}$$

where where $\|\cdot\|_0$ is the l_0 -norm defined as the number of non-zero entries called sparsity level and L indicates a given upper bound on the l_0 -norm. The orthogonal matching pursuit (OMP) algorithm can be used to solve the above optimization problem [31]. The STD output will be:

$$D_{STD}(\boldsymbol{x}) = r_b(\boldsymbol{x}) - r_t(\boldsymbol{x}) \tag{33}$$

where

$$r_b(\boldsymbol{x}) = \|\boldsymbol{x} - \boldsymbol{A}^b\,\hat{\boldsymbol{\alpha}}^b\|_2 \tag{34}$$

$$r_t(\boldsymbol{x}) = \|\boldsymbol{x} - \boldsymbol{A}^t\,\hat{\boldsymbol{\alpha}}^t\|_2 \tag{35}$$

STD has no explicit assumption on statistical data distribution.

2.3.Feature extraction

Feature reduction is necessary in many pattern recognition applications. Its advantages is increasing the classification accuracy, decreasing the computation time, allowing data visualization and better understating of data. Feature reduction can be done in two general ways: feature extraction and feature selection. In feature selection, just a few number of features are selected by using a search algorithm and a selection criterion and the reminder of them are discarded. In feature extraction, usually a transformation is applied to the data to project it to a new feature space [32]. Then, some features from the new feature space are taken. Each of the feature extraction and feature selection methods have their advantages and disadvantages and can be useful dependent on data and application. In feature selection, the physical and real meaning of features are preserved while in feature extraction the real meaning of data is removed. However, the feature selection approach may loss parts of information due to discarding some features while the feature extraction approach uses an extraction of data which obtained from all available features. In the following subsections, some feature extraction and feature selection methods are represented.

2.3.1.Principal component analysis (PCA)

The basic idea of PCA is dimensionality reduction through retaining the variance (energy) of data as much as possible [33]. This is achieved by projection to a new set of uncorrelated variables called principal components (PCs). The PCs are ordered so that the first ones contain the most of variation present in all of the original variables. Let data as a $f \times n$ matrix $\boldsymbol{X}$ with n samples and f features. Data becomes zero-mean by subtracting mean from the data samples. Then, the covariance matrix is estimated by [53]:

$$\boldsymbol{C}_x = \frac{1}{n-1}\boldsymbol{X}\boldsymbol{X}^T \tag{36}$$

where $\boldsymbol{C}_x$ quantifies the correlation between each pair of measurements in $\boldsymbol{X}$. The goal is redundancy reduction. To this end, the data $\boldsymbol{X}$ should be transformed to $\boldsymbol{Y}$ such that $\boldsymbol{C}_y$ be a diagonal matrix. The PCA transform does this task. The PCs of $\boldsymbol{X}$ are the eigenvectors of $\boldsymbol{X}\boldsymbol{X}^T$. PCA is an unsupervised feature extraction method. In the following, some supervised feature extraction methods are represented.

2.3.2.Linear discriminant analysis (LDA)

PCA searches for directions having the largest variance while it does not consider the sample labels for class separation. To solve this problem, LDA has been introduced which maximizes the Fisher objective function as follows [53]:

$$\max_{\boldsymbol{w}} J(\boldsymbol{w}) = \frac{\boldsymbol{w}^T \boldsymbol{S}_B \boldsymbol{w}}{\boldsymbol{w}^T \boldsymbol{S}_w \boldsymbol{w}} \tag{37}$$

where $\boldsymbol{S}_B$ and $\boldsymbol{S}_w$ are the between-class and within-class scatter matrices, respectively defined by [34]:

$$\boldsymbol{S}_B = \sum_{k=1}^{c} (\boldsymbol{\mu}_k - \boldsymbol{\mu})(\boldsymbol{\mu}_k - \boldsymbol{\mu})^T \tag{38}$$

$$\boldsymbol{S}_w = \sum_{k=1}^{c} \sum_{i=1}^{n_k} (\boldsymbol{x}_{ik} - \boldsymbol{\mu}_k)(\boldsymbol{x}_{ik} - \boldsymbol{\mu}_k)^T \tag{39}$$

where $\boldsymbol{x}_{ik}$ is i th sample of k th class, c is the number of classes, $\boldsymbol{\mu}_k$ is the mean of class k and $\boldsymbol{\mu}$ denotes the overall mean of data samples. The columns of LDA projection matrix, i.e., $\boldsymbol{w}$ are obtained by solving (37). LDA has two main disadvantages: it can extract maximum $c-1$ features and it fails to work if the number of training samples be limited.

2.5.3. Clustering based feature extraction (CBFE)

CBFE is a supervised feature extraction method [35]. In contrast to methods such as PCA and LDA that use the covariance matrices, the CBFE method only uses the first order statistics, i.e., mean vector. So, it has superior performance when sufficient training samples are not available. In addition, CBFE can extract any arbitrary number of features. CBFE considers a vector containing the mean values of training samples in individual classes corresponding to each variable (feature). Then, a clustering approach such as k-means algorithm is used to cluster the obtained vectors. The mean of features whose associated vectors located in a cluster is considered as an extracted feature. In other words, the number of clusters determines the number of extracted features. Let the mean matrix of training samples as follows [35]:

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{a}_1 \\ \boldsymbol{a}_2 \\ \vdots \\ \boldsymbol{a}_f \end{bmatrix} \tag{40}$$

where

$$\boldsymbol{a}_i = \begin{bmatrix} m_{i1} & m_{i2} \dots & m_{ic} \end{bmatrix}; i = 1, \dots, f \tag{41}$$

is the vector assigned to feature i, f is the number of features and c is the number of classes. f vectors are clustered. For an instance, with $f = 10$ dimensional data and $k = 3$, the k-means clustering is done and three following clusters are achieved:

$$clust_1 = \begin{bmatrix} \boldsymbol{a}_1 \\ \boldsymbol{a}_3 \\ \boldsymbol{a}_5 \\ \boldsymbol{a}_7 \end{bmatrix}, clust_2 = \begin{bmatrix} \boldsymbol{a}_2 \\ \boldsymbol{a}_8 \\ \boldsymbol{a}_9 \\ \boldsymbol{a}_{10} \end{bmatrix}, clust_3 = \begin{bmatrix} \boldsymbol{a}_4 \\ \boldsymbol{a}_6 \\ \boldsymbol{a}_7 \end{bmatrix} \tag{42}$$

$clust_1$ means that features 1, 3, 5 and 7 are similar because their associated vectors belong to a cluster. So, mean of features 1, 3, 5 and 7 is considered as an extracted feature. The similar process is done for other clusters.

2.3.4. Median-mean and feature line embedding (MMFLE)

Similar to LDA, MMFLE tries to maximize class separability using discriminant analysis [36]. But, MMFLE uses two main alternatives to solve some difficulties of LDA. To deal with the negative effect of outliers in calculation of mean in LDA, the median-mean line (MML) measurement is substituted in MMFLE. In addition, the feature line (FL) distance metric is used to enlarge the training set. The MMFLE uses the following scatter matrices [36]:

$$\mathbf{S}_w = \boldsymbol{S}_w^{MML} + \lambda \boldsymbol{S}_w^{FL} \tag{43}$$

$$\boldsymbol{S}_b = \boldsymbol{S}_b^{MML} + \lambda \boldsymbol{S}_b^{FL} \tag{44}$$

where λ is a small free parameter that controls the impact of MML and FL metrics, $\boldsymbol{S}_w^{MML}$ and $\boldsymbol{S}_b^{MML}$ are the within-class and between-class scatter matrices computed by using MML metric and $\boldsymbol{S}_w^{FL}$ and $\boldsymbol{S}_b^{FL}$ are the scatter matrices computed by FL metric.

2.4.Feature selection

2.4.1.Advanced binary ant colony optimization (ABACO)

One of meta-heuristic algorithms is ant colony optimization (ACO) that has been inspired by the foraging behavior of ants [37]. Real ants lay some pheromone on the path when they find a food source. The amount of this pheromone depends on quality, quantity and distance of the food source. When another ant encounters the laid trail, can detect it. Then, the ant reinforces the trail by laying some other pheromone on that path. The more the ants follow a trail, the more the trail becomes attractive. ACO can be used

for feature selection where the features are treated as the graph nodes for constructing a graph model. The search of feature subset is done based on the constructed graph. The nodes are fully connected. In the ACO algorithm, the probability decision is made based on 1-artificilal pheromone trail which shows the history of previous good moves and 2-the local heuristic information which expresses goodness of visibility of the edge. The probabilistic transition rule combines these two factors as follows [38]:

$$p_{ij}^{k}(t)=\begin{cases}\frac{\tau_{ij}^{\alpha}\eta_{ij}^{\beta}}{\sum_{l}\tau_{il}^{\alpha}\eta_{il}^{\beta}} & ; \quad \text{if } i \text{ and } j \text{ are admissible nodes}\\ 0; \quad o.w.\end{cases} \tag{45}$$

where transition probability from feature i and j for ant k in iteration t is indicated by $p_{ij}^{k}(t)$. The value of pheromone trail and the heuristic visibility of edge $i-j$ are indicated by τ_{ij} and ηij, respectively. The free parameters α and β are used for controlling the trade-off between the pheromone value and heuristic information. In the binary ant colony optimization (BACO), each solution is obtained as a vector of binary bits where the ants have to decide a bit is 0 or 1. The advanced BACO (ABACO) is a combination of BACO and ACO. It does not require to know the number of selected features (difficulty of ACO). In addition, the ants are fully connected and ants can simultaneously observe all the features (contrary to BACO) to decide that select a feature or not.

Different criteria can be used for heuristic information measurement. Among them, ABACO uses F-score and three correlation based approaches. Correlation is the most widely used statistic for describing the relation degree between two variables.

2.4.2.Relief-F

The relief-F algorithm has an easy principal to understand. It works based on this principal that objects with similar attributes would be liked to put in a class. To this end, a sample is randomly selected: R_i. Then, two nearest neighbors of R_i are searched: one sample, called nearest hit H, is from the same class and other one, called nearest miss M, is from the different class. The quality of each feature A, denoted by w(A), is updated depending on the value of that feature for R_i, M and H. If the feature value is different for two samples R_i, and H, it means attribute A separates two samples with the same class that is not desirable. So the value of w(A) is decreased. In contrast, if the value of A be different for two samples of R_i and M with different classes, it means that A is a good feature which discriminates between two different classes. So, w(A) is increased. This process is repeated m times where m is a predefined parameter [39]-[40].

Redundant features are not discriminated by relief-F. In addition, the performance of the relief-F algorithm is decreased if few data is available [41].

2.4.3.Feature selection with adaptive structure learning (FSASL)

We would like to select features that faithfully preserve the data intrinsic structure. The data structure is estimated by data features where the existence of noisy or redundant features result in an inaccurate or unreliable data structure. In other words, from one hand, we need an accurate data structure for identification of informative features and from the other hand, we need informative features to provide an accurate estimate of data structure [42].

FSASL is an unsupervised wrapper algorithm that simultaneously performs both of the structure learning and feature selection. It works based on linear regression. The main difficulty of FSASL is its high computational complexity [41]. To provide an adaptive global structure learning, assume $\boldsymbol{X}=\{x_1,x_2,\ldots,x_n\}\in R^{d\times n}$ be the data matrix containing n samples with d features. For extraction of the global structure of data, the sparse reconstruction coefficeints can be used [42]. Each sample x_i can be represented as a linear combination of other samples in $\boldsymbol{X}$ with $\boldsymbol{s}_i$ as the corresponding sparse weight vector. The sparse weight matrix $\boldsymbol{S}\in R^{n\times n}$ can be achieved by solving the following optimization problem [42]:

$$\min_{\boldsymbol{S}}\sum_{i=1}^{n}\left[\left\|x_i-Xs_i{}^2\right\|+\alpha\left\|\boldsymbol{s}_{i1}\right\|\right]$$
$$\text{subject to } \boldsymbol{S}_{ii}=0 \tag{46}$$

where parameter α balances the reconstruction errors and the sparsity constraint. The features should be selected such a way that preserve the sparse and global reconstruction structure. To this end, a transformation matrix $\boldsymbol{w}\in R^{d\times m}$ is defined to get [42]:

$$\min_{\boldsymbol{S},\boldsymbol{w}}\sum_{i=1}^{n}\left[\left\|\boldsymbol{w}^T\boldsymbol{x}_i-\boldsymbol{w}^T\boldsymbol{X}\boldsymbol{s}_i\right\|^2+\alpha\left\|\boldsymbol{s}\right\|_1+\gamma\left\|\boldsymbol{w}\right\|_{21}\right]$$
$$\text{subject to } \boldsymbol{S}_{ii}=0,\ \boldsymbol{w}^T\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{w}=\boldsymbol{I} \tag{47}$$

where γ denotes the regularization parameter and $\left\|\boldsymbol{w}\right\|_{21}=\sum_{i=1}^{d}\sqrt{\sum_{j=1}^{m}w_{ij}^{2}}$. With this solution, not only the global structure captured by sparsity weight matrix $\boldsymbol{S}$ helps to search relevant features but also by eliminating the unfavorable

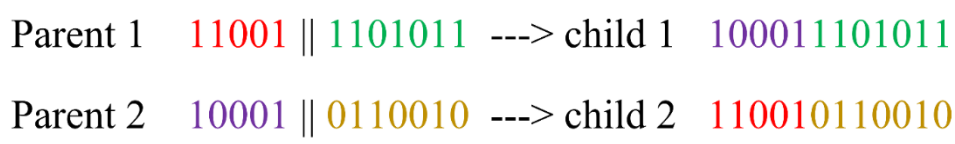

Fig. 5. The one-point crossover operator.

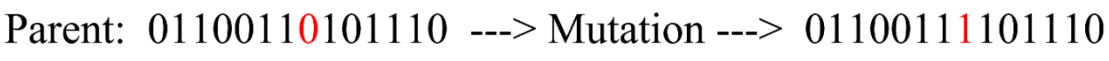

Fig. 6. The mutation operator.

features, a better estimation of the global structure is provided.

2.4.4. Least absolute shrinkage and selection operator (LASSO)

Let $\left(\boldsymbol{x}_i, y_i\right); i=1,\ldots,N$ where $\boldsymbol{x}_i=\left(x_{i1},\ldots,x_{ip}\right)^T$ is the regressors and y_i is the response for i th observation. The ordinary least square (OLS) minimizes the residual squared error. But, OLS has two main disadvantages. The first one is prediction accuracy. OLS provides estimations with low bias but large variance [43]. Some coefficients can be shrinked or set to zero, and thus, the prediction accuracy is improved. The variance is reduced by sacrificing a little bias. The second disadvantage is difficulty of inter prediction. When there are a large number of predictors, it is desirable that select a smaller subset exhibiting the most demanded effects. Each of two standard techniques, ridge regression and subset selection, have drawbacks for improving the OLS algorithm. Although the subset selection allows having interpretation models but because of dropping regressors from the model, it is extremely variable. In contrast, ridge regression shrinks coefficients and so, it is more stable. But, it does not provide an easy interpretable model because it does not set any coefficient to zero.

To deal with the discussed problems, LASSO has been introduced. It sets some coefficients to zero and shrinks the others. Therefore, it provides appropriate features in terms of both ridge regression and subset selection.

2.4.5. Genetic algorithm (GA)

One of the major heuristic algorithms that belongs to evolutionary methods is GA. GA is used to find the global optimum solution in various optimization problems. It is also used for feature selection in different applications. There are the following main steps in a GA [44]-[45]:
1)Generating an initial population
2)Fitness evaluation
3)Selection process
4)Crossover
5)Mutation
6)Repeat steps 3-6.

Each of these steps are briefly explained in the following:
1)Generating an initial population

At first, a coding structure has to be determined. A solution known as a chromosome is usually coded as a string of {0,1}. These components of chromosomes are called genes. Let have m solutions with N features as $x_i; i=1,\ldots,m$ where x_i is denoted by a string of {0,1} of length N.
2)Fitness evaluation

A fitness measure such as mutual information between the class variable and feature subset is considered. If the candidate solution satisfies the decided fitness value, the result is achieved. Otherwise, the following steps are followed.
3)Selection process

To select the individual x_i from the available solutions $\{x_1,\ldots,x_m\}$, the following probability is computed:

$$p\left(x_i\right)=\frac{f\left(x_i\right)}{\sum_{j=1}^{m} f\left(x_j\right)} \tag{48}$$

where $p\left(x_i\right)$ is the probability that x_i be a member of the next generation.

4)Crossover

The crossover operator generates child or new chromosomes from two parent chromosomes by combining information of parents. Different crossover methods can be used. For instance, the one-point crossover is illustrated in Fig 5. According to this crossover method, a random number c is selected between 1 and N. By appending the last $N-c$ componnets of the first parent chromosomes to the first c components of the second one, the first child chromosome is generated. By appending the last $N-c$ components of the

second parent chromosome to the first c components of the first parent chromosome, the second child chromosome is generated.

5)Mutation

The mutation operator is applied on each individual by perturbing the bit string. A simple and usual way is to generate a number v between 1 and N and make a random change in v th element of the string as shown in Fig. 6.

6)Repeat the process

After generation of a new child population, it is evaluated in terms of the fitness value. If it fails, the steps 3-6 are repeated to reach the maximum number of generations.

2.4.6. Sequential backward selection (SBS)

There are various search algorithms. Exponential algorithms evaluate several subsets that exponentially grow with the dimensionality of data [46]. Methods such as exhaustive search, beam search and brand and bound belong to this group [47]. The sequential algorithms remove or add features sequentially. These methods may trap in local minima. Sequential forward selection (SFS) and sequential backward selection (SBS) belong to this group. Randomized algorithms utilize the randomness in the search procedure to avoid the local minima. Methods such as GA belong to this group. SFS is known as the simplest greedy search method that starts from an empty set. It sequentially adds the feature which maximizes a given objective function in combination with the features that have already been selected. SFS performs well when the optimal subset is small. SBS is implemented in the opposite direction of SFS. It starts from a full set and it sequentially removes the features which degrade the objective function value. SBS works well when the optimal feature subset is large. While the main drawback of SFS is that it is unable to delete the feature that becomes obsolete after addition of new feature set, the main disadvantage of SBS is that it is unable to evaluate the goodness of a feature after discarding it.

3- EXPERIMENTS

In this section, at first the evaluation measures and telecom datasets are introduced. Then, the experimental results are discussed.

3.1.Evaluation measures and datasets

Six measures are used for evaluation of customer churn prediction as a two-class classification problem. Five of these measures are computed by the confusion matrix where the confusion matrix is constructed as [48]-[49]:

Actual class	Predicted class	
	1	0
1	TP	FN
0	FP	TN

where 1 (positive) is the label of class churn and 0 (negative) is the label of class non-churn. TP, TN, FP, FN are defined as:

True positive (TP): the number of correctly classified positive samples.

True negative (TN): the number of correctly classified negative samples.

False positive (FP): the number of incorrectly classified negative samples.

False negative (FN): the number of incorrectly classified positive samples.

The following measures are computed by using the confusion matrix [48]-[49]:

Sensitivity (recall): accuracy of class positive (churn), i.e., fraction of churn samples which are correctly classified as churn:

$$Sensitivity = \frac{TP}{TP + FN} \tag{49}$$

Specificity: accuracy of class negative (non-churn), i.e., the fraction of non-churn samples which are correctly classified:

$$Specificity = \frac{TN}{TN + FP} \tag{50}$$

Precision: reliability of class positive (churn), i.e., the number of correctly identified churns over the total number of churns identified by the used method:

$$Precision = \frac{TP}{TP + FP} \tag{51}$$

Accuracy: overall accuracy of the classifier:

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} \tag{52}$$

F-measure: a composite measure of accuracy and reliability of positive (churn) class computed by:

$$F - measure = 2.\frac{Precision.Sensitivity}{Precision + Sensitivity} \tag{53}$$

AUC: The performance of two-class classification methods or target detection ones are usually investigated by the receiver operating characteristic (ROC) curves. The ROC curve indicates the relationship between the probability of detection (PD) and the false alarm rate (FAR) given by:

$$PD = \frac{N_{cd}}{N_t}, \quad FAR = \frac{N_{fd}}{N} \tag{54}$$

Table 1. Classification results for BigML dataset.

Method	Sensitivity	Specificity	Precision	Accuracy	F-measure	AUC
DT	0.76	0.96	0.74	0.93	0.75	0.87
LR	0.66	0.83	0.39	0.81	0.49	0.83
RF	**0.82**	**0.99**	**0.94**	**0.97**	**0.88**	**0.94**
FFNN	0.69	0.98	0.82	0.94	0.75	**0.94**
LSTM	0.59	0.97	0.75	0.91	0.66	0.91
SVM	0.68	0.98	0.85	0.94	0.75	**0.94**
NN	0.46	0.93	0.52	0.86	0.49	0.70

Table 2. Target detection results for BigML dataset.

Method	Sensitivity	Specificity	Precision	Accuracy	F-measure	AUC
MSD	0	**1.00**	0	0.86	NaN	0.54
ASD	**0.82**	0.78	0.37	0.78	**0.51**	**0.86**
OSP	0.01	0.97	0.04	0.84	0.01	0.46
SAM	0.25	0.97	0.57	**0.87**	0.35	0.79
KSAM	0.21	0.96	0.47	0.86	0.29	0.74
CEM	0.21	0.97	0.52	0.86	0.30	0.83
STD	0.07	0.99	**0.59**	0.86	0.13	0.80

Table 3. Feature extraction results for BigML dataset.

Method	Sensitivity	Specificity	Precision	Accuracy	F-measure	AUC
PCA	0.44	0.91	0.45	0.85	0.45	0.73
LDA	0.26	0.91	0.32	0.82	0.29	0.69
CBFE	**0.76**	**0.95**	**0.71**	**0.92**	**0.73**	**0.90**
MMFLE	0.45	0.92	0.48	0.85	0.46	0.70

Table 4. Feature selection results for BigML dataset

Method	Sensitivity	Specificity	Precision	Accuracy	F-measure	AUC
ABACO	0.79	0.95	**0.73**	**0.93**	**0.76**	0.87
Relief-F	0.76	0.93	0.65	0.91	0.70	0.88
FSASL	0.44	0.91	0.44	0.84	0.44	0.73
LASSO	0.76	0.93	0.64	0.91	0.70	0.89
GA	**0.81**	0.95	0.72	**0.93**	**0.76**	**0.90**
SBS	0.15	**0.98**	0.62	0.87	0.24	0.62

where N is the total number of samples, N_t is total number of targets (churns), N_{cd} is the number of correctly detected samples, and finally N_{fd} represents the number of falsely detected targets. A detector achieving a higher PD with respect to others, at the same FAR, is preferred compared to them. A better detector results in a higher area under ROC curve (AUC) value.

Three datasets are used for evaluation of customer churn prediction methods: BigML[1], Telecom customer (kaggle)[2] and Telcom customer churn dataset[3]. The BigML dataset is a publicly available telecom data acquired by a company in the US. This data has 3333 samples with 18 features where 14.5% of them are labeled as churn, i.e., the churn rate of data is about 14.5%. This dataset contains the characteristics of telephony account features and usage. Some considered features are: number of messages, total day minutes, total day calls, total day charge, total night calls, total night charge, total

1 https://bigml.com/user/cesareconti89/gallery/dataset/58cfbada49c4a13341003cba
2 https://www.kaggle.com/abhinav89/telecom-customer/downloads/tele
3 https://www.kaggle.com/blastchar/telco-customer-churn/version/1

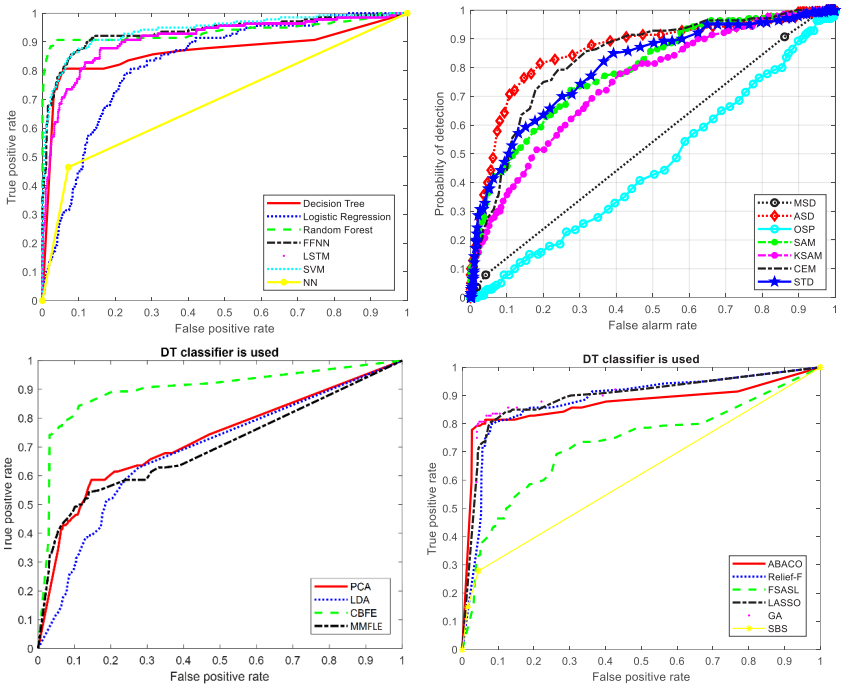

Fig. 7. ROC curves for BigML dataset.

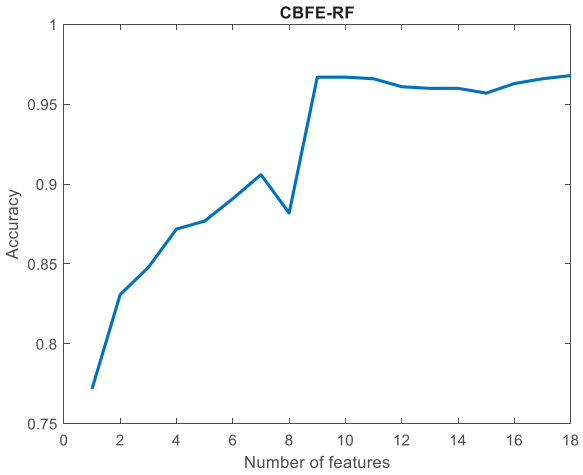

Fig. 8. RF classification accuracy versus the number of features extracted by CBFE in BigML dataset.

Table 5. Classification results for kaggle dataset.

Method	Sensitivity	Specificity	Precision	Accuracy	F-measure	AUC
DT	0.54	0.57	0.55	0.55	0.54	0.55
LR	0.23	**0.88**	**0.65**	0.56	0.34	0.62
RF	0.60	0.63	0.61	0.61	**0.61**	**0.66**
FFNN	0.58	0.65	0.62	**0.62**	0.60	**0.66**
LSTM	**0.65**	0.52	0.57	0.58	0.60	0.61
SVM	0.54	0.66	0.61	0.60	0.57	0.64
NN	0.50	0.55	0.52	0.53	0.51	0.53

Table 6. Target detection results for kaggle dataset.

Method	Sensitivity	Specificity	Precision	Accuracy	F-measure	AUC
MSD	0	**1.00**	0	0.51	NaN	0.50
ASD	(out of memory)					
OSP	0	**1.00**	0	0.51	NaN	0.47
SAM	**0.21**	0.85	0.59	**0.54**	**0.31**	0.56
KSAM	0.18	0.86	0.55	0.52	0.27	0.53
CEM	0.00	**1.00**	**1.00**	0.51	0.00	**0.62**
STD	0.00	**1.00**	0.25	0.51	0.00	0.54

Table 7. Feature extraction results for kaggle dataset.

Method	Sensitivity	Specificity	Precision	Accuracy	F-measure	AUC
PCA	**0.51**	0.54	0.52	0.53	0.51	0.53
LDA	0.50	0.52	0.50	0.51	0.50	0.52
CBFE	**0.51**	**0.56**	**0.53**	**0.54**	**0.52**	**0.54**
MMFLE	**0.51**	0.54	0.52	0.53	0.51	0.53

Table 8.: Feature selection results for kaggle dataset.

Method	Sensitivity	Specificity	Precision	Accuracy	F-measure	AUC
ABACO	0.53	**0.58**	0.55	0.55	0.54	0.56
Relief-F	0.53	**0.58**	0.55	0.55	0.54	0.57
LASSO	0.52	0.55	0.53	0.53	0.52	0.54
SBS	**0.55**	**0.58**	**0.56**	**0.56**	**0.55**	**0.59**
FSASL	Out of memory					
GA	High computation time					

international calls and total international charge.

Telecom customer (kaggle) dataset contains approximately 100000 samples and 100 features. Various important factors are recorded and considered for describing the customers of telecom industry and attributes of the telecom industry. The percentage of churn samples is about 50%. Some considered attributes in this dataset are as follows: mean monthly revenue, mean number of monthly minutes of use, mean total monthly recurring charge, mean number of directory assisted calls, mean number of dropped voice calls, mean number of blocked voice calls, mean number of unanswered data calls, mean number of customer care calls, mean unrounded minutes of use of outbound wireless to wireless voice calls, mean number of call forwarding calls, number of active subscribers in household, total minutes of use over the life of the customer, average monthly revenue over the previous six months and number of adults in household. The target variable is churn where it explains whether the customer will churn or not.

The Telco customer churn dataset contains 703 samples and 19 features. Different information such as customer account information (payment method, monthly charges, how long the customer has been with operator), demographic information (age, gender and whether customer has partners or not) and services which customer has signed up for (phone, internet, online backup) are included in the data features. Customers who have left the service within the last month are marked as churn.

70% of each dataset is used for training and the remainder is used for testing. Note that all of experiments are done with the same training samples in each dataset to have a fair comparison between different methods.

3.2.Experimental results

At first, the customer churn prediction results for BigML dataset are reported. In the first experiment, different classifiers (DT, LR, RF, FFNN, LSTM, SVM and NN) are compared together. Comparison of classifiers with respect to 6 evaluation measures are represented in Table 1. The best result in each column is shown as bold. Among various methods, RF provides the best results in terms of all measures. Generally, after RF, DT, FFNN and SVM provide good results compared to other methods. NN has the weakest efficiency in identification of churns.

In the second experiment, performance of different target detection methods (MSD, ASD, OSP, SAM, KSAM, CEM and STD) are evaluated (see Table 2). Among them, MSD fails to identify churn. The highest overall accuracy and AUC are obtained by SAM and ASD, respectively. Generally, ASD is the best choice

because it significantly outperforms other target detection methods from the view of churn identification (by providing the highest sensitivity value).

In the third experiment, the efficiency of four feature

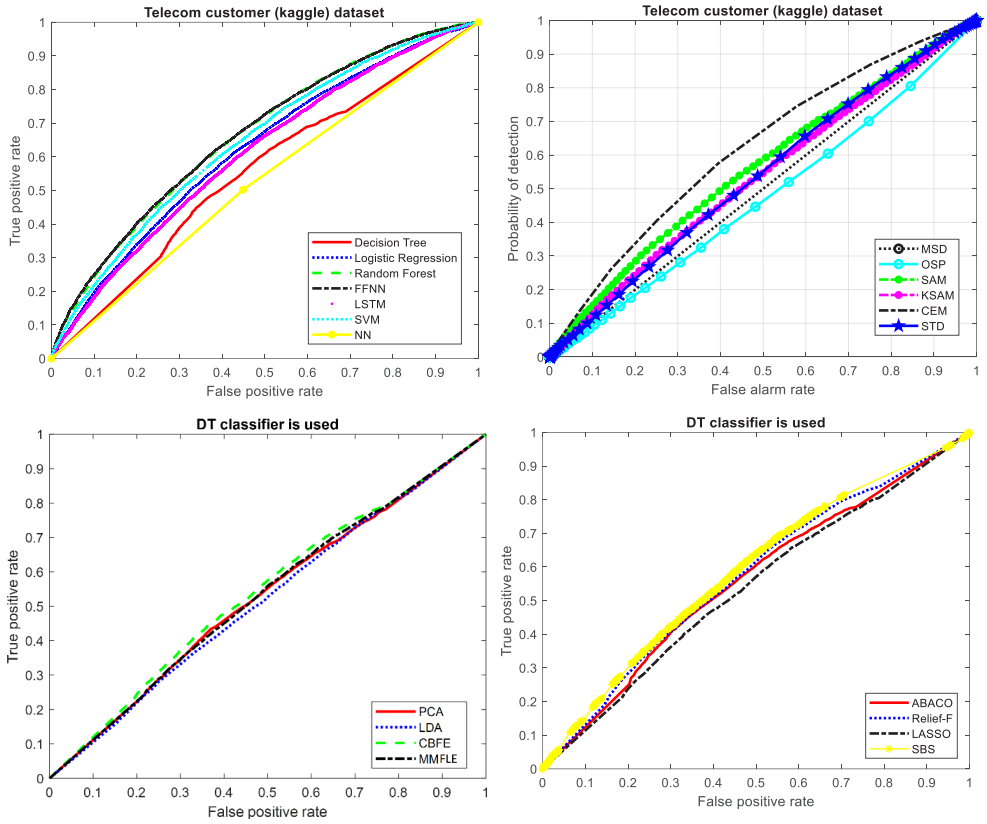

Fig. 9. ROC curves for kaggle dataset.

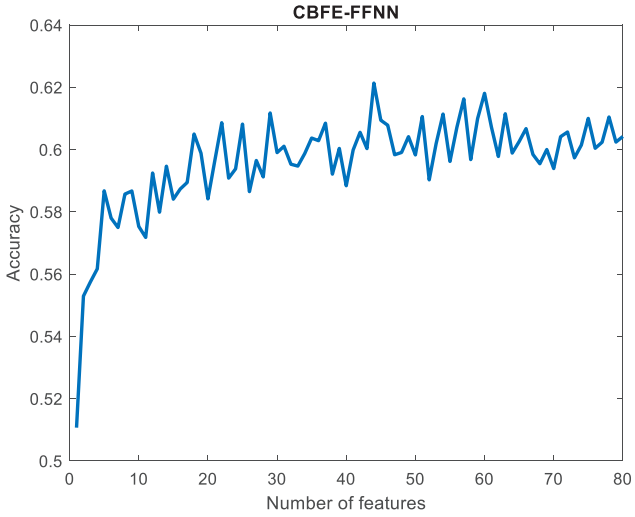

Fig.10: FFNN classification accuracy versus the number of features extracted by CBFE in kaggle dataset.

extraction methods (PCA, LDA, CBFE and MMFLE) are evaluated as feature reduction before churn classification. To have a fair comparison between different feature extraction methods, 10 features are extracted by each method. The extracted features are fed to a DT for classification. The results are reported in Table 3. As seen, the best performance is related to CBFE while the worst results are obtained by LDA.

In the fourth experiment, the feature selection methods (ABACO, relief-F, FSASL, LASSO, GA and SBS) are compared together. The features selected by each method are given to the same DT classifier to have a fair comparison (see Table 4). Generally, GA and ABACO rank first and second, respectively. SBS approximately fails to identify churns.

The ROC curves related to four above experiments, associated with Tables 1-4, are shown in Fig. 7. According to the experimental results for BigML dataset, we found:

Table 9. Classification results for Telco customer churn dataset.

Method	Sensitivity	Specificity	Precision	Accuracy	F-measure	AUC
DT	0.51	0.83	0.53	0.74	0.52	0.74
LR	**0.71**	0.79	0.56	0.77	**0.63**	0.83
RF	0.50	0.90	0.65	0.79	0.56	0.82
FFNN	0.51	0.92	0.70	**0.81**	0.59	**0.84**
LSTM	0.53	0.89	0.65	0.79	0.58	0.83
SVM	0.44	**0.93**	**0.71**	0.80	0.54	0.78
NN	0.49	0.81	0.49	0.72	0.49	0.65

Table 10.: Target detection results for Telco customer churn dataset.

Method	Sensitivity	Specificity	Precision	Accuracy	F-measure	AUC
MSD	0.00	**1.00**	**1.00**	0.72	0.00	0.48
ASD	**0.80**	0.59	0.42	0.64	**0.55**	0.79
OSP	0.06	0.73	0.08	0.54	0.07	0.23
SAM	0.50	0.86	0.58	0.76	0.54	0.81
KSAM	0.34	0.95	0.71	**0.78**	0.46	0.81
CEM	0.29	0.96	0.74	**0.78**	0.41	**0.82**
STD	0.14	0.97	0.60	0.74	0.22	0.71

Table 11. Feature extraction results for Telco customer churn dataset.

Method	Sensitivity	Specificity	Precision	Accuracy	F-measure	AUC
PCA	**0.52**	0.83	0.53	0.74	0.52	0.72
LDA	0.51	0.83	0.53	0.74	0.52	**0.73**
CBFE	0.51	**0.84**	**0.55**	**0.75**	**0.53**	0.72
MMFLE	0.51	0.83	0.54	0.74	0.52	**0.73**

Table 12. Feature selection results for Telco customer churn dataset.

Method	Sensitivity	Specificity	Precision	Accuracy	F-measure	AUC
ABACO	0.51	0.82	0.52	0.74	0.52	0.71
Relief-F	0.48	0.83	0.52	0.74	0.50	**0.73**
FSASL	0.40	0.84	0.49	0.72	0.44	0.67
LASSO	0.33	**0.89**	0.53	0.74	0.41	**0.73**
GA	**0.52**	0.85	**0.57**	**0.76**	**0.54**	**0.73**
SBS	0.34	0.79	0.38	0.67	0.36	0.58

·The best classification method: RF
·The best target detection method: ASD
·The best feature extraction method: CBFE
·The best feature selection method: GA

Moreover, it is found that the classification methods generally work better than target detection methods; and among various methods, RF is the best choice for customer churn prediction. To assess the effects of the number of extracted features, CBFE is applied before RF and the accuracy of RF is measured in different

number of extracted features. The results are shown in Fig. 8. As seen, with increasing the number of extracted features, the RF accuracy is improved and the highest accuracy (97%) is obtained with 18 features, i.e., when no feature reduction is occurred. However, RF can achieve 97% accuracy without applying any feature reduction methods such as CBFE. The best feature selection method, i.e., GA, is also applied to RF and in average 96% accuracy is obtained that is less than 97%. GA selects in average 10 features in each run. So, the results show that the feature reduction methods have not any positive effect in classification improvement of RF in BigML dataset.

The above experiments are reported for Telecom customer (kaggle) dataset. The classification methods are compared together in Table 5. The highest accuracy and AUC are obtained by FFNN. RF ranks second in terms of accuracy and AUC. The lowest accuracy and sensitivity are obtained by NN

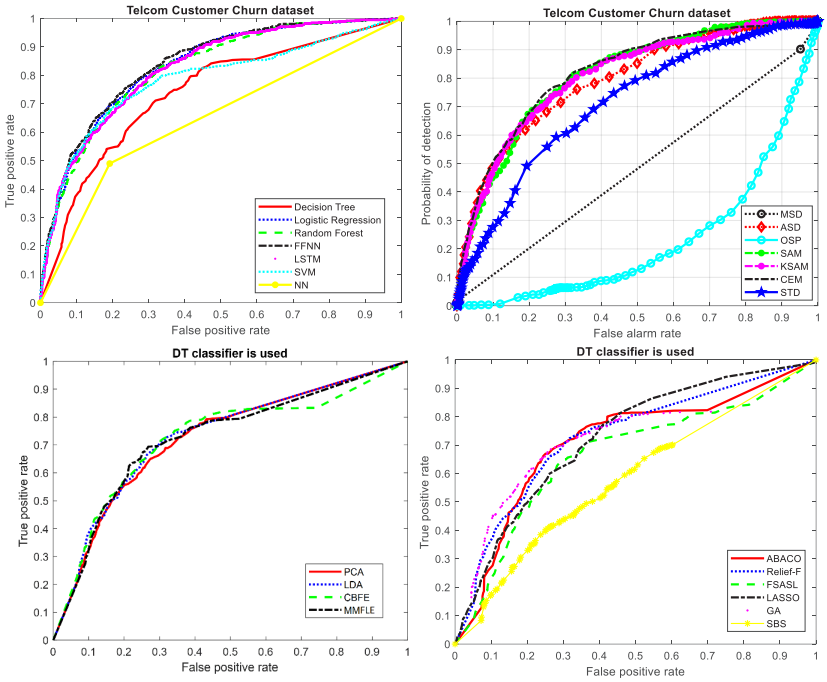

Fig. 11. ROC curves for Telco customer churn dataset.

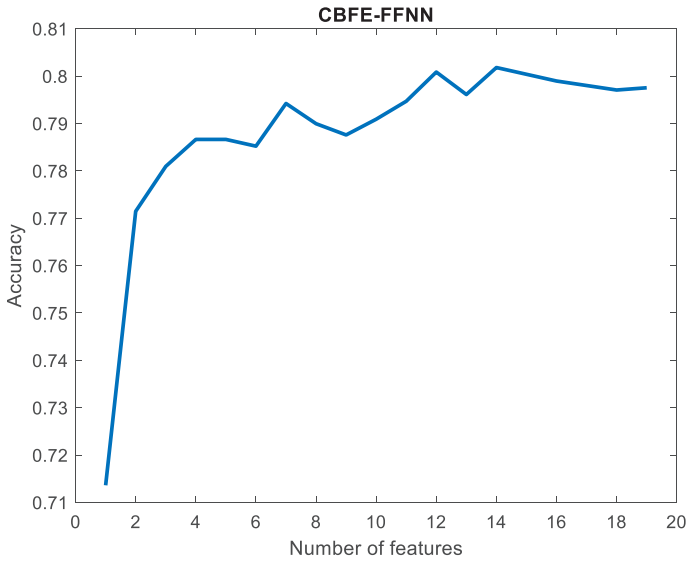

Fig. 12. FFNN classification accuracy versus the number of features extracted by CBFE in Telco customer churn

and LR,

respectively. Among different classifiers, LSTM network identifies churns with the highest accuracy, i.e., the highest sensitivity value.

The target detection algorithms are compared together and the results are shown in Table 6. ASD results in out of memory error in MATLAB, and so, it is removed from the comparisons. Other methods except SAM and KSAM also fail to identify churn. Generally, target detection methods are not good choices for customer churn prediction in kaggel dataset.

Different feature extraction methods are compared (Table 7) and CBFE provides the best results. Comparison of feature selection methods are shown in Table 8. FSASL results in out of memory error, and so it is removed from the comparison. In addition, due to high computation time of GA, it is set aside. Among remained methods, SBS shows relatively better classification results compared to other methods.

dataset.

The ROC curves of kaggle dataset associated with Tables 5-8 are shown in Fig. 9. According to the experimental

results, among various methods, FFNN is the best classifier and CBFE is the best feature extraction method. Combination of them when CBFE is used as feature reduction before the FFNN classifier is assessed. Fig. 10 shows accuracy of FFNN versus the number of features extracted by CBFE. The maximum accuracy 62% is obtained with 44 features. FFNN can obtain 62% accuracy alone, i.e., without CBFE, which it shows that feature reduction has not any positive effect in classification performance of the kagge dataset. According to the experiments results for kaggle dataset, we found:

·The best classification method: FFNN
·The best target detection method: SAM
·The best feature extraction method: CBFE
·The best feature selection method: SBS

The above experiments are also done for Telco customer churn dataset. The comparison of classifiers are shown in Table 9. The highest accuracy and AUC value are achieved by FFNN. The highest sensitivity and F-measure are obtained by LR. NN acquires the lowest overall accuracy. Different target detection methods are also compared together (see Table 10). CEM and KSAM provide the highest accuracy. But, ASD is the best method in terms of sensitivity (detection of churns). Among various feature extraction methods and feature selection ones, CBFE and GA rank one (see Table 11 and 12). The ROC curves, associated with Tables 9-12 are shown in Fig. 11.

Accuracy of FFNN versus the number of features extracted by CBFE is shown in Fig. 12. Generally, with increasing the number of features, the classification accuracy is improved. The highest accuracy (80%) is obtained with 14 features that is a little bit lower than that of the only use of FFNN without feature reduction (81%). The best feature selection method is GA according to Table 12. The performance of the best classifier (FFNN) is assessed when GA is used as a preprocessing step for feature reduction. GA in average selects 8 features from the dataset. The classification accuracy of FFNN obtained by the selected features is 93% that is 12% more than that of the only use of FFNN without feature reduction. In contrast to two previous datasets, which feature reduction has not positive effect in the classification accuracy, the feature selection method of GA provides significant improvement in classification accuracy of classifier (FFNN) in Telco customer churn dataset. According to the experimental results for Telco customer churn dataset, we found:

·The best classification method: FFNN
·The best target detection method: CEM, ASD
·The best feature extraction method: CBFE
·The best feature selection method: GA

4- CONCLUSION

Various machine learning methods were evaluated for customer churn prediction. The algorithms are divided into four groups: classification, target detection, feature extraction and feature selection. Generally, classification methods outperform target detection ones. Among classification methods, RF and FFNN show the highest prediction accuracy. Among target detection methods, ASD and CEM outperform others; and among feature selection methods, GA works better than the others. Three telecom datasets were experimented. In two datasets, feature reduction (feature extraction or feature selection) was not effective in the prediction results. But, in one dataset, GA as a feature selection method provided significant improvement in classification accuracy of FFNN. As a brief and general conclusion, FFNN and RF methods as classifier together with GA as a feature selector are the best candidates for customer churn prediction.

REFERENCES

[1] Shirazi, F., Mohammadi, M.. A big data analytics model for customer churn prediction in the retiree segment. International Journal of Information Management, 48 (2019), 238-253.

[2] Ahn, J.-H. , Han, S. P., Lee, Y.S. Customer churn analysis: Churn determinants and mediation effects of partial defection in the Korean mobile telecommunications service industry. Telecommunications Policy, 30 (2006), 552–568.

[3] Kim, H.-S., Yoon., C.-H.. Determinants of Subscriber Churn and Customer Loyalty in the Korean Mobile Telephony Market. Telecommunications Policy, 28 (2004), 751-765.

[4] Keramati, A., Ardabili, S. M.S.. Churn analysis for an Iranian mobile operator. Telecommunications Policy, 35 (4) (2011), 344-356.

[5] Kisioglu, P., Topcu, Y. I. Applying Bayesian Belief Network approach to customer churn analysis: A case study on the telecom industry of Turkey. Expert Systems with Applications, 38 (6) (2011), 7151-7157.

[6] Maldonado, S., Flores, Á., Verbraken, T., Baesens, B., Weber, R.. Profit-based feature selection using support vector machines – General framework and an application for customer retention. Applied Soft Computing, 35 (2015), 740-748.

[7] Amin, A., Anwar, S., Adnan, A., Nawaz, M., Alawfi, K., A. H., Huang, K. Customer churn prediction in the telecommunication sector using a rough set approach. Neurocomputing, 237 (2017), 242-254.

[8] Caigny, A. D., Coussement, K., De Bock, K.W. A new hybrid classification algorithm for customer churn prediction based on logistic regression and decision trees. European Journal of Operational Research, 269 (2) (2018), 760-772.

[9] Keramati, A., Jafari-Marandi, R., Aliannejadi, M., Ahmadian, I., Mozaffari, M., Abbasi, U. Improved churn prediction in telecommunication industry using data mining techniques. Applied Soft Computing, 24 (2014), 994-1012.

[10] Tsai, C.-F., Lu, Y.-H. Customer churn prediction by hybrid neural networks. Expert Systems with Applications, 36 (10) (2009), 12547-12553.

[11] Vafeiadis, T., Diamantaras, K.I., Sarigiannidis, G., Chatzisavvas, K.Ch. A comparison of machine learning techniques for customer churn prediction. Simulation Modelling Practice and Theory, 55 (2015), 1-9.

[12]De Bock, K. W., Poel, D. V. d. An empirical evaluation of rotation-based ensemble classifiers for customer churn prediction. Expert Systems with Applications, 38 (10) (2011), 12293-12301.
[13]Idris, A., Rizwan, M., Khan, A. Churn prediction in telecom using Random Forest and PSO based data balancing in combination with various feature selection strategies. Computers & Electrical Engineering, 38 (6) (2012), 1808-1819.
[14]Tan, Steinbach, Kumar. Introduction to Data Mining, chapter 4, Classification: Basic Concepts, Decision Trees, and Model Evaluation (2004).
[15]Shang, Z., Deng, T., He, J., Duan, X. A novel model for hourly PM2.5 concentration prediction based on CART and EELM. Science of The Total Environment, 651 (Part 2) (2019), 3043-3052.
[16]Cutler, A. Random Forests for Regression and Classification, Utah State University, Ovronnaz, Switzerland, September 15-17 (2010).
[17]Chen, L., Su, W., Feng, Y., Wu, M., She, J., Hirota, K. Two-layer fuzzy multiple random forest for speech emotion recognition in human-robot interaction. Information Sciences, 509 (2020), 150-163.
[18]Akca, Z. and Kaya, R. On the Taxicab Trigonometry. Jour. of Inst. of Math& Comp. Sci. (Math. Ser) 10 (3) (1997), 151-159.
[19]Ben-Hur A., Weston J. A User's Guide to Support Vector Machines. In: Carugo O., Eisenhaber F. (eds) Data Mining Techniques for the Life Sciences. Methods in Molecular Biology (Methods and Protocols), 609 (2010), Humana Press.
[20]Svozil, D., Kvasnicka, V., Pospichal, J. Introduction to multi-layer feed-forward neural networks. Chemometrics and Intelligent Laboratory Systems, 39 (1) (1997), 43-62.
[21]Hochreiter, S. Long Short-Term Memory, Neural Computation, 9 (8) (1997), 1735-1780.
[22]Sood, A., Long Short-Term Memory, Accessed in Sept. 2019, available on: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=2ahUKEwiUn7G2-cjkAhWBSBUIHZjBAKoQFjAAegQIBBAC&url=http%3A%2F%2Fpages.cs.wisc.edu%2F~shavlik%2Fcs638%2FlectureNotes%2FLong%2520Short-Term%2520Memory%2520Networks.pdf&usg=AOvVaw2LVZ1s0x0E87xrCxzImfi8.
[23]Kwon, H., Nasrabadi, N.M. A Comparative Analysis of Kernel Subspace Target Detectors for Hyperspectral Imagery. EURASIP Journal on Advances in Signal Processing, Article number: 029250 (2006) (2007), https://doi.org/10.1155/2007/29250.
[24]Scharf, L. L., Friedlander, B., Matched subspace detectors. *IEEE Transactions on Signal Processing*, 42, (8) (1994), 2146-2157.
[25]Harsanyi, J. C. and Chang, C.-I. Hyperspectral image classification and dimensionality reduction: an orthogonal subspace projection approach. *IEEE Transactions on Geoscience and Remote Sensing*, 32 (4) (1994), 779-785.
[26]Kraut, S., Scharf, L. L., McWhorter, L. T. Adaptive subspace detectors. *IEEE Transactions on Signal Processing*, 49 (1) (2001), 1-16.
[27]Zhao, R., Z., Shi, Zou, Z., Zhang, Z. Ensemble-Based Cascaded Constrained Energy Minimization for Hyperspectral Target Detection. Remote Sens., 11 (11) (2019), 1310.
[28]Kruse, F.A., Lefkoff, A.B., Boardman, J.W., Heidebrecht, K.B., Shapiro, A.T., Barloon, P.J., Goetz, A.F.H. The spectral image processing system (SIPS)—interactive visualization and analysis of imaging spectrometer data. Remote Sensing of Environment, 44 (2–3) (1993),145-163.
[29]Camps-Valls, G. Kernel spectral angle mapper. *Electronics Letters*, 52 (14) (2016), 1218-1220.
[30]Zhang, Y., Du, B., Zhang, L. A Sparse Representation-Based Binary Hypothesis Model for Target Detection in Hyperspectral Images. *IEEE Transactions on Geoscience and Remote Sensing*, 53 (3) (2015), 1346-1354.
[31]Cui, M., Prasad, S. Sparse representation-based classification: Orthogonal least squares or orthogonal matching pursuit?. Pattern Recognition Letters, 84 (2016), 120-126.
[32]Imani, M., Ghassemian, H. Binary coding based feature extraction in remote sensing high dimensional data. Information Sciences, 342 (2016), 191-208,
[33]Fukunaga, K. Introduction to Statistical Pattern Recognition. Academic Press Inc., San Diego (1990).
[34]Liang, Z., Shi, P. Kernel direct discriminant analysis and its theoretical foundation. Pattern Recognition, 38 (3), 445-447 (2005).
[35]Imani, M., Ghassemian, H. Band Clustering-Based Feature Extraction for Classification of Hyperspectral Images Using Limited Training Samples. IEEE Geoscience and Remote Sensing Letters, 11 (8) (2014), 1325-1329.
[36]Imani, M., Ghassemian, H. Feature Extraction Using Median-Mean and Feature Line Embedding. International Journal of Remote Sensing, 36 (17) (2015), 4297-4314.
[37]Omran, M. G.H., Al-Sharhan, S. Improved continuous Ant Colony Optimization algorithms for real-world engineering optimization problems. Engineering Applications of Artificial Intelligence, 85 (2019), 818-829.
[38]Kashef, S., Nezamabadi-pour, H. An advanced ACO algorithm for feature subset selection. Neurocomputing, 147 (2015), 271-279.
[39]Robnik-Šikonja, M., Kononenko, I. Machine Learning Journal, 53 (2003), 23-69.
[40]Megchelenbrink, W. Relief-based feature selection in bioinformatics: detecting functional specificity residues from multiple sequence alignments, Master thesis in information science, supervised by Elena Marchiori and Peter Lucas, Radboud University Nijmegen (2010).
[41]Roffo, G., Feature Selection Library (MATLAB Toolbox), Feature Selection Library (FSLib) (2018), arXiv:1607.01327.
[42]Du, L., Shen, Y.-D., Unsupervised Feature Selection with

Adaptive Structure Learning (2015), arXiv:1504.00736v1 [cs.LG].

[43] Tibshirani, R., Regression Shrinkage and Selection via the Lasso. Journal of The Royal Statistical Society, Series B (1994).

[44] Stripling, E., Broucke, S., Antonio, K., Baesens, B., Snoeck, M., Profit maximizing logistic model for customer churn prediction using genetic algorithms. Swarm and Evolutionary Computation, 40 (2018), 116-130.

[45] Sivakumar, S., Chandrasekar, C., Feature Selection Using Genetic Algorithm with Mutual Information. International Journal of Computer Science and Information Technologies, 5 (3) (2014), 2871-2874.

[46] Peng, H.-Y., Jiang, C.-F., Fang, X., Liu, J.-S., Variable selection for Fisher linear discriminant analysis using the modified sequential backward selection algorithm for the microarray data. Applied Mathematics and Computation, 238 (2014), 132-140.

[47] Yan, C., Liang, J., Zhao, M., Zhang, X., Zhang, T., Li, H., A novel hybrid feature selection strategy in quantitative analysis of laser-induced breakdown spectroscopy, Analytica Chimica Acta, 1080 (2019), 35-42.

[48] Amin, A., Anwar, S., Adnan, A., Nawaz, M., Alawfi, K., Hussain, A., Huang, K., Customer churn prediction in the telecommunication sector using a rough set approach. Neurocomputing, 237 (2017), 242-254.

[49] Tunga, M. A., Karahoca, A., Detecting GSM churners by using Euclidean Indexing HDMR. Applied Soft Computing, 27 (2015), 38-46.

[50] Li, P., Li, S., Bi, T., Liu, Y., Telecom customer churn prediction method based on cluster stratified sampling logistic regression. *International Conference on Software Intelligence Technologies and Applications & International Conference on Frontiers of Internet of Things*, Hsinchu (2014), 282-287.

[51] Othman, N. H., Lee, K. Y., Radzo, A. R. M. l, Mansor, W., Rashid, U. R. M., Optimal PCA-EOC-KNN Model for Detection of NS1 from Salivary SERS Spectra. *2018 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*, Bangkok (2018), 204-208

[52] Bishop, C. M., Pattern Recognition and Machine Learning, Springer (2006).

[53] Fukunaga, K., *Introduction to Statistical Pattern Recognition*, 2nd ed. New York: Academic (1990).

HOW TO CITE THIS ARTICLE

M.Imani, Customer Churn Prediction in Telecommunication Using Machine Learning: A Comparison Study , AUT J. Model. Simul., 52(2) (2020) 229-250.

DOI: 10.22060/miscj.2020.18038.5202