J Inf Process Syst, Vol.15, No.6, pp.1306~1325, December 2019
https://doi.org/10.3745/JIPS.04.0149

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Feature Selection Using Submodular Approach for Financial Big Data

Girija Attigeri*, Manohara Pai M. M.**, and Radhika M. Pai***

Abstract

As the world is moving towards digitization, data is generated from various sources at a faster rate. It is getting humungous and is termed as big data. The financial sector is one domain which needs to leverage the big data being generated to identify financial risks, fraudulent activities, and so on. The design of predictive models for such financial big data is imperative for maintaining the health of the country's economics. Financial data has many features such as transaction history, repayment data, purchase data, investment data, and so on. The main problem in predictive algorithm is finding the right subset of representative features from which the predictive model can be constructed for a particular task. This paper proposes a correlation-based method using submodular optimization for selecting the optimum number of features and thereby, reducing the dimensions of the data for faster and better prediction. The important proposition is that the optimal feature subset should contain features having high correlation with the class label, but should not correlate with each other in the subset. Experiments are conducted to understand the effect of the various subsets on different classification algorithms for loan data. The IBM Bluemix Big Data platform is used for experimentation along with the Spark notebook. The results indicate that the proposed approach achieves considerable accuracy with optimal subsets in significantly less execution time. The algorithm is also compared with the existing feature selection and extraction algorithms.

Keywords

Classification, Correlation, Feature Subset Selection, Financial Big Data, Logistic Regression, Submodular Optimization, Support Vector Machine

1. Introduction

The Indian financial sector is being reformed through several initiatives of the Government of India such as cashless transactions, complete digitization, etc. This results in every transaction and activity leaving a digital footprint. Thus, the digital data generated is very large as it includes digital signatures, images, text, and other forms as well. If this is utilized properly, several financial predictions can be posited towards strengthening the economy.

Analyzing such data is significant for the government and other organizations in order to make well-informed decisions towards betterment. However, gaining insight from such data has its own limitations as it is not ready for analysis.

Manuscript received July 20, 2018; first revision October 2, 2018; second revision November 19, 2018; accepted February 2, 2019.
Corresponding Author: Manohara Pai M. M. (mmm.pai@manipal.edu)
* Dept. of Information and Communication Technology, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, India ({girija.attigeri, mmm.pai, radhika.pai}@manipal.edu)

The financial sector produces a huge volume of quotes, trading data, and transaction data. The stock exchange alone produces terabytes of data every day. Financial data has high velocity as it is generated at the rate of 200 or more transactions per second [1]. It also includes various sources of data such as corporate banking, trading, institutional reference data, market data, and many other sources. Also, the challenging aspect of financial data is the continual stream of rules and regulations, which brings new sources and complex metrics into the financial system. Integrating the data from various sources adds in noise and irrelevant attributes, which is a veracity aspect in big data. These make the financial data more interesting from the perspective of big data making it more popular as financial big data. However, irrelevant and noisy attributes provide no useful information for analysis. Useful attributes contribute towards potential analysis and effective usage of time. The process of identifying these attributes is referred to as feature selection. Considering the task at hand, one might need to select the right subset of features of data. This means identifying relevant and useful attributes from a large pool of data using data pre-processing techniques.

Data preprocessing is an essential step for quality outcome from the model under consideration [2]. While preparing data for analytics, the major concerns are data cleaning, integrity of the data, selecting important features, etc., as large numbers of features increase the computational cost of analytical algorithms. This can be solved by reducing the dimension of the features [3], by choosing features useful for modelling.

Data with selected features are used for building an analysis model, which can be used for prediction and decision making in the considered domain. The present paper aims at analyzing loan data, which predicts the loan as non-performing assets (NPA) or not by building a predictive model using machine learning algorithms. Predicting this is very crucial for the banking sector, as early identification helps to take the necessary action to prevent loss. The effectiveness of prediction is evaluated using statistical analysis. For building the prediction model for loan data, feature subset selection is done through the submodular approach. Feature subset selection is identifying or constructing a subset having 'k' features from a given set of 'n' features. Submodularity is a property that exhibits an inherent diminishing returns, which means an element when added to a smaller subset has more advantage than adding it to a larger subset. The forward selection of features can be done by setting submodular objective function [4]. Submodularity has several applications in machine learning and artificial intelligence such as feature selection, inference, optimized information retrieval, and clustering [5]. It has been effectively applied in sensor placement, data subset selection, and set cover problems.

The main contribution of the paper is to use submodular optimization for feature subset selection to improve the outcome of prediction algorithms for big data. The implementation of this approach is carried out using Big Data techniques by devising the Hadoop MapReduce algorithm. This splits the data, and carries computations in parallel in the Map phase. The reduce phase combines solutions to get correlation values, which are used for selecting useful features. The specific purposes of the paper are:

(i) Applying submodular based feature selection on Big data using the MapReduce algorithm; and
(ii) Analyzing and illustrating the proposed approach over the prediction models on loan data with respect to time and accuracy.

In the remainder of the paper, a review of the existing work is explained in Section 2. In Section 3, the methodology used for feature selection algorithms is discussed. Section 4 describes the execution of the submodular feature selection and its evaluation using classification algorithms. Empirical results and analysis are discussed in Section 5. A summary of the work done is presented in Section 6.

2. Related Work

In this section, the literature review of the concepts of submodularity, feature selection algorithms, and big data is presented. Li et al. [3] presented various feature selection methods such as similarity based, sparse-learning-based, information-theoretical-based, and statistical-based methods. They also mentioned that scalability, stability, and model selection as the open problems when dealing with feature selection. Arguello [4] presented a survey on feature selection algorithms. The author discussed stepwise forward selection, backward selection, submodular optimization, and other parameter minimization methods for feature selection and presented the results for simple toy data. Fattah [6] proposed statistical feature selection using the term 'distribution' for text categorization and evaluated performance. Kira and Rendell [7] used the relief score for feature selection. The method uses forward selection avoiding heuristic searches for features. The authors tested the algorithm on LED and parity domain data. Fallahpour et al. [8] used the wrapper based feature selection method with sequential floating forward selection. They tested using the support vector machine (SVM) classifier for bankruptcy prediction. Wright et al. [9] applied different feature selection methods on college student data having 30 attributes. The identified features were used for predicting the mean income of postgraduate students having undergraduate debt obligations.

Kim [10] proposed the feature selection method based on distributional differences in each positive and negative data. Maldonado et al. [11] proposed the feature selection method using acquisition costs based on the mixed integer linear programming. The purpose was to select less than 10 variables for loan analysis, which was done based on the SVM classifier.

Krause and Cevher [12] analyzed the greedy algorithms for the dictionary selection problem, which generalizes the subset selection for the prediction of multiple variables. They used the approximate submodularity notion to provide additive approximation guarantees. But their analysis was for a more general problem, than for the feature subset selection. Bar et al. [13] used the convolutional neural network (CNN) for identifying the feature set. They extracted the most informative features for the given task from layers of CNN. They tested the approach on a radiograph dataset having 600 samples.

Similarly Iyer et al. [14] proposed a novel framework for both unconstrained and constrained submodular function optimization. The feature subset selection using the submodular optimization problem was found of great efficacy in a number of areas in machine learning, such as speech data subset selection [15], feature subset selection [16], problems on social influence, and sensor placement [17]. Even though these problems were NP-hard, a study by Nemhauser et al. [18] indicated that the submodular optimization problem could be resolved by a simple greedy algorithm. The results proved that the greedy algorithm initialized with an empty set, iteratively added elements (features) with the highest marginal gain.

Hall [19] emphasized on the hypothesis, which stated that a good feature subset contained features which highly correlated with the label of the class and uncorrelated with each other. The authors used the correlation analysis based feature selection for synthetic and real data such as golf data. The features selected were used for decision tree algorithms C4.5 and ID3 for predictions. Recent research effort is concentrating in the direction of feature selection for big data. Pouramirarsalani et al. [20] presented the hybrid feature selection algorithm called as whale and used the genetic algorithm for fraud detection in the e-banking system.

Wang et al. [21] proposed a pre-processing method using the wrapper based feature selection for traffic data. It defined feature selection as an optimization function. The proposed method had exponential time complexity. Clustering based feature selection was proposed by Gangurde [22] on a theoretic graph. The approach first constructed a neighborhood graph and then found minimum spanning tree of the data points, which resulted in a representative set of features, but the implementation details and empirical results did not emphasize much on the algorithm.

Sarlin [23] emphasized the importance of data reduction algorithms, and provided a comparative analysis of the results obtained for the financial data. Bar and Zaelit [24] applied the feature space transformation method using the principal component analysis on bank data. They used the transformed feature for classification to predict private companies' retailed earnings, but the experimentation was carried out on small data.

3. Background

In this section, the concepts used for the current work are discussed. Section 3.1 describes the basic terminologies used for feature selection and its importance to handle big data prediction problems. In Section 3.2, the implementation frameworks for big data such as MapReduce, Hadoop, and Spark are discussed. Section 3.3 briefs about the submodular optimization approach.

3.1 Feature Selection

Feature selection is an essential and frequent practice in data pre-processing. It is a vital component of the machine learning and data analytics process [2]. In statistical analysis and machine learning terminologies, it is also called as variable selection or attribute selection. It is a process of constructing an optimal subset by selecting features, which are relevant for specific analysis and removing irrelevant, redundant or noisy data. It is intended to speed-up machine learning algorithms, improve efficiency, and increase comprehensibility. Irrelevant features are features that do not provide any useful information, and redundant features are features that do not add more information than the currently selected features. The feature selection process involves the following steps:

- Generation of subset: It is a procedure, which selects a candidate feature at each step using one of the methods such as forward, backward, weighing or random selection.
- Subset evaluation: The generated candidate subset is evaluated against the criteria. It tests the goodness of the candidate subset. There are two ways of evaluations, namely, dependent and independent. The dependent evaluation criteria involve the underlying mining/analytical algorithms. Independent evaluation is done without involving the mining algorithms.
- Stopping criteria: The selection of features has to stop at some point using the stopping criteria such as accuracy being greater than threshold, etc.
- Validation of results: The feature selection process has to be validated by conducting various tests and comparisons.

There are two methods for feature selection: feature subset selection and feature extraction.

The first method chooses a subset of relevant features for analysis model. It has different approaches such as filters, wrappers, and embedded methods.

The filter methods select features without considering any learning. This is fast and efficient, but misses features, which can be useful when combined with other features. A general algorithm for the filter method is as follows; Given a dataset D={X, L}, where, X is a feature set and L is the class label, the algorithm starts with one of the subsets X′= Ø. In each of the following iterations, the generated subset is evaluated with an independent measure. If this measure is better than the previously computed measure, then it is marked as the optimal subset. These steps are iterative and executed until the stopping condition is not satisfied. At last, the optimal subset is identified by the algorithm. The filter based feature selection approach can be designed and implemented by changing the identification method and evaluation metric of the subset. Some algorithms using the filter approach are the sequential feature selection [25], forward selection [26], set cover [27], relief score [28], etc.

Wrappers are supervised feature selection algorithms. The embedded techniques embed feature selection along with a predictive algorithm. It uses a learning algorithm for subset evaluation.

The embedded approach works with the learning algorithm at lower computational cost. It captures the feature dependencies along with the class labels. It examines correlations between features, which are defined as input and output. It checks for efficient features locally that permit improvement. It makes use of the independent measure or selection criteria to select optimal subsets for a defined subset size.

Feature extraction transforms the feature set and extracts relevant transformed set suitable for the analysis model. Transformation methods can be of two types such as linear or nonlinear. The linear methods work on the assumption that the data can be transformed on a lower dimensional linear subspace. Nonlinear methods assume that significant data has non-linear distribution within higher-dimensional space.

3.2 Big Data

The term ‘big data’ refers to data that cannot be processed or analyzed by conventional or traditional processing tools [2]. Organizations today have access to enormous amounts of data in raw form or in semi-structured or unstructured form and hence, are not able to leverage any value from it. Big data technologies are tools, techniques, and architectures intended to extract value from enormous volumes of a variety of data generated at high speed, allowing analysis with insight.

Volume refers to the size of the data. It is a relative term. Small organizations generate and store gigabytes to terabytes of information, whereas global enterprises generate, store, and intend to analyze petabytes to exabytes of data. The volume of the data continues to grow, regardless of the size of the organization. In order to get insight for decision making, companies store all sorts of data related to finance, environment, health, and so on. Organizations analyze this voluminous data to gain competitive advantage and for efficient decision making.

Today data is generated from various sources and in a variety of forms, because of the extensive use of sensors, smart devices, social networking, etc. Hence, the data becomes more complex and is not possible to store in a traditional system. Classifying of data based on type include structured traditional relational data, semi-structured, and unstructured data.

Structured data can be grouped into a relational schema as rows and columns within a standard database. It allows queries and responses to get usable information based on parameters and requirements. Semi-structured data does not conform to an explicit or fixed schema. Such data is self-describing and has markers or tags, which impose hierarchical structure among the records and fields

within the data. Web-logs and social media and RSS feeds are some examples of semi-structured data. Unstructured data cannot be easily indexed or represented into relational tables. Since it does not have any structure, it is difficult for fetching and analyzing. Text, audio, images, and video files are examples of unstructured data.

With big data in mind, the focus is not only on how much data is collected, but also how fast it can be analyzed to get better decisions. The velocity of data is represented as the speed or frequency of the generation of data and knowledge delivery. It is typically considered as how quickly the data arrives, is stored, and how fast it can be retrieved. Velocity signifies data in motion representing the speed at which the data is moving. The increase in information stream sources and sensor network deployments have led to increased flow of data, which need to be analyzed at a faster rate, but which traditional systems fail to do.

Hadoop and Spark are frameworks designed for big data solutions. Hadoop provides a distributed environment for storage and processing of data in a batch mode. It makes use of MapReduce for processing data, which is scalable to deal with more than ten thousand nodes. It makes use of a special form of DAG, which operates with Map and Reduce functions on key/value pairs. It parallelizes the jobs for faster processing using multithreading across the nodes of the cluster.

Apache Spark is a cluster computing, open-source framework. It makes use of in-memory technology. It is a Java virtual machine (JVM)-based fast, distributed, and scalable data processing engine. The Spark framework is designed for real- time data analytics. It is faster in processing voluminous data as it exploits in-memory computations and other optimizations. The Spark has proven to be 100 times faster than MapReduce when it is run using in-memory analytics compared with batch mode, which is only 10 times faster [29,30].

3.3. Submodular Optimization Function

Submodular functions are a class of discrete functions that have diminishing property and studied in mathematics, operation research, and economics. It is formulated and stated as function: $2^p \rightarrow \mathbb{R}$ that returns a real value for any subset $V \subset P$. It is said to be submodular only if $f(M) + f(N) > f(M \cup N) + F(M \cap N)$ where, $M, N \subset P$ and has diminishing returns. Diminishing returns states that $f\ (M\ U\ \{i\}) - f\ (M) \geq f(N\ U\ \{i\}) - f(N)$ where, $M \subseteq N \subset V$ and $i \in V \backslash N$; which indicates marginal gain of adding *i* to set *V*. The submodular function can inherently model notions of coverage, diversity, and information in various applications. It can be optimized extremely by using a simple algorithm such as greedy [12,29]. The formal representation for feature selection using submodular optimization is:

Let

Data [m X n]: m is the number of features and n is the number of observations.

Each observation is the [feature, response] pair.

Model M is built to predict response using features.

Problem: Selecting predictive features to maximize the model fit.

Model selection problem is to minimize the prediction errors measured using the Error Sum of Squares (ESS) as shown in Eq. (1).

$$\underset{\beta}{\text{argmax}}(\text{ESS}(X_\beta) = \|\text{y-}\bar{y}\ \|^2 = \sum_{i=1}^{n}(y_i - \bar{y}_i)^2 \quad (1)$$

where, β indicates the number of features selected for the model.

For choosing features forward, stepwise selection can be used, which uses greedy techniques to get an optimal set of features with the criteria "At each step from the remaining features choose the one which is highly correlated with the predictor variable". The aim of selecting a subset is to maximize the model fit. Model fit is measured by computing the coefficient of determination as shown in Eq. (2).

$$\mathbb{R}^2 = 1 - \frac{ESS(X_S)}{ESS(Y)} \tag{2}$$

X_s is the dataset with feature subset S.

Forward selection works well when adding a feature to the subset, and improves the performance of the model than the sum of performances of the models with individual features. Eqs. (3) and (4) represent the property of the submodular function for improving the model by adding set X_s to X_T.

$$\Delta_T(S) = \mathbb{R}^2(S \cup T) - \mathbb{R}^2(T) \tag{3}$$

If S = A U B then

$$\Delta_T(A) + \Delta_T(B) \geq \Delta_T(S) \tag{4}$$

If (AUB) improves model performance, then Eq. (4) requires that either A or B improve the performance in isolation. This represents the property of the submodular function.

The correlation based feature selection using submodular optimization aims at finding the goodness of a feature for classification by diminishing the joint entropy as the optimization function. It is a supervised approach. The feature is considered to be good if it highly correlates with the class label and least correlates with the other features. A feature subset with higher accuracy for the underlying classification algorithm is considered as the optimal subset of features. The correlation factor is computed using the information theory. Three measures are computed using the equations shown in (5), (6), and (7).

- Entropy: It is a measure of uncertainty of a feature. Entropy is calculated using Eq. (5).

$$E(X) = -\sum P(x_i) \log_2 P(x_i) \tag{5}$$

- Information gain (IG): Knowledge gain obtained for feature X, if the feature Y is already known. The information gain is computed using Eq. (6).

$$IG(X/Y) = E(X) - E(X/Y) \tag{6}$$

- Symmetrical uncertainty (SU): Probability with which one variable can be predicted given other variables. It is computed using Eq. (7).

$$SUF(X,Y) = 2[IG(X/Y)/E(X) + E(Y)] \tag{7}$$

The procedure for forming a subset of features based on information gain is shown in Algorithm 1. δ and η are the thresholds set for selecting the features. The algorithms check for features' correlation with the class label and with the other features. A feature is selected if its correlation value with the class label is higher than δ and its correlation with the other attributes is less than η. After computing information gain, the submodular optimization function is applied to evaluate the performance of the subset.

Submodular Optimization Functions for Feature Selection is formulated as:

Let
F be a set of features F= f1, f2 …….. fn ;
A submodular function $g: 2^F \rightarrow \mathbb{R}$ measures the goodness of feature subset S; (g is a prediction algorithm)
k is the total number of features to be selected.
Optimization problem: $S^* = \underset{s \subseteq F}{argmax}\, g(S)$ subject to $|S| \leq k$
Select the subset that has maximum accuracy in prediction.

Algorithm 1. Subset construction using correlation

INPUT:
Data with all Features, Class Label
OUTPUT: Set F of k features
PROCEDURE:

1. Read Dataset
2. Compute IG(X_i, C)
3. For each feature i
   {
   Compute IG(X_i,C)
   If(IG(X_i,C) ≥ δ) : Add X_i to F;
   }
4. For each feature X_i and X_j in F
   {
   Compute SU(X_i, X_j)
   If(SU(X_i, X_j) ≥ η) : Remove X_i from F;
   }
5. Return F

4. Methodology

This section describes the methodology and architecture used for feature selection using the big data platform, MapReduce approach for submodular based feature selection, logistic regression, and SVM. Fig. 1 shows the methodology followed. The data is collected from secondary sources and then prepared using the preprocessing techniques to remove redundant and missing data. Then the important features are selected using the submodular based feature selection algorithm. The data with only the selected features is further used for the classification model. The data is divided into training data and testing data. The classification models are built using the training data. The efficiency of the models is evaluated using the test data. The evaluation metrics such as accuracy and ROC (receiver operating characteristics) are recorded. The process is again repeated for the next subset of features. All the evaluation metrics are analyzed for various subsets of the feature to select the best subset.

The big data system architecture is used for loan data analysis as shown in Fig. 2. The system architecture is designed using the reference architecture as proposed by Paakkonen and Pakkala [31]. In this, the ellipses indicate the data sources, the rectangles indicate the process, and the arrow indicates the process flow. A dataset is stored in the HDFS. It is then used for the feature selection algorithms. The

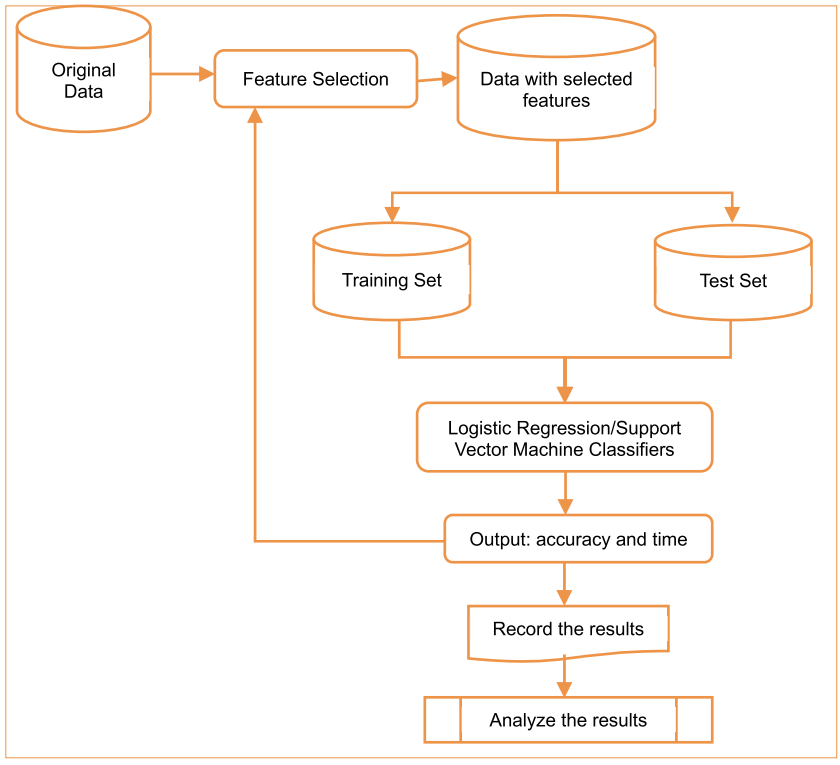

Fig. 1. Methodology used for analysis.

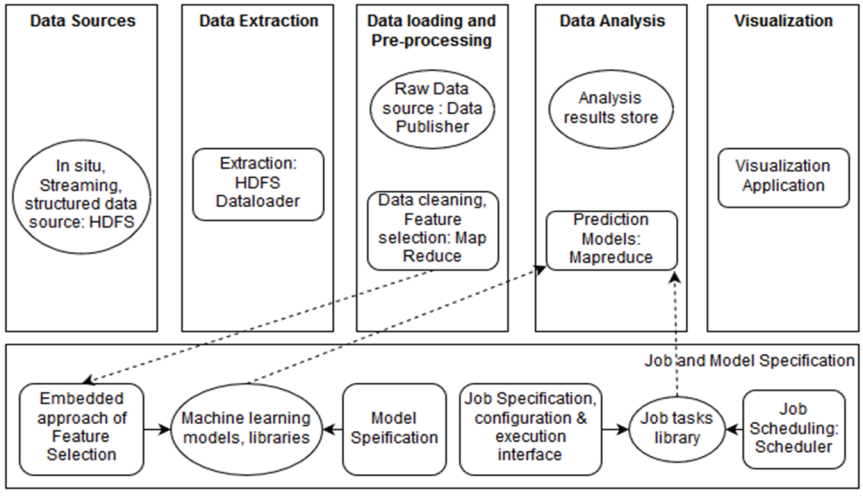

Fig. 2. Big data system architecture for loan data analysis

features selected are stored in a database. The HDFS file is considered as an in situ data store (immobile) and structured data source for the loan data analysis. The HDFS data-loader fetches information from the HDFS file to be stored into a raw data store and the Data publisher. The dataset with the selected features is given as input to build the machine learning classification and the prediction models. These models are tested and the evaluation metrics are recorded. The models are designed and evaluated with

different feature subsets. An analysis of the optimal subset is performed based on the evaluation metrics computed. The results are stored in the database referred to as the Analysis results store. The task configuration and execution interface is set considering the job specification.

The MapReduce job descriptions for computing the entropy of an attribute is shown in Algorithm 2. Consider the dataset (X,y) where, X is the feature vector {x1, x2...xn} and y = {0,1} is the output variable. The dataset is represented as $\{X,y\}^m$ for m rows of data. Two classification models are considered for testing the feature selection algorithm. The logistic regression model returns the probability of y = {0, 1} given x and θ using the hypothesis function given in Eq. (8).

$$h(x, \Theta) = y' = \frac{1}{1+ exp^{(\Theta^T.x)}} \tag{8}$$

The logistic regression model is obtained using the gradient descent algorithm shown in Eq. (9).

Repeat {

$$\theta_j = \theta_j - (\alpha \frac{1}{m} \sum_{i=1}^{M} (h_\theta(x^i) - y^i)\, x_j^i \tag{9}$$

Simultaneously update θ_j for {j = 0…..n}

}

The cost estimation of the model is done using Eq. (10).

$$J(x, \theta) = \sum_{i=1}^{M} \log y_i + \sum_{i=1}^{M} \log(1 - y_i) \tag{10}$$

Algorithm 2. MapReduce E(X) computation

INPUT

Loan data set with m number of samples and n number of features

OUTPUT

E(X) for each feature

MAP FUNCTION

1. Read the loan data file
2. Read each row of the file
3. For each feature i

```
{
  Emit(i_value, (Number of positive samples, Number of negative samples))
}
```

REDUCE FUNCTION

For each feature

1. Compute ps (positive labels)
2. Compute ns (negative labels)
3. ts = ps + ns
4. $E(X_i) = -\{\frac{ps}{ts} \log\left(\frac{ps}{ts}\right) + \frac{ns}{ts} \log\left(\frac{ns}{ts}\right)\}$
5. Emit(i, $E(X_i)$)

The mapper function takes the input file having features, label, and initial values of theta as (X, y, θ). The JobTracker initiates the map tasks at each node. Each map function operates on the local copy of the data chunk and updates the theta values in temp variable using the computations shown in Eq. (11). The

mapper emits these temp values. For multiple iterations, the key value pair emitted from the mapper will be (iteration number, (temp, and count)).

The reducer gets input as (iteration number, iterator (temp, count)). It combines all the temp values and computes the global values of theta as shown in Eq. (12).

$$temp_j^{(i)} = \sum (h_\theta(x^i) - y^i)\, x_j^{(i)} \tag{11}$$

Key (h,θ_j)

$$\theta_j = \theta_j - \frac{1}{\sum count} \sum temp_j \qquad \text{For } all\ j = 0,1 \ldots n \tag{12}$$

The SVM is yet another machine learning model considered for the present study. It is used for classification. The objective of SVM is to train a model by obtaining hyperplanes and support vectors. Hyperplane is a linear partition of the feature space into two categories given by the equation $y_i(W^T x_i + b) \geq 1$. These are used to assigns new objects into a particular class. The optimization problem of SVM is given as:

$$\text{Minimizing } (W^T, b)\{\frac{\|W^T\|^2}{2} + C \sum_i \zeta_i \text{ subject to } y_i(W^T x_i + b) \geq 1 - \zeta_i \text{ for any } i = 0. \ldots n$$

ζ : Slack variable that allows some objects to fall off the margin, and penalizes them

C: The parameter to control the trade-off amongst the slack variable penalty and width of the margin

Support vectors (SV) are the points nearest to the margins of the hyperplane.

The MapReduce algorithm for training the SVM model considers the training data, which is then submitted at the master node. The master node divides the data across all the cluster nodes. To compute the weight and support vectors, the map and reduce functions are instantiated by the job tracker at all the nodes of the cluster. The map job processes the data available at the node and emits localized weight and support vectors. These are aggregated in the reduce function to compute the final weight and support vector for the SVM model. The input and output of the map and reduce functions are:

SVM Mapper input: {X, y}, SVM Mapper output: {key, {W, SV}}

SVM Reducer input: {key, iterator of localized {W, SV}}, SVM Mapper output: {global {W, SV}}

5. Experimental Results

The dataset and methods used for the experimentation, software and hardware configurations, and the results obtained are discussed in this section.

5.1 Dataset and Methods

An empirical study was performed using the loan classification dataset to evaluate the effect of the submodular optimization based feature selection algorithm. The experiments were carried out on the loan datasets having approximately more than two lakh instances with nearly eight hundred features. The sample features are amount of the loan, account number, number of payments, employment status, purpose, designation, delinquency, house type, and others. The main aim is to predict whether the performance of the loan is good or bad. This prediction helps in analyzing whether the loan will turn into

a NPA or not. Firstly, attributes having almost all zeros are removed. Then, the feature selection algorithm is applied and 'k' best features on the remaining dataset are obtained. The effectiveness of the feature subset is tested using classification algorithms such as Logistic Regression and SVM. The performance of the classification models with reduced dimensions is analyzed.

The metrics used for evaluating the performance of the classifiers with different subset of features are:

(i) Area under the curve (AUC): It is the area under the ROC curve. It is computed using Eq. (13).

$$AUC = \frac{TPR+TNR}{2} \tag{13}$$

where, TPR is true positive rate and TNR is true negative rate.

(ii) Runtime for training: Time measured in seconds to build the classifier using the training data. Test runtime is much less compared with the training time as the classifier is already built. Hence, only the model training runtimes are shown.

5.2 Hardware and Software Used

The experiments are conducted on Hadoop cluster with twenty computing nodes. Each node has the configuration as specified below:

(i) Processor: Intel Core i7-4790
(ii) Clock speed: 3.3 GHz.
(iii) Cores: 4 per processor
(iv) Cache: 8 MB
(v) RAM: 8 GB
(vi) Network: InfiniBand

The master node in Hadoop hosts the NameNode and the JobTracker. The Namenode manages the storage component of the Hadoop and the HDFS. It coordinates with the slave nodes by means of their respective DataNode processes. The JobTracker manages the processing in each compute node through the TaskTrackers. The TaskTrackers execute the MapReduce jobs in each node. The same configuration is used by Spark. It has a master process and worker processes. The master process is present on the master node and is responsible for managing the cluster and assigning the jobs. The worker nodes reside in the slave nodes. The worker processes execute the jobs on the slave machines. Both of these frameworks work on the HDFS. The software used for the experiments were the Hadoop 2.6.1-MapReduce 2 with Apache Spark 2.2.0 on the Ubuntu 14.04 operating system. The MapReduce configuration has 48 max maps tasks and 20 max reducer tasks.

5.3 Results and Discussion

In this section, the results of CFS-SO (correlation based feature selection using the submodular optimization) with classification models and its comparison with the existing algorithm is discussed.

5.3.1 Result analysis of the CFS-SO algorithm

The experiments are conducted to form different subsets using CFS-SO. These subsets are used for the classification task. The algorithms used are the logistic regression (LR) and the SVM. The subset of

features obtained are evaluated using these classification algorithms. With CFS-SO approach, the sample features and information gain are obtained as shown in Table 1. The distribution of information gain for 60 features is shown in Fig. 3.

Table 1. Information gain

Sl. No.	Attribute No.	IG
1	632	0.003358
2	532	0.002738
3	14	0.002684
4	758	0.002513
5	7	0.002493
6	143	0.002458
7	593	0.002402
8	278	0.002398
9	69	0.002395
10	557	0.002383

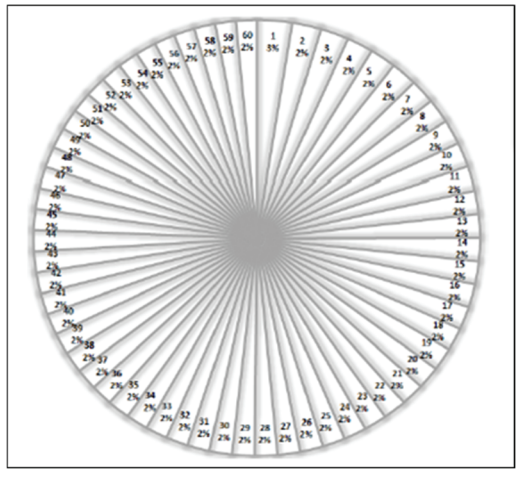

Fig. 3. Distribution of information gain for 60 features.

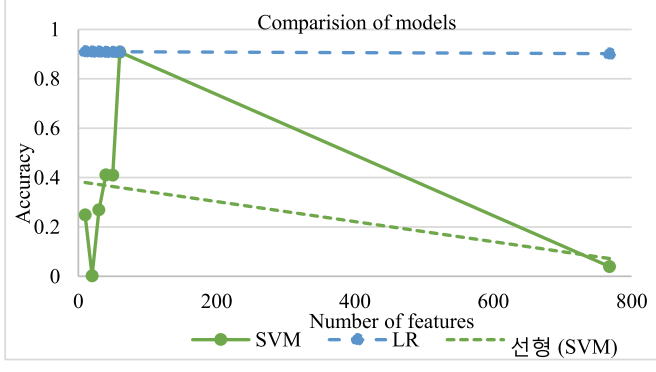

Fig. 4. Comparison of SVM and LR models with different numbers of features using CFS-SO.

Fig. 4 shows that LR consistent performance compared with the SVM. However, the SVM trend line shows declining performance as the number of attributes increase. The SVM kernel function transforms the data into higher dimensional space so that the data becomes linearly separable for the classification task. When the data is high dimensional with many features, it is difficult to clearly define linear separability and hence, the performance of the SVM declines.

The ROC graphs obtained for the SVM are depicted in Fig. 5. It shows that for a subset with 60 features, the performance of the SVM is good, but for others it does not exhibit good performance. Hence, for the SVM, the subset with sixty features is an optimal subset. The ROC graphs obtained for LR are shown in Fig. 6. It shows consistent performance for all the subsets, except for the subset with features, which indicate variation in specificity and sensitivity. Hence, the SVM shows good performance when 60 top features are considered. The LR presents stable performance for all the subsets.

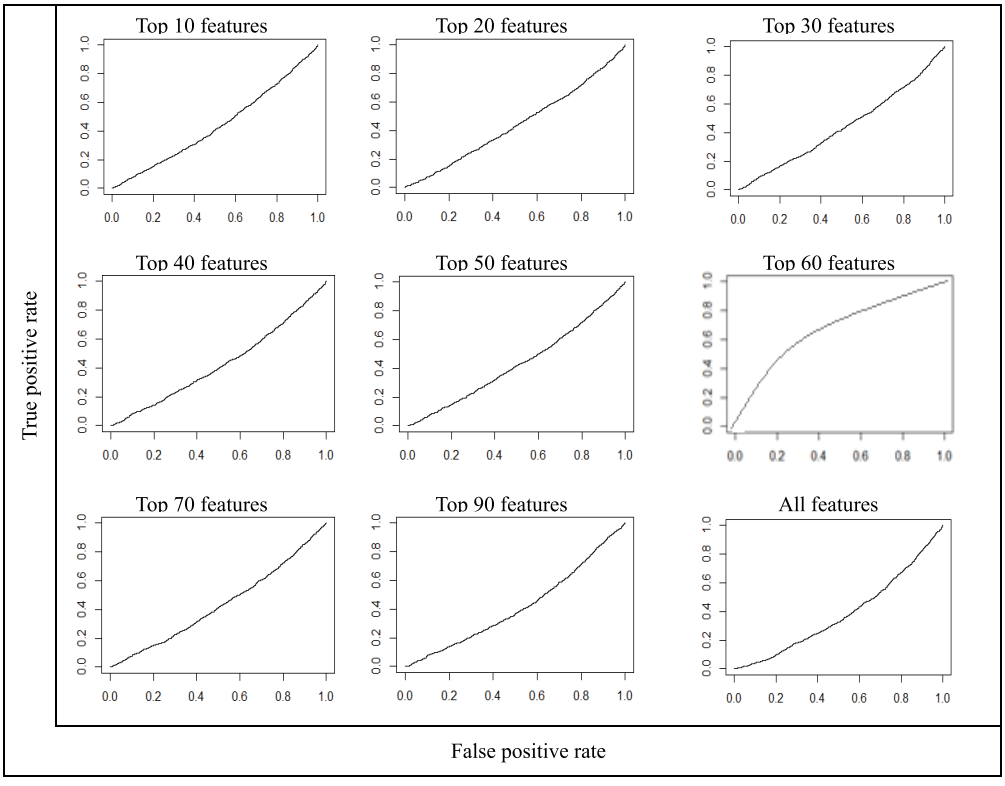

Fig. 5. Comparison of ROC curves for SVM model with different numbers of features.

The time complexity of the feature selection algorithm is $\Theta(n^2)$. The complexity of building LR model is $\Theta(mnp)$, where, m is the number of rows, n is the number of features, and p is the number of iterations until convergence. The time complexity for model building for SVM [32] is $O((mn)^3)$. Hence, as the features decrease, the time also decreases, at least linearly. It is also observed empirically that when all the features are considered, the time taken is very high, and decreases significantly when the features are reduced. With the experiments conducted, it can be inferred that reducing the dimension does not deter the performance of the classification algorithms when an optimal subset of features is considered. It also shows exponential improvement in computational time as shown in Fig. 7.

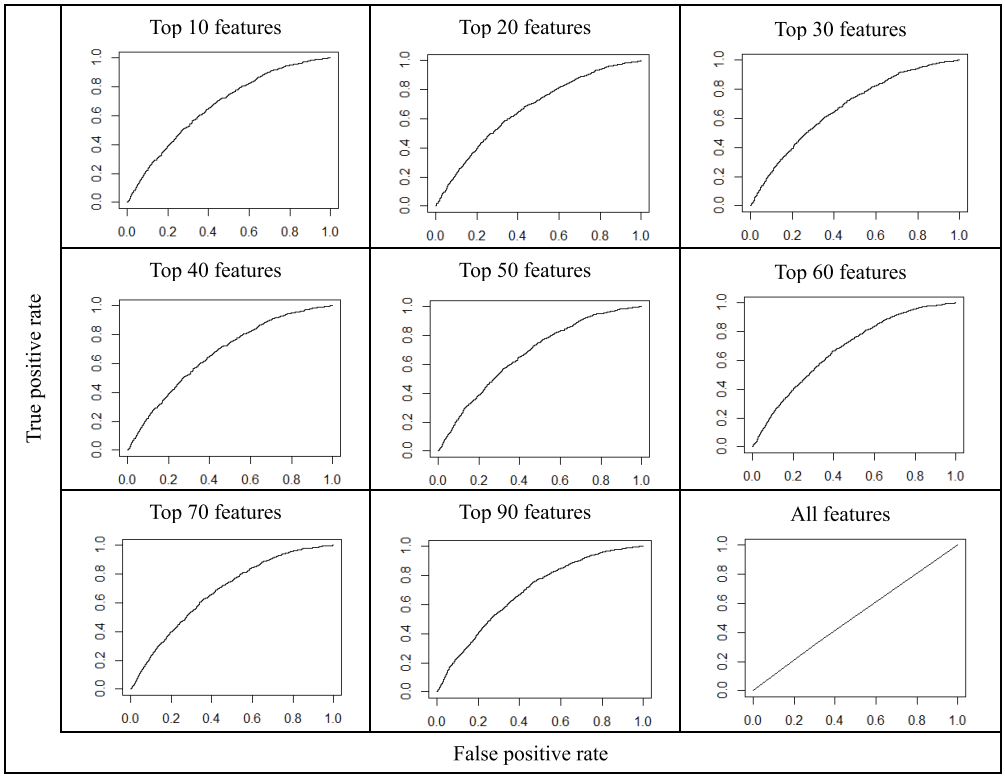

Fig. 6. Comparison of ROC curves for LR model with different numbers of features.

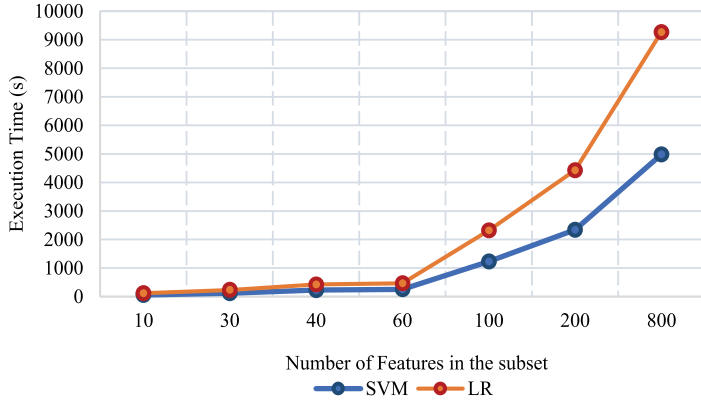

Fig. 7. Execution time comparison for LR and SVM models for different sets of features.

5.3.2 Comparison of CFS-SO, LASSO, and PCA algorithms

The experiment is repeated to identify the optimal subset leading to significant improvement in the accuracy of the model. The results in Table 2 show that the top 60 best optimal features provide 91% accuracy in both the LR and SVM algorithms compared considering all the features. In order to carry out comparative analysis with the existing standard methods of feature selection, least absolute shrinkage and selection operator (LASSO) [33,34] and feature extraction algorithm Principal Component Analysis

(PCA) were implemented on the loan data set. The LASSO works on the principle of fitting a linear equation which best suits for computing the class variable. In the process, the features which get higher coefficients can be considered as features of high importance. In order to reduce the dimension of the feature set, the PCA transforms the given feature set on to another feature space using the eigen-values and the eigen-vectors of the mean adjusted input feature set. Hence, after transformation, it is difficult to know which component maps to which feature. Thus, the importance of the variables or features in the original data set cannot be analyzed. The CFS-SO allows the analysis of the importance of individual features with respect to the predictor variable effectively.

Table 2. Performance comparison of classification models with CFS-SO, LASSO, and PCA feature selection and extraction algorithms

Feature selection algorithm	Classification algorithm	Number of features/Principal components						
		10	**20**	**30**	**40**	**50**	**60**	**All**
PCA	SVM	0.40	0.40	0.56	0.67	0.912	0.921	0.908
	LR	0.068	0.08	0.87	0.82	0.913	0.919	0.808
LASSO	SVM	0.50	0.59	0.59	0.63	0.82	0.86	0.04
	LR	0.64	0.634	0.64	0.64	0.84	0.81	0.90
CFS-SO	SVM	0.25	0.0028	0.27	0.411	0.41	0.91	0.04
	LR	0.911	0.91	0.91	0.91	0.909	0.90	0.90

With the PCA, both the SVM and the LR work fine for principal components numbering more than 40. With CFS-SO, the LR shows better performance, whereas the SVM performs better for the top 60 features. With the LASSO, the LR outperforms the SVM. It can be inferred from both the CFS-SO and the LASSO that for the considered dataset, linear correlation with the predictor variable suits well and hence, the LR shows better performance. Considering the CFS-SO and LASSO, it can be observed that CFS-SO performs better.

The best accuracies obtained for LR and SVM after applying PCA, LASSO, and CFS-SO are shown in Fig. 8. The accuracies obtained without applying the feature selection considering all the features is shown in Fig. 9. These figures indicate the importance of the CFS-SO feature selection for improvising the accuracy of the prediction models.

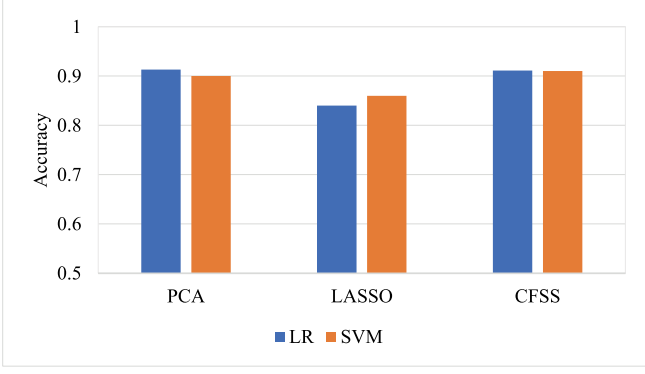

Fig. 8. Best accuracies obtained after applying PCA, LASSO, and CFS-SO.

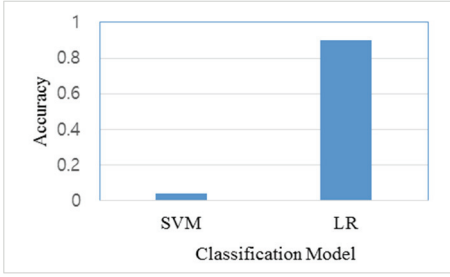

Fig. 9. Accuracies obtained without applying feature selection.

6. Conclusion and Future Work

The prediction models for financial health play a key role in controlling financial irregularities in a country's economy. It is significant to employ such prediction models leveraging all the data available in the financial domain. Important predictions would be to know financial failure condition, fraudulent activities, bankruptcy and NPA, etc. The data for such models is not readily available as it contains noisy and redundant features as well. Hence, the data needs to be pre-processed and a right representative set of the data should be prepared. The present work focuses on understanding the suitability of the correlation based method using submodular optimization for the selection of features on voluminous data. First, the data is preprocessed by handling null values and converting the categorical data to numerical data. Then, the right subset of features is identified, which aids in predictive analysis of bad loans. The performance of the prediction algorithm is used as the evaluation metric for choosing the right subset. The experimental results show that subsets with optimal number of features do not deter the performance of the classification models. Such feature sets reduce the computational time exponentially. The performance comparison of the classification models with CFS-SO, LASSO, and PCA algorithms indicate that models with CFS-SO perform better than with LASSO. The CFS-SO can be chosen over PCA to retain the original input features.

The big data technology is used to exercise the relevance of the approach for the problem addressed. Big data preprocessing for improvement of predictive modelling is an essential step towards prediction of financial data analytics for NPA and fraud. Any future work can comprise of building of suitable predictive models for fraud detection utilizing pre-processed data using the above proposed approach.

References

[1] T. Seth and V. Chaudhary, "Big data in finance," in *Big Data: Algorithms, Analytics, and Applications*. Boca Raton, FL: CRC Press, 2015, pp. 329-356.

[2] I. Taleb, R. Dssouli, and M. A. Serhani, "Big data pre-processing: a quality framework," in *Proceedings of 2015 IEEE International Congress on Big Data*, New York, NY, 2015, pp. 191-198.

[3] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature selection: a data perspective," *ACM Computing Surveys*, vol. 50, no. 6, article no. 94, 2018.

[4] B. Arguello, "A survey of feature selection methods: algorithms and software," PhD dissertation, University of Texas at Austin, TX, 2015.

[5] A. Krause, "SFO: a toolbox for submodular function optimization," *Journal of Machine Learning Research*, vol. 11, pp. 1141-1144, 2010.

[6] M. A. Fattah, "A novel statistical feature selection approach for text categorization," *Journal of Information Processing Systems*, vol. 13, no. 5, pp. 1397-1409, 2017.

[7] K. Kira and L. A. Rendell, "A practical approach to feature selection," in *Machine Learning Proceedings 1992*. St. Louis, MO: Elsevier, 1992, pp. 249-256.

[8] S. Fallahpour, E. N. Lakvan, and M. H. Zadeh, "Using an ensemble classifier based on sequential floating forward selection for financial distress prediction problem," *Journal of Retailing and Consumer Services*, vol. 34, pp. 159-167, 2017.

[9] E. Wright, Q. Hao, K. Rasheed, and Y. Liu, "Feature selection of post-graduation income of college students in the United States," 2018; https://arxiv.org/abs/1803.06615.

[10] S. D. Kim, "A feature selection technique based on distributional differences," *Journal of Informaion Processing System*, vol. 2, no. 1, pp. 23-27, 2006.

[11] S. Maldonado, J. Perez, and C. Bravo, "Cost-based feature selection for support vector machines: an application in credit scoring," *European Journal of Operational Research*, vol. 261, no. 2, pp. 656-665, 2017.

[12] A. Krause and V. Cevher, "Submodular dictionary selection for sparse representation," in *Proceedings of the 27th International Conference on Machine Learning (ICML)*, Haifa, Israel, 2010, pp. 567-574.

[13] Y. Bar, I. Diamant, L. Wolf, S. Lieberman, E. Konen, and H. Greenspan, "Chest pathology identification using deep feature selection with non-medical training," *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, vol. 6, no. 3, pp. 259-263, 2018.

[14] R. Iyer, S. Jegelka, and J. Bilmes, "Fast semidifferential-based submodular function optimization," *Proceedings of the 30th International Conference on Machine Learning (ICML)*, Atlanta, GA, 2013, pp. 855-863.

[15] K. Wei, Y. Liu, K. Kirchhoff, and J. Bilmes, "Using document summarization techniques for speech data subset selection," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Atlanta, GA, 2013, pp. 721-726.

[16] A. Krause and C. Guestrin, "A note on the budgeted maximization of submodular functions," Carnegie Mellon University, *Technical Report No. CMU-CALD-05-103*, 2005.

[17] D. Kempe, J. Kleinberg, and E. Tardos, "Maximizing the spread of influence through a social network," in *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Washington, DC, 2003, pp. 137-146.

[18] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions - I," *Mathematical Programming*, vol. 14, no. 1, pp. 265-294, 1978.

[19] M. A. Hall, "Correlation-based feature selection for machine learning," PhD dissertation, The University of Waikato, Hamilton, New Zealand, 1999.

[20] A. Pouramirarsalani, M. Khalilian, and A. Nikravanshalmani, "Fraud detection in E-banking by using the hybrid feature selection and evolutionary algorithms," *International Journal of Computer Science and Network Security*, vol. 17, no. 8, pp. 271-279, 2017.

[21] Y. Wang, W. Ke, and X. Tao, "A feature selection method for large-scale network traffic classification based on spark," *Information*, vol. 7, article no. 6, 2016.

[22] H. D. Gangurde, "Feature selection using clustering approach for big data," *International Journal of Computer Applications*, vol. 2014, no. 4, pp. 1-3, 2014.

[23] P. Sarlin, "Data and dimension reduction for visual financial performance analysis," *Information Visualization*, vol. 14, no. 2, pp. 148-167, 2015.

[24] H. S. Bhat and D. Zaelit, "Forecasting retained earnings of privately held companies with PCA and L1 regression," *Applied Stochastic Models in Business and Industry*, vol. 30, no. 3, pp. 271-293, 2014.

[25] I. Pisica, G. Taylor, and L. Lipan, "Feature selection filter for classification of power system operating states," *Computers & Mathematics with Applications*, vol. 66, no. 10, pp. 1795-1807, 2013.

[26] H. Liu and H. Motoda, *Feature Selection for Knowledge Discovery and Data Mining.* New York, NY: Springer Science & Business Media, 2012.

[27] M. Dash, "Feature selection via set cover," in *Proceedings 1997 IEEE Knowledge and Data Engineering Exchange Workshop*, Newport Beach, CA, 1997, pp. 165-171.

[28] A. Arauzo-Azofra, J. M. Benitez, and J. L. Castro, "A feature set measure based on relief," in *Proceedings of the 5th International Conference on Recent Advances in Soft Computing*, Nottingham, UK, 2004, pp. 104-109.

[29] X. Meng, J. Bradley, B. Yavuz, E. Sparks, S. Venkataraman, D. Liu, et al., "MLlib: machine learning in Apache Spark," *The Journal of Machine Learning Research*, vol. 17, pp. 1-7, 2016.

[30] K. Noyes, "Five things you need to know about Hadoop v. Apache Spark," 2015; https://www.infoworld.com/article/3014440/five-things-you-need-to-know-about-hadoop-v-apache-spark.html.

[31] P. Paakkonen and D. Pakkala, "Reference architecture and classification of technologies, products and services for big data systems," *Big Data Research*, vol. 2, no. 4, pp. 166-186, 2015.

[32] A. Abdiansah and R. Wardoyo, "Time complexity analysis of support vector machines (SVM) in LibSVM," *International Journal Computer and Application*, vol. 128, no. 3, pp. 28-34, 2015.

[33] J. Giersdorf and M. Conzelmann, "Analysis of feature-selection for LASSO regression models," 2017; https://www.ni.tu-berlin.de/fileadmin/fg215/teaching/nnproject/Lasso_Project.pdf.

[34] V. Fonti and E. Belitser, "Feature selection using lasso," VU Amsterdam Research Paper in Business Analytics, 2017; https://beta.vu.nl/nl/Images/werkstuk-fonti_tcm235-836234.pdf

Girija Attigeri https://orcid.org/0000-0002-0899-5781

She is currently Assistant Professor (Selection Grade) in the Department of Information and Communication Technology, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, India. She has received B.E. and M.Tech. degrees from the Visvesvaraya Technological University, Karnataka, India. She has 12 years of experience in teaching and research.

Manohara Pai M. M. https://orcid.org/0000-0003-2164-2945

He is a Professor and Associate Director of Research and Consultancy at the Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal. He has experience of 26 years in research, academics, and industry. He has received his Ph.D. from the University of Mysore at Karnataka, India. His research interests span big data analytics, wireless sensor networks, internet of things, cloud computing, and intelligent transportation system. He has publications in reputed international conferences and journals. He has six patents to his name and has authored two books. He has supervised four PhD and 80 plus post-graduate students. He was visiting professor of ESIGELEC-IRSEEM at the University of Rouen, France. He is the investigator for several projects funded by the Government of India and by various industries. He is IEEE Senior member and Chair of IEEE, Mangalore Subsection.

Radhika M. Pai https://orcid.org/0000-0002-0916-0495

She is a Professor at the Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, in the Department of Information and Communication Technology. She has experience in research, academics, and industry of about 25 years. She has received her Ph.D. from the National Institute of Technology, Karnataka, India. Big data analytics, database systems, data mining and warehousing and operating systems are her major research interests. She has publications in reputed international conferences and journals. She has received grants from the Government of India.