Publication pour la présente candidature LAFPT

Titre	A Large-Scale Filter Method for Feature Selection Based on Spark
Auteurs	Reine Marie Ndéla Marone ; Fodé Camara ; **Samba Ndiaye**
Référence	2017 IEEE 4th International Conference on Soft Computing & Machine Intelligence (ISCMI)
Editeur	IEEE
Pages	16 - 20
Année	2018
DOI	10.1109/ISCMI.2017.8279590
URL	https://ieeexplore.ieee.org/document/8279590
Index	https://www.scopus.com/authid/detail.uri?authorId=6701604512
ISBN	978-1-5386-1314-6
Encadreur	Oui
Extrait d'une thèse	Oui

IEEE.org | IEEE Xplore Digital Library | IEEE-SA | IEEE Spectrum | More Sites
Cart (0) | Create Account | Personal Sign In

IEEE Xplore® Digital Library
> Institutional Sign In
IEEE

Browse My Settings Get Help Subscribe

All
Enter keywords or phrases (Note: Searches metadata only by default. A search for 'smart grid' = 'smart AND grid')
Advanced Search | Other Search Options

A large-scale filter method for feature selection based on spark

3 Author(s) Reine Marie Marone ; Fodé Camara ; Samba Ndiaye View All Authors

58
Full
Text Views

Abstract:
Recently, enormous volumes of data are generated in information systems. That's why data mining area is facing new challenges of transforming this "big data" into useful knowledge. In fact, "big data" relies low density of information (low data quality) and data redundancy, which negatively affect the data mining process. Therefore, when the number of variables describing the data is high, features selection methods are crucial for selecting relevant data. Features selection is the process of identifying the most relevant variables and removing those are redundant and irrelevant. In this paper, we propose a parallel, scalable feature selection algorithm based on mRMR (Max-Relevance and Min-Redundancy) in Spark, an in-memory parallel computing framework specialized in computation for large distributed datasets. Our experiments using real-world data of high dimensionality demonstrated that our proposition scale well and efficiently with large datasets.

Published in: 2017 IEEE 4th International Conference on Soft Computing & Machine Intelligence (ISCMI)

Date of Conference: 23-24 Nov. 2017

Date Added to IEEE *Xplore*: 05 February 2018

ISBN Information:
Electronic ISBN: 978-1-5386-1314-6
DVD ISBN: 978-1-5386-1313-9
Print on Demand(PoD) ISBN: 978-1-5386-1315-3

INSPEC Accession Number: 17560792

DOI: 10.1109/ISCMI.2017.8279590

Publisher: IEEE

Conference Location: Port Louis, Mauritius

More Like This

Temporal Multiple Correspondence Analysis for Big Data Mining in Soccer Videos
2015 IEEE International Conference on Multimedia Big Data
Published: 2015

Reducing Data Complexity in Feature Extraction and Feature Selection for Big Data Security Analytics
2018 1st International Conference on Data Intelligence and Security (ICDIS)
Published: 2018

View More

IEEE
See the top organizations patenting in technologies mentioned in this article
ORGANIZATION 4
ORGANIZATION 3
ORGANIZATION 2
ORGANIZATION 1
Click to Expand >
Provided by: InnovationQ PLUS
POWERED BY IEEE AND IP.COM
A PATENT SEARCH AND ANALYTICS TOOL

2017 4th IEEE International Conference on Soft Computing and Machine Intelligence

A Large-Scale Filter Method for Feature Selection Based on Spark

Reine Marie Marone[1], Fodé Camara[2], Samba Ndiaye[1]
[1] Departement of mathematics, Cheikh Anta Diop University, Dakar, Senegal
[2] Department of mathematics, Alioune Diop University, Bambey, Senegal
e-mail: fode.camara@uadb.edu.sn, reine.marie.marone@ucad.edu.sn

Abstract— **Recently, enormous volumes of data are generated in information systems. That's why data mining area is facing new challenges of transforming this "big data" into useful knowledge. In fact, "big data" relies low density of information (low data quality) and data redundancy, which negatively affect the data mining process. Therefore, when the number of variables describing the data is high, features selection methods are crucial for selecting relevant data. Features selection is the process of identifying the most relevant variables and removing those are redundant and irrelevant. In this paper, we propose a parallel, scalable feature selection algorithm based on mRMR (Max- Relevance and Min-Redundancy) in Spark, an in-memory parallel computing framework specialized in computation for large distributed datasets. Our experiments using real-world data of high dimensionality demonstrated that our proposition scale well and efficiently with large datasets.**

Keywords-feature selection, filter method, parallel computing, apache spark, mRMR, SVM

I. INTRODUCTION

Feature selection is very important task in data mining that tries to remove irrelevant and redundant features from original data [1]. It is widely used in many applications such as genes selection, anomaly detection, pattern recognition and many others fields. For example, in anomaly detection, feature selection permits to identify the most relevant features contained by a network packet and decreases the time taken to classify network packets either as normal or anormalous [2].

Unfortunately, as large-scale datasets are usually adopted nowadays, most existing feature selection algorithms do not scale well, and their efficiency significantly deteriorates or even becomes inapplicable [3]. Parallel processing can help alleviate this problem, effectively allowing users to work with Big Data [3]. Efficient distributed programming frameworks, such as MapReduce [3] along with its open-source implementation Apache Hadoop, have been proposed to manage the problem of Big Data. However the MapReduce parallel programming with Apache Hadoop causes very high I/O overhead for iterative computations because it is a disk-based model [4]. Then, Apache Hadoop is not suited for the features selection algorithms, which need iterative computation [4]. More recently, Apache Spark [4] has been presented as an alternative to Hadoop and is designed to overcome the disk I/O limitations and improve the performance of large-scale data processing [4].

That is the reason why in this paper, we propose a parallel, scalable feature selection method based on mRMR algorithm, that we call PSF-mRMR (for Parallel Scale Filter method based on mRMR), on the shared memory parallel environment Spark to improve it performance. Our experimental results demonstrated that the proposed algorithm can scale well and efficiently process large datasets.

The rest of the paper is organized as follows. Section II discusses related works. In section III, we formulate the problem.

Section IV gives the details of our proposition. Section V presents our algorithm and Section VI the working environment. In Section VII, we evaluate the performance of our algorithm. Section VIII concludes the paper and gives some future works.

II. RELATED WORKS

Feature selection is a dimensionality reduction method that aims to choose a subset of relevant features that has the lowest dimension and describes properly a given problem with minimum performance degradation.

In general feature selection methods can be classified into 3majors categories: Filter, Wrapper and embedded [5].

In the wrapper methods the "usefulness" of a subset of features is evaluated on the basis of the classifier performance [5].

Embedded methods exploit intrinsic characteristics of a given model to guide the feature selection process, and choose features which best contribute to the accuracy performance of the model [5].

In Filter methods features are selected on the basis of characteristics, which determine their relevance or discriminant powers with the outcome variable [5, 6]. Filter methods offers better computational complexity but do not take account the interactions among the variables, which cannot be ignored. Although many faster filter methods based on information theory, specifically mutual information and svm feature weights to mathematically evaluate the relevance and redundancy of data have been proposed in literature, optimizing their implementation through efficient parallelization is also crucial for challenge ultrahigh dimensional issues in big data [7]. This triggered researchers to exploit parallelism within feature selection algorithms in order to improve modeling task (prediction, recognition, classification), decrease the training time, and develop generalization through overfitting. Many filters methods algorithms have been implemented on Spark improving both the classification accuracy and its runtime when dealing with big data problems.

In [1] authors parallelize a broad group of well-known information theory-based methods in Apache Spark.

Experimental results for a broad set of real-world datasets point to competitive performance (in terms of generalization and efficiency) when dealing with ultra-high-dimensional datasets that are huge in terms of both number of features and instances.

The work in [8] proposes a toolkit named Manchester AnalyticS Toolkit (MAST), which provides an efficient, parallel and scalable implementation of feature selection techniques, based on information theory. MAST is able to process a dataset of 100 million examples and 100,000 features in under 10 minutes on a four socket server which each socket containing an 8-core Intel Xeon E5-4620 processor.

Authors in [9] propose a filter feature selection algorithm based on evolutionary computation. This method uses the MapReduce paradigm to obtain subsets of features from big datasets. The algorithm implemented on the framework spark, decomposes the original dataset in blocks of instances and learn from them in the map phase; then, in the reduce phase the obtained partial results are merged into a final vector of feature weights; a threshold is used to determine the selected subset of features. The experiments show that, this algorithm improves both the classification accuracy and its runtime when dealing with big data problems.

In [10], authors proposes an efficient feature selection method FSMS for network traffic based on Spark. In this method, the complete feature set is firstly preprocessed based on Fisher score, and a sequential forward search strategy is employed for subsets. The Spark computing framework along with continuous iterations then selects the optimal feature subset. This method significantly reduces the modeling and classification time for the classifier.

What comes out is that the methods presented in the state of art deal with the complex iterative computations because many of them include iteratively one or many features into a feature subset.

Unlike to these methods our algorithm select a subset of relevants and non redundant features in only one single pass which permit us to reduces more significantly the learning time while keeping a good classification accuracy.

III. Problem Definition

We address two-class classification problems, the target class label $l \in \{0, 1\}$. F is the given feature set $\{f_1,..,f_p\}$. An instance X is denoted by a p-dimensional vector $(x_1,..,x_p)$, where x_j is denoted the value of the feature f_j of X. Let $J(E, T)$ be the objective function which evaluates the subset E of F using the data T. The subset E_1 is better than E_2 if $J(E_1, T) > J(E_2, T)$.

In this paper, we assume p so large as in the big data context, and we proposed a large-scale filter method: PSF-mRMR for Parallel Scale Filter method based on mRMR (Maximum Redundancy and Maximum Relevancy). We used the well-known parallel computing framework, Apache Spark™, to implemente the algorithm.

IV. Our Propasal

Several algorithms like mRMR have been proposed in the literature in order to maximize the relevancy of a feature subset and minimizing the redundancy among the features.

A. mRMR

mRMR is a method that aim to maximizes the relevancy of features with the target label l while minimizing the redundancy between features [11]. Let f_i and f_j be two variables in F. $MI(f_i, f_j)$ represents the measure of mutual information between the variables f_i and f_j. $MI(l, f_i)$ denotes the measure of mutual information between the class label l and f_i.

The redundancy of a feature subset is determined by the mutual information among the features. The redundancy among the variables in F is given by

$$W_I(F) = \frac{1}{|F|^2} \sum_{f_i, f_j \in F} MI(f_i, f_j) \tag{1}$$

The relevance of the variables in F with respect to l is computed as

$$V_I(F) = \frac{1}{|F|} \sum_{f_i \in F} MI(l, f_i) \tag{2}$$

The maximally relevant and minimally redundant set of feature S^* among all sets S in F is obtained by optimizing the conditions in equations (1) and (2) as follows:

$$S^* = \arg\max_{S \subseteq F} [V_I(F) - W_I(F)] \tag{3}$$

B. Our Method (PSF-MRMR)

In the litterature many experiments show that a feature ranking using weights from linear SVM (support vector machines) models gives good performances, even when the training and testing data are not identically distributed [12]. For this reason, in our method PSF-mRMR, we use the ranking measure proposed by the authors in [13], which combine linear support vector machines and the mRMR criterion to rank the features for better's results. Let $\beta \in [0, 1]$ determines the tradeoff between SVM ranking and mRMR ranking, $R_{F,i}$ the relevancy of feature i in the set F on classification given by

$$R_{F,i} = \frac{1}{|F|} \sum_{l} MI(l, i) \tag{4}$$

And $Q_{F,i}$ the redundancy of feature i in the set F on classification given by

$$Q_{F,i} = \frac{1}{|F|^2} \sum_{i' \in F, i' \neq i} MI(i,i') \qquad (6)$$

Let ω_i denotes the SVM weight of the attribute *i*.

For *i*-th feature, the ranking measure r_i is given by

$$r_i = \beta|\omega_i| + (1-\beta)\frac{R_{F,i}}{Q_{F,i}} \qquad (5)$$

- **Dataset Format**

The input dataset is in the *libsvm* format, in others words for each instance *j* we first have the label l_i which takes either the value 0 or the value 1, then we have an attribute a_i which can appear followed of two points (**:**) and its value v_j^i for this instance *Ij*. Some features may not appear in some instances when the dataset is sparse.

Initial Dataset
l_1 a_1: v_1^1 ..a_n: v_n^1
..
l_m a_1: v_1^m ..a_n: v_n^m

Figure 1. Initial dataset in libsvm format

where *n* and *m* represent respectively the number of features and the number of instances in the dataset.

To make our algorithm less expensive in term of time consuming and more efficient, we transform the dataset into the following format:

Transformed Dataset
v_1^1 .. v_n^m
..
v_n^1 .. v_n^m

Figure 2. Transformed Dataset

In the new dataset obtained after transformation, we have for each instance *Ij* and each attribute a_i the value v_i^j.

Our algorithm works mainly with the transformed dataset. The initial dataset is used primarily to return a set of attributes in the libsvm format for the step of classification.

V. OUR ALGORITHM

Our proposed algorithm, called PSF_mRMR**,** is a feature selection method based on Spark, a parallel programming framework. Let *D* denote the input dataset (with *n* features and *m* instances) and *K* the number of features to return. Let β be the tradeoff between SVM ranking and mRMR ranking**,** and *p* number of partitions for the dataset**.** *F* denotes the feature space. The output *D'* will be the optimal subset of *K* attributes with max r_i score.

Our algorithm can be broken into six steps:

- **Step 1: distribute features among the worker**

In this stage, a set of values of each feature a_i in each instance *Ij* is constructed. This is done with the following statement:

1. Construct *values*={{ v_i^1 ,.., v_i^m }, i=1 to n }

Then, the feature space *F* is decomposed into blocks of features executed in parallel on each worker node. This corresponds to the following statements:

2. Create *p* sets of feature subspace sub_w, *w = 1..p* from the entire feature space *F*.
3. Each sub_w will be send to a unique worker (between the *p* workers).

- **Step 2: associate features and labels**

Each feature of each block will be associated with each other feature of the entire space of features F in order to compute the mutual information between them. This is done simultaneously on the workers by creating for each attribute a_i of each block several sets.

Let { v_i^1 ,.., v_i^m } be the set of values of a_i in each instance of the dataset(these values are directly accessible in the transformed dataset), { v_j^1 ,.., v_j^m } is the set of values in each instance for a_j and { l_1 ,.., l_m } is the set of the class labels.

For each feature a_i on a block and for each other feature a_j of the entire space of features F, map a_i as follows:

a_i=>{ a_i, { v_i^1 ,.., v_i^m }, { v_j^1 ,.., v_j^m } , { l_1 ,.., l_m }}

We call the set consisting of the {a_i, { v_i^1 ,.., v_i^m }, { v_j^1 ,.., v_j^m }, { l_1 ,.., l_m }}obtained *r2sub*.

- **Step 3: Calculates the mutual information between feature and class label**

In this stage, we use the sets obtained in the previous step for each feature a_i in order to calculate its mutual information M_{ij} with another feature a_j but also its mutual information R_i with the class label. We then obtain a new set that we called *r3sub*. Each element in *r3sub* consisting of a feature, its mutual information with another feature and also its mutual information with the class label. This correspond to the following instructions:

Foreach element *el* $\subset$ *r2sub*

1. *rdd* [(a_i, M_{ij} , R_i)] = *mapToPair* (*el*=>{ a_i, M_{ij} , R_i })

2. $\mathrm{M}_{ij} = MutualInformation\,(\{v_i^1,..,v_i^m\},\{v_j^1,..,v_j^m\})$

3. $R_i = MutualInformation\,(\{v_i^1,..,v_i^m\},\{l_1,..,l_m\})/n$

where $l_{k\in 1..m}$ *represents the label of class in instance k.*

EndForeach

- **Step 4 : for each feature, sum mutual information with others features**

In the 4th step (reduce step) for each feature a_i, algorithm sums its mutual information with the other features of the dataset (in order to obtain the redundancy), while keeping the mutual information with the class label (for the relevance). A new set is then obtained and we call it *r4sub*. Each element in r4sub consisting of $\{a_i, sumM_{ij}, R_i\}$, where a_i is the feature, $sumM_{ij}$ is the sum of mutual information between a_i and the other features of the space of features F and R_i the mutual information between a_i and the class label.

This correspond to the following instructions:

Foreach element $(a_i, \mathrm{M}_{ij}, R_i) \in r3sub$

1. $rdd\,[(a_i, sumM_{ij}, R_i)] = reduceByKey\,(\_+\_)$

2. $$sumM_{ij} = \sum_{i=1}^{n} M_{ij}$$

EndForeach

- **Step 5 : calculate the relevance of each features and his redundancy with others features**

5th stage consists of compute the ranking measure r_i given in (4) which combine the redundancy and the relevance. Then send all r_i values to the master.

This correspond to the following instructions:

Foreach element $(a_i, sumM_{ij}, R_i) \subset r4sub$

1. $rdd\,[(a_i, r_i)] = mapToPair\,(\{a_i, sumM_{ij}, R_i\} \Rightarrow \{a_i, r_i\})$

2. $r_i = \beta + weight + ((1-\beta) * (R_i / Q_i))$

3. $Q_i = sumM_{ij} / (n*n);$

/ weight represents the SVM weight of attribute* a_i **/*

EndForeach

4. *Workers send* r_i *to the master*

- **Step 6 : return the optimal subset of features**

Finally, master collects, orders the features and returns those with highest scores r_i. This is done by the following instructions:

On the master:

1. *Collect and take ordered*
2. *Return D': optimal subset of K features in D with highest scores* r_i.

VI. DATA DESCRIPTION

We used support vector machine as classifier and LibSVM as the support vector machine tool.

We used three benchmark real-world datasets chosen from mldata.org [14]. Some informations of those datasets are presented in Table 2.

TABLE I. CHARACTERISTICS OF BENCHMARK DATASETS

NAME	NUMBER OF FEATURES	NUMBER OF INSTANCES
DUKE	7129	86
OVARIAN	15154	253
BREAST	24481	97

The experiments were performed on a cluster consisting of 4 nodes, where each node has 8 cores running at 2.60 GHz, with 56 GB memory and a 382 GB disk, then on a cluster of 6nodes with the sames parameters. The computing nodes are all running at the linux.

VII. EXPERIMENTAL RESULTS

For the evaluation of the classification accuracy of our proposition, Table 3 and Figure3 illustrate the results obtained by our algorithm using different percentages of the original set of features.

TABLE II. CLASSIFIER ACCURACY OF BENCHMARK DATASETS

PERCENTAGE FEATURES TAKEN FROM DATASET	CLASSIFIER ACCURACY		
	BREAST	DUKE	OVARIAN
100%	0,7526	0,5227	0,7549
75%	0,6598	0,9773	0,8221
50%	0,7629	0,9773	0,4744
25%	0,866	0,9773	0,8537

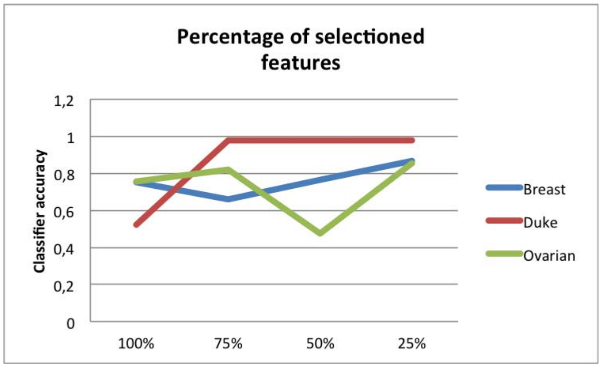

Figure 3. Classifier accuracy of datasets.

What is especially remarkable is that for all datasets, the classification accuracy is much better for a subset of 25 percent of features.

After discussing the performance of our proposition in terms of classification accuracy we also study his scalability.

To do this, we varied the number of cores, and perform all tests with the same conditions.

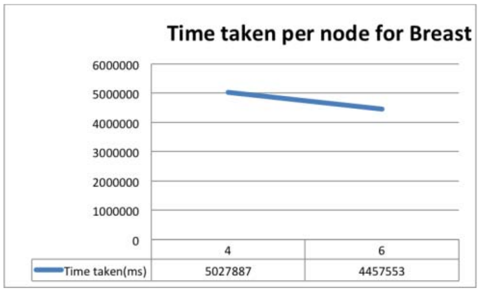

Figure 4. Scalabilty of Breast dataset

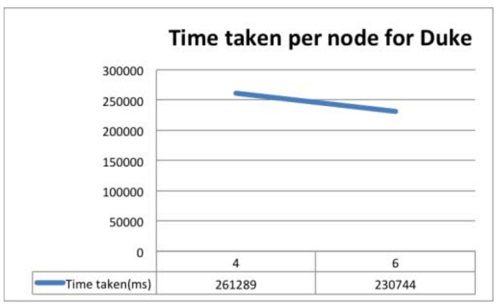

Figure 5. Scalabilty of Duke dataset

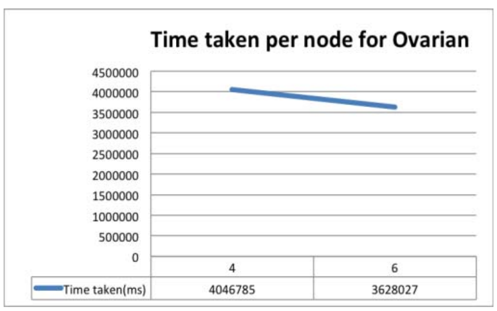

Figure 6. Scalability of Ovarian dataset

The performance results demonstrate that our solution offers good computational efficiency. The time of selecting features decreases significantly for the majority of datasets when the number of nodes increases.

VIII. Conclusion

In this paper, we proposed a novel scalable parallel filter method based on Spark. In our proposal, the Spark computing framework calculates the relevance of each feature regarding to class label and his redundancy relative to other features. Then, the most relevant attributes and less redundant is selected in just one single pass.

Experimental results demonstrated that our algorithm achieves a great performance improvement in scaling well and processing efficiently large datasets by selecting relevant attributes for classification problem.

In the future, we plan to experiment PSF-mRMR algorithm with more large datasets and with other filter methods like Relief, as well as with other classifiers such as *kNN*.

Acknowledgment

We have recipient of a Microsoft azure sponsored account. We would like to thanks Microsoft for this opportunity that have helped us to evaluating the performance of our algorithm. Without their Apache Spark cluster, we would never have made it to this point.

References

[1] Sergio Ramırez-Gallego, Hector Mourino-Talın, David Martınez-Rego,Veronica Bolon-Canedo, Jose Manuel Benıtez, Amparo Alonso-Betanzos and Francisco Herrera. An Information Theory-Based Feature Selection Framework for Big Data under Apache Spark.Journal of latex class files, vol. 13, no. 9, september 2014.

[2] Vaishali Chahar, Rita Chhikara, Yogita Gigras and Latika Singh. Significance of Hybrid Feature Selection Technique for Intrusion Detection Systems.Indian Journal of Science and Technology, Vol 9(48), DOI: 10.17485/ijst/2016/v9i48/105827, December 2016.

[3] Zhao Z., Cox J., Duling D., Sarle W. (2012) Massively Parallel Feature Selection: An Approach Based on Variance Preservation. In: Flach P.A., De Bie T., Cristianini N. (eds) Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2012. Lecture Notes in Computer Science, vol 7523. Springer, Berlin, Heidelberg.

[4] Dilpreet Singh and Chandan K Reddy. A survey on platforms for big data analytics. J Big Data. 2015; 2(1): 8.Published online 2014 Oct 9.

[5] Chuan Liu, Wenyong Wang, Qiang Zhao andMartin Konan. A new feature selection method based on a validity index of feature subset.Pattern Recognition Letters,Volume 92, 1 June 2017, Pages 1-8.

[6] Wenyan Z, Xuewen L , Jingjing W. Feature Selection for Cancer Classi cation Using Microarray Gene Expression Data. Biostat Biometrics Open Acc J. 2017;1(2): 555557.

[7] Jaseena K.U. and Julie M. David.Issues, challenges, and solutions: big data mining.Sixth International Conference on Networks & Communications,DOI: 10.5121/csit.2014.41311.

[8] Anthony Kleerekoper, Michael Pappas, Adam Pocock, Gavin Brown, Mikel Lujan. A scalable implementation of information theoretic feature selection for high dimensional data. Big Data (Big Data), 2015 IEEE International Conference,USA.

[9] Daniel Peralta, Sara del Río,Sergio Ramírez-Gallego,Isaac Triguero, Jose M. Benitezand Francisco Herrera.Evolutionary Feature Selection for Big Data Classification: A MapReduce Approach.Mathematical Problems in Engineering Volume 2015 (2015), Article ID 246139, 11 pages.

[10] Yong Wang,Wenlong Ke and Xiaoling Tao.A Feature Selection Method for Large-Scale Network Traffic Classification Based on Spark.Information (2078-2489) . 2016, Vol. 7 Issue 1, p1-11. 11p.

[11] Monalisa Mandal, Anirban Mukhopadhyay. An Improved Minimum Redundancy Maximum Relevance Approach for Feature Selection in Gene Expression Data. Jul 2016 · IEEE/ACM Transactions on Computational Biology and Bioinformatics.

[12] Yin-Wen Chang and Chih-Jen Lin. Feature ranking using linear SVM.Proceedings of the Workshop on the Causation and Prediction Challenge at WCCI 2008, PMLR 3:53-64, 2008.

[13] Piyushkumar A. Mundra and Jagath C. Rajapakse.SVM-RFE With MRMR Filter for Gene Selection.IEEE transactions on nanobioscience, vol. 9, no. 1, march 2010.

[14] http://mldata.org/repository/data/viewslug/ovarian-cancer-nci-pbsii-data/.