Ahmed Hussein Ali
Mahmood Zaki Abdullah

NOVEL APPROACH FOR BIG DATA CLASSIFICATION BASED ON HYBRID PARALLEL DIMENSIONALITY REDUCTION USING SPARK CLUSTER

Abstract *The big data concept has elicited studies on how to accurately and efficiently extract valuable information from a huge dataset. The major problem during big data mining is data dimensionality, which is due to the large number of dimensions in such datasets. This major consequence of high data dimensionality is that it affects the accuracy of machine learning (ML) classifiers; it also results in the wasting of time due to the presence of several redundant features in a dataset. This problem can be possibly solved using a fast feature reduction method. Hence, this study presents a fast HP-PL that is a new hybrid parallel feature reduction framework that utilizes spark to facilitate feature reduction on shared/distributed-memory clusters. An evaluation of the proposed HP-PL on the CICIDS2017 dataset showed the algorithm to be significantly faster than the conventional feature reduction techniques. The proposed technique required* $>$ *1 minute to select 4 dataset features from over 79 features and 3,000,000 samples on a 3-node cluster (a total of 21 cores). For the comparative algorithm, more than two hours was required to achieve the same feat. In the proposed system, Hadoop's distributed file system (HDFS) was used to achieve distributed storage, while Apache Spark was used as the computing engine. The model development was based on a parallel model with full consideration of the high performance and throughput of distributed computing. Conclusively, the proposed HP-PL method can achieve good accuracy with less memory and time compared to the conventional methods of feature reduction. This tool can be publicly accessed at https://github.com/ahmed/Fast-HP-PL.*

Keywords big data, dimensionality reduction, parallel processing, Spark, PCA, LDA

Citation Computer Science 20(4) 2019: 411–429

1. Introduction

The increased generation of data from many sources has resulted in an exponential increase in the volume of generated data (increasing to the Petabyte scale). Globally, the reported rate of data increase has been put to 40 zeta byte per year and has been predicted by the International Data Corporation (IDC) to increase to more than 40 ZB by the year 2020 [16, 26]. Figure 1 presents the comparative increase in digital data over time (measured in years).

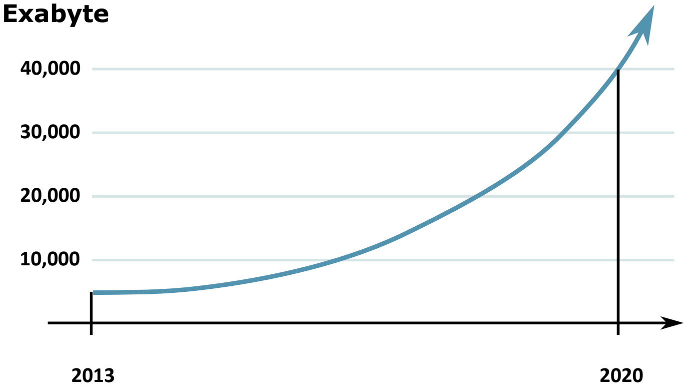

Figure 1. Global generation of digital data

In computer science, Data Mining (DM) is the process of using computational knowledge to analyze a huge set of data to discover the hidden trend or pattern in a dataset [27] that might not be noticed in the unprocessed data. The high level of complexity and volume of the available data have made DM and knowledge extraction necessary tasks during data processing [3]. The number of attributes or features of a dataset is referred to its dimensionality. For instance, attributes of medical data could include blood pressure, cholesterol level, weight, etc. A dataset with more than 100 attributes is said to be a high dimensional dataset, as the number of attributes in a high dimensional dataset is usually more than the sample sizes [17]. Such a huge volume of attributes makes the computation of high dimensional datasets extremely difficult. The generation of high dimensional data occurs at a faster rate from various sources, such as geological data, social networks, healthcare data, government agencies, ecological data, etc. This view is strengthened by evidence provided in a survey in the libSVM database from the 1990s where the maximum data dimensionality was reported to be around 62,000. However, this skyrocketed to 16 million over the first 10 years of this century. Currently, data dimensionality has reached almost 29 million [26], and operations on such big data using the existing learning algorithms do not proceed well and have become a challenge towards high dimensional data analysis. Being that training algorithms strive to use all of the features in high dimensional data, their performance is often affected. Hence, the use of such learning frameworks on such datasets requires a reduction in the dimensionality of the big data. Dimensionality reduction involves the removal of redundant or "extra" information from a dataset to retain only the relevant data. This process increases the accuracy of data classification and significantly reduces the algorithmic computational cost. Conventional algorithms usually encounter high computational complexity in terms of time and space when handling high dimensional data. Thus, solving such computational problems using CPU-based frameworks is not ideal, as they are becoming inadequate with the growing of data and dimensionality. Achieving the desired outcome for such problems that involve high data dimensionality requires the integration of several core architectures, such as high-performance parallel computing techniques that will size-up to the computational requirements of learning frameworks.

During data dimensionality reduction, some features may be discarded, while weights may be assigned to the remaining ones; new features may also be defined, such as smaller feature sets whose values amount to the linear combinations of the values of the original features. Previous works have addressed the issue of accelerating Dimensionality Reduction (DR) algorithms such that they can explore large datasets within a short amount of time. However, such acceleration can be in the form of complexity reduction on the part of the algorithm itself. A standout example of this acceleration is that of the mRMR method [20], which, despite its wide acceptance, is still prone to the issue of high complexity when faced with a high number of features. To reduce runtime, a greedy variation called fastmRMR was recently suggested for checking the redundancy of only the selected features rather than that of the whole dataset. Another greedy approach is the simplification of the calculated score metric for each feature [21]. Feature reduction has reportedly been achieved using several approaches and memetic frameworks [30] or PSO [9, 23]. Meanwhile, the HPC hardware resources can be exploited to facilitate applications without the need for reducing the general algorithmic computational requirements. For example, the WEKA toolkit [12] consists of multithreaded support for some DR frameworks that ensures the exploitation of the numerous cores available in the existing processors.

This work presents a novel DR parallelization method called fast HP-PL, which is accelerated DR by exploiting the hardware resources of the current distributed-memory systems. The system is implemented based on a hybrid approach that utilizes Spark to work on different networked nodes; it also exploits several cores within the same node using multiple slaves. Being that the Principle Component Analysis(PCA) method does not need labels, it can be useful in finding a subspace with basis vectors that correspond to the maximum-variance directions in the original space. As a similar method to PCA and Linear Discriminant Analysis(LDA) creates linear surfaces to distinct groups without increasing variance. It finds underlying space vectors that must be class-specific. LDA also produces the major mean variations between the desired classes; it generates a linear combination when specific features are available to a specific subject. Samples from multiple classes can have two defined parameters; the first parameter is an intra-class scatter matrix, while the second parameter is an inter-class scatter matrix. The reduction of the intra-class distance measure and maximization of the inter-class measure are the objective function. The proposed HP-PL achieves hybrid DR by combining PCA and LDA algorithms on a Spark architecture. The aim is to find a mapping from the initially high-dimensional space to a low-dimensional space where the most relevant features are retained. The feature vectors dimension was reduced in this study to improve the classification speed and minimize the level of confusion.

The remaining part of this article is presented in the following manner: works related to this study are reviewed in Section 2, while Section 3 provides the relevance of the problem. In Section 4, the development of the process for the parallel implementation of the algorithm on Spark is presented, while Section 5 presents the demonstration of the DR methods. In Section 6, the suggested methodology is described, while the experimental setup is presented in Section 7. The datasets used in this study are detailed in Section 8, while Section 9 presents and discusses the achieved results. The conclusion and recommendations for future study are presented in Section 10.

2. Related work

During data mining, DR is performed as a data pre-processing step, using different techniques based on the data complexity and process requirements. Dimension reduction for classification tasks is done in a manner that will improve algorithmic computational efficiency while ensuring classification accuracy. Several DR techniques have been proposed and developed in previous studies; these DR techniques can be classified based on the classification approach into different categories mentioned in the first section of this paper. A review of the existing literature on the parallel implementation of different DR algorithms was conducted as follows:

A fast-mRMR-MPI was presented by Gonzalez *et al.* [11] as a new hybrid parallel implementation that combines MPI and OpenMP to facilitate FS on shared-memory clusters. The proposed scheme was evaluated experimentally on different scenarios and proven to be effective in selecting the same features as the initial version (fast-mRMR) but at a lower computational time. Although the level of improvement is a function of the data characteristics (more favorable in datasets with the numbers of features higher than samples), the performance of the proposed fast-mRMR-MPI was efficient in all of the evaluated numbers of nodes for the studied datasets. A study by Hyunsoo *et al.* [13] suggested numerous multi-class schemes that are based on the GLDA algorithms. These schemes exploited the advantage of the DR transformation matrix without the need for parameter optimization or more training. The study introduced a marginal LDC, a Bayesian LDC, and a 1-D BLDC for multi-class classification. New algorithms for improving the convergence speed of the incremental LDA algorithm were proposed by Chatterjee and Roychowdhury. They derived the projected algorithm via the use of the steepest descent and conjugate direction methods to optimize the step size in each iteration.

An approach in which a feature reduction algorithm is used to remove redundant features from a dataset before applying a supervised data mining technique was presented by Priyanka *et al.* [7]. The implementation of the proposed system was done using Spark on a UNSW-NB15 network dataset for the accurate, fast, and efficient detection of network intrusion in the Netflow records. This study employed two feature reduction frameworks (LDA and Canonical Correlation Analysis [CCA]) and seven common classifiers. Ayyad and Khalid [5] suggested a novel combination of two projection-based FR frameworks in the DWT domain. The hybridized algorithms are RW-LDA and SVD using the left and right singular vectors. PaMPa-HD was introduced by Daniele *et al.* [4] as a MapReduce-based frequent closed itemset mining algorithm for DR. It was suggested as a solution for data mining process parallelization and speeding up.

Furthermore, different techniques have been suggested for the easy configuration of the algorithmic parameters. For instance, Zoang [28] suggested a method for reducing the dimensionality in a high-dimensional time-series dataset, which is simpler when compared to the existing DFT- and DCT-based methods. As DR preserves planar geometric blocks, it can be applied in image processing and computer graphics. Nitika and Kriti [24] developed a new DR method using PCA and Feature Ranking. The performance of the proposed method relating to DR was evaluated on a breast cancer dataset. From the results, the proposed method effectively achieved DR without any effect on the classification cost and accuracy. Zebin *et al.* [29] presented DR techniques for cloud computing environments that are efficient in both data storage and data preprocessing.

The study also presented a parallel and distributed implementation of the PCA technique for hyperspectral DR in cloud computing environments. These ensured the applicability of the conventional PCA algorithm for parallel and distributed computing through an optimization process before implementing it on a real cloud computing environment. Muhammad *et al.* [26] analyzed LDA- and PCA-based FS algorithms using data sourced from two types of gas sensors (an in-house fabricated 4×4 tin-oxide gas array sensor and seven commercial Figaro sensors). The performance of both approaches was evaluated using a decision tree-based classifier, while the implementation of the software was done in MATLAB. The implementation of the hardware was done on the Zynq system-on-chip (SoC) platform.

A Parallel Random Forest (PRF) algorithm was presented by Jianguo *et al.* [6] for big data on the Apache Spark platform. The optimization of the PRF algorithm was based on a hybrid combination of data-parallel and task-parallel optimization. A vertical data-partitioning method was performed during the data-parallel optimization to minimize the cost of the data communication. Alaa and Hasan [8] developed two face-recognition schemes; one scheme (called PCA-NN) was based on PCA and FFNN, while the second scheme (called LDA-NN) was based on LDA and FFNN. Both systems involve two phases – a PCA or LDA pre-processing phase, and an NN classification phase. Tayde and Patil [27] strived to present a solution to the DR problem using a Hadoop and Spark distributed framework. The study utilized Feature Selection (FS), pattern matching, and SVM-based classification in Hadoop and Spark.

3. Significance of problem

The existing classifiers cannot efficiently handle the high dimensionality associated with today's real-world data [25, 26]. Hence, many data analyzers require a preprocessing step called feature reduction, which is considered to be an important data preparation step that facilitates the accurate and efficient extraction of valuable knowledge from high dimensional data. Assume Q and L to represent the number of features and number of samples, respectively.

At a fixed Q, a larger L will imply more constraints; the ensuing correct hypothesis is presumed more reliable. However, at a fixed L, the impact of a reduced Q will be similar to that of a significant increase in the number of instances. Theoretically, DR will exponentially reduce the hypothesis space. Assume four binary (i.e., 0 1) features N1, N2, N3, N4, and class C (maybe negative or positive). If there are four instances in the training data (i.e., $L = 4$), only one-fourth of all the possible number of instances $2^4 = 16$, and the hypothesis space will have a size of $(2^2)4 = 65; 536$. However, there are only two relevant features to the target concept; the hypothesis space will have a size of $(2^2)2 = 16$, implying that the hypothesis space has been exponentially reduced.

Having been left with only two features, learning can be perfected using the only four available instances if no instance is repeated in the reduced training dataset. The complexity of the model built with two features will also be less than a model built with four features. Thus, the hypothesis space can be effectively reduced via feature selection; the number of training instances can also be virtually increased, making the creation of a perfect compact model possible.

4. Spark

Spark was developed to support iterative algorithms that are not well supported by MapReduce [14]. This includes most of the iterative ML frameworks (such as k-means). Even though these applications are supported by Spark, it still maintains the fault tolerance of MapReduce. The Spark engine [14] can be executed in several environments (such as Mesos clusters or Hadoop) and has been applied in query processing (SQL), advanced analytics, and large data streaming in different data stores. The performance of Spark is 10× higher than that of Hadoop in iterative ML workloads. Regarding the system architecture (Fig. 2), the workflow is controlled by a Spark master node, while the Spark worker nodes execute the job submitted to the Spark master node. Spark is mainly based on the resilient distributed data set (RDD) concept, which represents a set of objects (read-only) distributed across machines. An RDD may be cached in memory across machines and could be reutilized in several MapReduce-like parallel operations. Fault tolerance can be achieved by rebuilding from the other RDDs at the expense of data partitioning. Spark is built with a set of more than 80 high-level built-in operators.

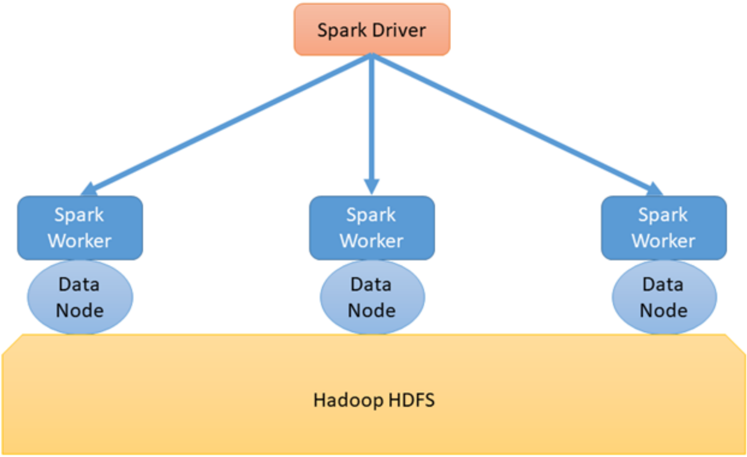

Figure 2. Spark Architecture

5. Dimensionality reduction

This is a Big Data (BD) era with thousands of observations and variables that need to be analyzed [6, 19] to identify the vital patterns and extract important information for business/profit-making as well as for making informed decisions or even conducting some basic research. Often, there are several variables (hundreds to tens of thousands) in a sample, and the major task during Feature Extraction (FE) and Feature Selection (FS) is to minimize the data dimensionality as much as possible while ensuring that no important information need is lost for the task at hand.

5.1. Principle Component Analysis (PCA)

PCA is one of the feature extraction techniques for extracting important features (called components) from a large set of the dataset [1, 18]. With the PCA techniques, a set-off element in the low dimensional dataset is extricated from a high-dimensional dataset in a bid to get the maximum of information. PCA turns out to be more important with fewer factors and more valuable when handling three or more dimensional datasets. This is usually done on a symmetric correlation or covariance matrix. For instance, assume M to be a matrix whose rows are the point in space; then, the MTM and eigenpairs of that point can be computed. E is the matrix whose columns (as the eigenvectors) are arranged in a way that makes the largest eigenvalue come first. Consider Matrix L to have $M^T M$ eigenvalues along the diagonal in a manner that presents the largest value firsts and the 0's in the other entries; then, even though

$M^TMe = \lambda e = e\lambda$ for each eigenvector e and its corresponding eigenvalue λ, it is assumed that:

$$M^TM = EL \tag{1}$$

Observably, ME represents the points of M that are transformed into another coordinate space where the first axis corresponds to the largest eigenvalue.

5.2. Linear Discriminant Analysis (LDA)

LDA is an important linear feature extraction method [5, 8] that aims to find a set of the most discriminant projective vectors for mapping HD samples onto an LD space. All of the samples in the projective feature space will have the maximum inter-class scatter and minimum intra-class scatter, while the remaining test samples from the other classes will be easy to classify.

LDA [22] linearly combines classes with maximum inter-class mean differences. The aim of LDA is to solve optimal discrimination projection matrix W_{opt}:

$$W_{opt} = arg_w max = \frac{|W^T s_b W|}{|W^T s_w W|} \tag{2}$$

or

$$W_{opt} = arg_w max = \frac{|W^T s_b W|}{|W^T s_t W|} \tag{3}$$

where S_b, S_w, and S_t are the scatter matrixes:

$$S_b = \sum_{i=1}^{j} n_i(\mu_i - \mu)(\mu_i - \mu)^T \tag{4}$$

$$S_w = \sum_{i} n_i(x_i - \mu_{k1})(x_i - \mu_{k1})^T \tag{5}$$

where

S_b – inter-class scatter matrix,
S_w – intra-class scatter matrix,
S_t – total scatter matrix, $S_t = S_b + S_w$,
μ_i – mean feature vector of class i,
n_i – number of samples in class i.
J – overall number of samples in complete dataset,
x_i – sample's feature vector,
μ_{k1} – vector of class to which x_i belongs.

6. Proposed HP-PL approach

This paper suggests a new approach for big data processing. This section focused on the methodology of the proposed new method, focusing on the combination of two appearance-based methods (PCA and LDA) for data reduction. One of the major advancements in ML in the past decade is the ensemble method. This method achieves a more suitable pre-processing step in order to build a highly accurate classifier or find an efficient dataset. This is achieved by combining several moderate DR methods. The PCA and LDA methods are two successful methods for the construction of an ensemble pre-processing step. In this study, both techniques were used to reduce the effect of overfitting in single data reduction. The ensemble PCA and LDA were used to improve the pre-processing-based big data classification. Owing to the large size of big datasets, the processing, feature vector extraction, and data classification will require a great deal of time using one machine. Thus, this problem was overcome by using Hadoop and Spark (distributed frameworks) in the proposed model. The proposed system is presented as a block diagram in Figure 3.

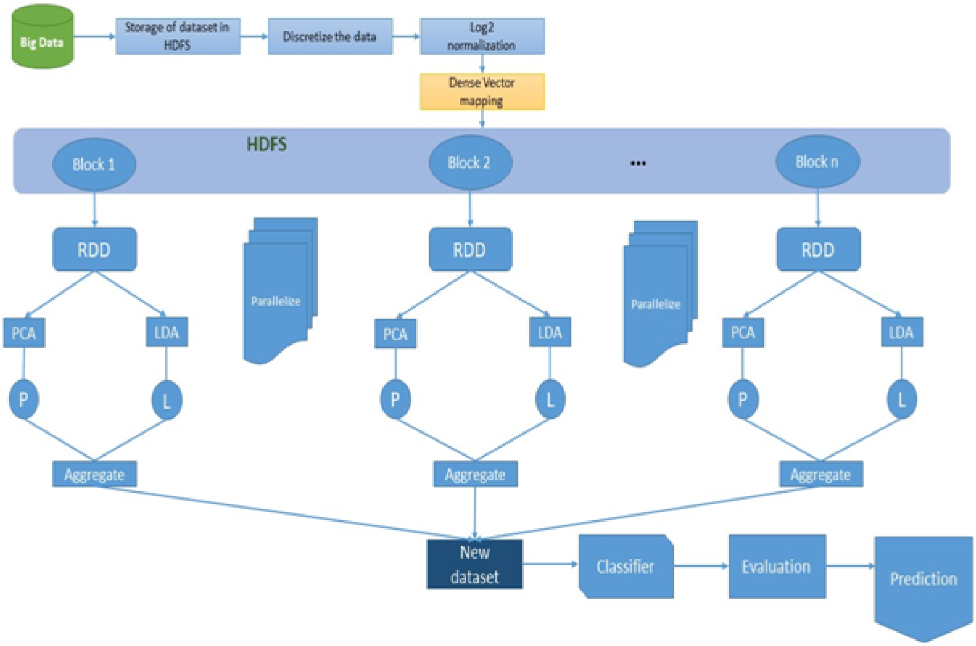

Figure 3. Proposed architecture for HP-PL

It consists of the following steps:

- Storage of dataset in HDFS.
- Discretize data (0–127).
- Normalization of these dissimilarity distance vectors.
- Convert dataset into vector dense (Scala Programming) to be suitable for parallelizing implementation.
- Partitioning dataset into N blocks according to number of slaves in Spark cluster.
- Spark engine (MLIib) consuming data as Resilient Distributed Dataset RDD.

- Apply PCA, LDA on loaded RDD dataset (user-selected number of components) output of ensemble reduction method will be set of eigenvectors (also called as Components) P from PCA, and L from LDA.
- Aggregate the parts of obtained dataset.
- New dataset.
- Apply classifier to validate approach.
- Evaluation.
- Prediction.

The preprocessing step of the proposed method is as follows:

i. HDFS is used in the Hadoop applications as the primary system for data storage, which utilizes a NameNode and DataNode framework to launch a distributed file system; this ensures data access across highly scalable Hadoop clusters. HDFS encourages rapid data transfer between compute nodes. It receives data, then fractionates the data information into separate blocks before distributing them to various cluster nodes to enable highly efficient parallel processing.

ii. Numerical data discretization is an important data pre-processing method for data mining and knowledge discovery. This is considered to be a data reduction mechanism that owes to its ability to reduce data from a large numerical value domain to a subset of categorical values. Data discretization is done using several algorithms on the Apache Spark platform; it improves the learning speed and accuracy of the learning methods.

iii. Owing to the heterogeneous nature of the dissimilarity distance scores output from the different recognition methods, these scores need to be transformed via score normalization into a single domain before they are combined. This work mapped the distance vector to the $[0, 1]$ range using either MIN-MAX or log 2 normalization techniques. The end-points of the range of the distance is specified by quantities D_{max} andD_{min}, and D_n is given by:

$$D_n = \frac{D - D_{min}}{D_{max} - D_{min}} \tag{6}$$

where $D_{min} = min(d_1, d_2, \ldots\ldots d_m)$ and $D_{max} = max(d_1, d_2, \ldots\ldots d_m)$.

iv. Both the integer-typed and 0-based indices as well as the double-typed values of a local vector are stored on a single machine. However, MLlib supports both dense and sparse local vectors. A dense vector is supported by a double array that represents its entry values; for a sparse vector, this is supported by indices and values (two parallel arrays). For instance, the dense and sparse formats of vector (1.0, 0.0, 3.0) can be represented as [1.0, 0.0, 3.0] and (3, [0, 2], [1.0, 3.0]), respectively; where 3 represents the vector size.

Apache Spark installation on a multi-node cluster needs multiple nodes; for this, we can use multiple machines. The master computer (which includes the main process) manages all views and reads each part of the big file and sends it to the specific slave process that will be received and processed by HP-PL. Figure 4 illustrates the proposed architecture for HP-PL in preprocessing and manipulating big data.

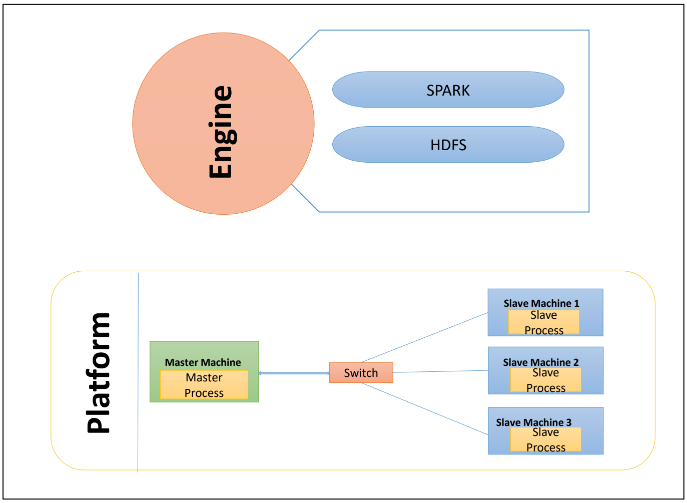

Figure 4. Proposed architecture for HP-PL

7. Experimental design and analysis

The experimental clusters consist of one master and three slaves. For the Hadoop--based version, the serial version was written in Java, while that of Spark was written in Scala. Table 1 shows the deployment of the nodes in the cluster.

Table 1
Deployment of cluster nodes

Node	CPU	Memory	Network	JVM version	Hadoop version	Scala version	Spark version
Master	Core 8	16 GB	1 GB/s	1.7.0	1.2.1	2.10.4	1.6.0
Slave 1-3	Core 8	16 GB	1 GB/s	1.7.0	1.2.1	2.10.4	1.6.0

8. Data set description

A dataset with both normal and attack behaviors is required for the performance evaluation of the proposed scheme. Several studies have been done using older benchmark data sets, which do not offer realistic output performance. The dataset from UCI KDD is a labeled comma-separated file. With the help of Spark context, the file was converted into an RDD (Resilient Distributed Dataset), which is the data type in the memory for computation. The dataset used in our experiment to access the classifier for network intrusion detection is the KDDCup'99 dataset [17], which was developed by MIT at Lincoln's laboratory. The description of the KDDCup'99 dataset is presented in Tables 2 and 3.

Table 2
KDD attack description

Class	Training size	[%]	Test size	[%]
Normal	972,781	19.850	60,593	19.480
DOS	3,883,390	79.270	231,455	74.410
Probe	41,102	00.830	4,166	01.330
U2R	52	00.0010	245	00.070
R2L	1,106	00.020	14,57	04.680
Total	4,898,431	100.0	311,029	100.0

Table 3
Statistics of redundant records in the KDD train set

	Original record	Distinct record	Reduction rate [%]
Attacks	3,925,650	262,178	93.320
Normal	972,781	812,814	16.440
Total	4,898,431	1,074,992	78.050

Another dataset used in this study is the CICIDS 2017 dataset (see Table 4), which consists of 5 days of network activity data with 225,745 packages and > 80 data features; the data was collected over 7 days of monitored network activity. The attack simulation in the dataset is partitioned into seven attack classes (comprised of Brute Force, Botnet, Heart Bleed, DoS, Web, DDoS, and Infiltration attacks).

Table 4
Overall characteristics of CICIDS2017 dataset

Dataset Name	CICIDS2017
Dataset type	Multi-class and binary class
Year of release	2017
Total number of distinct instances	3,000,000
Number of features	79
Number of distinct classes	2

9. Results and discussion

This section presents the evaluation of the experiments. The novel HP-PL approach was evaluated on several available datasets on the ML repository of UCI. Spark was used on a Yarn cluster that consists of three nodes. The cluster is armed with Spark 1.6.0 and Hadoop 1.2.1, and each node has an Intel® CORE8 CPU HQ processor and 16G RAM. Table 5 presents the datasets employed in the experiments.

Table 5

Datasets info

Dataset	Info			
	#samples	*#features*	*#classes*	*source*
KDD99	4,000,000	42	23	UCI
CICIDS2017	3,000,000	79	2	Canadian Institute for Cybersecurity

To further prove the classification performance of HP-PL, it was evaluated on a subset of the above datasets, which are standard datasets for testing and evaluating state-of-the-art ML algorithms.

This section presents the results of the experiments on the proposed HP-PL. The comparison was based on three different classifiers (Random Forest, Naïve Bayes, and Hoffeding Tree). Our approach was tested by one of the recent types of datasets called CICIDS2017 (see Table 6).

Table 6

The CICIDS dataset statistics

Statistic type	Ratio [%]
Kappa statistic	0.9708
Mean absolute error	0.0133
Root mean squared error	0.0794
Relative absolute error	5.3155
Root relative squared error	22.4504

CICIDS is considered to be a large volume of data with high dimensions. Each type of classifier was tested on 4 samples (12 comparisons). The samples are all features of CICIDS2017, 4-component PCA, 4-component LDA, and our proposed approach. The comparison of each sample was based on a set of statistical methods such as accuracy, TPR, FPR, Precision, Recall, F-measure, the Matthews correlation coefficient, MCC, the receiver operating characteristics curve (ROC Curve), and the precision-recall curve (PRC). Table 7 shows that the proposed method produced results as good as those using PCA and LDA for different classifiers.

Table 7

Performance evaluation and results based on cross-validation statistics

RandomForest									
All features of CICIDS2017									
Accuracy	TPR	FPR	Prec.	Rec.	F-M	MCC	ROC Area	PRC Area	C
99.8356	0.999	0.005	0.999	0.999	0.999	0.993	1.000	1.000	0
	0.995	0.001	0.994	0.995	0.994	0.993	1.000	0.999	1
Weighted Average	0.998	0.004	0.998	0.998	0.998	0.993	1.000	1.000	–

Table 7 (cont.)

PCA-4									
Accuracy	TPR	FPR	Prec.	Rec.	F-M	MCC	ROC Area	PRC Area	C
99.7991	0.999	0.006	0.999	0.999	0.999	0.992	1.000	1.000	0
	0.994	0.001	0.993	0.994	0.993	0.992	1.000	0.999	1
Weighted Average	0.998	0.006	0.998	0.998	0.998	0.992	1.000	1.000	–
LDA-4									
Accuracy	TPR	FPR	Prec.	Rec.	F-M	MCC	ROC Area	PRC Area	C
99.8196	0.999	0.005	0.999	0.999	0.999	0.993	1.000	1.000	0
	0.995	0.001	0.993	0.995	0.994	0.993	1.000	0.999	1
Weighted Average	0.998	0.005	0.998	0.998	0.998	0.993	1.000	1.000	–
HP-PL									
Accuracy	TPR	FPR	Prec.	Rec.	F-M	MCC	ROC Area	PRC Area	C
99.824	0.999	0.005	0.999	0.999	0.999	0.993	1.000	1.000	0
	0.995	0.001	0.993	0.995	0.994	0.993	1.000	0.999	1
Weighted Average	0.998	0.004	0.998	0.998	0.998	0.993	1.000	1.000	–
NaïveBayes									
All features of CICIDS2017									
Accuracy	TPR	FPR	Prec.	**Rec.**	F-M	MCC	ROC Area	PRC Area	C
81.3604	0.783	0.008	0.998	0.783	0.878	0.583	0.964	0.994	0
	0.992	0.217	0.440	0.992	0.609	0.583	0.964	0.795	1
Weighted Average	0.814	0.039	0.916	0.814	0.838	0.583	0.964	0.965	–
PCA-4									
Accuracy	TPR	FPR	Prec.	Rec.	F-M	MCC	ROC Area	PRC Area	C
14.6707	0.000	0.000	1.000	0.000	0.000	0.002	0.500	0.853	0
	1.000	1.000	0.147	1.000	0.256	0.002	0.500	0.147	1
Weighted Average	0.147	0.147	0.875	0.147	0.038	0.002	0.500	0.750	–

Table 7 (cont.)

<table>
<tr><td colspan="10">LDA-4</td></tr>
<tr><td>Accuracy</td><td>TPR</td><td>FPR</td><td>Prec.</td><td>Rec.</td><td>F-M</td><td>MCC</td><td>ROC Area</td><td>PRC Area</td><td>C</td></tr>
<tr><td rowspan="2">86.0227</td><td>0.988</td><td>0.882</td><td>0.867</td><td>0.988</td><td>0.923</td><td>0.228</td><td>0.616</td><td>0.920</td><td>0</td></tr>
<tr><td>0.118</td><td>0.012</td><td>0.624</td><td>0.118</td><td>0.199</td><td>0.228</td><td>0.616</td><td>0.241</td><td>1</td></tr>
<tr><td>Weighted Average</td><td>0.860</td><td>0.754</td><td>0.831</td><td>0.860</td><td>0.817</td><td>0.228</td><td>0.616</td><td>0.820</td><td>–</td></tr>
<tr><td colspan="10">HP-PL</td></tr>
<tr><td>Accuracy</td><td>TPR</td><td>FPR</td><td>Prec.</td><td>Rec.</td><td>F-M</td><td>MCC</td><td>ROC Area</td><td>PRC Area</td><td>C</td></tr>
<tr><td rowspan="2">48.024</td><td>0.391</td><td>0.002</td><td>0.999</td><td>0.391</td><td>0.562</td><td>0.292</td><td>0.575</td><td>0.910</td><td>0</td></tr>
<tr><td>0.998</td><td>0.609</td><td>0.220</td><td>0.998</td><td>0.360</td><td>0.292</td><td>0.575</td><td>0.216</td><td>1</td></tr>
<tr><td>Weighted Average</td><td>0.480</td><td>0.091</td><td>0.885</td><td>0.480</td><td>0.533</td><td>0.292</td><td>0.575</td><td>0.808</td><td>–</td></tr>
<tr><td colspan="10">HoeffdingTree</td></tr>
<tr><td colspan="10">All features of CICIDS2017</td></tr>
<tr><td>Accuracy</td><td>TPR</td><td>FPR</td><td>Prec.</td><td>Rec.</td><td>F-M</td><td>MCC</td><td>ROC Area</td><td>PRC Area</td><td>C</td></tr>
<tr><td rowspan="2">95.4853</td><td>0.967</td><td>0.115</td><td>0.980</td><td>0.967</td><td>0.973</td><td>0.826</td><td>0.988</td><td>0.998</td><td>0</td></tr>
<tr><td>0.885</td><td>0.033</td><td>0.821</td><td>0.885</td><td>0.852</td><td>0.826</td><td>0.988</td><td>0.924</td><td>1</td></tr>
<tr><td>Weighted Average</td><td>0.955</td><td>0.103</td><td>0.957</td><td>0.955</td><td>0.956</td><td>0.826</td><td>0.988</td><td>0.987</td><td>–</td></tr>
<tr><td colspan="10">PCA-4</td></tr>
<tr><td>Accuracy</td><td>TPR</td><td>FPR</td><td>Prec.</td><td>Rec.</td><td>F-M</td><td>MCC</td><td>ROC Area</td><td>PRC Area</td><td>C</td></tr>
<tr><td rowspan="2">95.4853</td><td>0.967</td><td>0.115</td><td>0.980</td><td>0.967</td><td>0.973</td><td>0.826</td><td>0.988</td><td>0.998</td><td>0</td></tr>
<tr><td>0.885</td><td>0.033</td><td>0.821</td><td>0.885</td><td>0.852</td><td>0.826</td><td>0.988</td><td>0.924</td><td>1</td></tr>
<tr><td>Weighted Average</td><td>0.955</td><td>0.103</td><td>0.957</td><td>0.955</td><td>0.956</td><td>0.826</td><td>0.988</td><td>0.987</td><td>–</td></tr>
<tr><td colspan="10">LDA-4</td></tr>
<tr><td>Accuracy</td><td>TPR</td><td>FPR</td><td>Prec.</td><td>Rec.</td><td>F-M</td><td>MCC</td><td>ROC Area</td><td>PRC Area</td><td>C</td></tr>
<tr><td rowspan="2">97.2453</td><td>0.977</td><td>0.053</td><td>0.991</td><td>0.977</td><td>0.984</td><td>0.895</td><td>0.895</td><td>0.999</td><td>0</td></tr>
<tr><td>0.947</td><td>0.023</td><td>0.875</td><td>0.947</td><td>0.910</td><td>0.895</td><td>0.894</td><td>0.958</td><td>1</td></tr>
<tr><td>Weighted Average</td><td>0.972</td><td>0.048</td><td>0.974</td><td>0.972</td><td>0.973</td><td>0.895</td><td>0.895</td><td>0.993</td><td>–</td></tr>
</table>

Table 7 (cont.)

HP-PL									
Accuracy	TPR	FPR	Prec.	Rec.	F-M	MCC	ROC Area	PRC Area	C
99.272	0.996	0.029	0.995	0.996	0.996	0.971	0.999	1.000	0
	0.971	0.004	0.979	0.971	0.975	0.971	0.999	0.993	1
Weighted Average	0.993	0.025	0.993	0.993	0.993	0.971	0.999	0.999	–

10. Conclusion and future work

Dimensionality reduction is one data mining method that has nowadays become a common and important step in ML due to the continuous increase in the generation of a dataset from different fields. This paper proposed fast HP-PL as a new parallel tool for facilitating DR and improving big data classification accuracy by leveraging the computational capabilities of distributed-memory clusters. The hybrid HP-PL proposed in this study was executed on Apache Spark.

From the experimental evaluation, fast HP-PL was found to select the best features using the lowest computational time when compared to the other benchmark frameworks. Although the level of improvement depends on the features of the dataset, the proposed fast HP-PL performed well on the number of nodes evaluated for the tested datasets. The proposed tool in this study can be accessed at https://github.com/ahmed/Fast-HP-PL. Our future work will focus on the flexibility of the fast HP-PL to enable it to accept more input data formats; we will also focus on the development of parallel versions of other algorithms for feature reduction.

The applications of the proposed algorithms include online applications where the complete dataset is not available but rather presented as a stream provision of efficient DR for applications for high dimensionality applications (such as bioinformatics, medical, weather forecast, biometrics, financial, text categorization, etc).

Acknowledgements

The authors are grateful to ICCI, informatics Institute for Postgraduate Studies (IIPS-IRAQ) for their moral support. Many thanks for Mustansiriya university (www.uomustansiriyah.edu.iq)for its support to finish this paper. Special thanks to the anonymous reviewers for their valuable suggestions and constructive comments.

References

[1] Agarwal S., Ranjan P., Ujlayan A.: Comparative analysis of dimensionality reduction algorithms, case study: PCA. In: *2017 11th International Conference on Intelligent Systems and Control (ISCO)*, pp. 255–259, IEEE, 2017.

[2] Akbar M.A. et al.: An Empirical Study for PCA and LDA Based Feature Reduction for Gas Identification, *IEEE Sensors Journal*, vol. 16(14), pp. 5734–5746, 2016.

[3] Ali A.H., Abdullah M.Z.: Recent Trends in Distributed Online Stream Processing Platform for Big Data: Survey. In: *2018 1st Annual International Conference on Information and Sciences (AiCIS)*, pp. 140–145, IEEE, 2018.

[4] Apiletti D., Baralis E., Cerquitelli T., Garza P., Pulvirenti F., Michiardi P.: A Parallel MapReduce Algorithm to Efficiently Support Itemset Mining on High Dimensional Data, *Big Data Research*, vol. 10, pp. 53–69, 2017.

[5] Ayyad M., Khalid C.: New fusion of SVD and Relevance Weighted LDA for face recognition, *Procedia Computer Science*, vol. 148, pp. 380–388, 2019.

[6] Chen J., Li K., Tang Z., Bilal K., Yu S., Weng C., Li K.: A Parallel Random Forest Algorithm for Big Data in a Spark Cloud Computing Environment, *IEEE Transactions on Parallel and Distributed Systems*, vol. 28(4), pp. 919–933, 2016.

[7] Dahiya P., Srivastava D.K.: Network Intrusion Detection in Big Dataset Using Spark, *Procedia Computer Science*, vol. 132, pp. 253–262, 2018.

[8] Eleyan A., Demirel H.: PCA and LDA Based Face Recognition Using Feedforward Neural Network Classifier. In: *International Workshop on Multimedia Content Representation, Classification and Security*, pp. 19–206, Springer, 2006.

[9] Ghamisi P., Benediktsson J.A.: Feature Selection Based on Hybridization of Genetic Algorithm and Particle Swarm Optimization, *IEEE Geoscience and Remote Sensing Letters*, vol. 12(2), pp. 309–313, 2015.

[10] Ghassabeh Y.A., Rudzicz F., Moghaddam H.A.: Fast incremental LDA feature extraction, *Pattern Recognition*, vol. 48(6), pp. 1999–2012, 2015.

[11] González-Domínguez J., Bolón-Canedo V., Freire B., Touriño J.: Parallel feature selection for distributed-memory clusters, *Information Sciences*, vol. 496, pp. 399–409, 2019.

[12] Hall M., Frank E., Holmes G., Pfahringer B., Reutemann P., Witten I.H.: The WEKA data mining software: an update, *ACM SIGKDD Explorations Newsletter*, vol. 11(1), pp. 10–18, 2009.

[13] Kim H., Drake B.L., Park H.: Multiclass classifiers based on dimension reduction with generalized LDA, *Pattern Recognition*, vol. 40(11), pp. 2939–2945, 2007.

[14] Liu P., Zhao H.-h., Teng J.-y., Yang Y.-y., Liu Y.-f., Zhu Z.-w.: Parallel naive Bayes algorithm for large-scale Chinese text classification based on a spark, *Journal of Central South University*, vol. 26(1), pp. 1–12, 2019.

[15] Lok U.-W., Song P., Trzasko J.D., Borisch E.A., Daigle R., Chen S.: Parallel Implementation of Randomized Singular Value Decomposition and Randomized Spatial Downsampling for Real-Time Ultrafast Microvessel Imaging on a Multi--Core CPUs Architecture. In: *2018 IEEE International Ultrasonics Symposium (IUS)*, pp. 1–4, IEEE, 2018. https://doi.org/10.1109/ULTSYM.2018.8579678.

[16] Mallios X., Vassalos V., Venetis T., Vlachou A.: A Framework for Clustering and Classification of Big Data Using Spark. In: Debruyne C., Panetto H., Meersman R., Dillon T., Kühn E., O'Sullivan D., Ardagna C.A. (eds.), *On the Move to Meaningful Internet Systems: OTM 2016 Conferences. OTM 2016*, Lecture Notes in Computer Science, vol. 10033, pp. 344–362, Cham, Springer, 2016.

[17] Mohammed M.A., Hasan R.A., Ahmed M.A., Tapus N., Shanan M.A., Khaleel M.K., Ali A.H.: A Focal load balancer based algorithm for task assignment in a cloud environment. In: *2018 10th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, pp. 1–4, IEEE, 2018.

[18] Nick W., Shelton J., Bullock G., Esterline A., Asamene K.: Comparing dimensionality reduction techniques. In: *SoutheastCon 2015*, pp. 1–2, IEEE, 2015.

[19] Patil S.V., Kulkarni D.B.: A Review of Dimensionality Reduction in High--Dimensional Data Using Multi-core and Many-core Architecture. In: *Workshop on Software Challenges to Exascale Computing*, pp. 54–63, Springer, 2018.

[20] Ramírez-Gallego S., Lastra I., Martínez-Rego D., Bolón-Canedo V., Benítez J.M., Herrera F., Alonso-Betanzos A.: Fast-mRMR: Fast Minimum Redundancy Maximum Relevance Algorithm for High-Dimensional Big Data, *International Journal of Intelligent Systems*, vol. 32(2), pp. 134–152, 2017.

[21] Raza M.S., Qamar U.: A parallel rough set based dependency calculation method for efficient feature selection, *Applied Soft Computing*, vol. 71, pp. 1020–1034, 2018.

[22] Sadaghiyanfam S., Kuntalp M.: Comparing the Performances of PCA (Principal Component Analysis) and LDA (Linear Discriminant Analysis) Transformations on PAF (Paroxysmal Atrial Fibrillation) Patient Detection. In: *Proceedings of the 2018 3rd International Conference on Biomedical Imaging, Signal Processing*, pp. 1–5, ACM, 2018.

[23] Salih A.-H.A., Ali A.H., Hashim N.Y.: Jaya: An Evolutionary Optimization Technique for Obtaining the Optimal Dthr Value of Evolving Clustering Method (ECM), *International Journal of Engineering Research and Technology*, vol. 11(12), pp. 1901–1912, 2018.

[24] Sharma N., Saroha K.: A novel dimensionality reduction method for cancer dataset using PCA and Feature Ranking. In: *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 2261–2264, IEEE, 2015.

[25] Szabó Z., Lőrincz A.: Fast Parallel Estimation of High Dimensional Information Theoretical Quantities with Low Dimensional Random Projection Ensembles. In: *International Conference on Independent Component Analysis and Signal Separation*, pp. 146–153, Springer, 2009.

[26] Tanwar S., Ramani T., Tyagi S.: Dimensionality Reduction Using PCA and SVD in Big Data: A Comparative Case Study. In: Patel Z., Gupta S. (eds.), *Future Internet Technologies and Trends. ICFITT 2017. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, vol. 220, pp. 116–125, Springer, Cham, 2017.

[27] Tayde S.S., Patil N.: A Novel Approach for Genome Data Classification Using Hadoop and Spark Framework. In: *Emerging Research in Computing, Information, Communication, and Applications*, pp. 333–343, Springer, 2016.

[28] Thanh H.C.: Parallel Dimensionality Reduction Transformation for Time-Series Data. In: *2009 First Asian Conference on Intelligent Information and Database Systems*, pp. 104–108, IEEE, 2009.

[29] Wu Z., Li Y., Plaza A., Li J., Xiao F., Wei Z.: Parallel and Distributed Dimensionality Reduction of Hyperspectral Data on Cloud Computing Architectures, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 9(6), pp. 2270–2278, 2016.

[30] Zhu Z., Ong Y.-S., Dash M.: Wrapper–filter Feature Selection Algorithm Using a Memetic Framework, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37(1), pp. 70–76, 2007.

Affiliations

Ahmed Hussein Ali
Ph.D. candidate, ICCI, Informatics Institute for Postgraduate Studies, Baghdad, Iraq,
ORCID ID: https://orcid.org/0000-0002-3202-1928

Mahmood Zaki Abdullah
Computer Eng., College of Engineering, Mustansiriyah University, Baghdad, Iraq,
ORCID ID: https://orcid.org/0000-0002-3191-3780

Received: 25.07.2019
Revised: 10.09.2019
Accepted: 10.09.2019