Performance Evaluation of Data-driven Intelligent Algorithms for Big data Ecosystem

Muhammad Junaid1 · Sajid Ali2 · Isma Farah Siddiqui4 · Choonsung Nam5 · Nawab Muhammad Faseeh Qureshi3 · Jaehyoun Kim3 · Dong Ryeol Shin1

Accepted: 4 November 2021
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract
Artificial intelligence, specifically machine learning, has been applied in a variety of methods by the research group to transform several data sources into valuable facts and understanding, allowing for superior pattern identification skills. Machine learning algorithms on huge and complicated data sets, computationally expensive on the other hand, processing requires hardware and logical resources, such as space, CPU, and memory. As the amount of data created daily reaches quintillion bytes, a complex big data infrastructure becomes more and more relevant. Apache Spark Machine learning library (ML-lib) is a famous platform used for big data analysis, it includes several useful features for machine learning applications, involving regression, classification, and dimension reduction, as well as clustering and features extraction. In this contribution, we consider Apache Spark ML-lib as a computationally independent machine learning library, which is open-source, distributed, scalable, and platform. We have evaluated and compared several ML algorithms to analyze the platform’s qualities, compared Apache Spark ML-lib against Rapid Miner and Sklearn, which are two additional Big data and machine learning processing platforms. Logistic Classifier (LC), Decision Tree Classifier (DTC), Random Forest Classifier (RFC), and Gradient Boosted Tree Classifier (GBTC) are four machine learning algorithms that are compared across platforms. In addition, we have tested general regression methods such as Linear Regressor (LR), Decision Tree Regressor (DTR), Random Forest Regressor (RFR), and Gradient Boosted Tree Regressor (GBTR) on SUSY and Higgs datasets. Moreover, we have evaluated the unsupervised learning methods like K-means and Gaussian Mixer Models on the data set SUSY and Hepmass to determine the robustness of PySpark, in comparison with the classification and regression models. We used "SUSY," "HIGGS," "BANK," and "HEPMASS" dataset from the UCI data repository. We also talk about recent developments in the research into Big Data machines and provide future research directions.

Keywords Big Data · Predictive analytic · Machine Learning · PySpark ML-lib · Rapid Miner · SK-Learn

Nawab Muhammad Faseeh Qureshi
faseeh@skku.edu

Extended author information available on the last page of the article

Published online: 23 August 2022
1 Introduction

The rapid improvement in industrial operations and technology, the internet of things, smart gadgets, and social media, digital data has grown in volume and complexity at a rapid rate [1–4]. Big data is a collection of records with a large volume and a rate of exponential growth over time [5]. For example, the New York Stock Exchange market creates over 1TB of new data per day, social media creates more than 500TB, and a Jet engine creates around 10TB of data in only 30 minutes, according to the report. The International Data Corporation keeps track of how much data is generated throughout the world. The data is rising exponentially during the year, as seen in Fig. 1 [6]. If current trends continue, data science agencies project that by 2030, there will be around 78 Zeta bytes of digital data [7].

Due to the quantity and nature of big data, traditional approaches are no longer able to cope. In addition, conventional techniques for data analysis cannot handle vast amounts of data in real-time. As the amount of dynamic data being processed grows every day, researchers are attempting to create new algorithms that exceed traditional approaches totally. The objective is to create a better computing system to rapidly and economically process and analyze large-scale data. These architectures need sophisticated computing algorithms capable of concurrently processing and monitoring data. Apache Spark is one of the most robust frameworks for managing and successfully handling the real-time streaming of large data. The framework addresses the problems of reliability, speed, and amount of data in big data analytic to delivers exceptional quality, and resource pooling power [8] as shown in Figs. 2 and 3.

Apache Spark is a clustering computation framework that supports Java, R, Scala, and Python [9]. It is built on top of the Hadoop platform and operates in the background to facilitate the delivery of trustworthy, consistent, and high-speed computing on huge data [10–12]. PySpark ML-lib is developed on top of Spark to take advantage of Spark’s advantages, such as its in-memory columnar data format, enabling data transmission between Java Virtual Machine (JVM) and Python methods at a low cost. Python users who work with Pandas and NumPy data will benefit from it. PySpark will be more
beneficial in conjunction with other python libraries with only minor setup adjustments. In addition, the machine learning functions and syntax are comparable to those of the Python script. This improves the advancement of in-memory programming approaches and top-level data processing frameworks [13, 14]. PySpart frameworks have demonstrated successfully in stochastic modeling [15], healthcare informative [16], text mining [17], genomics data analysis [18, 19], etc.

Sci-kit-learn also offers an easy-to-use and reliable Python learning software for large data analysis. It supports Python computational and scientific libraries such as NumPy, python, and SciPy, and supports such techniques as vector support, random forests, and k-neighbors [20]. The Rapid Miner also provides advanced data processing, machine learning, in-depth learning, text mining, and predictive analytical application. It includes all facets of the processes of machine learning, including data processing, visualization of outcomes, model evaluation, and optimization [21]. It is used by business, industrial and science, education.

This study investigates the impact of data volume on computational capability. Based on the findings, the study identifies potential big data machine learning-oriented prototype layouts and their benefits. This paper aims to show the PySpark ML-lib and its capabilities of applying machine learning on big data from the computational aspect. We highlight the advantages of such a highly scalable machine learning architecture by rigorous testing and comparison to other frameworks such as SK-learn and Rapid Miner.

We highlight the advantages of such a highly scalable machine learning architecture by rigorous testing and comparison to other frameworks such as SK-learn and Rapid Miner. The following is a quick summary of our major contributions:

- Although machine learning and its science and business applications have been studied over the last two decades. Research has been limited on big data machine learning. We are evaluating Apache Spark ML-lib big data learning analysis of millions of data records. The main part of the paper is to provide the machine learning community with a fascinating yet complex topic that allows the big data community to move on a new and exciting path to the rapidly expanding application machine learning industry.
- We perform several real-world big data trials to examine a number of qualitative and quantitative facets of PySpark ML-lib. A comparative analysis is also carried out with the Rapid Miner API, MATLAB and the Library of SK-learn, which has been widely used in science. We evaluate varied traditional models for the study of Big Data Machine learning methods, classification, and regression, including Logistic/linear, Gradient boost tree, Random Forest, and Decision Tree. Furthermore, We evaluated
unsupervised machine learning approaches like Kmeans and Gaussian Mixture Model (GMM).

The rest of the article is as follows: The literature review in ML-lib and Apache Spark is presented in following Sect. 2. Section 3 starts with an overview of big data technologies and a detailed explanation of Apache Spark, Rapid-Miner, and SK-learn as powerful big data platforms. Sections 4 deals with the methods used in the study, broadly classified as supervised and unsupervised learning. Section 5 presents an explanation of data sets and the data pre-processing involved in preparing them and the analysis results. At last, future work and conclusion are presented in Sect. 6.

2 Related Works

Godson Michael [22] explains the Internet of Things (IoT) in which packets are only transmitted over the network for correspondence if both the sender and the recipient are connected to the internet. To accomplish these goals, they proposed architecture that employs open-source technology such as Apache Kafka for real and virtual message processing, and Apache Spark for streaming, processing, and structuring real-time and historical data. The stored data is presented in a more appealing and understandable way using the Dashing framework. In [23], authors focused on Apache Spark’s benefits compared to the Hadoop Map Reduction and the use of time-series analyses in real-time analysis. It takes plenty of time and storage for this method of reading and writing data on the hard drive and HDFS. They used Apache Spark, which is an open-source project for faster queries, IT, and Real-time performance processing. The focus of this paper is also to study the time series Hadoop and Spark, which calculates and analyzes data in real-time. In [24], the author’s survey aims to provide a comprehensive overview of numerous optimization strategies for improving Spark’s generality and efficiency. They present the Spark programming paradigm and computation method, address the benefits and drawbacks of Spark, and do a literature review and classification of different solving strategies. Furthermore, they incorporate numerous data storage and analysis methods, machine learning algorithms, and Spark-supported applications in the article. V Srinivas [25] explored Spark as a general data analytic tool for distributing clusters based on Hadoop. In addition to reducing maps, it provides memory calculations to increase storage efficiency and handling. Structured data can be processed on Hive and streamed from HDFS, Flume, Twitters, and Kafka. It runs over an existing Hadoop cluster and is accessible through the Hadoop database (HDFS). They also note the modular nature of the API (based on distributed object sets forwarding) making it easy to customize. Jobs may be grouped and reviewed locally into portable repositories. Mehdi.A [26] analyzed in order to improve open-source Big Data Spark machine learning They even make a distinction between Spark and Weka in this article. In terms of efficiency, speed, and data handling capacity, Spark clearly outperforms Weka. Weka has a user-friendly GUI as well as a large number of built-in algorithms. A. Shoro [27] explored Apache Spark software for big data analysis and its capabilities for Twitter’s diverse giant records streaming properties. they considered Apace spark to be very successful after using data sets for Twitter sentiment analysis. They also figure out that Spark is difficult to use and that data visualizations should be improved. S. Salloum, 2016 [28] performed evaluated
various classification data mining strategies based on data from the University of California Machines Learning Repository (UCI). Several data mining strategies were examined and time complexity have been computed for each classification.

3 Big Data Technologies

In recent years, increasingly important has been creating Big Data apps. In actual fact, an increasing number of companies in many sectors depend on huge data collection. The Big Data is less cost-efficient, on the contrary, in traditional data methods and networks. They have a bad response time and lack of scalability, performance and accuracy. A great deal was undertaken to solve the complex problems of big data. This has led to the emergence of various delivery networks and innovations. Hadoop, Spark, NO-SQL, Sklearn and Weka libraries, Hive, Cloud, and Rapid Miner technologies are gaining popularity. These technologies are computer software tools for extracting, managing, and analyzing data from a massively complex and large data collection that traditional management tools would never be able to handle [29–34]. However, in such a setting, selecting among a variety of technologies may be time consuming and difficult. Many factors should be taken into account, including technology compatibility, deployment complexity, cost, efficiency, performance, dependability, support, and security issues [35]. We go through the technologies that are currently being employed in the research study in more detail in the subheading.

3.1 PySpark ML-LIB

Apache Spark is a scalable, fast and storage large data processing platform built at University of California, allowing developers of distributed applications to use Java, Python, Scala and R languages for programming. Apache Spark Streaming, Apache Spark SQL, Apache Spark GraphX and Apache Spark ML-lib are the four most used libraries [19, 26, 36]. In an environment with highly tolerant defects and a lot analysis, stream processing takes place during Apache Spark Streaming, the Spark simple planning module. Apache Spark SQL makes relational queries in mine various database structures with the query-frames data statement principle [37]. Apache GraphX is a graph processing tool for Apache Spark. It offers programming models spread to treat two basic data structures: graphs and arrays. Apache Spark ML-lib is one of 55 machine-learning algorithms that provide information and parallel processes to a vast range of data analyses. The Library uses different algorithms for machine learning such as sorting, classification, regression and reduction [38].

Many engineering and machine-learning scientists have embarked on advance spark ML-lib to assist the big data analysis ecosystem. We look at some of the latest Apache Spark ML-lib applications advancements in this area. [39] CVST, a modular open source framework was proposed. The framework is intended for smart transport applications production. The data analysis part of ML-lib is used to process and feed data to the front end [40]. Built an infrastructure of institutional information systems to provide resources for evaluating student data patterns. The recommended framework forecasts and recommends courses using Apache Spark ML-lib for the next half year [41]. The bigNN, which can handle extremely high biomedical penalty rate categorization, is another fascinating big Data Analytic component that was produced by Apache Spark [42].
3.2 Rapid Miner

The first Rapid Miner, known in 2001 as YALE, was developed by Ralf Klinkenberg, Ingo Mierswa, and Simon Fischer. All the applications which can be carried out in Rapid Miner are statistical modelling, pre-processing data, market analysis, optimising and prediction analysis. It has eight operators and the servers that follow. (a) The process control operator provides a range of parameters, such as logging and random number generator initialization settings of a worldwide significance to the process. (b) The process control operator offers a collection of settings that are of global importance to the process, such as logging and random number generator startup settings. (c) Utility operator is used to insert a process into another process. (d) To access the repositories, utilize the repository access operator. (e) CSV files are read using the Import operator. (f) Almost any type of Input Output Object may be written to a file using Export operator. (g) An example set's characteristics can be renamed using the data transformation operator. (h) For prediction, the modeling operator is utilized. (i) The evaluation operator separates the data set into three categories: training, testing, and estimating performance. Rapid Miner Studio is a code-free platform for designing complex analytic procedures that speed up Hadoop cluster computations [43–45]. We list the most recent advancement in the big data community using Rapid Miner [46]. gave a review of existing data mining algorithms for predicting diabetes disease. Rapid Miner is mostly used to diagnose diabetes, according to their analysis [45]. used Rapid Miner data mining technology will be used to classify massive data and analyze people’s attitudes and opinions on the Facebook social networking site [47]. tested decision trees, regression, SVM, k-NN and Naive Bayes on breast cancer detection using Rapid Miner.

3.3 Sci‑Kit Learn

Sci-Kit learning is a Python program that incorporates a wide variety of cutting-edge learning methods with medium-sized challenges, unattended and monitored. This kit concentrates on educating non-specialists about deep learning using a high-level vocabulary for general purposes. Priority is given to the following: ease of use, consistency, documentation and API reliability. Sci-kit learn works well in big data and support a variety of algorithms [48]. It has few limitations and is licensed under the BSD so it can be used both in the academic and business fields [49–51]. This study in [52] examines hospital health-care systems using machine learning and presents a thorough management strategy and [53] used SK-learn to create comprehensive graphs of the Covid-19 data and forecasted machine learning models to categorize gene sequences so that mutations could be detected fast. The broader study has conducted using big data sets to do textural analysis of hundreds of thousands of legislative speech actions [54].

3.4 MATLAB

MatLab is a numerical computations programming language that is mostly used by engineers and data analysts. It is a unified, strong platform for dealing with Big data [55, 56]. We may choose from a number of toolboxes to expand on the fundamental capabilities that come with it. Matlab is available for use on Unix, Macintosh, and Windows systems, as well as personal computers for students. The fact that MatLab makes data
visualization so simple for the user is its most significant benefit. MatLab enables users to obtain a more natural read on their data instead of depending on some fundamental understanding of coding or computer science principles. Users may retain strict control over the way their data is displayed by using the easy-to-read style in which data is provided both before and after analysis. MatLab has a significant number of dedicated users, including several institutions and a few businesses with the financial means to purchase a license [57]. Despite its widespread usage in colleges, MatLab is simple to learn for novices who are just getting started with programming languages since the package contains everything you’ll need when we buy it. When using Python, we must install additional packages. Simulink, a fundamental element of the MatLab package for which there is currently no suitable equivalent in other programming languages, is one component of MatLab [58].

In the next subsection, We will go through the various machine learning algorithms, as well as datasets and data pre-processing methodologies.

4 Methods

 Supervised learning is a form of machine learning that trains models using label data. Models finds mapping function for mapping the input variable (X) to the output variable in supervised learning (Y). Supervised instruction requires monitoring to train an example which is equivalent in the presence of a teacher, when a pupil learns things. Two kinds of problems can be taught: classification and regression. In fact, supervised learning could be used to evaluate risk, classify images, detect fraud, spam etc. Two types of problems: (a) classification and (b) regression are subject to supervised learning. Random Forest, Decision Trees, Logistic Regression, Support vector Machines are examples of classification while Regression Trees, Non-Linear Regression, Bayesian Linear Regression, Polynomial Regression are famous examples of classification algorithms.

 Unsupervised learning is another approach in which patterns derived from unlabeled input. Objective of Unsupervised learning is to discover the structure and patterns of the input data. Unsupervised learning requires no supervision. Rather, it searches for data models on its own. Two types of problems may include unattended learning: clustering and Association. Unsupervised training cannot be extended to regression or classification problems directly, since we have input data but no equivalent outcome data. The objective is to identify a fundamental data set layout the data in a compact format based on similarities. Some famous algorithms are K-means clustering, Hierarchical clustering, Neural Networks, Apriori algorithm, Principle Component Analysis, KNN (k-nearest neighbors) [59].

4.1 Supervised Learning

Logistic Classifier: It’s a way of forecasting the results of In conventional logistic regression, a binomial or multivariate classification algorithm is used. As the logic regression passes logit function, a linear combination of our input characteristics using gradient descent is minimized. Although logistic regression was introduced in the 1950s as a traditional statistical classification method [60–62].

Consider the model p=P(Y=1), which has two predictors, $x_1$ and $x_2$, and one binary response variable, Y. We assume that the predictor variables and the log-odds of the
event \( Y = 1 \) have a linear relationship. This linear relationship may be expressed mathematically as follows:

\[
l = \log_e \frac{p}{1 - p} = C_0 + C_1 x_1 + C_2 x_2
\]

(1)

where \( C_i \) are parameters/coefficients of the model.

**Decision Tree** They are one of the analytical, processing and studying mathematical simulation approaches. To draw conclusions on the target value from expectations of an object, it uses a decision tree. Classification trees are tree structures in which a different value set may be taken from the target variable; the leaves represent classes, and branches represent functions which correspond to certain labels. Given their understanding and ease, decision trees are one of the most common master algorithms for learning [63, 64].

The majority of decision tree algorithms work top-down, selecting a variable at each step that optimally separates the collection of objects. Different algorithms employ various metrics to determine what is "best." These are used to assess the homogeneity of the target variable across subgroups. These metrics are applied to each potential subgroup, and the resulting values are averaged to offer a measure of the split’s quality.

**Random Forest** Breiman developed the Random Forest ensemble learning algorithm. An ensemble learner approach creates a large number of independent learners and then combines their performance. Random Forest employs a variation of the bagging method. Each classifier in Bagging is designed independently using a bootstrap sample of the input data. A judgment is taken at a node split in a standard decision tree classifier based on all function attributes. The best parameter at each node in a decision tree in Random Forest, on the other hand, is made up of a randomly chosen number of features. This random collection of features allows Random Forest to scale well not only when there are several features per feature vector, but also when there are few features per feature vector [65–67]. Train a classification or regression tree \( R_b \) on given training set \( X \) with responses \( Y \). The number of trees, \( B \) is a hyper-parameter. After training, summing the predictions from all the various regression trees on \( x' \) may be used to make predictions for unknown samples \( x' \):

\[
\hat{R} = \frac{1}{B} \sum_{b=1}^{B} R_b(x')
\]

(2)

**Gradient Boosted Tree** Gradient Boosted Trees have a lot in common with Random Forests. They take advantage of decision trees series, then make the forecast based on the weighted scoring from trees. The main distinction with Gradient Boosted Trees is that the first tree is used to project, and then an additional tree is inserted after it has been validated to mitigate error, i.e. the first tree’s failure. One tree at a time is introduced, each one mitigating the loss of the one before it until a stable model is created. Because of the dependency of earlier trees, trees are introduced sequentially rather than in parallel. Due to the dependency on previous tree calculations trees are added sequentially and not in parallel. This reduces distributed calculation’s output advantage. Gradient boosted tree regressors and classifiers were tested on limited data to measure computational time and precision for our objectives [68].

Gradient boosting fits a decision tree \( D_n(x) \) at the \( nth \) step. Let \( L_n \) be the number of leaves. For input \( x \), the outcome of \( D_n(x) \) may be represented as the sum:
where $P_{Ln}$ is the predicted value in the region $R_{Ln}$.

**Linear Regression** A linear technique to modelling the connection between a response and one or more explanatory factors is called linear regression. Simple linear regression is used when there is only one explanatory variable, whereas multiple linear regression is used when there are more. Let the response variable is $y$ and the independent features are $x_0, x_1, ..., x_n$. The model accept the feature vector in the form as

$$ y_i = w_0 + w_1 x_{i1} + ... + w_n x_{in} + \epsilon_i $$

where $w_0, ..., w_n$ is weight matrix that linearly multiply with the features. $\epsilon$ is the random variable that indicate the noise to the linear equation. To evaluate regression, there are several proposed metrics in the literature and one of these metrics is Root Mean Squared Error (RMSE).

**Root Mean Squared Error** measures how near a regression line is to a set of points. The Root Mean Square Error (RMSE) is the standard deviation of the prediction errors (residuals). The prediction error is a measure of how distant the data points are from the regression line; i.e., how spread out the data points are from the regression line. In other words, lesser value of RMSE shows how tightly the data is clustered around the line of best fit. In climatology, forecasting, and regression analysis, root mean square error is widely used to check experimental results.

$$ RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} y_{residual}^2} $$

### 4.2 Unsupervised Learning

**K-means clustering** The general principle of clustering is to identify in the data some underlying structure, also called classes of related objects. K For clustering, a similarity metric in the form of Euclidean distance is used, regardless of the platform. The basic concept of K Means is also referred to as divisive or partial clustering to start from each data point by dividing a larger cluster into smaller groups dependent on user feedback K (or the number of clusters). A core called the centroid is in every cluster. In each cluster the total number of centros is always the same as K. The algorithm searches for and assigns data points to the nearest cluster. As all data points in each centroid (which represents each cluster here) are allocated, the central values are recalculated and the procedure repeats before the clusters meet a convergence criterion. For each cluster, centroids are a New Mean and a convergence criterion is a consistency or a compact cluster measurement. Let the data be $x_0, ..., x_m$, the Kmeans clustering algorithm will divide the m observations to k ($\leq m$) sets. Therefore, the aim is:

$$ \arg \min_{i=1}^{k} \sum_{x \in S_i} ||x - \mu_i||^2 = \arg \min_{i=1}^{k} |S_i| VarS_i $$
where $\mu_i$ is the mean and $S_i$ is the $i^{th}$ set in $k$.

**Gaussian mixture model** These are probabilistic models for a general population representing usually distributed sub-populations. Mixture models usually don’t need to know which sub-population a data point belongs to such that the model will automatically learn the sub-population. This is a method of unchecked learning because the task of sub-popularity is unknown. GMMs were also widely used in the object tracking of certain items, which predicted the number of mixture compounds and their means at each frame in a video series. GMMs were used for the extraction of features from speech data. The gaussian model weights defined as $\phi_k$ with $\sum_{i=1}^{K} \phi_i = 1$ as a constraint, such that the total probability distribution is normalized to 1. The total probability distribution given as:

$$P(x) = \sum_{i=1}^{K} \phi_i N(x|\mu_i, \sigma_i) \quad (7)$$

where $\sigma$ is the variance and $N$ can be written as:

$$N(x|\mu_i, \sigma_i) = \frac{1}{\sigma_i \sqrt{2\pi}} \exp\left(-\frac{(x - \mu_i)^2}{2\sigma_i^2}\right) \quad (8)$$

**Silhouette Score** The term ”silhouette” refers to a method of interpreting and assessing consistency within data clusters. The process includes a brief graphic representation of each object’s classification. The silhouette is between 1 and +1, with a high value indicating that the item fits well with its own cluster but not with the clusters around it. If the majority of the objects have a high value, the cluster architecture is advantageous. There may be too many or too few clusters in the setup if some points have low or negative values. The Silhouette score is calculated as follows:

$$S_i = \frac{q(i) - p(i)}{\max(p(i), q(i))} \quad (9)$$

For $i^{th}$ data point, $S_i$ is the Silhouette coefficient, where $p(i)$ represents the average distance between $i$ and all other data points in the cluster to which $i$ belongs, and $q(i)$ represents the lowest average distance between $i$ and all clusters to which $i$ does not belong.

## 5 Results and Discussion

### 5.1 System Environment

We performed our experiment on Windows 10, 64 bit Operating System. The hardware configuration is Intel CPU core i7-9700K with 48GB DDR4 memory.

### 5.2 Datasets

Four various broad data sets, namely Hepmass, SUSY, Higgs, and Bank marketing have been used in our analysis. The raw data sets have downloaded from UCI Machine Learning Repository. The ”HEPMASS” data includes powerful physics experiments to look for exotic particles and a bi-classification task. The ”SUSY” data are referencing to distinguish between a signal process that creates super-symmetric particles and a background process.
that do not match with the phenomenon. The "HIGGS" data are collection of signal samples that used to determine whether a particular signal emits Higgs Bosons or does not. The Bank data set related to the customer in a marketing campaigns whether a person will subscribe to a bank service or not [69]. Table 1 shows the detailed characteristics of each data set. The total number of classes in each data set is two, have real values.

5.3 Data Prepossessing

We have used four sets of data for our experiments and have split them in 7:3 ration, training and testing, respectively. Except for banking marketing, others are numeric data type. Therefore, We used the technique One hot encoding in PySpark because many machine learning algorithms do not operate on string and other data types. In addition, the classes were imbalanced, we applied oversampling the bank data to avoid the problem of overfitting [70]. The raw data sets were in text format, we converted them to CVS format. The categorical encoding method has used in SK-learn library to convert the label to a numerical vector. For Rapid miner and MATLAB use auto preparation data modeling to handle the data. In case of PySpark, a dense vector has created from the data frame. The dense vector has two columns, namely, label and features. The label columns contains the original class in the data frame, whereas, the features compose of all the detests features in a single length vector [71].

5.4 Results

We have concentrated on Logistic, Decision Tree, Gradient-boosted Tree, and Random Forest, the supervised classification (Fig. 4) and Regression approaches (Fig. 6), as well as the unsupervised methods, such as K-means and Gaussian Mixture Model. All machine learning classification and clustering algorithms are applied in three different environments: PySpark Ml lib, Rapid miner, MATLAB and Sci-Kit learn. The results of the classification methods comparison are shown in Fig. 4. The graph depicts the ROC in each data set as it is applied to classification algorithms in various environments. The accuracy in each set of data in different environmental settings is depicted in Fig. 5.

In the case of PySpark, the Gradient-boosted tree classifier has a maximum ROC value of 0.91, and the highest accuracy of about 92 percent for the bank data set. Rapid Miner also suggests a Gradient-boost tree to achieve maximum accuracy in a small data set. In contrast, the random forest classifier was more accurate, about 90 percent, than other classifiers in the SK-learn environment. In constrast, MATLAB perform better than sklearn and rapidminer in all the cases. Whereas, it has similar performance with pyspark. One should consider using PySpark to achieve the highest accuracy.

| Dataset   | Characteristics | Attributes | Data record | Size (MB) |
|-----------|-----------------|------------|-------------|-----------|
| Bank marketing | Multivariate    | Real       | 76946       | 6.12      |
| SUSY      | Multivariate    | Real       | 50000000    | 2220      |
| HEPMASS   | Multivariate    | Real       | 10500000    | 7230      |
| HIGGS     | Multivariate    | Real       | 110000000   | 7480      |
Fig. 4 Classification algorithms Comparison of SK-learn, Rapid miner, MATLAB and Apache Spark ML-lib with respect to ROC parameter

Fig. 5 Classification algorithms Comparison of SK-learn, Rapid miner, MATLAB and Apache Spark ML-lib with respect to Accuracy
Considering the medium data sets, PySpark has achieved the highest accuracy of SUSY data set, about 80 percent with Gradient-boost (ROC = 0.829), and Hepmass data set about 89 percent with Random Forest method (ROc = 0.832). Rapid Miner reports that Gradient-boosted tree is better with both data sets, having accuracy, 64 percent (SUSY) and 86 percent (Hepmass), and ROC, 0.819 and 0.944. Similarly, Sklearn also suggests a Gradient-boosted Tree. However, the ROC value is lesser than PySpark and Rapid Miner. The second option would be MATLAB to get the adequate accuracies in comparison to other ML frameworks. The highest ROC value on MATLAB is 0.916 for the gradient-based tree for big data. It achieved 0.929 ROC value for the small dataset.

The random forest on the PySpark platform achieved maximum accuracy of 89.4 percent with a ROC of approximately 0.70 (Maximum) for the largest data set, HIGGS, in our study. Whereas, Gradient-boosted tree on Rapid Miner and Sklearn gives 62.4 percent and 71.4 percent accuracy with 0.68 and 0.71 ROC values, respectively. However, MATLAB perform better than sklearn and rapidminer with highest accuracy of 90.78 percent for bank (small) dataset, maximum average 84.66 percent for big datasets (Higgs, Susy, and Hepmass).

Moreover, we used ML regression approaches to investigate the impact of large data on the platforms. To determine the strength of various platforms, we used the Susy and Higgs data sets only. Linear regression and random forest achieve 0.20 RMSE for Susy data, whereas MATLAB outperforms the SK learn package with a minimum error of 0.26 as shown in Fig. 6. In the context of linear regression, PySpark and MATLAB is better than Rapid Miner with error difference of 0.05 for huge data, such as Higgs. PySpark and Rapid Miner have reported least error of 0.175 and 0.18, respectively. On Sklearn, MATLAB RapidMiner, and PySpark, Fig. 7 displays the time spent by each method. PySpark has outperformed other platforms in all of the techniques.

In addition to classification and regression models, we have evaluated the unsupervised machine learning methods such as K-means and Gaussian Mixer Model on SUSY and Hepmass data set to check the robustness of PySpark against Sklearn and MATLAB. We analyse the clustering approaches in terms of time complexity and Silhouette Score. Fig. 8 shows a comparison in the time require to cluster the SUSY and Hepmass data sets. In both data sets, the Gaussian Mixer Model (GMM) takes less time than the K-means. PySpark takes 38.5 percent less time than the Sklearn and MATLAB on K-means methods, and 27 per cent less on GMM. The graphs depicts that, in unsupervised machine learning,
the PySpark ML-lib outperforms the SK-learn and MATLAB package. With the SUSY and Hepmass data sets, Fig. 9 shows the Silhouetter Score between Kmeans and GMM. PySpark surpassed Sklearn and MATLAB with a silhouetter score of 86.57 percent using GMM. Sklearn performs marginally better than PySpark for Hepmass data.

Findings of our research indicate Apache Spark ML-lib is a powerful platform for big data analytic. Rapid miner, MATLAB and Sklearn, on the other hand, are slower than
Apache Spark ML-lib when dealing with large amounts of data. This analogy might not be fair due to the various file systems and configurations used by Apache Spark ML-lib, rapid miner, and Sci-kit learn library.

However, our aim is to demonstrate how Spark performs with massive data sets. Rapid miner, MATLAB and Sklearn are used as a valid baseline that is widely accepted in the research community. Rapid miner, MATLAB and Sklearn have a number of advantages over Spark, including: (1) Users have access to a large number of documentation and resources; (2) It is very simple and convenient to use for non-expert users. (3) Rapid miner, MATLAB provides an excellent user interface and GPU computing support. Sklearn, MATLAB and rapid miner also have a large selection of machine learning techniques.

Figure 5 shows that the Apache Spark ML-lib is able, as predicted, to be faster than the Sci-Kit Learn library used and that the test has been carried out during the operational period, which corresponds to either the grading algorithms or the clustering method showed that Apache Pyspark ML-lib is faster.

6 Future Work and Conclusion

In order to turn this wide range of ordered and even non-structured data into highly qualified information and facts, data processing has to evolve on a growing number of occasions. Digital data is being collected. This paper analyzes and contrasts the latest technologies in big data. The aim is to help consumers pick and apply the right mix of big data technologies for their particular application. It not only presents a high-level overview of various Big Data technologies, but it also compares them using many large data sets and well-known machine learning methods and tools such as Rapid miner and Sci-Kit learn ML library. It classifies and evaluates the key features, advantages, drawbacks, and applications of various technologies.

Using a number of larger data sets, we want to incorporate more realistic tests for most of the Apache Spark ML-lib components in the future and also, We are aiming for an experimental evaluation of ML-lib Apache Spark using a series of broad data sets of diverse characteristics on a selection of programming languages, tools (e.g. Python, Scala, Weka and R), clusters or on various hardware and device setups. Also, we will calculate the hardware resources (Memory, CPU, GPU etc) used by the different platforms during the experimental evaluation.

Acknowledgements This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2021R1A2C1011198) and INHA UNIVERSITY Research Grant.

References

1. D’silva, G.M., Khan, A., & Bari, S., et al. Real-time processing of iot events with historic data using apache kafka and apache spark with dashing framework, in 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT) (IEEE, 2017), pp. 1804–1809
2. Maheshwar, R.C., & Haritha, D. Survey on high performance analytics of bigdata with apache spark, in *2016 International Conference on Advanced Communication Control and Computing Technologies (ICACCCCT)* (IEEE, 2016), pp. 721–725.

3. Al-Barznji, K., & Atanassov, A. (2018). Big Data Sentiment Analysis Using Machine Learning Algorithms, in *Proceedings of 26th International Symposium “Control of Energy, Industrial and Ecological Systems, Bankia, Bulgaria* (2018).

4. Jagadish, H. V. (2015). Big data and science: Myths and reality. *Big Data Research*, 4(1), 1–5.

5. Kejela, G., Esteves, R.M., & Rong, C. Predictive analytics of sensor data using distributed machine learning techniques, in *2014 IEEE 6th international conference on cloud computing technology and science* (IEEE, 2014), pp. 626–631.

6. Zhou, L., Pan, S., Wang, J., & Vasilakos, A. V. (2017). Machine learning on big data: Opportunities and challenges. *Neurocomputing*, 237, 350.

7. Seagate. [https://www.seagate.com/](https://www.seagate.com/)

8. Assefi, M., Behravesh, E., Liu, G., & Tafti, A.P. Big data machine learning using apache spark MLlib, in *2017 IEEE International Conference on Big Data (Big Data)* (IEEE, 2017), pp. 3492–3498.

9. Aziz, K., Zaidouni, D., & Bellafkih, M. Real-time data analysis using Spark and Hadoop, in *2018 4th International Conference on Optimization and Applications (ICOA)* (2018), pp. 1–6. [https://doi.org/10.1109/ICOA.2018.8370593](https://doi.org/10.1109/ICOA.2018.8370593)

10. Shoro, A. G & Soomro, T. R. (2015). “Big data analysis: Apache spark perspective”, *Global Journal of Computer Science and Technology*, 15(1).

11. Armbrust, M., Das, T., Torres, J., Yavuz, B., Zhu, S., Xin, R., Ghodsi, A., Stoica, I., Zaharia, M. (2018). Structured Streaming: A Declarative API for Real-Time Applications in Apache Spark, in *Proceedings of the 2018 International Conference on Management of Data* (2018), SIGMOD '18, p. 601–613. [https://doi.org/10.1145/3183713.3190664](https://doi.org/10.1145/3183713.3190664)

12. L’heureux, A., Grolinger, K., Elyamany, H. F., & Capretz, M. A. (2017). Machine learning with big data: Challenges and approaches. *IEEE Access*, 5, 7776.

13. Reyes-Ortiz, J. L., Oneto, L., & Anguita, D. (2015). Big data analytics in the cloud: Spark on hadoop vs mpi/openmp on beowulf. *Procedia Computer Science*, 53, 121.

14. Dahiya, P., Chaithra, B., & Kumari, U. (2017). Survey on big data using Apache Hadoop and Spark. *International Journal of Computer Engineering In Research Trends*, 53, 121.

15. Jonnalagadda, V. S., Srikanth, P., Thumati, K., & Nallamala, S. H. (2016). A review study of apache spark in big data processing. *International Journal of Computer Science Trends and Technology (IJICT)*, 4(3), 93.
26. Zaharia, M., Xin, R. S., Wendell, P., Das, T., Armbrust, M., Dave, A., et al. (2016). Apache spark: A unified engine for big data processing. *Communications of the ACM*, 59(11), 56.

27. Athmaja, S., Hanumanthappa, M., & Kavitha, V. A survey of machine learning algorithms for big data analytics, in *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIECS)* (IEEE, 2017), pp. 1–4

28. Venkataraman, S., Panda, A., Ousterhout, K., Armbrust, M., Ghodsi, A., Franklin, M.J., Recht, B., & Stoica, I. (2017). Drizzle: Fast and adaptable stream processing at scale, in *Proceedings of the 26th Symposium on Operating Systems Principles*, pp. 374–389

29. Al-Saqqa, S., Al-Naymat, G., & Awajan, A. (2018). A large-scale sentiment data classification for online reviews under apache spark. *Procedia Computer Science*, 141, 183

30. Emani, C. K., Cullot, N., & Nicolle, C. (2015). Understandable big data: A survey. *Computer Science Review*, 17, 70

31. Omar, H. K., & Jumaa, A. K. (2019). Big Data Analysis Using Apache Spark MLlib and Hadoop HDFS with scala and java. *Kurdistan Journal of Applied Research*, 4(1), 7

32. Hafez, M.M., Shehab, M.E., El Fakharany, & E., et al. Effective selection of machine learning algorithms for big data analytics using apache spark, in *International Conference on Advanced Intelligent Systems and Informatics* (Springer, 2016), pp. 692–704

33. Qureshi, N.M.F., Bashir, A.K., Siddiqui, I.F., Abbas, A., Choi, K., & Shin, D.R. A knowledge-based path optimization technique for cognitive nodes in smart grid, in *2018 IEEE global communications conference (GLOBECOM)* (IEEE, 2018), pp. 1–6

34. Siddiqui, I. F., Qureshi, N. M. F., Chowdhry, B. S., & Uqaili, M. A. (2019). Edge-node-aware adaptive data processing framework for smart grid. *Wireless Personal Communications*, 106(1), 179.

35. Qureshi, N.M.F., Siddiqui, I.F., Abbas, A., Bashir, A.K., Choi, K., Kim, J., & Shin, D.R. Dynamic container-based resource management framework of spark ecosystem, in *2019 21st international conference on advanced communication technology (ICACT)* (IEEE, 2019), pp. 522–526

36. pyspark.mllib package – PySpark 2.0.0 documentation. [https://spark.apache.org/docs/2.0.0/api/python/pyspark.mllib.html](https://spark.apache.org/docs/2.0.0/api/python/pyspark.mllib.html)

37. Siddiqui, I. F., Qureshi, N. M. F., Chowdhry, B. S., & Uqaili, M. A. (2020). Pseudo-cache-based IoT small files management framework in HDFS cluster. *Wireless Personal Communications*, 113(3), 1495.

38. Meng, X., Bradley, J., Yavuz, B., Sparks, E., Venkataraman, S., Liu, D., Freeman, J., Tsai, D., Amde, M., Owen, S., et al. (2016). Milib: Machine learning in apache spark. *The Journal of Machine Learning Research*, 17(1), 1235.

39. Park, W. H., Shin, D. R. & Qureshi, N. M. F. (2021). Effective emotion recognition technique in NLP task over nonlinear big data cluster. *Wireless Communications and Mobile Computing*, 2021, 5840759

40. Lee, M.S., Kim, E., Nam, C.S., & Shin, D.R. Design of educational big data application using spark, in *2017 19th International Conference on Advanced Communication Technology (ICACT)* (IEEE, 2017), pp. 355–357

41. Ye, Z., Tafti, A. P., He, K. Y., Wang, K., & He, M. M. (2016). Sparktext: Biomedical text mining on big data framework. *PloS One*, 11(9), e0162721.

42. Tafti, A.P., Behravesh, E., Assefi, M., LaRose, E., Badger, J., & Mayer, J. A. Doan, D. Page, P. Peissig, bigNN: An open-source big data toolkit focused on biomedical sentence classification, in *2017 IEEE International Conference on Big Data (Big Data)* (IEEE, 2017), pp. 3888–3896

43. RapidMiner Best Data Science and Machine Learning Platform. [https://rapidminer.com/](https://rapidminer.com/)

44. Qureshi, N. M. F., Shin, D. R., Siddiqui, I. F. & Chowdhry, B. S. (2017). Storage-tag-aware scheduler for hadoop cluster. *IEEE Access*, 5, 13742–13755.

45. Siddiqui, I. F., Qureshi, N. M. F., Shaikh, M. A., Chowdhry, B. S., Abbas, A., Bashir, A. K. & Lee, S. U. J. (2019). Stuck-at fault analytics of IoT devices using knowledge-based data processing strategy in smart grid. *Wireless Personal Communications*, 106(4), 1969–1983.

46. Prekopcsak, Z., Makrai, G., Henk, T., & Gaspar-Papanek, C. Radoop: Analyzing big data with rapidminer and hadoop, in *Proceedings of the 2nd RapidMiner community meeting and conference (RCOMM 2011)* (Citeseer, 2011), pp. 1–12

47. Wagan, S. A., Junaid, M., Qureshi, N. M. F., Shin, D. R. & Choi, K. (2020). Comparative survey on big data security applications, A blink on interactive security mechanism in apache ozone. In 2020 *Global Conference on Wireless and Optical Technologies (GCWOT)* (pp. 1–6). IEEE.

48. Jo, J. M. (2019). Effectiveness of normalization pre-processing of big data to the machine learning performance. *The Journal of the Korea institute of electronic communication sciences*, 14(3), 547.

49. scikit-learn: machine learning in python – scikit-learn 0.24.2 documentation. [https://scikit-learn.org](https://scikit-learn.org)
50. Qureshi, N. M. F., Farah, I., Siddiqui, B. S. C. & Shin, D. R. (2022). Intelligent MapReduce technique for energy harvesting through IoT devices. *Energy Harvesting in Wireless Sensor Networks and Internet of Things*, p. 259.

51. Chary, D. (2020). Review on Advanced Machine Learning Model: Scikit-Learn

52. Qureshi, N. M. F., Siddiqui, I. F., Unar, M. A., Uqaili, M. A., Nam, C. S., Shin, D. R., Kim, J., Bashir, A. K. & Abbas, A. (2019). An aggregate mapreduce data block placement strategy for wireless IoT edge nodes in smart grid. *Wireless personal communications*, 106(4), 2225–2236.

53. Park, W., Qureshi, N. M. F. & Shin, D. R. (2022). Pseudo NLP joint spam classification technique for big data cluster. *Computers, Materials and Continua*, 71(1), 517–535.

54. Fournier-Tombs, E., & MacKenzie, M. K. (2021). Big data and democratic speech: Predicting deliberative quality using machine learning techniques. *Methodological Innovations*, 14(2), 20597991211010416.

55. Ergün, B., & Şahin, C. Laser point cloud segmentation in MATLAB, in *MATLAB* (IntechOpen, 2021)

56. Junaid, M., Wagan, S. A., Qureshi, N. M. F., Nam, C. S. & Shin, D. R. (2020). Big data predictive analytics for apache spark using machine learning. In *2020 Global Conference on Wireless and Optical Technologies (GCWOT)* (pp. 1–7). IEEE.

57. Ozgur, C. (2021). Matlab vs. Python vs. R | Journal of Data Science | School of Statistics, Renmin University of China. https://jds-online.org/journal/ JDS/ article/ 402/ info

58. Kamangar, Z. U., Siddiqui, I. F., Arain, Q. A., Kamangar, U. A. & Qureshi, N. M. F. (2021). Personality characteristic-based enhanced software testing levels for crowd outsourcing environment. *KSII Transactions on Internet and Information Systems (TIIS)*, 15(8), 2974–2992.

59. Mohit, R. R. V., Katoch, S., Vanjare, A., & Omkar, S. (2015). Classification of complex UCI datasets using machine learning algorithms using hadoop. *International Journal of Computer Science and Software Engineering (IJCSSE)*, 4(7), 190.

60. Peng, H., Liang, D., & Choi, C. (2013). Evaluating parallel logistic regression models, in *2013 IEEE International Conference on Big Data* , pp. 119–126. https://doi.org/10.1109/BigData.2013.6691743

61. Duan, R., Ning, Y., Shi, J., Carroll, R.J., Cai, T., & Chen, Y. (2021). On the global identifiability of logistic regression models with misclassified outcomes, arXiv preprint arXiv:2103.12846

62. Abarda, A., Bentaleb, Y., El Moudden, M., Dakkon, M., Azhari, M., Zerouaoui, J., Ettaki, B. (2018). Solving the problem of latent class selection, in *Proceedings of the International Conference on Learning and Optimization Algorithms: Theory and Applications* (2018), pp. 1–6

63. Charbuty, B., & Abdulazeez, A. (2021). Classification based on decision tree algorithm for machine learning. *Journal of Applied Science and Technology Trends*, 2(01), 20.

64. Sajja, V.R., Lakshmi, P.J., Naik, D.B., Kalluri, H.K. Student Performance Monitoring System Using Decision Tree Classifier, in *Machine Intelligence and Soft Computing* (Springer, 2021), pp. 393–407

65. Azhari, M., Abarda, A., Alaloui, A., Ettaki, B., & Zerouaoui, J. (2020). Detection of pulsar candidates using bagging method. *Procedia Computer Science*, 170, 1096.

66. Azhari, M., Alaloui, A., Abarda, A., Ettaki, B., & Zerouaoui, J. A comparison of random forest methods for solving the problem of pulsar search, in *The Proceedings of the Third International Conference on Smart City Applications* (Springer, 2019), pp. 796–807

67. Cowling, T. E., Cromwell, D. A., Bellot, A., Sharples, L. D., & van der Meulen, I. (2021). Logistic regression and machine learning predicted patient mortality from large sets of diagnosis codes comparably. *Journal of Clinical Epidemiology*, 133, 43.

68. Friedman, J. H. (2002). Stochastic gradient boosting. *Computational statistics & Data Analysis*, 38(4), 367.

69. UCI machine learning repository: Bank marketing data set. https://archive.ics.uci.edu/ml/datasets/bank+marketing

70. Fajardo, V. A., Findlay, D., Jaïswal, C., Yin, X., Houmanfar, R., Xie, H., Liang, J., She, X., & Emerson, D. (2021). On oversampling imbalanced data with deep conditional generative models. *Expert Systems with Applications*, 169, 114463.

71. Jiang, Z., Pan, T., Zhang, C., & Yang, J. (2021). A new oversampling method based on the classification contribution degree. *Symmetry*, 13(2), 194.

**Publisher’s Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.
Muhammad Junaid graduated from the Department of Computer and Electrical Engineering, Sungkyunkwan University, Suwon, South Korea, in 2020 with a master’s degree in computer science. He is currently pursuing a Ph.D. degree in the same department. His current research interests include machine learning, deep learning, biomedical image processing, Big data, and explainability. As an enthusiastic researcher, his goal is to combine state-of-the-art machine learning, and deep learning algorithms with Medical domain problems.

Sajid Ali graduated from JNTU in Hyderabad, India, with a bachelor’s degree in mechanical engineering in 2014, and from Osmania University in Hyderabad, India, with a master’s degree in automation and robotics in 2017. He began working as an assistant professor in the department of mechanical engineering at NMR Engineering College in Hyderabad, India, in 2017. He has been pursuing his Ph.D. degree in electrical and computer engineering at Sungkyunkwan University in South Korea since March 2019. His current research interests include machine learning, deep learning, biomedical image processing, meta learning, and explainability are some of his current research interests.

Isma Farah Siddiqui received her B.E. degree in Software Engineering from Mehran University of Engineering and Technology, Pakistan, in 2006 and M.E. degree in Information Technology from Mehran University of Engineering and Technology, Pakistan, in 2008. She received her Ph.D. from Department of Computer Science and Engineering at Hanyang UniversityERICA, South Korea. Currently, she is serving as Associate Professor in the Department of Software Engineering at Mehran University of Engineering and Technology. Her research interests include Smart Environment, Semantic Web, IoT and Big Data.
ChoonSung Nam received B.S., M.S., and Ph.D. degrees in Software from SangMyung University, SoongSil University, 2007, and in Computer Science from Sunkyunkwan University, 2011, in Korea. He had worked for unmanned vehicle in Sunkyunkwan University as a Postdoc during 2011~2014. He had worked for VANET and UI/UX for 3D-Touch in Yonsei University as a Postdoc during 2014~2016. Currently, he is working for Drone, Force-Touch and e-Call Service in Inha University as a assistant professor. His research interests include VANET, Sensor Networks, Drone, IoT, UI/UX for 3D-Touch and so on.

Nawab Muhammad Faseeh Qureshi is an Assistant Professor at Sungkyunkwan University, Seoul, South Korea. He received Ph. D. in Computer Engineering from one of the TOP five South Korean universities Sungkyunkwan University, South Korea. He was awarded the 1st Superior Research Award from the College of Information and Communication Engineering that is known as the most prestigious graduation award based on his research contributions and performance. He has served as Guest Editor in many prestigious Journals. Also, he has served as General Chair Workshop in conferences that includes IEEE Wireless Communications and Networking Conference (WCNC2020) and IEEE Globecom 2020. In addition to that, he has served as Proceedings Chair in Global Conference on Wireless & Optical Technologies consecutively for the fifth time. He is a reviewer of various prestigious IEEE and ACM Journals. He has evaluated several theses as external Ph.D. thesis evaluator. Also, he has facilitated several institutes and conferences with keynotes and Webinars on Big data analysis and Modern Technology convergence. He is an active Senior Member of IEEE, KSII (Korean Society for Internet Information), and IEICE (Institute of Electronics, Information and Communication Engineers). His research interests include big data analytics, machine learning, deep learning, context-aware data processing of the Internet of Things, and cloud computing.

Jaehyoun Kim received his B.S. degree in mathematics from Sungkyunkwan University, Seoul, Korea, M.S. degree in computer science from Western Illinois University and Ph.D. degrees in computer science from Illinois Institute of Technology in USA. He was a Chief Technology Officer at Kookmin Bank in Korea before he joined the Department of Computer Education at Sungkyunkwan University in March 2002. Currently he is a professor at Sungkyunkwan University. His research interests include software engineering and architecture, e-Learning, SNS and communication, internet business related policy and computer based learning.
Dong Ryeol Shin received the B.S., M.S., and Ph.D. degrees in Electrical Engineering from the Sungkyunkwan University in 1980, the Korea Advanced Institute of Science and Technology (KAIST) in 1982, and the Georgia Institute of Technology in 1992, respectively. During 1992–1994, he had worked for Samsung Data Systems, Korea, where he was involved in the research of Intelligent Transportation Systems. Since 1994, he has been with the Department of Computer Science and Engineering at Sungkyunkwan University where he is currently a Professor in Network Research Group. His current research interests lie in the areas of mobile network, ubiquitous computing, cloud computing, and bioinformatics. And he is actively involved in the security of vehicular area networks, and the implementation and analysis of big data platform, applicable to 3D image processing of robotic arms.

Authors and Affiliations

Muhammad Junaid¹ · Sajid Ali² · Isma Farah Siddiqui⁴ · Choonsung Nam⁵ · Nawab Muhammad Faseeh Qureshi³ 🌐 · Jaehyoun Kim³ · Dong Ryeol Shin¹

Muhammad Junaid
mjunaid@skku.edu

Sajid Ali
sajidali@skku.edu

Isma Farah Siddiqui
isma.farah@faculty.muet.edu.pk

Choonsung Nam
namgun99@inha.ac.kr

Jaehyoun Kim
jaekim@skku.edu

Dong Ryeol Shin
drshin@skku.edu

¹ Department of Electrical and Computer Engineering, Sungkyunkwan University, Suwon, South Korea
² Department of Computer Science and Engineering, Sungkyunkwan University, Suwon, South Korea
³ Department of Computer Education, Sungkyunkwan University, Seoul, South Korea
⁴ Department of Software Engineering, Mehran University of Engineering and Technology, Jamshoro, Pakistan
⁵ Department of Software Convergence Engineering, Inha University, Incheon, South Korea