Large scale data analysis using MLlib

By Ahmed Hussein Ali

TELKOMNIKA Telecommunication, Computing, Electronics and Control
Vol. 19, No. 5, October 2021, pp. 1735~1746
ISSN: 1693-6930, accredited First Grade by Kemenristekdikti, Decree No: 21/E/KPT/2018
DOI: 10.12928/TELKOMNIKA.v19i5.21059

Large scale data analysis using MLlib

Ahmed Hussein Ali[1], Maan Nawaf Abbod[2], Mohammed Khamees Khaleel[3], Mostafa Abdulghafoor Mohammed[4], Tole Sutikno[5]
[1]ICCI, Informatics Institute for Postgraduate Studies, Baghdad, Iraq
[1]Computer Science Department, AL Salam University College, Iraq
[2,4]Imam Aadham University College, Iraq
[1,2,3]Department of Computer, College of Education, AL-Iraqia University, Iraq
[5]Department of Electical Engineering, Universitas Ahmad Dahlan, Yogyakarta, Indonesia

Article Info

Article history:

Received Jul 9, 2021
Revised Sep 4, 2021
Accepted Sep 12, 2021

Keywords:

Big data
Data analysis
Machine learning
Open source
Parallel processing
Spark MLlib

ABSTRACT

Recent advancements in the internet, social media, and internet of things (IoT) devices have significantly increased the amount of data generated in a variety of formats. The data must be converted into formats that is easily handled by the data analysis techniques. It is mathematically and physically expensive to apply machine learning algorithms to big and complicated data sets. It is a resource-intensive process that necessitates a huge amount of logical and physical resources. Machine learning is a sophisticated data analytics technology that has gained in importance as a result of the massive amount of data generated daily that needs to be examined. Apache Spark machine learning library (MLlib) is one of the big data analysis platforms that provides a variety of outstanding functions for various machine learning tasks, spanning from classification to regression and dimension reduction. From a computational standpoint, this research investigated Apache Spark MLlib 2.0 as an open source, autonomous, scalable, and distributed learning library. Several real-world machine learning experiments are carried out in order to evaluate the properties of the platform on a qualitative and quantitative level. Some of the fundamental concepts and approaches for developing a scalable data model in a distributed environment are also discussed.

Corresponding Author:

Ahmed Hussein Ali
Department of Computer
AL-Iraqia University
A'dhamia Haibat Khaton Street, 22 Mahalla 308, Baghdad, Iraq
Email: msc.ahmed.h.ali@gmail.com

1. INTRODUCTION

The application of information technology in several fields, such as computing, networking, and storage capacity, has seen significant advancements in the previous decade [1]–[6]. The result of this advancement is the emergence of a new scientific paradigm: the era of huge data collection and exploration, which has evolved into a scientific discovery approach that is on an equal footing with conventional theoretical analysis, experimental designs, and computer simulation. The amount of data generated and stored has expanded dramatically over the last two decades as a result of the development of the internet of things (IoT), artificial intelligence, cloud computing, and other cutting-edge computer technologies [7]–[19]. Because more than 6000 tweets are sent out per second on Twitter, and the similar trend can be found on

Facebook, WhatsApp, and other social media platforms. Social media and the Internet have made substantial contributions to this rate of data generation [20]–[24]. Large amounts of data are generated by application/web servers from their logs at the organizational level. Other systems also contribute to the increased rate of data generation. As a result, data has evolved into an essential component of business existence. Because of the increased level of digital data generation, combined with the increasing complexity of the data. It has become impossible to process them using conventional data processing methods; as a result, efforts are now being directed toward developing advanced computing infrastructures that can handle data volumes and complexity of this magnitude (referred to as parallel and distributed processing) [25]–[42].

Managing vast amounts of data is a difficult task that demands the development of more complex systems in order to achieve accurate and timely enormous data analysis [43]–[45]. In order to process big data analytics problems in a timely and reliable manner, infrastructure for big data has been developed, allowing for high-quality performance and resource availability for self-service and convenience of use on demand. There are numerous machine learning frameworks for large data analysis presently accessible; they are relevant to different scientific domains and have been shown to be useful in healthcare informatics genetic data analysis, text exploration, and random picture modeling, among other applications. Apache Spark machine learning library (MLlib) is an open-source, in-demand, and independent library for big data analysis using machine learning techniques [46]. It has the advantage of having an automatic data balancing and a distributed design, making it a good choice for big data analysis. A collection of dominating people in occupations for numerous machine learning tasks, such as classification, regression, base compilation and extraction (and dimensional reduction), is introduced by Apache Spark [47]–[51]. Despite the fact that numerous research have been conducted on machine learning and its usefulness, ML libraries for big data analysis, such as Apache Spark MLlib, have received little attention. Perhaps this is the first study to look at libraries for large data analysis that are based on machine learning techniques. Big data analytics is primarily concerned with the advancement of computer infrastructures in such a way that data mining and analysis can be completed quickly and efficiently [45], [52]–[56]. It is the primary driving force behind the existing business. Because large data analytics is a computationally intensive operation, the user experience during large data analytics is influenced by the setup of different software and devices [57]–[66].

Several big data processing techniques have been recommended since the last decade due the failure of the conventional processing methods to handle the large volume of data generated daily from business and industrial processes [67]. As a result, researchers have been concentrating their efforts on developing more effective methods of obtaining value-added information from large amounts of data. There are many different types of studies in the area of big data processing models. For example, data flow models such as MapReduce, which facilitate data processing utilizing a variety of operators while sharing stable storage systems, are one type of study [68]–[74]. Resilient distributed datasets (RDDs) are a more efficient data sharing abstraction from stable storage since they do not require data copying, which saves money. In most high-level application programming interfaces (APIs) for data flow systems, integrated language APIs [75]–[81] are provided, which allow the user to interact with "parallel groups" through operators such as map and join. Parallel groups on these systems, on the other hand, either represent files on disk or the temporary data sets that were used for query plan expression on these systems. Despite the fact that systems have the ability to convey data via the operators in the same query, data exchange through inquiries has proven to be inefficient. As a result, Spark's API is built on the parallel summation model, which is convenient to implement. It does not claim to be the first to use an integrated interface language, but by including RDDs as a storage layer behind this interface, it can support a larger range of applications.

The systems in the following category are those that provide high-level interfaces for specialized applications that require data sharing, as described above. Pregel [82] provides support for redundancy diagramming applications, whilst Twister [83] and HaLoop [84] are iterative MapReduce programs, respectively. These frameworks only provide data sharing for the calculation styles that are supported, and they do not provide an universal abstracting framework. They can only be used by the user to share selected data from specified operations. For example, a user cannot load data into memory using Pregel or Twister and then select the query to run on it after it has been loaded. The fact that RDDs expressly provide distributed storage means that it can be used to enable applications that are not currently supported by these specialized systems, such as interactive data mining. According to [85], it is proposed a methodology that demonstrates to be an upgrade over standard big data analytics methods that use either Hadoop/Spark or deep learning as distinct components. Lunga *et al.* [85] proposed a framework that makes use of Spark's distributed computing capabilities as well as deep learning architecture for multiple layers perceptron (MLP) using cascade learning to train multiple layers perceptrons is proposed. A framework for in-depth training learning models with Apache Spark has been created and developed in [47], [48], [50], [51], [57], [69], [86]–[92]. This framework shortens the training time by taking advantage of the advantages of both data and parity modeling at the same time. It is possible to create data parallelism by distributing training data across many

Spark block machines and replicating models on each device [86]. Each model goes through its training in parallel with the data part. The parallelism model is implemented by distributing each replica of the deep neural network model over the spark group in a layer-by-layer fashion.

The impact of various software and hardware configurations on the problem of big data processing is explored in this research. The focus of the presentation is on the capabilities and advantages of Apache Spark MLlib 2.0 as a large data analytics tool, particularly in relation to Hadoop. This study is developed as a means of providing insight into the usage of machine learning libraries in big data analysis from the standpoint of industry. This work opens the door to other elements of big data analysis utilizing machine learning methods, which is regarded to be a rapidly expanding study topic. Several real-world tests is carried out to investigate the qualitative and quantitative aspects of Apache Spark MLlib 2.0. Moreover, a comparative research is carried out using the massive online analytics (MOA) library, which is a well-known Java-based machine learning library that is widely used in the industry. Furthermore, the performance of several commonly used machine learning models for big data analysis is examined, and compared across a variety of software and hardware settings. The remaining part of this article is arranged thus: section 2 introduces Apache Spark MLlib. The method and components of the investigated Apache Spark MLlib 2.0 are presented in section 3, while the results and discussion of the features and benchmarking are presented in section 4, and conclusion is presented in section 5.

2. APACHE SPARK MLLIB 2.0

This is a scalable and fast big data processing engine that was first developed by the AMPLab at the University of California, Berkeley [93]–[95]. It may be used to construct distributed applications in a variety of computer languages including Java, Python, and other programming languages [96]–[105]. When it is installed, it includes four major libraries: Apache Spark structured query language (SQL), Apache Spark Streaming, Apache Spark MLlib, and Apache Spark GraphX. These libraries are described in more detail below [106]–[108]. However, despite the fact that the most basic scheduling Spark modules are Apache Spark Streaming, which is fault tolerant and performs high level analytics, Apache Spark SQL performs relational queries for a variety of mining databases because it incorporates a data abstraction model known as data frames [109]. It is important to note that Apache Spark GraphX [110] is a high-level Apache Spark processing library that can handle two commonly used data structures utilizing distributed arithmetic models. Apache Spark MLlib provides >55 scalable machine learning algorithms for big data analytics, taking advantage of the advantages of both data and the data collection method. As well as enabling the implementation of numerous machine learning strategies, such as grouping and regression; classification; rule extraction; and dimensional reduction. It also enables the rapid and simple creation of machine learning approaches for large-scale applications [67], [111]–[116].

A suite of multiple-language APIs is also available on the Apache Spark MLlib [117] platform for the evaluation and deployment of a wide range of machine learning techniques. In recent years, several changes have been made to multiple areas of data science solutions [118], [119], and a number of academics have committed attention to the creation of the components of Apache Spark MLlib for big data analytics. Figure 1 depicts the development side of Apache Spark MLlib track 2.0, with a unique number of anchors assigned to each release of the library [120]–[122]. This section discusses some of the recent improvements in Apache Spark MLlib applications, including some of the new features introduced. In order to aid in the development of smart transportation applications, a scalable and open-source platform known as connected vehicles and smart transportation (CVST) has been proposed by a number of researchers. The proposed CVST is built of four essential components: data distribution, resource management, business intelligence, and application. The business intelligence component is in charge of data analytics, and it makes use of MLlib to process and transmit data to the front end. According to the findings of the study [107], [123], [124], an architectural design for academic information system services for students enrollment pattern analysis should be considered. This system makes use of MLlib to anticipate the suggested courses for the forthcoming semester, which is a powerful prediction tool.

Sparktext is a text mining framework developed by Ye *et al*. [125] for use with Apache Spark learning and flow algorithms in conjunction with the Cassandra NoSQL database [90], [126]–[129]. The database was built using a big collection of medical publications for the purpose of cancer type classification. Aurora [130] demonstrated how to analyze web-sourced mobile data using Apache's K-algorithm Spark MLlib, which is based on the Spark algorithm. The study gave an effective technique of determining the number of grid users based on the grouping of latitude and longitude information, which was based on the results of the investigation. When learning human behaviors, the study by [122] provided ALMD, which performs feature description by monitoring the appearance and movement randomly based on the usage of the Apache Spark ML library and the usage of Apache Spark ML library. Assefi *et al.* [131] have described the construction of a framework for demographics analysis utilizing next-generation data sequencing as a

case study. In order to optimize the system, it is necessary to update the resource estimator and optimize the components. The system was developed entirely on Apache Spark, and as a result, it takes advantage of the favorable aspects of the MLlib and other Spark components. BigNN was developed by Assefi *et al.* [131] as another fascinating feature of big data analytics on Apache Spark. It is capable of handling biomedical strings on a very large scale, which is very useful in the healthcare industry. MLlib can be implemented using programs written in R, Scala, Python, and Java, among other programming languages. Vector, LabeledPoint, and rating are the core data abstractions used by MLlib; as a result, the pedestrian and other statistical components of MLlib work on data represented by these abstractions. Observational data features are captured using the vector type, which represents an index set of double type values with a zero-index of the int type. The vector type is used to record the observational data features.

A vector of length n might theoretically represent a note with n properties, which would imply that it represents an object in a file with N dimensions. The vector type offered by MLlib differs from the vector type supplied by the Scala set library in that the vector type in MLlib implements the digital vector concept from linear algebra, but the vector type in the Scala set library does not. MLlib is capable of handling both dense and sparse vector types. In addition, because the MLlib Vector type is considered an adjective, it cannot be instantiated directly by the application; instead, the factory methods given by MLlib must be utilized to construct an instance of either the sparse vector class or the dense vector class. It should be noted that the factory methods for creating instances of the dense vector or sparse vector classes are already specified in the vectors object, which is convenient.

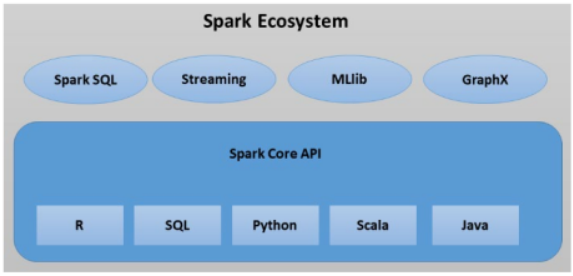

Figure 1. MLlib in spark ecosystem

3. METHOD

Spark's machine learning library, MLlib, has been under heavy development since its inception, and unlike the Spark core. It is still not in a fully stable state with regard to its overall API and design. As of Spark version 1.2.0, a new, experimental API for MLlib has been released under the ml package (whereas the current library resides under the MLlib package). Figure 1 shows the Spark ecosystem with MLlib. This new API aims to enhance the APIs and interfaces for models as well as feature extraction and transformation so as to make it easier to build pipelines that chain together steps that include feature extraction, normalization, dataset transformations, model training, and cross-validation. Since the new API is still experimental, it may be subject to major changes in the next few Spark releases. Over time, the various feature-processing techniques and models that we will cover will simply be ported to the new API; however, the core concepts and most underlying code will remain largely unchanged.

This section summarized the tests carried out on the six datasets listed in Table 1. The findings were provided in terms of the processing time for MLlib and MOA when both programs were run on the same hardware. The performance of Apache Spark MLlib 2.0 was compared and evaluated on six distinct large datasets obtained from the University of California, Irvine's machine learning repository. The experimental setup used in this work consisted of a standalone Spark cluster that makes use of an HDFS storage system and Apache Zeppelin 0.7.1 as an editor, both of which were developed by the authors. The Spark cluster is made up of the following components: a master node that runs a driver software; three worker nodes; and a data node (includes1 worker node that executes on the master node). Similar to the design illustrated in Table 2, the three nodes had a similar configuration. The three worker nodes each had a memory capacity of 48 GB, and each worker node was configured with four executors (each with a memory capacity of 4 GB) and two CPUs. Each worker in the master node was configured with three executors (each with a size of 5 GB) and two cores, as shown in the diagram. A total of 16 GB of RAM was allocated to the driver process.

The MLlib was run on a Scala 2.11.8 PL in a Spark 2.2.1 cluster, with Hadoop 2.7.3 serving as the distributed storage device, and the results were published. The amount of RAM available to the executors in each worker node was changed by employing the optimal number of data partitions in order to obtain the fastest possible execution time. Table 1 describes the characteristics of the datasets that were used in this investigation in terms of the amount of attributes, records, and classes that they contained.

Table 1. Dataset description

Data	No of record	No of attributes	No of classes
Covtype	581012	54	7
Covtype-2	581012	54	2
Higgs	11,000,000	28	2
Botnet Attacks	7,062,606	115	10
Dota2	102944	116	2
SUSY	5,000,000	18	2

Table 2. System description

Parameter	Specification
Operating system	Windows10
CPU	Intel® Core™ i7-6700 CPU @ 3.40 GHz with 8 logical cores
Memory	16 GB
No. of workers	3
Computational framework	Apache Spark 2.2.1
Compatible framework	Hadoop
DSS	HDFS (Hadoop 2.7.3)
Code development editor	Apache Zeppelin 0.7.1
Coding language	Scala 2.11.8

4. RESULTS AND DISCUSSION

The implementation process was kicked off by first defining the Spark context for the program that was selected. As previously stated, this is the primary point of entry for Spark functionality, and it must be given before attempting to create the RDDs. The three Spark Context parameters, which are the application name, the number of cores, and the URL of the cluster, were also supplied in the configuration. In addition, the name of the application should be significant in order to clearly identify the program's objective. To specify the name of an application for a local cluster, the keyword "local" is used. Worker nodes are responsible for processing work in Spark and, as previously stated, the number of worker nodes to be formed is dictated by the number of cores available. The following step is to train the model using the training data and to provide the parameters that are accessible for the supervised machine learning methods that have been selected (support vector machines (SVM), decision tree, and logistic regression). The parameters for the decision tree, SVM, and logistic regression methods were shown in Tables 3, 4, and 5, respectively.

The testing of the trained model on the testing set is the next step; this was accomplished using the "predict" method which was implemented using the "map" transformation of Spark for each row of the test set. The comparison of the computational time of Apache Spark MLlib and MOA under different experimental conditions is shown in Figure 2.

Table 3. The decision tree classification technique relies on a number of parameters

Parameter	Explanation	Value used
maxBins	The required number of bins for finding the splits at each node; the default value is 32.	32
minInfoGain	The minimum info gain needed for the creation of a split; the default value is 0.0	0.15
numClasses	The required number of classes to execute classification tasks	2
maxDepth	The maximum tree depth; the default value is 5.	6
impurity	The required criterion for the selection of information gain (gini or entropy).	entropy

Table 4. The parameters that were used in the SVM classification algorithm

Parameter	Explanation	Value used
validateData	Data must be validated by the algorithm before training	TRUE
iterations	The number of considered iterations	1000
numClasses	The number of considered classes; the default value is 2.	2

Table 5. The parameters that were used in the logistic regression algorithm

Parameter	Explanation	Value used
validateData	Data must be validated by the algorithm before training	TRUE
iterations	The number of considered iterations	1000
numClasses	The number of considered classes; the default value is 2.	2

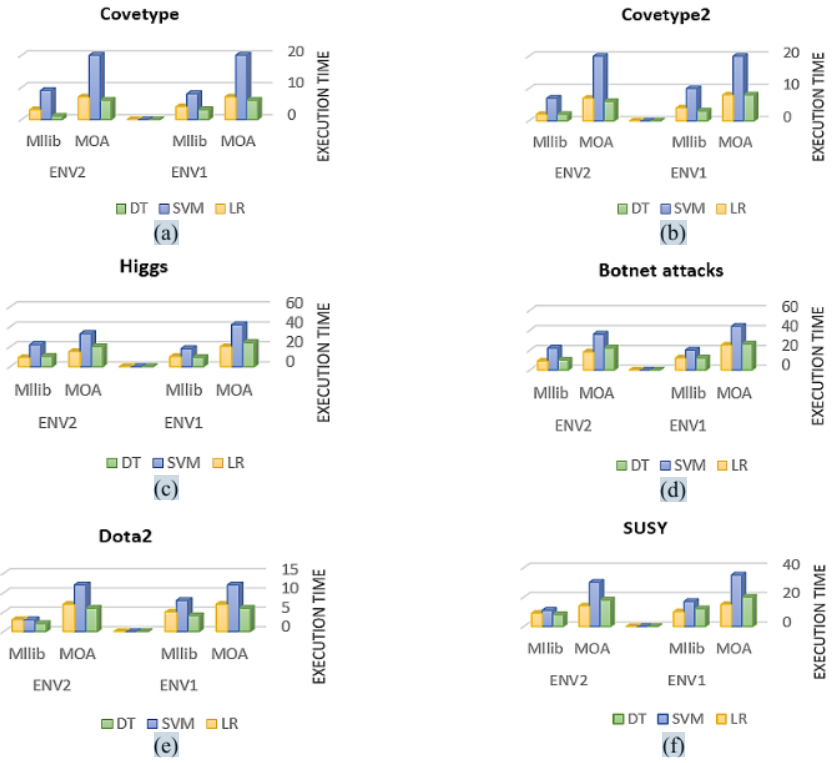

Figure 2. Computational time of Apache Spark MLlib and MOA under different datasets: (a) Covetype, (b) Covetype2, (c) Higgs, (d) Botnet attacks, (e) Dota2 and (f) SUSY database

There was a close similarity in the area under the ROC for both Apache Spark MLlib and MOA as the difference between them was not statistically significant. However, the little difference between them could be due to the detailed parametric settings of each classifier during the random selection of the test and train datasets. Obviously, Apache Spark MLlib was faster than MOA based on the observed computational times of the classifiers; however, the clustering method showed statistically significant differences between the Apache Spark MLlib and MOA.

5. CONCLUSION

Data generation has increased at an alarming rate in recent years, necessitating advancements in data analytics and processing tools in order to enable the extraction of relevant information from vast amounts of organized and unstructured data. Big data machine learning techniques, which are believed to be efficient in pattern finding, can be used to more efficiently handle this challenge. Apache Spark MLlib is a widely used machine learning library for big data, and it is a powerful tool for big data analytics. As proved in this study, it provides excellent performance in terms of computational time. Massive online analytics (MOA), on the other hand, is slightly slower than Apache Spark MLlib during big data analysis; however, because the classifiers use different configurations and file systems, the comparison may not be appropriate. MLlib was implemented on the Spark distributed file system, whereas the MOA classifier was implemented on the

Hadoop distributed file system, the comparison may not be appropriate. Because we want to demonstrate how well Spark performs on large data sets using MOA as a benchmark, it is assumed that there are many MOA features that Spark cannot compete with, such as the availability of a large pool of resources and documents for MOA users, the ease with which non-experts can implement MOA, and the presence of a good graphical user interface in MOA, among other things. These characteristics are the reason why MOA supports a variety of machine learning techniques.

ACKNOWLEDGEMENTS

The authors would like to thank ICCI, Informatics Institute for Postgraduate Studies, Iraqia University, and Al Salam University College for their facilities and support; and Universitas Ahmad Dahlan to support this collaborative research.

REFERENCES

[1] V. Chang, "An ethical framework for big data and smart cities," *Technol. Forecast. Soc. Change*, vol. 165, p. 120559, 2021, doi: 10.1016/j.techfore.2020.120559.

[2] S. A. Alomari, M. S. Alzboon, M. S. Al-Batah, and B. Zaqaibeh, "A novel adaptive schema to facilitates playback switching technique for video delivery in dense LTE cellular heterogeneous network environments," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 5, pp. 5347–5367, 2020, doi: 10.11591/ijece.v10i5.pp5347-5367.

[3] S. Reshma and Chetanaprakash, "Advancement in infotainment system in automotive sector with vehicular cloud network and current state of art," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 2, pp. 2077–2087, 2020, doi: 10.11591/ijece.v10i2.pp2077-2087.

[4] W. N. W. Abd Manan and M. A. Salamat, "Concept of minimizing the response time for reducing dynamic data redundancy in cloud computing," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 15, no. 3, pp. 1597–1602, 2019, doi: 10.11591/ijeecs.v15.i3.pp1597-1602.

[5] M. El Ghmary, T. Chanyour, Y. Hmimz, and M. O. C. Malki, "Efficient multi-task offloading with energy and computational resources optimization in a mobile edge computing node," *International Journal of Electrical and Computer Engineering*, vol. 9, no. 6, pp. 4908–4919, 2019, doi: 10.11591/ijece.v9i6.pp4908-4919.

[6] K. K. Chakravarthi and V. Vijayakumar, "Workflow scheduling techniques and algorithms in IaaS cloud: A survey," *International Journal of Electrical and Computer Engineering*, vol. 8, no. 2, pp. 1256–1268, 2018, doi: 10.11591/ijece.v8i2.pp1256-1268.

[7] M. R. Belgaum, Z. Alansari, S. Musa, M. M. Alam, and M. S. Mazliham, "Role of artificial intelligence in cloud computing, IoT and SDN: Reliability and scalability issues," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 5, pp. 4458–4470, 2021, doi: 10.11591/ijece.v11i5.pp4458-4470.

[8] J. Karande and S. Joshi, "DEDA: An algorithm for early detection of topology attacks in the internet of things," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 2, pp. 1761–1770, 2021, doi: 10.11591/ijece.v11i2.pp1761-1770.

[9] F. Arena and G. Pau, "An overview of big data analysis," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 4, pp. 1646–1653, 2020, doi: 10.11591/eei.v9i4.2359.

[10] K. F. Mahmmod, M. M. Azeez, and Z. H. Ismael, "Design an active verification mechanism for certificates revocation in OCSP for internet authentication," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 4, pp. 4208–4216, 2020, doi: 10.11591/ijece.v10i4.pp4208-4216.

[11] H. Hosseinian, H. Shahinzadeh, G. B. Gharehpetian, Z. Azani, and M. Shaneh, "Blockchain outlook for deployment of IoT in distribution networks and smart homes," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 3, pp. 2787–2796, 2020, doi: 10.11591/ijece.v10i3.pp2787-2796.

[12] A. Ouacha and M. El Ghmary, "Virtual machine migration in mec based artificial intelligence technique," *IAES International Journal of Artificial Intelligence*, vol. 10, no. 1, pp. 244–252, 2021, doi: 10.11591/ijai.v10.i1.pp244-252.

[13] A. H. Ali and M. Z. Abdullah, "A parallel grid optimization of SVM hyperparameter for big data classification using spark Radoop," *Karbala Int. J. Mod. Sci.*, vol. 6, no. 1, pp. 5-18, 2020, doi: 10.33640/2405-609X.1270.

[14] Pronika and S. S. Tyagi, "Performance analysis of encryption and decryption algorithm," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 23, no. 2, pp. 1030–1038, 2021, doi: 10.11591/ijeecs.v23.i2.pp1030-1038.

[15] Y. A. Ghani Alyouzbaki and M. F. Al-Rawi, "Novel load balancing approach based on ant colony optimization technique in cloud computing," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 4, pp. 2320–2326, 2021, doi: 10.11591/eei.v10i4.2947.

[16] A. M. N. G. Molk, M. R. Aref, and R. R. Khorshiddoust, "Analysis of design goals of cryptography algorithms based on different components," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 23, no. 1, pp. 540–548, 2021, doi: 10.11591/ijeecs.v23.i1.pp540-548.

[17] Y. N. Seitkulov, S. N. Boranbayev, G. B. Ulyukova, B. B. Yergaliyeva, and D. Satybaldina, "Methods for secure cloud processing of big data," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 22, no. 3, pp. 1650–1658, 2021, doi: 10.11591/ijeecs.v22.i3.pp1650-1658.

[18] M. F. Falah *et al.*, "Comparison of cloud computing providers for development of big data and internet of things application," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 22, no. 3, pp. 1723–1730, 2021, doi: 10.11591/ijeecs.v22.i3.pp1723-1730.

[19] H. Kim, J. Park, and H. Jung, "Automatic control system based on IoT data identification," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 19, no. 3, pp. 1525–1532, 2020, doi: 10.11591/ijeecs.v19.i3.pp1525-1532.

[20] W. Budiharto and M. Meiliana, "Prediction and analysis of Indonesia Presidential election from Twitter using sentiment analysis," *J. Big data*, vol. 5, no. 1, pp. 1–10, 2018, doi: 10.1186/s40537-018-0164-1.

[21] T. Yigitcanlar, N. Kankanamge, and K. Vella, "How are smart city concepts and technologies perceived and utilized? A systematic geo-Twitter analysis of smart cities in Australia," *J. Urban Technol.*, vol. 28, no. 1–2, pp. 135–154, 2021, doi: 10.1080/10630732.2020.1753483.

[22] H. Shirdastian, M. Laroche, and M.-O. Richard, "Using big data analytics to study brand authenticity sentiments: The case of Starbucks on Twitter," *Int. J. Inf. Manage.*, vol. 48, pp. 291–307, 2019, doi: 10.1016/j.ijinfomgt.2017.09.007.

[23] Y. Alemami, M. A. Mohamed, S. Atiewi, and M. Mamat, "Speech encryption by multiple chaotic maps with fast fourier transform," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 6, pp. 5658–5664, 2020, doi: 10.11591/ijece.v10i6.pp5658-5664.

[24] A. Alrumaih, A. Al-Sabbagh, R. Alsabah, H. Kharrufa, and J. Baldwin, "Sentiment analysis of comments in social media," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 6, pp. 5917–5922, 2020, doi: 10.11591/ijece.v10i6.pp5917-5922.

[25] P. Anita, M. Devi, and P. Anita, "High performance modified bit-vector based packet classification module on low-cost FPGA," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 5, pp. 3855–3863, 2021, doi: 10.11591/ijece.v11i5.pp3855-3863.

[26] N. Sridevi and M. Meenakshi, "Efficient reconfigurable architecture for moving object detection with motion compensation," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 23, no. 2, pp. 802–810, 2021, doi: 10.11591/ijeecs.v23.i2.pp802-810.

[27] P. Veeresh, R. Praveen Sam, and C. Shoba Bindu, "Reliable fault tolerance system for service composition in mobile Ad Hoc network," *International Journal of Electrical and Computer Engineering*, vol. 9, no. 4. pp. 2523–2533, 2019, doi: 10.11591/ijece.v9i4.pp2523-2533.

[28] E. A. Abdessamad, N. Bahri, A. Mansouri, N. Masmoud, and A. Ali, "Area and power efficient VLSI architecture of mode decision in integer motion estimation for HEVC video coding standard," *International Journal of Electrical and Computer Engineering*, vol. 9, no. 4, pp. 2469–2480, 2019, doi: 10.11591/ijece.v9i4.pp2469-2480.

[29] L. E. Aik, T. W. Hong, and A. K. Junoh, "An improved radial basis function networks based on quantum evolutionary algorithm for training nonlinear datasets," *IAES International Journal of Artificial Intelligence*, vol. 8, no. 2, pp. 120–131, 2019, doi: 10.11591/ijai.v8.i2.pp120-131.

[30] C. Jittawiriyanukoon, "Granularity analysis of classification and estimation for complex datasets with MOA," *International Journal of Electrical and Computer Engineering*, vol. 9, no. 1, pp. 409–416, 2019, doi: 10.11591/ijece.v9i1.pp409-416.

[31] S. Alzaeemi, M. A. Mansor, M. S. Mohd Kasihmuddin, S. Sathasivam, and M. Mamat, "Radial basis function neural network for 2 satisfiability programming," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 18, no. 1, pp. 459–469, 2019, doi: 10.11591/ijeecs.v18.i1.pp459-469.

[32] K. Suresh and R. J. Kannan, "Review of advancements in multi-tenant framework in cloud computing," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 11, no. 3, pp. 1102–1108, 2018, doi: 10.11591/ijeecs.v11.i3.pp1102-1108.

[33] N. Edward and J. Elcock, "Task scheduling in heterogeneous multiprocessor environments-an efficient ACO-based approach," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 10, no. 1, pp. 320–329, 2018, doi: 10.11591/ijeecs.v10.i1.pp320-329.

[34] B. S. Sabrina, K. Hamoudi, and K. Salim, "Bi-objective scheduling with cooperating heuristics for embedded real-time systems," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 9, no. 3, pp. 789–798, 2018, doi: 10.11591/ijeecs.v9.i3.pp789-798.

[35] A. B. Ibrahim, C. Z. Zulkifli, S. A. Ariffin, and N. H. Kahar, "High frequency of low noise amplifier architecture for WiMAX application: A review," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 3, pp. 2153–2164, 2021, doi: 10.11591/ijece.v11i3.pp2153-2164.

[36] B. U. V Prashanth, M. R. Ahmed, and M. R. Kounte, "Design and implementation of DA FIR filter for bio-inspired computing architecture," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 2, pp. 1709–1718, 2021, doi: 10.11591/ijece.v11i2.pp1709-1718.

[37] L. K. Ramasamy, S. Kadry, and S. Lim, "Selection of optimal hyper-parameter values of support vector machine for sentiment analysis tasks using nature-inspired optimization methods," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 1, pp. 290–298, 2021, doi: 10.11591/eei.v10i1.2098.

[38] H. F. S. Saipol and N. Alias, "Numerical simulation of DIC drying process on matlab distributed computing server," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 20, no. 1, pp. 338–346, 2020, doi: 10.11591/ijeecs.v20.i1.pp338-346.

[39] G. B. Pallavi and P. Jayarekha, "An efficient resource sharing technique for multi-tenant databases," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 3, pp. 3216–3226, 2020,

doi: 10.11591/ijece.v10i3.pp3216-3226.
[40] N. R. Latha and G. R. Prasad, "Memory and I/O optimized rectilinear Steiner minimum tree routing for VLSI," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 3, pp. 2959–2968, 2020, doi: 10.11591/ijece.v10i3.pp2959-2968.
[41] N. H. Ja'afar and A. Ahmad, "Algorithm development and hardware implementation for medical image compression system: A review," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 18, no. 3, pp. 1331–1341, 2020, doi: 10.11591/ijeecs.v18.i3.pp1331-1341.
[42] M. Aljarah, M. Shurman, and S. H. Alnabelsi, "Cooperative hierarchical based edge-computing approach for resources allocation of distributed mobile and IoT applications," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 1, pp. 296–307, 2020, doi: 10.11591/ijece.v10i1.pp296-307.
[43] N. Razali, S. Ismail, and A. Mustapha, "Machine learning approach for flood risks prediction," *IAES International Journal of Artificial Intelligence*, vol. 9, no. 1, pp. 73–80, 2020, doi: 10.11591/ijai.v9.i1.pp73-80.
[44] H. Bagheri and A. A. Shaltooki, "Big data: Challenges, opportunities and cloud based solutions," *International Journal of Electrical and Computer Engineering*, vol. 5, no. 2, pp. 340–343, 2015, doi: 10.11591/ijece.v5i2.pp340-343.
[45] N. Nizamuddin and A. Abugabah, "Blockchain for automotive: An insight towards the IPFS blockchain-based auto insurance sector," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 3, pp. 2443–2456, 2021, doi: 10.11591/ijece.v11i3.pp2443-2456.
[46] R. Bandi, J. Amudhavel, and R. Karthik, "Machine learning with PySpark – Review," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 12, no. 1, pp. 102–106, 2018, doi: 10.11591/ijeecs.v12.i1.pp102-106.
[47] M. Armanur Rahman, J. Hossen, A. Sultana, A. A. Mamun, and N. A. A. Aziz, "A smart method for spark using neural network for big data," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 3, pp. 2525–2534, 2021, doi: 10.11591/ijece.v11i3.pp2525-2534.
[48] S. Hartini, Z. Rustam, G. S. Saragih, and M. J. S. Vargas, "Estimating probability of banking crises using random forest," *IAES International Journal of Artificial Intelligence*, vol. 10, no. 2, pp. 407–413, 2021, doi: 10.11591/IJAI.V10.I2.PP407-413.
[49] M. M. Akawee, M. A. M. Al-Obaidi, H. M. T. Al-Hilfi, S. I. Jassim, and T. Sutikno, "An efficient hybrid model for secure transmission of data by using efficient data collection and dissemination (EDCD) algorithm based WSN," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 20, no. 1, pp. 545–551, 2020, doi: 10.11591/ijeecs.v20.i1.pp545-551.
[50] N. P. Shetty, J. Shetty, R. Narula, and K. Tandona, "Comparison study of machine learning classifiers to detect anomalies," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 5, pp. 5445–5452, 2020, doi: 10.11591/IJECE.V10I5.PP5445-5452.
[51] R. A. Ibrahem Alhayali, M. A. Ahmed, Y. M. Mohialden, and A. H. Ali, "Efficient method for breast cancer classification based on ensemble hoffeding tree and naïve Bayes," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 18, no. 2, pp. 1074–1080, 2020, doi: 10.11591/ijeecs.v18.i2.pp1074-1080.
[52] A. V Vitianingsih, N. Suryana, and Z. Othman, "Spatial analysis model for traffic accident-prone roads classification: A proposed framework," *IAES International Journal of Artificial Intelligence*, vol. 10, no. 2, pp. 365–373, 2021, doi: 10.11591/ijai.v10.i2.pp365-373.
[53] K. A. Shakil, M. Alam, and S. Khan, "A latency-aware max-min algorithm for resource allocation in cloud," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 1, pp. 671–685, 2021, doi: 10.11591/ijece.v11i1.pp671-685.
[54] N. Seman and N. A. Razmi, "Machine learning-based technique for big data sentiments extraction," *IAES International Journal of Artificial Intelligence*, vol. 9, no. 3, pp. 473–479, 2020, doi: 10.11591/ijai.v9.i3.pp473-479.
[55] B. C. Manujakshi and K. B. Ramesh, "Framework for cost-effective analytical modelling for sensory data over cloud environment," *International Journal of Electrical and Computer Engineering*, vol. 9, no. 5, pp. 3822–3832, 2019, doi: 10.11591/ijece.v9i5.pp3822-3832.
[56] H. Ashraf, M. Alenezi, M. Nadeem, and Y. Javid, "Security assessment framework for educational ERP systems," *International Journal of Electrical and Computer Engineering*, vol. 9, no. 6, pp. 5570–5585, 2019, doi: 10.11591/ijece.v9i6.pp5570-5585.
[57] A. Boukhalfa, N. Hmina, and H. Chaoui, "Parallel processing using big data and machine learning techniques for intrusion detection," *IAES International Journal of Artificial Intelligence*, vol. 9, no. 3, pp. 553–560, 2020, doi: 10.11591/ijai.v9.i3.pp553-560.
[58] U. Narayanan, V. Paul, and S. Joseph, "A novel approach to big data analysis using deep belief network for the detection of android malware," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 16, no. 3, pp. 1447–1454, 2019, doi: 10.11591/ijeecs.v16.i3.pp1447-1454.
[59] N. H. M. Ariffin, F. Ahmad, and U. M. Haneef, "Acceptance of mobile payments by retailers using UTAUT model," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 19, no. 1, pp. 149–155, 2020, doi: 10.11591/ijeecs.v19.i1.pp149-155.
[60] J. I. Naser and A. J. Kadhim, "Multicast routing strategy for SDN-cluster based MANET," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 5, pp. 4447–4457, 2020, doi: 10.11591/ijece.v10i5.pp4447-4457.
[61] K. Nagarathna, "Energy-aware strategy for data forwarding in IoT ecosystem," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 5, pp. 4863–4871, 2020, doi: 10.11591/ijece.v10i5.pp4863-4871.
[62] A. Bhawiyuga, S. A. Kharisma, B. J. Santoso, D. P. Kartikasari, and A. P. Kirana, "Cloud-based middleware for

supporting batch and stream access over smart healthcare wearable device," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 5, pp. 1990–1997, 2020, doi: 10.11591/eei.v9i5.1978.
[63] C. C. Uchenna, N. Jamil, R. Ismail, L. K. Yan, and M. A. Mohamed, "Malware threat analysis techniques and approaches for iot applications: A review," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 3, pp. 1558–1571, 2021, doi: 10.11591/eei.v10i3.2423.
[64] N. S. M. Pakhrudin, M. Kassim, and A. Idris, "A review on orchestration distributed systems for IoT smart services in fog computing," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 2, pp. 1812–1822, 2021, doi: 10.11591/ijece.v11i2.pp1812-1822.
[65] S.-T. Nam, C.-Y. Jin, and S.-Y. Shin, "A forecasting of stock trading price using time series information based on big data," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 3, pp. 2548–2554, 2021, doi: 10.11591/ijece.v11i3.pp2548-2554.
[66] Y. I. Alzoubi, A. Al-Ahmad, and A. Jaradat, "Fog computing security and privacy issues, open challenges, and blockchain solution: An overview," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 6, pp. 5081–5088, 2021, doi: 10.11591/ijece.v11i6.pp5081-5088.
[67] K. Radha, B. T. Rao, S. M. Babu, K. T. Rao, V. K. Reddy, and P. Saikiran, "Service level agreements in cloud computing and big data," *International Journal of Electrical and Computer Engineering*, vol. 5, no. 1. pp. 158–165, 2015, doi: 10.11591/ijece.v5i1.pp158-165.
[68] A. H. Katrawi, R. Abdullah, M. Anbar, and A. K. Abasi, "Earlier stage for straggler detection and handling using combined CPU test and LATE methodology," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 5, pp. 4910–4917, 2020, doi: 10.11591/ijece.v10i5.pp4910-4917.
[69] J. Santosh Kumar, B. K. Raghavendra, S. Raghavendra, and Meenakshi, "Performance evaluation of Map-reduce jar pig hive and spark with machine learning using big data," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 4, pp. 3811–3818, 2020, doi: 10.11591/ijece.v10i4.pp3811-3818.
[70] M. B. Masadeh, M. S. Azmi, and S. S. S. Ahmad, "Available techniques in hadoop small file issue," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 2, pp. 2097–2101, 2020, doi: 10.11591/ijece.v10i2.pp2097-2101.
[71] A. Bhaskar and R. Ranjan, "Optimized memory model for hadoop map reduce framework," *International Journal of Electrical and Computer Engineering*, vol. 9, no. 5, pp. 4396–4407, 2019, doi: 10.11591/ijece.v9i5.pp4396-4407.
[72] U. Narayanan, V. Paul, and S. Joseph, "A light weight encryption over big data in information stockpiling on cloud," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 17, no. 1, pp. 389–397, 2019, doi: 10.11591/ijeecs.v17.i1.pp389-397.
[73] J. Parsola, D. Gangodkar, and A. Mittal, "Post event investigation of multi-stream video data utilizing hadoop cluster," *International Journal of Electrical and Computer Engineering*, vol. 8, no. 6, pp. 5089–5097, 2018, doi: 10.11591/ijece.v8i6.pp.5089-5097.
[74] D. C. Vinutha and G. T. Raju, "An accurate and efficient scheduler for hadoop mapreduce framework," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 12, no. 3, pp. 1132–1142, 2018, doi: 10.11591/ijeecs.v12.i3.pp1132-1142.
[75] M. Isard and Y. Yu, "Distributed data-parallel computing using a high-level programming language," in *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, 2009, pp. 987–994.
[76] C. Chambers *et al.*, "FlumeJava: easy, efficient data-parallel pipelines," *ACM Sigplan Not.*, vol. 45, no. 6, pp. 363–375, 2010.
[77] J. Farooq, P. Sharma, and R. S. Kumar, "A BIM-based detailed electrical load estimation, costing and code checking," *International Journal of Electrical and Computer Engineering*, vol. 8, no. 5, pp. 3484–3495, 2018, doi: 10.11591/ijece.v8i5.pp3484-3495.
[78] C.-Y. Song and E.-S. Cho, "A service-oriented cloud modeling method and process," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 1. pp. 962–977, 2020, doi: 10.11591/ijece.v10i1.pp962-977.
[79] R. Kraleva, M. Sabani, V. Kralev, and D. Kostadinova, "An approach to designing and developing an LMS framework appropriate for young pupils," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 2, pp. 1577–1591, 2020, doi: 10.11591/ijece.v10i2.pp1577-1591.
[80] S. Islam, Y. S. Nugroho, and M. Javed Hossain, "What network simulator questions do users ask? A large-scale study of stack overflow posts," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 21, no. 3, pp. 1622–1633, 2021, doi: 10.11591/ijeecs.v21.i3.pp1622-1633.
[81] F. Renaldi, I. Santikarama, E. C. Djamal, and A. J. Maulidin, "Service landscape for private universities in Indonesia based on service oriented architecture and cloud technology," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 22, no. 1, pp. 497–506, 2021, doi: 10.11591/ijeecs.v22.i1.pp497-506.
[82] G. Malewicz *et al.*, "Pregel: a system for large-scale graph processing," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, 2010, pp. 135–146.
[83] J. Ekanayake *et al.*, "Twister: a runtime for iterative mapreduce," in *Proceedings of the 19th ACM international symposium on high performance distributed computing*, 2010, pp. 810–818.
[84] Y. Bu, B. Howe, M. Balazinska, and M. D. Ernst, "HaLoop: Efficient iterative data processing on large clusters," *Proc. VLDB Endow.*, vol. 3, no. 1–2, pp. 285–296, 2010.
[85] D. Lunga, J. Gerrand, L. Yang, C. Layton, and R. Stewart, "Apache spark accelerated deep learning inference for large scale satellite image analytics," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 13, pp. 271–283, 2020, doi: 10.1109/JSTARS.2019.2959707.
[86] P. Ganesh, K. Sailaja Kumar, D. Evangelin Geetha, and T. V Suresh Kumar, "Performance evaluation of cloud

service with hadoop for twitter data," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 13, no. 1, pp. 392–404, 2019, doi: 10.11591/ijeecs.v13.i1.pp392-404.
[87] M. A. Rahman, A. Hossen, J. Hossen, C. Venkataseshaiah, T. Bhuvaneswari, and A. Sultana, "Towards machine learning-based self-tuning of hadoop-spark system," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 15, no. 2, pp. 1076–1085, 2019, doi: 10.11591/ijeecs.v15.i2.pp1076-1085.
[88] S. Shetty, B. Dinesh Rao, and S. Prabhu, "Growth of relational model: Interdependence and complementary to big data," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 2, pp. 1780–1795, 2021, doi: 10.11591/ijece.v11i2.pp1780-1795.
[89] J. G. Shanahan and L. Dai, "Large scale distributed data science using apache spark," in *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 2015, pp. 2323–2324.
[90] N. D. Zaki, N. Y. Hashim, Y. M. Mohialden, M. A. Mohammed, T. Sutikno, and A. H. Ali, "A real-time big data sentiment analysis for iraqi tweets using spark streaming," *Bull. Electr. Eng. Informatics*, vol. 9, no. 4, pp. 1411–1419, 2020.
[91] Y. Choubik, A. Mahmoudi, M. M. Himmi, and L. El Moudnib, "STA/LTA trigger algorithm implementation on a seismological dataset using hadoop mapreduce," *IAES International Journal of Artificial Intelligence*, vol. 9, no. 2, pp. 269–275, 2020, doi: 10.11591/ijai.v9.i2.pp269-275.
[92] S. Rajagopal, P. P. Kundapur, and K. S. Hareesha, "A predictive model for network intrusion detection using stacking approach," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 3, pp. 2734–2741, 2020, doi: 10.11591/ijece.v10i3.pp2734-2741.
[93] X. Meng *et al*., "Mllib: Machine learning in apache spark," *J. Mach. Learn. Res*., vol. 17, no. 1, pp. 1235–1241, 2016.
[94] E. Alomari, I. Katib, and R. Mehmood, "Iktishaf: A big data road-traffic event detection tool using Twitter and spark machine learning," *Mob. Networks Appl.*, pp. 1–16, 2020, doi: 10.1007/s11036-020-01635-y.
[95] A. H. Ali, and M. Z. Abdulah, "A survey on vertical and horizontal scaling platforms for big data analytics," *Int. J. Integr. Eng.*, vol. 11, no. 6, pp. 138–150, 2019, doi: 10.30880/ijie.2019.11.06.015.
[96] F. L. Khaleel, N. S. Ashaari, T. S. M. T. Wook, and A. Ismail, "Programming learning requirements based on multi perspectives," *International Journal of Electrical and Computer Engineering*, vol. 7, no. 3, pp. 1299–1307, 2017, doi: 10.11591/ijece.v7i3.pp1299-1307.
[97] B.-E. B. Semlali, C. El Amrani, and S. Denys, "Development of a Java-based application for environmental remote sensing data processing," *International Journal of Electrical and Computer Engineering*, vol. 9, no. 3, pp. 1978–1986, 2019, doi: 10.11591/ijece.v9i3.pp1978-1986.
[98] A. G. SFakhar, K. A Fauzan, H. M Saad, H. R. Affendi, and K. H. Fen, "Development of a portable community video surveillance system," *International Journal of Electrical and Computer Engineering*, vol. 9, no. 3, pp. 1814–1821, 2019, doi: 10.11591/ijece.v9i3.pp1814-1821.
[99] B. Hassan, R. Amina, L. Amine, L. Elhoussine, and M. Azouazi, "A regexcriteria api to complete the power of regular expressions engine," *International Journal of Electrical and Computer Engineering*, vol. 9, no. 4, pp. 3185–3193, 2019, doi: 10.11591/ijece.v9i4.pp3185-3193.
[100] G. Mappatao, I. M. Z. Bautista, M. K. Orsos, M. A. Ribo, and J. Castillo, "Development of a remote tending system for analog broadcast transmitters," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 15, no. 3, pp. 1474–1484, 2019, doi: 10.11591/ijeecs.v15.i3.pp1474-1484.
[101] G. P. Mappatao, I. M. Z. S. Bautista, M. K. J. Orsos, M. A. C. Ribo, and J. C. Castillo, "Remote tending of modern broadcast transmitters," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 15, no. 3, pp. 1491–1500, 2019, doi: 10.11591/ijeecs.v15.i3.pp1491-1500.
[102] A. Al-Mnayyis, S. A. Alasal, M. Alsmirat, Q. B. Baker, and S. Al Zu'bi, "Lumbar disk 3D modeling from limited number of MRI axial slices," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 4, pp. 4101–4108, 2020, doi: 10.11591/ijece.v10i4.pp4101-4108.
[103] M. J. Alam and T. M. ShahzahanAli, "A smart login system using face detection and recognition by ORB algorithm," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 20, no. 2, pp. 1078–1087, 2020, doi: 10.11591/ijeecs.v20.i2.pp1078-1087.
[104] Y. Zahidi, Y. E. Younoussi, and Y. Al-Amrani, "A powerful comparison of deep learning frameworks for Arabic sentiment analysis," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 1, pp. 745–752, 2021, doi: 10.11591/ijece.v11i1.pp745-752.
[105] Y. Zahidi, Y. El Younoussi, and Y. Al-Amrani, "Different valuable tools for Arabic sentiment analysis: a comparative evaluation," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 1, pp. 753–762, 2021, doi: 10.11591/ijece.v11i1.pp753-762.
[106] A. H. Ali and M. Z. Abdullah, "Recent trends in distributed online stream processing platform for big data: Survey," in *2018 1st Annual International Conference on Information and Sciences (AiCIS)*, 2018, pp. 140–145, doi: 10.1109/AiCIS.2018.00036.
[107] M. A. Mohammed and N. ŢĂPUŞ, "A novel approach of reducing energy consumption by utilizing enthalpy in mobile cloud computing," *Stud. Informatics Control*, vol. 26, no. 4, pp. 425–434, 2017, doi: 10.24846/v26i4y201706.
[108] A. Peng and H. Liu, "Hybrid Program Recommendation Algorithm Based on Spark MLlib in Big Data Environment," in *Proceedings of the 9th International Conference on Computer Engineering and Networks*, 2021, pp. 489–498, doi: 10.1007/978-981-15-3753-0_48.
[109] H. Zhou, G. Sun, S. Fu, L. Wang, J. Hu, and Y. Gao, "Internet Financial Fraud Detection Based on a Distributed Big Data Approach With Node2vec," in IEEE Access, vol. 9, pp. 43378-43386, 2021, doi: 10.1109/ACCESS.2021.3062467.

[110] A. Sasikanthr, K. Samatha, N. Deshai, B. Sekhar, and S. Venkatramana, "Research on Advanced Streaming Processing on Apache Spark," *Int. J. Ind. Eng. Prod. Res.*, vol. 32, no. 1, pp. 133–141, 2021, doi: 10.22068/ijiepr.32.1.133.

[111] M. F. Darmawan, A. F. Z. Abidin, S. Kasim, T. Sutikno, and R. Budiarto, "Random forest age estimation model based on length of left hand bone for asian population," *Int. J. Electr. Comput. Eng.*, vol. 10, no. 1, pp. 549–558, 2020, doi: 10.11591/ijece.v10i1.pp549-558.

[112] J. A. Jupin, T. Sutikno, M. A. Ismail, M. S. Mohamad, and S. Kasim, "Review of the machine learning methods in the classification of phishing attack," *Bulletin of Electrical Engineering and Informatics*, vol. 8, no. 4, Institute of Advanced Engineering and Science, pp. 1545–1555, 2019, doi: 10.11591/eei.v8i4.1922.

[113] S. R. M-Dawam and K. R. Ku-Mahamud, "Reservoir water level forecasting using normalization and multiple regression," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 14, no. 1, pp. 443–449, 2019, doi: 10.11591/ijeecs.v14.i1.pp443-449.

[114] M. Z. Osman, M. A. Maarof, M. F. Rohani, N. N. A. Sjarif, and N. S. A. Zulkifli, "A multi-color based features from facial images for automatic ethnicity identification model," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 18, no. 3. pp. 1383–1390, 2020, doi: 10.11591/ijeecs.v18.i3.pp1383-1390.

[115] L. K. Xin and A. Abdullah, "Deep learning in non coding variant (a brief overview)," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 18, no. 3, pp. 1432–1438, 2020, doi: 10.11591/ijeecs.v18.i3.pp1432-1438.

[116] S. K. Jameel, S. Aydin, and N. H. Ghaeb, "Local information pattern descriptor for corneal diseases diagnosis," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 6, pp. 4972–4981, 2021, doi: 10.11591/ijece.v11i6.pp4972-4981.

[117] A. H. Ali and M. Z. Abdullah, "A novel approach for big data classification based on hybrid parallel dimensionality reduction using spark cluster," *Comput. Sci.*, vol. 20, no. 4, 2019, doi: 10.7494/csci.2019.20.4.3373.

[118] P. D. Hung, T. D. Hanh, and V. T. Diep, "Breast cancer prediction using spark MLlib and ML packages," in *Proceedings of the 2018 5th International Conference on Bioinformatics Research and Applications*, 2018, pp. 52–59, doi: 10.1145/3309129.3309133.

[119] Z. Zhang, J. Jiang, W. Wu, C. Zhang, L. Yu, and B. Cui, "Mllib*: Fast training of glms using spark mllib," in *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, 2019, pp. 1778–1789, doi: 10.1109/ICDE.2019.00194.

[120] S.-K. Lee and J.-H. Yu, "Success model of project management information system in construction," *Autom. Constr.*, vol. 25, pp. 82–93, 2012, doi: 10.1016/j.autcon.2012.04.015.

[121] B. Belabbess, "Automation of anomaly detections in real time by combining numeric and semantic processing." Université Paris-Est, 2018.

[122] M. A. Uddin, J. B. Joolee, A. Alam, and Y.-K. Lee, "Human action recognition using adaptive local motion descriptor in spark," *IEEE Access*, vol. 5, pp. 21157–21167, 2017, doi: 10.1109/ACCESS.2017.2759225.

[123] M. A. Mohammed, I. A. Mohammed, R. A. Hasan, N. Ţăpuş, A. H. Ali, and O. A. Hammood, "Green Energy Sources: Issues and Challenges," in *2019 18th RoEduNet Conference: Networking in Education and Research (RoEduNet)*, 2019, pp. 1–8, doi: 10.1109/ROEDUNET.2019.8909595.

[124] M. A. Mohammed, Z. H. Salih, N. Ţăpuş, and R. A. K. Hasan, "Security and accountability for sharing the data stored in the cloud," in *2016 15th RoEduNet Conference: Networking in Education and Research*, 2016, pp. 1–5, doi: 10.1109/RoEduNet.2016.7753201.

[125] Z. Ye, A. P. Tafti, K. Y. He, K. Wang, and M. M. He, "SparkText: Biomedical Text Mining on Big Data Framework," *PLoS One*, vol. 11, no. 9, p. e0162721, Sep. 2016, [Online]. Available: https://doi.org/10.1371/journal.pone.0162721.

[126] N. Q. Mohammed, M. S. Ahmed, M. A. Mohammed, O. A. Hammood, H. A. N. Alshara, and A. A. Kamil, "Comparative Analysis between Solar and Wind Turbine Energy Sources in IoT Based on Economical and Efficiency Considerations," in *2019 22nd International Conference on Control Systems and Computer Science (CSCS)*, 2019, pp. 448–452, doi: 10.1109/CSCS.2019.00082.

[127] R. A. I. Alhayali, M. A. Ahmed, Y. M. Mohialden, and A. H. Ali, "Efficient method for breast cancer classification based on ensemble hoffeding tree and naïve Bayes," *Indonesian Journal of Electrical Engineering and Computer Science,* vol. 18, no. 2, pp. 1074–1080, 2020, doi: 10.11591/ijeecs.v18.i2.pp1074-1080.

[128] Z. H. Salih, G. T. Hasan, and M. A. Mohammed, "Investigate and analyze the levels of electromagnetic radiations emitted from underground power cables extended in modern cities," in *2017 9th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, 2017, pp. 1–4, doi: 10.1109/ECAI.2017.8166452.

[129] Z. H. Salih, G. T. Hasan, M. A. Mohammed, M. A. S. Klib, A. H. Ali, and R. A. Ibrahim, "Study the Effect of Integrating the Solar Energy Source on Stability of Electrical Distribution System," in *2019 22nd International Conference on Control Systems and Computer Science (CSCS)*, 2019, pp. 443–447, doi: 10.1109/CSCS.2019.00081.

[130] S. Arora, "Analyzing mobile phone usage using clustering in Spark MLLib and Pig," *Int. J. Adv. Res. Comput. Sci.*, vol. 8, no. 1, pp. 144–146, 2017, doi: 10.26483/ijarcs.v8i1.2869.

[131] M. Assefi, E. Behravesh, G. Liu, and A. P. Tafti, "Big data machine learning using apache spark MLlib," in *2017 IEEE International Conference on Circuits and Systems (ICCS) international conference on big data (big data)*, 2017, pp. 3492–3498, doi: 10.1109/BigData.2017.8258338.

Large scale data analysis using MLlib

ORIGINALITY REPORT

19%
SIMILARITY INDEX

PRIMARY SOURCES

1	journal.uad.ac.id Internet	356 words — 8%
2	Mehdi Assefi, Ehsun Behravesh, Guangchi Liu, Ahmad P. Tafti. "Big data machine learning using apache spark MLlib", 2017 IEEE International Conference on Big Data (Big Data), 2017 Crossref	170 words — 4%
3	Ahmed Hussein Ali, Mahmood Zaki Abdullah. "A Parallel Grid Optimization of SVM Hyperparameter for Big Data Classification using Spark Radoop", Karbala International Journal of Modern Science, 2020 Crossref	107 words — 3%
4	hdl.handle.net Internet	57 words — 1%
5	Sk Kamaruddin, Vadlamani Ravi, Pritman Mayank. "Chapter 19 Parallel Evolving Clustering Method for Big Data Analytics Using Apache Spark: Applications to Banking and Physics", Springer Science and Business Media LLC, 2017 Crossref	21 words — < 1%
6	"Knowledge Innovation Through Intelligent Software Methodologies, Tools and Techniques", IOS Press, 2020 Crossref	12 words — < 1%

7	Ahmed Hussein Ali, Mahmood Zaki Abdullah. "A novel approach for big data classification based on hybrid parallel dimensionality reduction using spark cluster", Computer Science, 2019 Crossref	10 words — < 1%
8	digital.lib.washington.edu Internet	10 words — < 1%
9	publisher.uthm.edu.my Internet	10 words — < 1%
10	www.ijrte.org Internet	10 words — < 1%
11	arxiv.org Internet	9 words — < 1%
12	link.springer.com Internet	9 words — < 1%
13	web-tools.uts.edu.au Internet	9 words — < 1%
14	www.spiedigitallibrary.org Internet	9 words — < 1%
15	www.instaclustr.com Internet	8 words — < 1%
16	www.nat-hazards-earth-syst-sci-discuss.net Internet	8 words — < 1%

EXCLUDE BIBLIOGRAPHY ON