JOIV : Int. J. Inform. Visualization, 5(4) - December 2021 456-460

INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv

# A Review on Big Data Stream Processing Applications: Contributions, Benefits, and Limitations

Shaimaa Safaa Ahmed Alwaisi [a], Maan Nawaf Abbood [b], Luma Fayeq Jalil [c,*], Shahreen Kasim [d]
Mohd Farhan Mohd Fudzee [d], Ronal Hadi [e], Mohd Arfian Ismail [f]

[a] *Ministry of water resources/ planning and follow-up directorate, Baghdad, Iraq*

[b] *Imam Al-adham university College, Baghdad, Iraq*

[c] *AL Rasheed University College Computer Science Department Baghdad, Iraq*

[d] *Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia (UTHM), Malaysia*

[e] *Department of Information Technology, Politeknik Negeri Padang, West Sumatera, Indonesia*

[f] *Faculty of Computing, College of Computing and Applied Sciences, Universiti Malaysia Pahang, Malaysia*

*Corresponding author: [*]luma.jalil@alrasheedcol.edu.iq*

***Abstract*— The amount of data in our world has been rapidly keep growing from time to time. In the era of big data, the efficient processing and analysis of big data using machine learning algorithm is highly required, especially when the data comes in form of streams. There is no doubt that big data has become an important source of information and knowledge in making decision process. Nevertheless, dealing with this kind of data comes with great difficulties; thus, several techniques have been used in analyzing the data in the form of streams. Many techniques have been proposed and studied to handle big data and give decisions based on off-line batch analysis. Today, we need to make a constructive decision based on online streaming data analysis. Many researchers in recent years proposed some different kind of frameworks for processing the big data streaming. In this work, we explore and present in detail some of the recent achievements in big data streaming in term of contributions, benefits, and limitations. As well as some of recent platforms suitable to be used for big data streaming analytics. Moreover, we also highlight several issues that will be faced in big data stream processing. In conclusion, it is hoped that this study will assist the researchers in choosing the best and suitable framework for big data streaming projects.**

***Keywords*— Big data; machine learning; Spark; Kafka; data streaming.**

*Manuscript received 22 Dec. 2020; revised 5 Apr. 2021; accepted 21 Oct. 2021. Date of publication 31 Dec. 2021.*


## I. INTRODUCTION

In recent years, there has been a proliferation of applications That continuously generates huge amounts of data continuously and at an increasing rate. This is due, primarily, to the faster and lower cost hardware, [1] [2, 3] the emergence of new paradigms that thrive on user-generated data such as social networks, and the recognition of the importance of utilizing raw data (which was previously useless) in obtaining new information that are vital to deal with a variety of real-life issues. Distributed flow computing [4-6] is now an inevitable processing paradigm [7-10]. It allows to obtain continuous and real-time results from huge amounts of recent data. Data flow algorithms handle data continuous stream to store past records with only limited ability. Online machine learning covers methods that update their models after observing a new event and can instantly make predictions based on the updated model [11-16].

This paper provides a comprehensive and systematic introduction to these applications. The proposed work supplies a review of both the traditional and the modern framework, and then examines existing algorithms that use the flow of big data to identify their strengths and weaknesses. The survey will explore the recent achievements in big data streaming in term of contributions, benefits, and limitations.

## II. Material and Methods

This section describes the famous and widely applications and techniques used in the big data stream processing. In addition, several issued related to these applications were discussed.

### *A. Application on Big Data Stream Processing*

This section gives the review on the big data stream processing applications. The application includes Apache Storm, vertical and horizontal scaling platforms, Hadoop, Kafka, Spark, Vertical Scaling Platforms and HPC clusters.

*1) Apache Storm:* Apache Storm defines the calculations in terms of data streams flowing via a diagram of connected processing instances [17-22]. Such instances are in memory, can be replicated properly, and can be executed on multiple devices dynamical. The graph or diagram of inter-connected processes is named a topology. A single Storm topology consists of faucet (spout) that grout streams of data into the topology and thunderbolt that process and make a change on the data [10-14]. Topologies simplify the modeling of complex processes into several spouts and thunderbolt i.e., bolts. Tasks can be distributed and scaled by connecting several spouts and bolts with each other. Fig. 1 shows the architecture of Apache Strom.

*2) Vertical And Horizontal Scaling Platforms for Big Data Analytics:* Scaling refers to a system's ability to accommodate increases in data processing demands [23-26]. The big data processing scaling platforms can be grouped thus into two groups which are horizontal scaling and vertical scaling.

Horizontal Scaling involves the distribution of network loads to several servers (commodity machines may even be involved). It is also referred to as "scaling out". In horizontal scaling, multiple independent machines are combined to improve the data processing capability. Several operating systems are implemented on separate machines.

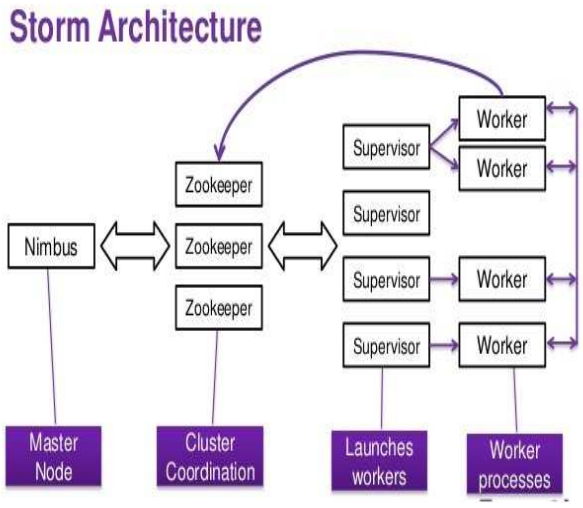


Fig. 1 Storm architecture

Vertical Scaling involves the capability of a single server is improved by introducing additional processors, faster memory and hardware; this form of scaling is also called "scaling up" and normally requires one operating system. Among the common horizontal scaling platforms are Apache Hadoop and MapReduce. Recently, research efforts have been dedicated to the development of next generation horizontal scaling tools, such as Spark which can address the issues of the currently existing scaling platforms [27-30]. Each of these scaling platforms will now be detailed in the next section.

*3) Hadoop:* Hadoop was designed for MapReduce and HDFS concepts as they are both involved in distributed computation. MapReduce is considered the basis of Hadoop and can implement parallel distributed data processing. As a UNIX-based Hadoop layer for data storage, HDFS is seen as Hadoop's own rack-aware filesystem. HDFS is developed on the concept of Google filesystem. Hadoop has the major feature of being able to distribute data computational processes cross various hosts; furthermore, it can perform parallel computations on applications that are close to their data. Data files on HDFS are duplicated as block sequences. Hadoop cluster merges different servers to improve I/O bandwidth, storage capacity, and computation power. Access to HDFS from applications can be done in different ways since they provide Java API for use in apps [31-35]. There are more than 40,000servers in the Yahoo! Hadoop clusters and they can store around 40 petabytes of app data. More than 100 organizations rely on Hadoop, with the largest Hadoop cluster containing 4,000 servers.

There are several advantages of Hadoop which can are the data source range and cost-effective. For the data source range, as it sources data from multiple sources, such data can be either unstructured or structured. Such data sources can be social media data, email data, or clickstream data. The conversion of data from different sources into a single format is time-consuming. This is not a necessary step with Hadoop as it can extract informative data from any data type and can implement other functions, such as data warehousing, fraud detection, and market trend analysis.

Meanwhile, for cost-effective, the data storage using the traditional storage methods is a costly venture as most of the benefits of the company will be spent on large data storage. The companies may even need to let-go large amounts of raw data to accommodate the new ones and this could lead to los of valuable information. However, such cost-related issues do not arise with Hadoop owing to its cost-efficiency regarding data storage. It can store data for a long time as there will be enough space to store all the newly generated raw data by a company. Future company directives can be implemented by just referring to the data archive; this is not so with the existing approaches where old data are normally deleted to accommodate new ones.

*4) Kafka:* Kafka was developed by the Apache Software Foundation as an open-source data processing technology that can be used due to its scalable storage layer; Kafka can effectively handle a huge volume of real-time data feeds [36]. It can be replicated, partitioned, or distributed to enhance its accuracy, and can be used as a publish-subscribe messaging system. The interesting attributes of Kafka are its scalability,

high throughput, and durability; a segment of its brokers can handle numerous megabytes of reads and writes per second from several applications. It can be easily scaled-up by adding more nodes to a cluster. It saves data on disk and runs as a collection of nodes to ensure data durability. Each node in Kafka, is called a broker. Messages that are propagated via Kafka are clustered into topics and apps that publish messages to a Kafka topic are called the producers while the consumers are apps that subscribes to Kafka processes and topics. Fig. 2 shows the architecture of Kafka.

*5) Spark:* The main reason for developing Spark is to support the iterative algorithms that are not fully compatible with MapReduce [37], such as the iterative ML frameworks like k means. Although Spark supports these applications, the fault tolerance of MapReduce is still retained. The execution of the Spark engine [38] can be done on various platforms, (such as Hadoop, and Mesos clusters); it has been applied in query processing (SQL), large data streaming, and advanced analytics in various databases. Spark performs better than Hadoop (10×) in iterative ML workloads; it relies on Spark master node for the workflow control. Furthermore, the Spark worker nodes performs the job that has been submitted to the Spark master node. Spark depend mainly on the RDD concept that represents a set of read-only objects distributed on several machines. Being that an RDD can be cached in memory across various machines, it could be reutilized in various MapReduce-like parallel computations.

This is a publicly available cluster computing framework which was first developed in 2014 by Matei Zaharia and later donated to the Apache Spark Foundation. The framework was built on HDFS [13]. It works on distributed processing of data, handing out data to separate worker nodes for processing. The worker nodes are managed by a master node which dispatches and schedules the distributed tasks. Hence, Spark requires a cluster manager and distributed storage system. Its faster in-memory data engine and developer-friendly API makes it the framework of choice. Apache Spark was developed as a faster alternative to. Hadoop MapReduce reads and writes from disk, which slows down the processing speed. Spark, on the other hand, stores the data in memory and reduces the read/write cycle. This results in running the applications 100x faster in memory and 10x faster on disk than Hadoop MapReduce [14]. the Spark Core is provided via Spark Streaming, giving programmers the chance of learning the project and switching between apps which manipulate the stored data in memory, on disk, or data arriving in real-time. Spark Streaming was also developed to provide an equivalent level of throughput, fault tolerance, and scalability as Spark Core [38-40]. Stream processing at a high level is all about the incessant processing of unbounded data streams. However, it is a difficult task to do this in a consistent and fault-tolerant manner [44]. However, there have been improvements in the stream processing engines such as Spark, Heron, Kafka, Flink, and Samza over the past few years which enables the development and operation of complex stream processing apps by businesses [40, 41]. Spark revolves around the concept of a resilient distributed dataset (RDD), which is a fault-tolerant collection of elements that can be operated on in parallel. The number of data pieces in a batch depends on the rate of incoming data and on the batch interval.

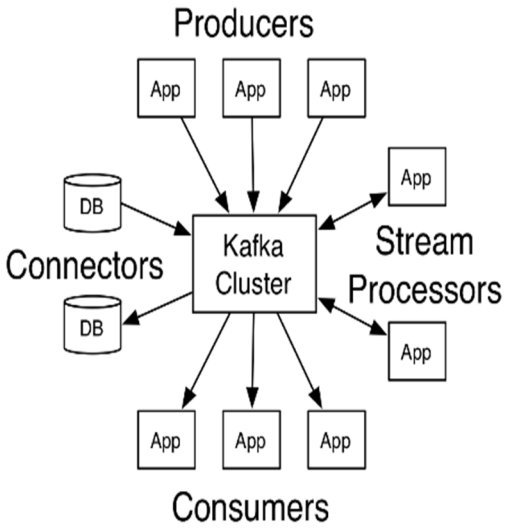


Fig. 2 Kafka architecture

*6) Vertical Scaling Platforms:* Vertical scaling in IT can be described as the building-up of resources while horizontal scaling can be considered as building-out of resources. Both forms of scaling vary in their principles and required resources. Vertical scaling can be basically thought of as the addition of more capability to a single component by the administrators. Vertical scaling is exemplified in the installment of additional memory or processing power on a single system. Among the existing vertical scale-up platforms are High Performance Computing Clusters (HPC), Graphics Processing Unit (GPU), Field Programmable Gate Arrays (FPGA), and Multicore processors [39].

*7) HPC Clusters:* As a parallel computing approach, this involves the connection of different computation elements (CPU, GPU) in a fast network. The commonly used HPC system framework is clusters; parallel computing is more robust on several servers than on specialized systems. HPC clusters [40], also called supercomputers or blade, are machines built with numerous cores with different cache, disk organization, and communication mechanism. They use built strongly with robust hardware that is optimized for speed and throughput. HPC clusters are the most efficient, cost-effective, and flexible HPC platform for performing HPC simulations [41].

### *B. Source of Waste in Data*

Several issues affect the quality of data and make it unclean (dirty). The following are some examples increasing the waste in data according to the literature [39]. These waste sources may exist in information to form the next lean phase related to unclean database management.

*1) Lack of Integrity Constraints:* Integrity constraints means that all instances of a database schema should follow the same procedures at all-time [32]. Lack of input standardization and details will result in poor identified constraints, which will affect the quality of the database and its integrity.

*2) Out-of-Date Values:* One of the most important features for the data is timeliness [5]. When integrating data from multiple data sources, different data sources might enter

data about the same entity at different points in time. Some data entries will lead to obsolete values. Changing the address of an employee over time is a good example of out-of-date values.[37]

*3) Heterogeneous Schemas:* In multiple data sources, different schemas are used to represent entities. In such heterogeneous multi-database systems, values and entities may be inconsistent or suffer from "a loss of a clear identity" [29].

*4) Different Data Entry Rules/ Format:* This issue occurs when integrating data from multiple data sources; hence, multiple sources might have different data entry requirements, rules or formats [34]. For instance, a person's full name might be entered starting with the first name in one data source, while starting with the last name in another one.

*5) Duplicate Data:* duplicate entities will cause a variety of problems affecting the database services such as: *Contradictory data:* Refers to multiple representations that may yield different information. For example, the telephone number of a customer may have the area code that reveals the territory of residence. If the city does not reflect the same area, then contradicting data occurs. [34]; *Overlapping data:* Occurs when integrating data from multiple data sources, the multiple representations may cover different or redundant data from different properties [40]. For example, a customer's telephone number may have the area code that reveals the geographical area. In reality, there is no need to add the area entity; however, if it exists, there is overlapping data; *Semantic structure:* This may occur in the domains that have rich semantic structures (i.e., cultural systems, religious data, etc.) in which multiple thesauri are used with kinds of relations between them [8].

## III. Result and Discussion

In this section, the question of how to detect the bottlenecks of latest work in big data streaming and how to fine tune its deployment is explained. The stream processing developers will investigate those answers in optimizing their Big Data processing architecture for a specific use case.

## IV. Conclusion

In this paper, we first presented a comprehensive study to detecting the bottleneck of big data stream processing. The big data framework that presented in the paper were spark, kafka, storm, and MOA. Second, we made a comparison for the recent work in big data streaming in term of contributions, benefits, and limitation. When the streaming data play an important role in big data mining, our work will help the big data streaming researcher to choose the best and suitable platform in big data streaming projects. In future work, we plan to build a promise framework to handle online big data streaming data based on kafka and spark.

## Acknowledgements

This work is supported by Ministry of Higher Education (MOHE) under Fundamental Research Grant Scheme (FRGS) reference code FRGS/1/2018/ICT04/UTHM/02/3 and from Universiti Malaysia Pahang for the financial sponsorship under Postgraduate Research Grants Scheme (PGRS) with grant No. PGRS190360.

## References

[1] J. Shao, F. Huang, Q. Yang, and G. Luo. (2017). Robust prototype-based learning on data streams IEEE Transactions on Knowledge and Data Engineering (pp. 978-991). vol. 30,.

[2] M. A. Ahmed, R. A. Hasan, A. H. Ali, and M. A. Mohammed. 2019.The classification of the modern arabic poetry using machine learning (pp. 2667-2674) Telkomnika, vol. 17.

[3] A. H. Ali. 2019.A Survey on Vertical and Horizontal Scaling Platforms for Big Data Analytics, ( pp. 138-150) International Journal of Integrated Engineerin vol. 11.

[4] W. I. Yudhistyra, E. M. Risal, I.-s. Raungratanaamporn, and V. Ratanavaraha, "Using Big Data Analytics for Decision Making: Analyzing Customer Behavior using Association Rule Mining in a Gold, Silver, and Precious Metal Trading Company in Indonesia," *International Journal on Data Science,* vol. 1, pp. 57-71, 2020.

[5] A. H. Ali. 2020. Fuzzy generalized Hebbian algorithm for large-scale intrusion detection system ( pp. 81-90) International Journal of Integrated Engineering, vol. 12.

[6] A. H. Ali and M. Z. Abdullah. (2018). Recent trends in distributed online stream processing platform for big data: Survey ( pp. 140-145) in 2018 1st Annual International Conference on Information and Sciences (AiCIS).

[7] R. A. Hasan and M. N. Mohammed.( 2017). A krill herd behaviour inspired load balancing of tasks in cloud computing (pp. 413-424) Studies in Informatics and Control, vol. 26.

[8] A. H. Ali and M. Z. Abdullah. (2019).A novel approach for big data classification based on hybrid parallel dimensionality reduction using spark cluster,Computer Science, vol. 20.

[9] M. A. H. Ali. (2018). An Efficient Model for Data Classification Based on SVM Grid Parameter Optimization and PSO Feature Weight Selection, International Journal of Integrated Engineering.

[10] S. Liang, E. Yilmaz, and E. Kanoulas.( 2018). Collaboratively tracking interests for user clustering in streams of short texts ( pp. 257-272) IEEE Transactions on Knowledge and Data Engineering, vol. 31.

[11] N. AlNuaimi, M. M. Masud, M. A. Serhani, and N. Zaki.( 2019). Streaming feature selection algorithms for big data: A survey," Applied Computing and Informatics.

[12] Y.-J. Lee, M. Lee, M.-Y. Lee, S. J. Hur, and O. Min.(2015) . Design of a scalable data stream channel for big data processing (pp. 537-540) in 2015 17th International Conference on Advanced Communication Technology (ICACT).

[13] P. Le Noac'H, A. Costan, and L. Bougé.(2017). A performance evaluation of Apache Kafka in support of big data streaming applications (pp. 4803-4806) in 2017 IEEE International Conference on Big Data (Big Data).

[14] S. Ramírez-Gallego, S. García, J. M. Benítez, and F. Herrera. (2018). A distributed evolutionary multivariate discretizer for big data processing on apache spark ( pp. 240-250) Swarm and Evolutionary Computation, vol. 38.

[15] O. A. Hammood, M. N. M. Kahar, W. A. Hammood, R. A. Hasan, M. A. Mohammed, A. A. Yoob, et al. (2020). An effective transmit packet coding with trust-based relay nodes in VANETs ( pp. 685-697) Bulletin of Electrical Engineering and Informatics, vol. 9.

[16] O. A. Hammood, M. N. M. Kahar, and M. N. Mohammed. (2017). Enhancement the video quality forwarding Using Receiver-Based Approach (URBA) in Vehicular Ad-Hoc Network(pp. 64-67) in 2017 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET).

[17] O. A. Hammood, M. N. M. Kahar, M. N. Mohammed, W. A. Hammood, and J. Sulaiman.( 2018) .The VANET-Solution Approach for Data Packet Forwarding Improvement (pp. 7423-7427) Advanced Science Letters, vol. 24.

[18] O. A. Hammood, N. Nizam, M. Nafaa, and W. A. Hammood. (2019). RESP: Relay Suitability-based Routing Protocol for Video Streaming in Vehicular Ad Hoc Networks," International Journal of Computers, Communications & Control, vol. 14.

[19] R. A. Hasan, M. A. Mohammed, Z. H. Salih, M. A. B. Ameedeen, N. Ţăpuş, and M. N. Mohammed. (2018). HSO: A Hybrid Swarm Optimization Algorithm for Reducing Energy Consumption in the Cloudlets ( pp. 2144-2154) Telkomnika, vol. 16..

[20] R. A. Hasan, M. A. Mohammed, N. Ţăpuş, and O. A. Hammood . (2017).A comprehensive study: Ant Colony Optimization (ACO) for

facility layout problem (pp. 1-8) in 2017 16th RoEduNet Conference: Networking in Education and Research (RoEduNet) .

[21] A. Bifet, R. Gavaldà, G. Holmes, B. Pfahringer, and F. Bach. (2018). 11 Introduction to MOA and Its Ecosystem.

[22] A. T. Vu, G. D. F. Morales, J. Gama, and A. Bifet. (2014).Distributed adaptive model rules for mining big data streams (pp. 345-353 in 2014 IEEE International Conference on Big Data (Big Data) .

[23] M. A. Mohammed and R. A. Hasan. (2017). Particle swarm optimization for facility layout problems FLP—A comprehensive study ( pp. 93-99) in 2017 13th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP).

[24] M. A. Mohammed, R. A. Hasan, M. A. Ahmed, N. Tapus, M. A. Shanan, M. K. Khaleel, et al(.2018). A Focal load balancer based algorithm for task assignment in cloud environment ( pp. 1-4) in 2018 10th International Conference on Electronics, Computers and Artificial Intelligence (ECAI).

[25] M. A. Mohammed, A. A. Kamil, R. A. Hasan, and N. Tapus.(2019). An Effective Context Sensitive Offloading System for Mobile Cloud Environments using Support Value-based Classification (pp. 687-698) Scalable Computing: Practice and Experience, vol. 20.

[26] M. A. Mohammed, I. A. Mohammed, R. A. Hasan, N. Ţăpuş, A. H. Ali, and O. A. Hammood. (2019). Green Energy Sources: Issues and Challenges in 2019 18th RoEduNet Conference: Networking in Education and Research (RoEduNet) ( pp. 1-8).

[27] A. Jain.( 2017 ). Mastering apache storm: Real-time big data streaming using kafka, hbase and redis: Packt Publishing Ltd,.

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

[29] S. A.-b. Salman, A.-H. A. Salih, A. H. Ali, M. K. Khaleel, and M. A. Mohammed.(2018). A New Model for Iris Classification Based on Naïve Bayes Grid Parameters Optimization (pp. 150-155). International Journal of Sciences: Basic and Applied Research (IJSBAR), vol. 40.

[30] M. A. Mohammed, Z. H. Salih, N. Ţăpuş, and R. A. K. Hasan. (2016). Security and accountability for sharing the data stored in the cloud (pp. 1-5). in 2016 15th RoEduNet Conference: Networking in Education and Research .

[31] M. A. Mohammed and N. ŢĂPUŞ . (2017) . A Novel Approach of Reducing Energy Consumption by Utilizing Enthalpy in Mobile Cloud Computing (pp. 425-434). Studies in Informatics and Control, vol. 26.

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

[33] P. Karunaratne, S. Karunasekera, and A. Harwood. (2017). Distributed stream clustering using micro-clusters on Apache Storm ( pp. 74-84). Journal of Parallel and Distributed Computing, vol. 108.

[34] M. A. A. Royida A. Ibrahem Alhayali, Yasmin Makki Mohialden, Ahmed H. Ali. (2020). Efficient method for breast cancer classification based on ensemble hoffeding tree and naïve Bayes (pp. 1074-1080). Indonesian Journal of Electrical Engineering and Computer Science, vol. 18.

[35] A.-H. A. Salih, A. H. Ali, and N. Y. Hashim. Jaya: An Evolutionary Optimization Technique for Obtaining the Optimal Dthr Value of Evolving Clustering Method (ECM).

[36] 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.(2017). in 2017 9th International Conference on Electronics, Computers and Artificial Intelligence (ECAI).

[37] Z. Alqadi, M. Abuzalata, Y. Eltous, and G. M. Qaryouti, "Analysis of fingerprint minutiae to form fingerprint identifier," *JOIV: International Journal on Informatics Visualization,* vol. 4, pp. 10-15, 2020.

[38] L. Shi, Y. Wu, L. Liu, X. Sun, and L. Jiang. (2018). Event detection and identification of influential spreaders in social media data streams (pp. 34-46). Big Data Mining and Analytics, vol. 1.

[39] T. Al-Khateeb, M. M. Masud, K. M. Al-Naami, S. E. Seker, A. M. Mustafa, L. Khan, et al. (2015). Recurring and novel class detection using class-based ensemble for evolving data stream. ( pp. 2752-2764). IEEE Transactions on Knowledge and Data Engineering, vol. 28.

[40] Arif, L. N. U., Barakbah, A. R., Sudarsono, A. & Edelani, R. 2019. Big Data Environment for Realtime Earthquake Data Acquisition and Visualization. JOIV: International Journal on Informatics Visualization, 3, 365-376.

[41] Y. Awasthi, "Press “A” for Artificial Intelligence in Agriculture: A Review," *JOIV: International Journal on Informatics Visualization,* vol. 4, pp. 112-116, 2020.