Scalable Machine Learning for IoT-Based Earthquake Data Analysis Using Distributed Big Data Frameworks

C Siva[1], K Yatheendra[2]

[1]P.G Scholar, Department of MCA, Sri Venkatesa Perumal College of Engineering & Technology, Puttur, E-mail: sivaramesh5119555@gmail.com, ORC-ID: https://orcid.org/0009-0004-1734-5192

[2]Assistant Professor, Department of CSE(AI & ML), Sri Venkatesa Perumal College of Engineering & Technology, Puttur, E-mail: k.yatheendra84@gmail.com, ORC-ID: https://orcid.org/0009-0003-1382-8587

Abstract: Internet of Things (IoT) data is growing at an exponential rate, which means that scalable and efficient big data analysis frameworks are needed. This is because standard machine learning methods can't handle huge datasets with real-time needs. The earthquake detection dataset includes details about events like size, location, alerts, and the chance of a tsunami. These are important details for predictive models. The dataset helps create automatic big data analysis services that are in line with the goals of Industry 4.0 and Society 5.0. This makes sure that accurate and timely information about environmental risks is shared. The AutoBDA framework combines Hadoop and Spark to allow distributed processing, which makes machine learning jobs more scalable. For predicting the size of an earthquake, Logistic Regression is used as the main method. The performance of the model is shown by an RMSE of 0.10 and an MAE of 0.07. This shows how well the distributed analysis approach works. A Flask-based deployment allows seamless user interaction, where test input values are analyzed and predictions are generated in real time using Spark's in-memory processing, reducing communication and computation costs. Ensemble algorithms, like Random Forest and Gradient Boosted Trees, can be used to improve speed even more. These algorithms are more accurate than others. Random Forest has an RMSE of 0.08 and an MAE of 0.04. Gradient Boosted Trees has the best results, with an RMSE of 0.03 and an MAE of 0.006. Adding healthcare and resource management to AutoBDA's list of services can make it more inclusive and flexible across many fields.

"Index Terms: *Automatic service composition, big data analysis, edge-computing, model-driven, industry 4.0, society 5.0".*

1. INTRODUCTION

The need for edge-based, data-driven solutions that can handle the complexity and scale of Big Data has grown a lot because of the fast growth of the Internet of Things (IoT) and the rising demand for high-quality services [1]. Big Data Analysis (BDA) used to be defined by the 3Vs: volume, velocity, and variety. Over time, the idea has grown to include value and variability, giving rise to the 5Vs. These stress its important role in advancing modern data science fields like machine learning (ML) and deep learning (DL) [2]. However, turning such huge amounts of multidimensional data into knowledge that can be used is hard because it takes a lot of computing power, subject knowledge, and time to process. In real-world

applications, these needs often make it harder to expand, include everyone, and change [3].

New architectures and optimization techniques have been the focus of recent research that tries to solve these problems. Model-driven architectures, for example, let system designers turn complicated computer processes into reusable software reference architectures. This makes the system more scalable and independent of the domain [4]. Also, effective job scheduling and optimization methods have been created to improve speed and cost-effectiveness in big data platforms like Apache Spark [2]. These include reinforcement learning-based methods for distributed environments. In the same way, ontology-driven models for privacy protection show how important it is to make systems that are not only useful but also safe and reliable [5]. These changes are in line with the bigger idea of Big Data as a Service (BDaaS), in which analytics tools are made available as services that can be used, scaled, and changed as needed [6].

At the same time, there has been a push to make BDA models work with paradigm-shifting ideas like Industry 4.0 and Society 5.0. Industry 4.0 focuses on connecting cyber-physical systems, automation, and smart analytics. Society 5.0, on the other hand, imagines technology-driven, people-centered solutions that make things more sustainable and open to everyone [7, 8]. All of these frameworks show how important it is to have flexible and all-encompassing BDA solutions that can handle different needs across fields and industries, especially for users on the edge who don't have a lot of technical knowledge.

To deal with these problems, the AutoBDA system adds an automatic service composition (ASC) and a model-driven software reference architecture (SRA). This method automates BDA processes with multiple steps, lowers the need for human help, and improves flexibility in a variety of settings. AutoBDA provides an open, scalable, and usable solution that meets the changing needs of Big Data and is in line with the goals of Industry 4.0 and Society 5.0. It does this by turning architectural decision processes into real system topologies.

2. LITERATURE REVIEW

In the past few years, a lot of new research has been done in the area of automated big data analytics. This is because datasets are getting more complicated, there are more computing environments, and people want smart, scalable, and cost-effective solutions. Siriweera, Paik, and Huang [9] made a big step forward in this area when they came up with a constraint-driven complexity-aware data science workflow for AutoBDA. Their main goal was to automate big data workflows with multiple steps by combining model-driven software reference design with automatic service composition. This made it so that people didn't have to do as much work by hand. The writers stressed that AutoBDA is flexible, open to everyone, and scalable, showing that it can be used in many areas while making the system simpler. Their method made architectural choices clear and usable again and again, which laid the groundwork for flexible workflow automation in big data systems.

In line with this vision, the Japanese government's Society 5.0 program [10] suggested a bigger social-technical framework in which technology, like AI and big data, helps with innovation that is focused on people. Society 5.0 builds on Industry 4.0 by imagining a super-smart society that solves social problems while also making business progress. This

idea emphasizes the significance of human-centered data-driven ecosystems by building resilience, adaptability, and inclusion into digital frameworks. It also gives developments like AutoBDA a way to make sure they are in line with social goals.

Siriweera and Naruse [11] did a survey on cloud robotics architecture and came up with a model-driven reference design for decentralized multicloud heterogeneous robotics platforms. This builds on previous architectural progress. Their work solved the problems of combining robotics with distributed cloud systems by suggesting designs that are modular, reusable, and not limited to one area. By applying the ideas of model-driven approaches to robots and multicloud platforms, they showed how architectural abstraction could lower differences and make it easier for systems to work together. This work fits very well with the AutoBDA idea of making solutions that are open to everyone and can be used in a variety of computing settings.

Another big problem in big data settings was solved by Chen, Jiao, Liu, and Wang [12]. They created EdgeDR, an online system for demand response in edge clouds, to deal with this problem. By allowing dynamic demand-response allocation, the writers came up with a way to make edge computing both efficient and fair. Their system was mainly about cutting down on energy use and making edge resources more cost-effective, which is very important for real-time data-driven services. EdgeDR shows how edge-aware design principles can improve the responsiveness, scalability, and sustainability of a system. This makes it work better with bigger frameworks like AutoBDA that aim to provide agile data analytics.

At the same time, Wu, He, Lin, and Mao [13] looked at the problems with federated learning in mobile edge computing systems, where client stability is often a problem. They came up with ways to speed up federated learning over clients that don't care about reliability. This would improve total performance and make sure that the system was stable. In settings with different clients and changing resource availability, their input is especially important. This fits with the larger goal of big data systems that are distributed and include everyone. Their work improves confidence and efficiency by fixing problems with federated learning. These are important for putting large-scale data-driven frameworks to use in the real world.

Dou, Zhang, Liu, and Chen [14] looked into another aspect of using big data. They came up with HireSome-II, a privacy-aware cross-cloud service building method. Their model dealt with the important issue of privacy in cross-cloud settings, where private data may move between different service providers. They talked about both scalability and reliability, which are very important in big data analytics, by suggesting ways to make service composition safe and effective. Their work shows how important it is to have privacy-protecting features that keep user trust and data protection even as systems become more distributed and automated.

Yu et al. [15] explored how to make clustering algorithms work better for big, heterogeneous systems. They came up with an automatic k-means clustering method for many-core supercomputers. By making scalable clustering algorithms, they were able to get around the speed problems that come with working with very large datasets on a variety of hardware. The suggested answer showed how new ideas in algorithms could work with improvements in

architecture to make sure that computer speed keeps up with the needs of growing data. Frameworks like AutoBDA depend on efficient core algorithms to support high-level automation, and these kinds of methods are needed to make the analytics layer work.

Garcia, Toril, Oliver, Luna-Ramirez, and Garcia [16] looked into how big data can be used in mobile networks. They looked into how big data analytics can be used for automatic Quality of Experience (QoE) management. Their study showed that network providers could use analytics to track, guess, and improve QoE in real time. By automating QoE management, they suggested a change from reactive to proactive optimization of networks. Their work shows how big data analytics can be useful in user-centered apps, which is in line with the goals of adaptability and inclusion set out in models like Society 5.0 [10].

Li, Lu, and Meng [17] created Bigprovision, a system that helps make the best use of resources for analytics tasks. They were the first to use provisioning frameworks for big data analytics. By managing resources on the fly in big data environments, their framework handled both efficiency and scalability. Performance and cost-effectiveness were both improved by Bigprovision. This shows how smart resource management can help analytics-as-a-service models grow. This work fits with model-based and service-oriented methods, which shows how important it is to be able to adapt when setting up big systems.

Finally, Sparks, Venkataraman, Kaftan, Franklin, and Recht [18] showed KeystoneML, a better path for big advanced analytics projects. KeystoneML worked on making machine learning processes more efficient by making changes at the system level that made them scalable and cut down on processing overhead. Their work directly helped frameworks like AutoBDA reach their goals by designing end-to-end analytics workflows with optimization in mind. These frameworks rely on seamless and automated workflow composition. KeystoneML showed that improvements in system-level engineering could work with new algorithms and building designs to make big data solutions that work really well.

3. MATERIALS AND METHODS

The suggested system, called AutoBDA, is created as a model-driven example of how to analyze large amounts of data automatically. This is shown by using earthquake monitoring services. For predictive modeling, it uses an earthquake dataset that has been preprocessed and split into training and testing subsets. The dataset includes seismic characteristics like magnitude, location, alert types, and tsunami indicators. To make sure it can grow and work well, the system combines Hadoop and Apache Spark, using both distributed and in-memory computing [9]. Spark MLlib is used to do machine learning tasks. Logistic Regression is the base method, and ensemble techniques like Random Forest and Gradient Boosted Trees (GBT) are added to make the results more accurate [12]. The system uses Automatic Service Composition (ASC) for process automation and Software Reference Architecture (SRA) for adaptability. It is set up through a Flask-based web interface that lets you make predictions in real time [10].

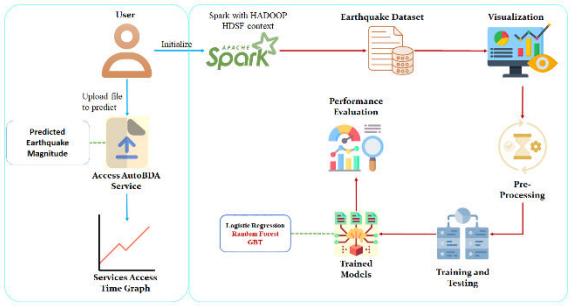

Fig.1 Proposed Architecture

The suggested system design (fig.1) shows how the AutoBDA framework can be used to predict the size of an earthquake using Spark in a Hadoop HDFS environment. The user starts the process by sending seismic data to the AutoBDA service. The data is then preprocessed and visualized to help the user understand it better. Machine learning models like Logistic Regression, Random Forest, and Gradient Boosted Trees (GBT) are then trained and tested on the dataset. Performance review checks how accurate predictions are, and outputs are made by trained models. Lastly, users can look at real-time forecasts and service access time [22] graphs to keep an eye on how well the system is working.

a) Dataset Collection:

The dataset for earthquakes used in this study has 782 rows and 24 columns with specific information about earthquakes that is needed for predictive modeling. It has information about the tsunami's title, magnitude, type of alert, importance, depth, location, latitude, longitude, continent, and country, as well as information about time such as the year, month, day, hour, minute, and second. These multidimensional features give information about both space and time, which makes preprocessing, training, and testing in distributed systems like Apache Spark and Hadoop more effective. This makes it possible to accurately predict the size of earthquakes on a large scale.

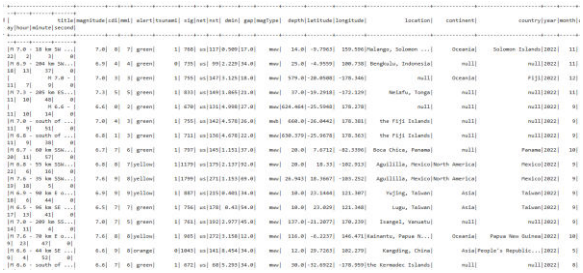

Fig.2 Dataset Collection

b) Visualization:

The visualization of the information gives a detailed look at earthquake characteristics through graphs, showing value distributions across columns, the ten countries most likely to experience earthquakes, the frequency of alert types, the severity levels, and the risks of tsunamis. These graphics make it easy to understand patterns in seismic data, which helps find outliers and important correlations before using machine learning models. By showing how often and where important characteristics like magnitude and alert categories appear, users can get a better idea of how the dataset is put together. This helps with smart preprocessing, building reliable models, and making more accurate predictions in large-scale data-driven analysis [19].

c) Data Processing:

Systematic preprocessing is done on the information to make sure it is good enough for predictive modeling. With Spark's StringIndexer, categorical characteristics like alerts are turned into numeric values, and the skip function takes care of missing values. After being processed, the dataset is made more uniform by changing some of its features. It is

then split into training and testing groups. This step keeps the review fair and stops model bias. Preprocessing [20] improves consistency, cuts down on errors, and sets up structured input. This makes it possible for scalable big data [21] workflows that can accurately predict the size of an earthquake using distributed processing tools like Apache Spark.

d) Training and Testing:

Logistic Regression is trained on 80% of the preprocessed earthquake dataset for prediction modeling. The other 20% is saved for testing. This supervised learning method predicts the size of earthquakes by looking at data from past events. Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) are used to measure the accuracy of the model. To improve the accuracy of predictions, ensemble extensions like Random Forest and Gradient Boosted Trees are also used. The way the models are trained and tested makes sure they are robust, which lets us compare models reliably and learn how to make earthquake forecasting work better in settings with a lot of data.

e) Algorithms:

Logistic Regression: Logistic Regression is used to describe the connection between features of an earthquake and its size. It gives us a starting point for making predictions about how likely it is that an earthquake will happen, which lets us look at past seismic data. It is easy to understand and use, which makes training on distributed datasets quick and effective. It also helps make accurate real-time predictions through Spark MLlib and lays the groundwork for further improvements with ensemble methods.

$$\hat{y}_i = \sigma(w^T x + b) = \frac{1}{1 + e^{-(w^T x + b)}} \quad (1)$$

Random Forest: By mixing several decision trees, Random Forest is used to make predictions more accurate. Each tree checks a certain group of seismic features, and when all of them are added together, the result lowers both overfitting and variance. This group method finds complicated patterns in earthquake records, which makes magnitude predictions more accurate. Integration with Spark makes sure that processing is spread out efficiently, allowing scalable and robust analysis across big IoT datasets. This makes the system better at making predictions than single-model approaches.

$$Gini = 1 - \sum_{i=1}^{C} (Pi)^2 \quad (2)$$

Gradient Boosted Trees (GBT): Gradient Boosted Trees build a group of weak learners over and over again, reducing the number of wrong predictions about earthquake sizes. It accurately models the non-linear relationships and connections in seismic data, making it more accurate. GBT handles large IoT datasets by using Spark's distributed computation to keep RMSE and MAE as low as possible. Its ability to guess well and quickly makes it good for real-time earthquake forecasting. The AutoBDA framework lets end users make quick, data-driven decisions.

4. EXPERIMENTAL RESULTS

RMSE: Root mean square error (RMSE) is a way to find the average difference between what a statistical model said would happen and what actually happened. It is the standard deviation of the residuals in math terms. The residuals show how far away the regression line is from the data points.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} ||y(i) - \hat{y}(i)||^2}{N}} \qquad (3)$$

MAE: Absolute Error is the amount of mistake when you measure something. It's the difference between what was recorded and what was "true." To give you an example, if the scale says you weigh 90 pounds but you know you really weigh 89 pounds, the scale is off by 1 pound.

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \qquad (4)$$

In Table (1), the performance review table uses RMSE and MAE metrics to measure how accurate predictions are. Gradient Boosting does better than others, getting the lowest errors and showing that it is more reliable for predicting earthquake magnitude.

Table.1 Performance Evaluation Table

Algorithm Name	RMSE	MAE
Logistic Regression	0.106836	0.079522
Random Forest	0.087549	0.047899
Gradient Boosting	**0.036851**	**0.006173**

Graph.1 Comparison Graph

Graph (1) shows how well different models work, with green bars showing RMSE and red bars showing MAE. Gradient Boosting displays the lowest values, which clearly proves that it is more accurate than other algorithms.

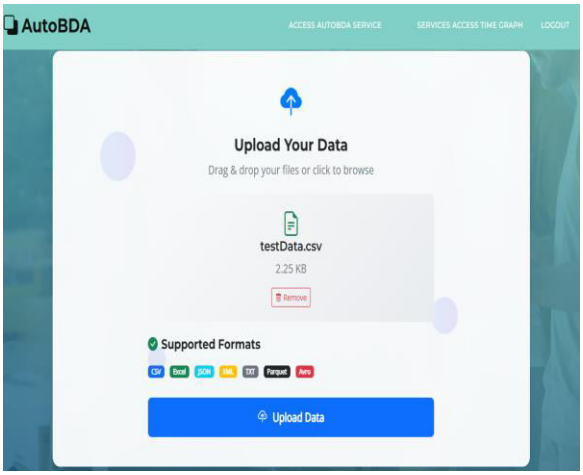

Fig.2 Upload Your Dataset

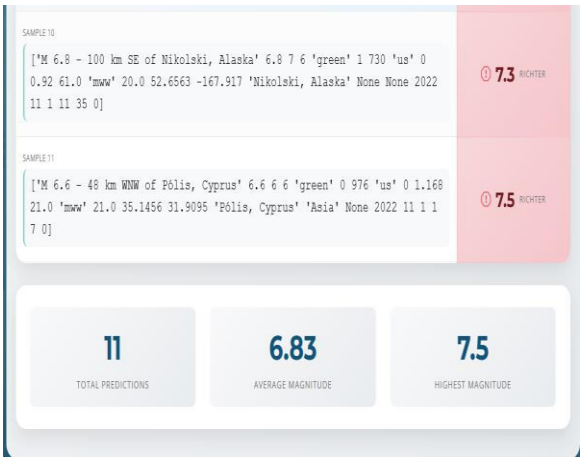

Fig.3 Final Result

Fig.4 Prediction Result

5. CONCLUSION

In conclusion, the AutoBDA framework shows a scalable and effective way to automate big data analysis by using Hadoop and Apache Spark for distributed and in-memory computation. This gets around the problems that traditional machine learning has with handling very large Internet of Things (IoT) datasets. The earthquake detection dataset, which included information about the size, location, kinds of alerts, and risks of tsunamis, was used to build predictive models and evaluate performance. Logistic Regression, which was done with Spark MLlib, was used as the base algorithm. It got 0.10 RMSE and 0.07 MAE, which showed that the system could handle big amounts of data fairly accurately. So that predictions could be made even better, ensemble learning methods were added. Random Forest got 0.08 RMSE and 0.04 MAE, and Gradient Boosted Trees did even better with 0.03 RMSE and 0.006 MAE. Deployment through a Flask-based web application made real-time forecast possible by letting users enter test data and get results right away thanks to Spark's efficient processing. In general, the system shows how mixing distributed frameworks, machine learning algorithms, and service-oriented deployment can lead to solutions that are flexible, accurate, and open to everyone, which is in line with the aims of Industry 4.0 and Society 5.0.

In the future, the AutoBDA framework could include more advanced deep learning models, like LSTM and CNN, that can pick up on complex temporal and spatial trends in seismic data and make predictions more accurate. Adding real-time IoT data streams and edge computing to the system can cut down on latency and make it more responsive. Adding automated alerts, multi-region forecasting, and cloud-native deployment will also make the platform more scalable, resilient, and accessible. This will allow for more disaster management apps and be more in line with the goals of Industry 4.0 and Society 5.0.

REFERENCES

[1] Wehrmeister, K., Pastor, A., Carreras Rodriguez, L., & Monti, A. (2025). Big Data Reference Architecture for the Energy Sector. Sustainability, 17(14), 6488.

[2] A D Venkatesh, K Bhaskar, G Swapna, & G Viswanath. (2025). Advanced Hybrid Learning

Architecture for Precision Cardiovascular Risk Assessment. In International Journal of Health Sciences and Pharmacy (IJHSP) (Vol. 9, Number 1, pp. 50–61). Zenodo. https://doi.org/10.5281/zenodo.15448632

[3] Ardagna, C. A., Bellandi, V., Bezzi, M., Ceravolo, P., Damiani, E., & Hebert, C. (2018). Model-based big data analytics-as-a-service: take big data to the next level. IEEE Transactions on Services Computing, 14(2), 516-529.

[4] Ranjan, R., Li, Z., Villari, M., Liu, Y., & Georgeakopoulos, D. (2020). Software-driven big data analytics: Guest editors' introduction. Computing, 102(6), 1409-1417.

[5] Castellanos, C., Correal, D., & Rodriguez, J. D. (2018, September). Executing architectural models for big data analytics. In European Conference on Software Architecture (pp. 364-371). Cham: Springer International Publishing.

[6] Ardagna, C. A., Bellandi, V., Bezzi, M., Ceravolo, P., Damiani, E., & Hebert, C. (2021). Model-based big data analytics-as-a-service: Take big data to the next level. IEEE Transactions on Services Computing, 14(2), 516–529.

[7] Ganesh, B. R. ., B M, P., Prasad K, K. ., Swapna, G., & G, Viswanath. (2025). Data Mining-Driven Multi-Feature Selection for Chronic Disease Forecasting. Journal of Neonatal Surgery, 14(5S), 108–124. https://doi.org/10.52783/jns.v14.1993

[8] Aceto, G., Persico, V., & Pescapé, A. (2019). A survey on information and communication technologies for Industry 4.0: State-of-the-art, taxonomies, perspectives, and challenges. IEEE Communications Surveys & Tutorials, 21(4), 3467–3501.

[9] Siriweera, A., Paik, I., & Huang, H. (2023). Constraint-driven complexity-aware data science workflow for AutoBDA. IEEE Transactions on Big Data, 9(6), 1438–1457.

[10] Cabinet Office, Government of Japan. (2022, November 11). Society 5.0. https://www8.cao.go.jp/cstp/english/society5%5F0/index.html

[11] Siriweera, A., & Naruse, K. (2021). Survey on cloud robotics architecture and model-driven reference architecture for decentralized multicloud heterogeneous-robotics platform. IEEE Access, 9, 40521–40539.

[12] Chen, S., Jiao, L., Liu, F., & Wang, L. (2022). EdgeDR: An online mechanism design for demand response in edge clouds. IEEE Transactions on Parallel and Distributed Systems, 33(2), 343–358.

[13] Wu, W., He, L., Lin, W., & Mao, R. (2021). Accelerating federated learning over reliability-agnostic clients in mobile edge computing systems. IEEE Transactions on Parallel and Distributed Systems, 32(7), 1539–1551.

[14] Dou, W., Zhang, X., Liu, J., & Chen, J. (2015). HireSome-II: Towards privacy aware cross-cloud service composition for big data applications. IEEE Transactions on Parallel and Distributed Systems, 26(2), 455–466.

[15] Naresh, M., Gudditti., M., Viswanath, & SunilKumarReddy, M.T. (2014). Distributed Utility-Based Energy Efficient Cooperative Medium Access Control in MANETS.

[16] Garcia, A. J., Toril, M., Oliver, P., Luna-Ramirez, S., & Garcia, R. (2019). Big data analytics for automated QoE management in mobile networks. IEEE Communications Magazine, 57(8), 91–97.

[17] Li, H., Lu, K., & Meng, S. (2015). Bigprovision: A provisioning framework for big data analytics. IEEE Network, 29(5), 50–56.

[18] Sparks, E. R., Venkataraman, S., Kaftan, T., Franklin, M. J., & Recht, B. (2017). KeystoneML: Optimizing pipelines for large-scale advanced analytics. In Proceedings of the IEEE 33rd International Conference on Data Engineering (pp. 535–546).

[19] Kumar, K., Udaya Suriya Rajkumar, D., Viswanath, G., & Mahalakshmi, J. (2024). A Hybrid Particle Swarm Optimization and C4.5 for Network Intrusion Detection and Prevention System. International Journal of Computing, 23(1), 109-115. https://doi.org/10.47839/ijc.23.1.3442

[20] Wu, D., Zhu, L., Lu, Q., & Sakr, S. (2018). HDM: A composable framework for big data processing. IEEE Transactions on Big Data, 4(2), 150–163.

[21] Yang, L., Yang, Y., Mgaya, G. B., Zhang, B., Chen, L., & Liu, H. (2021). Novel fast networking approaches mining underlying structures from investment big data. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 51(10), 6319–6329.

[22] Basanta-Val, P., Audsley, N. C., Wellings, A. J., Gray, I., & Fernández Garc, N. (2016). Architecting time-critical big-data systems. IEEE Transactions on Big Data, 2(4), 310–324.