INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD)
© 2026 IJNRD | Volume 11, Issue 6, June 2026 | ISSN: 2456-4184 | IJNRD.ORG

Earthquake Analysis and Prediction Using Automated Big Data Processing Framework

KADALI ROSHINI
Master Of Computer Applications
Ideal College Of Arts & Sciences,
Autonomous, Affiliated To Adikavi Nannaya University - Rajamahendravaram
Kakinada

M. KAMESWARA RAO
Assistant.Prof, Master Of Computer Applications
Ideal College Of Arts & Sciences,
Autonomous, Affiliated To Adikavi Nannaya University - Rajamahendravaram
Kakinada

Dr. V.S.V DEEPAK
HOD, department of computer science
Ideal College Of Arts & Sciences,
Autonomous, Affiliated To Adikavi Nannaya University - Rajamahendravaram
Kakinada

***Abstract*— Earthquake prediction using big data analytics has become an important research area due to the rapid growth of IoT-generated seismic information and the need for accurate disaster forecasting systems. This work focuses on enhancing an Automated Big Data Analysis framework by integrating advanced machine learning algorithms with distributed processing technologies such as Hadoop and Apache Spark. The framework performs large-scale data preprocessing, feature transformation, normalization, and automated analytical visualization to support efficient earthquake magnitude prediction. The existing system utilized Logistic Regression, which produced comparatively higher prediction errors. To improve performance, Random Forest and Gradient Boosting Tree algorithms were introduced as extension models. Experimental results show that the Gradient Boosting Tree model achieved the best performance with 0.03 RMSE and 0.006 MAE, significantly reducing prediction error compared to traditional methods.**

Keywords*— *Big Data, Earthquake, Apache Spark, Machine Learning

I. INTRODUCTION

Natural disasters such as earthquakes continue to cause severe damage to human life, infrastructure, and economic stability across the world. With the rapid growth of sensing technologies and Internet of Things devices, enormous volumes of seismic and environmental data are generated every day. Managing and analysing such massive datasets using conventional data processing techniques has become increasingly difficult due to limitations in scalability, processing speed, and storage efficiency. As a result, big data analytics has emerged as an important research area for handling complex and continuously growing earthquake-related information.

Recent advancements in distributed computing technologies have enabled researchers to process large-scale datasets more efficiently and perform faster analytical operations. Frameworks capable of distributed storage and parallel computation allow data scientists to extract meaningful insights from seismic records, sensor streams, and geographical observations. Machine learning techniques further contribute by identifying hidden patterns, correlations, and behavioural trends within earthquake datasets. These analytical methods support improved decision-making and help researchers understand seismic activity more effectively.

II. RELATED WORK

Dou et al. (2015) presented a privacy-aware cross-cloud service composition framework for big data applications. Their work focused on secure integration of distributed services while maintaining efficient data processing across multiple cloud platforms. The framework addressed challenges related to data privacy, interoperability, and scalable computation, which are critical issues in large-scale analytical environments. The study demonstrated that automated service composition can significantly improve flexibility and resource utilization in distributed big data systems. This research laid an important foundation for secure and adaptive analytical architectures used in modern automated big data frameworks.

Li et al. (2015) introduced the BigProvision framework to enhance resource provisioning in big data analytics environments. The authors emphasized dynamic allocation of computational resources to reduce processing delays and improve execution efficiency. Their framework supported scalable data analytics by optimizing resource management strategies in distributed systems. Similarly, Sparks et al. (2017) proposed KeystoneML, an optimized machine learning pipeline framework for large-scale analytics. Their work concentrated on improving machine learning workflow execution by reducing computational overhead and enhancing pipeline optimization. The framework demonstrated the importance of scalable machine learning architectures for handling massive datasets efficiently.

Ko et al. (2018) developed Closha, a workflow-based bioinformatics platform designed for processing large sequencing datasets. Their approach simplified workflow automation and improved management of heterogeneous data environments. Aceto et al. (2019) later explored the role of Industry 4.0 technologies and IoT-based communication systems in big data analytics. Their survey highlighted the increasing demand for scalable analytical frameworks capable of handling real-time industrial data streams. Garcia et al. (2019) further applied big data analytics to automated Quality of Experience management in mobile networks, demonstrating how intelligent analytical systems can improve service performance and operational reliability.

Zhao et al. (2020) proposed a multi-source urban big data analysis model for enterprise location recommendation. Their research integrated heterogeneous urban datasets to improve analytical decision-making accuracy. Ardagna et al. (2021) extended this concept by introducing a model-based Big Data Analytics-as-a-Service architecture that enabled flexible cloud-

based analytical operations. Islam et al. (2022) proposed a deep reinforcement learning-based Spark scheduling mechanism to optimize computational cost and processing efficiency in cloud environments. More recently, Siriweera et al. (2023) introduced a complexity-aware AutoBDA workflow framework that improved automation, scalability, and adaptability in big data analytics systems, providing strong support for intelligent distributed analytical applications.

Table: Summary of Key Literature Contributions and Their Impact on Current Research:

Author	Contribution	Impact on Research
Dou et al. (2015)	Developed secure cloud service composition for big data systems.	Improved security and service integration in big data analytics.
Li et al. (2015)	Proposed BigProvision for resource management in analytics.	Increased processing efficiency in distributed systems.
Sparks et al. (2017)	Introduced KeystoneML for machine learning pipeline optimization.	Enhanced large-scale machine learning performance.
Ko et al. (2018)	Created Closha workflow system for massive data analysis.	Simplified workflow automation and data handling.
Aceto et al. (2019)	Surveyed Industry 4.0 and IoT communication technologies.	Highlighted the need for scalable big data frameworks.
Garcia et al. (2019)	Applied analytics for mobile network quality management.	Improved automated network performance analysis.
Zhao et al. (2020)	Proposed multi-source urban big data analysis model.	Improved decision-making using multiple datasets.
Ardagna et al. (2021)	Developed Big Data Analytics-as-a-Service architecture.	Supported flexible cloud-based analytics systems.
Islam et al. (2022)	Designed deep learning-based Spark job scheduling.	Reduced processing time and computational cost.
Siriweera et al. (2023)	Proposed AutoBDA workflow for automated analytics.	Improved automation and scalability in big data systems.

III. PROPOSED APPROACH

Large-scale earthquake datasets collected from seismic sensors and monitoring systems require efficient analytical methods to process, manage, and predict earthquake magnitude accurately. The proposed approach develops an enhanced Automated Big Data Analysis framework by combining distributed computing technologies with advanced machine learning algorithms. Hadoop and Apache Spark are integrated into the framework to support high-speed parallel processing and efficient handling of massive earthquake datasets. This distributed environment reduces computational overhead and improves scalability during analytical operations.

The first stage of the framework performs data preprocessing and exploration. During this phase, missing values, noisy records, and redundant attributes are removed to improve data quality. Categorical attributes are transformed into numerical values using indexing methods, while normalization techniques are applied to maintain uniform data distribution. The processed dataset is then divided into training and testing sections using an 80:20 ratio for model development and evaluation.

For prediction analysis, the framework utilizes multiple machine learning algorithms. Logistic Regression is initially used as the baseline model for earthquake magnitude prediction. To improve analytical accuracy and reduce prediction error, the framework incorporates Random Forest and Gradient Boosting Tree algorithms as extension models. Random Forest increases prediction reliability through ensemble learning, while Gradient Boosting Tree improves model performance by minimizing residual prediction errors iteratively. Spark MLlib is employed to train and evaluate these models efficiently within the distributed environment.

Performance evaluation is carried out using Root Mean Square Error and Mean Absolute Error metrics. Experimental analysis shows that the Gradient Boosting Tree algorithm achieved lower RMSE and MAE values compared to Logistic Regression and Random Forest models. The trained prediction model is finally deployed through a Flask-based web application, enabling users to upload test data and obtain real-time earthquake magnitude predictions with improved accuracy and faster response time.

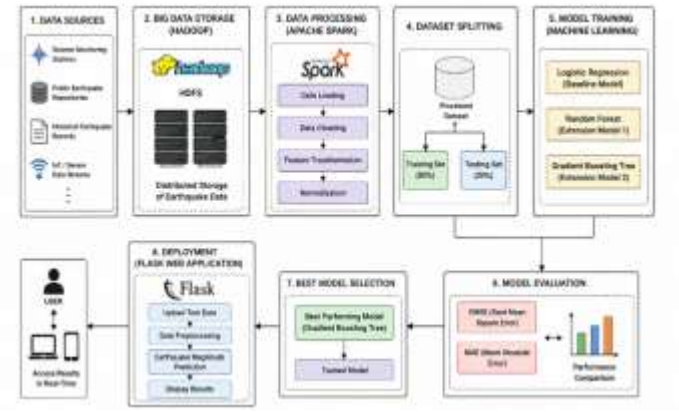

Figure 1: Big data earthquake prediction workflow

IV. METHODOLOGIES

Algorithm: Gradient Boosting Tree Based AutoBDA Earthquake Prediction

---

Input : Earthquake Dataset D
Output : Predicted Earthquake Magnitude

Step 1: Start

Step 2: Load earthquake dataset D using Apache Spark

Step 3: Perform data preprocessing
a) Remove missing values
b) Remove duplicate records
c) Convert categorical values to numeric
d) Normalize dataset values

Step 4: Split dataset into training and testing data
Training_Data = 80%

Testing_Data = 20%

Step 5: Initialize Gradient Boosting Tree model
GBT_Model ← GradientBoostingTree()

Step 6: Train model using Training_Data
GBT_Model.train(Training_Data)

Step 7: Apply trained model on Testing_Data
Predicted_Output ← GBT_Model.predict(Testing_Data)

Step 8: Calculate evaluation metrics
RMSE ← RootMeanSquareError(Actual, Predicted_Output)
MAE ← MeanAbsoluteError(Actual, Predicted_Output)

Step 9: Compare obtained RMSE and MAE values
with existing Logistic Regression
and Random Forest models

Step 10: If GBT error is minimum
Select GBT as Best Prediction Model

Step 11: Deploy trained GBT model using Flask Web Service

Step 12: Accept user test data through web interface

Step 13: Predict earthquake magnitude for uploaded data

Step 14: Display prediction results to user

Step 15: Stop

Dataset Collection

The methodology begins with collecting large-scale earthquake datasets from seismic monitoring sources and publicly available repositories. The dataset contains important earthquake-related attributes such as latitude, longitude, depth, magnitude, alert level, tsunami risk, and geographical location details. Since earthquake datasets are generated continuously from multiple sensing devices and monitoring systems, the collected data volume becomes very large and requires efficient storage and processing mechanisms. The collected dataset forms the foundation for analytical processing and machine learning-based prediction.

Distributed Big Data Environment Setup

Apache Hadoop and Apache Spark frameworks are configured to support distributed storage and parallel data processing. Hadoop Distributed File System (HDFS) is used to store massive earthquake datasets, while Spark enables in-memory distributed computation for faster analytical operations. This setup improves scalability and reduces processing time compared to traditional standalone systems.

Data Loading and Exploration

The earthquake dataset is loaded into the Spark environment using PySpark libraries. Initial exploration is performed to identify the number of records, feature types, missing values, and statistical distributions. Basic analytical operations such as mean, minimum, maximum, and standard deviation calculations are performed to understand dataset characteristics before preprocessing.

Data Cleaning

Raw earthquake datasets often contain incomplete records, null values, duplicated entries, and noisy information. During this step, unnecessary attributes are removed and missing values are handled using appropriate preprocessing methods. Cleaning the dataset improves analytical reliability and prevents prediction errors during model training.

Feature Transformation

Several dataset attributes contain categorical values such as alert types and location labels. These categorical features are converted into numerical representations using String Indexing techniques available in Spark MLlib. Numerical transformation is essential because machine learning algorithms require numeric input for model training and prediction.

Data Normalization

The dataset contains features with different value ranges, which can negatively affect model learning. Normalization techniques are applied to scale the dataset values into a uniform range. This process improves model stability, enhances convergence speed, and reduces bias caused by uneven feature distributions.

Dataset Splitting

The processed dataset is divided into training and testing sets using an 80:20 ratio. The training dataset is used for model learning, while the testing dataset is reserved for evaluating prediction performance. This separation ensures that the trained model is validated using unseen data for accurate performance measurement.

Logistic Regression

Initially, Logistic Regression is used as the baseline machine learning model for earthquake magnitude prediction. The algorithm learns relationships between earthquake parameters and target output values. Performance metrics such as RMSE and MAE are calculated to evaluate the prediction capability of the existing model.

Random Forest

To improve prediction accuracy, the first extension integrates the Random Forest algorithm. Random Forest combines multiple decision trees and generates predictions based on ensemble learning. This method reduces overfitting, improves prediction consistency, and produces lower error rates compared to Logistic Regression.

Gradient Boosting Tree

The proposed extension model utilizes the Gradient Boosting Tree algorithm to further improve earthquake prediction performance. The algorithm iteratively corrects prediction errors generated in previous stages and builds a strong predictive model. Gradient Boosting Tree improves learning efficiency and achieves lower RMSE and MAE values than both Logistic Regression and Random Forest models.

Performance Evaluation

The trained models are evaluated using Root Mean Square Error and Mean Absolute Error metrics. Lower RMSE and MAE values indicate better prediction performance and reduced deviation between actual and predicted earthquake magnitude values. Comparative analysis is performed among Logistic Regression, Random Forest, and Gradient Boosting Tree models to identify the most efficient approach.

VI RESULTS & DISCUSSION

	Algorithm Name	RMSE	MAE
0	Logistic Regression	0.106836	0.079522
1	Extension1 Random Forest	0.087549	0.047899
2	Extension2 Gradient Boosting	0.036851	0.006173

The experimental results demonstrate the effectiveness of the proposed extension models for earthquake magnitude prediction using the AutoBDA framework. Three machine learning algorithms were evaluated, namely Logistic Regression, Extension1 Random Forest, and Extension2 Gradient Boosting. Performance evaluation was carried out using Root Mean Square Error (RMSE) and Mean Absolute Error (MAE), where lower values indicate better prediction accuracy and reduced deviation between actual and predicted earthquake magnitudes.

From the obtained results, the existing Logistic Regression model produced an RMSE value of 0.106836 and an MAE value of 0.079522. These values indicate that the traditional model generated higher prediction error when processing large-scale earthquake datasets. To improve analytical performance, the first extension model using Random Forest was implemented. The Random Forest algorithm reduced the RMSE value to 0.087549 and the MAE value to 0.047899. This improvement shows that ensemble learning methods can provide better prediction stability and reduced error compared to the baseline model.

The second extension model based on Gradient Boosting achieved the best overall performance among all evaluated algorithms. The model produced a significantly lower RMSE value of 0.036851 and an MAE value of 0.006173. These results confirm that Gradient Boosting effectively minimized prediction errors and improved learning accuracy through iterative optimization. The graphical comparison generated from the code execution further verifies that the proposed Gradient Boosting extension outperformed Logistic Regression and Random Forest models in both RMSE and MAE evaluation metrics, making it the most suitable model for accurate earthquake prediction within the AutoBDA framework.

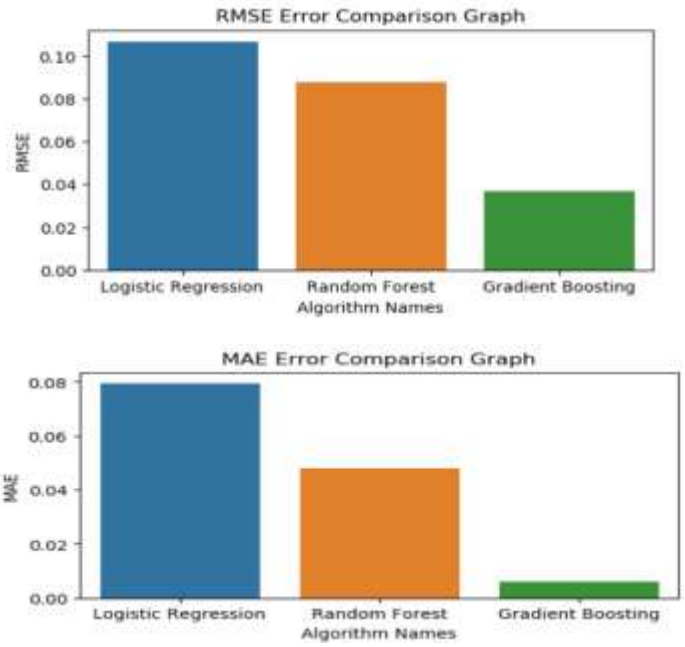

Figure 2: All Algorithms Performance Graph

The experimental analysis shows that integrating advanced machine learning algorithms with the AutoBDA framework significantly improves earthquake prediction performance. The existing Logistic Regression model was able to process earthquake data efficiently within the distributed Spark environment, but its prediction accuracy was limited due to higher RMSE and MAE values. This indicates that traditional linear models are less effective when handling complex seismic patterns and large-scale heterogeneous datasets.

The Random Forest extension improved prediction reliability by utilizing ensemble learning techniques. By combining multiple decision trees, the model reduced prediction variance and achieved better stability than Logistic Regression. The reduction in RMSE and MAE values confirms that Random Forest can capture nonlinear relationships within earthquake data more effectively. However, the model still showed moderate prediction error when compared with the Gradient Boosting approach.

Among all evaluated models, the Gradient Boosting Tree algorithm achieved the best performance with the lowest RMSE and MAE values. The iterative learning mechanism of Gradient Boosting enabled the framework to minimize residual errors and improve overall predictive accuracy. The results demonstrate that boosting techniques are highly effective for large-scale earthquake analytics. In addition, Apache Spark and Hadoop successfully reduced computational complexity through distributed processing, allowing faster model training and efficient handling of massive datasets within the AutoBDA framework.

VII. CONCLUSION

The developed AutoBDA framework successfully improved earthquake magnitude prediction by integrating distributed big data technologies with advanced machine learning algorithms.

Hadoop and Apache Spark enabled efficient processing of large-scale earthquake datasets through distributed storage and parallel computation. The framework performed effective data preprocessing, feature transformation, normalization, and automated analytical operations to support accurate prediction modelling. Experimental evaluation confirmed that the extension models outperformed the existing Logistic Regression approach. Among all tested algorithms, the Gradient Boosting Tree model achieved the best performance with minimum RMSE and MAE values, demonstrating higher prediction accuracy and reduced error. The Random Forest model also produced better results than the baseline model, proving the effectiveness of ensemble learning techniques for seismic data analysis.

REFERENCES

[1] G. Zhao et al., "Location recommendation for enterprises by multi-source urban big data analysis," IEEE Trans. Serv. Comput., vol. 13, no. 6, pp. 1115–1127, Nov./Dec. 2020.

[2] M. T. Islam, S. Karunasekera, and R. Buyya, "Performance and costefficient spark job scheduling based on deep reinforcement learning in cloud computing environments," IEEE Trans. Parallel Distrib. Syst., vol. 33, no. 7, pp. 1695–1710, Jul. 2022.

[3] A. S. S. Thenuwara Hannadige, "Architecture for intelligent big data analysis based on automatic service composition," Ph.D. dissertation, Division Inf. Syst., Univ. Aizu, 2019.

[4] S. Y. Chang and H.-C. Wu, "Divide-and-iterate approach to big data systems," IEEE Trans. Serv. Comput., vol. 15, no. 4, pp. 1967–1979, Jul./Aug. 2022.

[5] C. Ke, F. Xiao, Z. Huang, Y. Meng, and Y. Cao, "Ontology-based privacy data chain disclosure discovery method for big data," IEEE Trans. Serv. Comput., vol. 15, no. 1, pp. 59–68, Jan./Feb. 2022.

[6] C. A. Ardagna, V. Bellandi, M. Bezzi, P. Ceravolo, E. Damiani, and C. Hebert, "Model-based big data analytics-as-a-service: Take big data to the next level," IEEE Trans. Serv. Comput., vol. 14, no. 2, pp. 516–529, Mar./Apr. 2021.

[7] E. Tunstel et al., "Systems science and engineering research in the context of systems, man, and cybernetics: Recollection, trends, and future directions," IEEE Trans. Syst., Man, Cybern. Syst., vol. 51, no. 1, pp. 5–21, Jan. 2021.

[8] G. Aceto, V. Persico, and A. Pescapé, "A survey on information and communication technologies for industry 4.0: State-of-the-art, taxonomies, perspectives, and challenges," IEEE Commun. Surv. Tut., vol. 21, no. 4, pp. 3467–3501, Fourth Quarter 2019.

[9] A. Siriweera, I. Paik, and H. Huang, "Constraint-driven complexity-aware data science workflow for AutoBDA," IEEE Trans. Big Data, vol. 9, no. 6, pp. 1438–1457, Dec. 2023.

[10] Society 5.0 — Cabinet Office, Government of Japan. Accessed: Nov. 11, 2022. [Online]. Available: https://www8.cao.go.jp/cstp/english/society5%5F0/index.html

[11] A. Siriweera and K. Naruse, "Survey on cloud robotics architecture and model-driven reference architecture for decentralized multicloud heterogeneous-robotics platform," IEEE Access, vol. 9, pp. 40 521–40 539, 2021.

[12] S. Chen, L. Jiao, F. Liu, and L. Wang, "EdgeDR: An online mechanism design for demand response in edge clouds," IEEE Trans. Parallel Distrib. Syst., vol. 33, no. 2, pp. 343–358, Feb. 2022.

[13] W. Wu, L. He, W. Lin, and R. Mao, "Accelerating federated learning over reliability-agnostic clients in mobile edge computing systems," IEEE Trans. Parallel Distrib. Syst., vol. 32, no. 7, pp. 1539–1551, Jul. 2021.

[14] W. Dou, X. Zhang, J. Liu, and J. Chen, "HireSome-II: Towards privacyaware cross-cloud service composition for big data applications," IEEE Trans. Parallel Distrib. Syst., vol. 26, no. 2, pp. 455–466, Feb. 2015.

[15] T. Yu et al., "Large-scale automatic k-means clustering for heterogeneous many-core supercomputer," IEEE Trans. Parallel Distrib. Syst., vol. 31, no. 5, pp. 997–1008, May 2020.

[16] A. J. Garcia, M. Toril, P. Oliver, S. Luna-Ramirez, and R. Garcia, "Big data analytics for automated QoE management in mobile networks," IEEE Commun. Mag., vol. 57, no. 8, pp. 91–97, Aug. 2019.

[17] H. Li, K. Lu, and S. Meng, "Bigprovision: A provisioning framework for big data analytics," IEEE Netw., vol. 29, no. 5, pp. 50–56, Sep./Oct. 2015.

[18] E. R. Sparks, S. Venkataraman, T. Kaftan, M. J. Franklin, and B. Recht, "KeystoneML: Optimizing pipelines for large-scale advanced analytics," in Proc. IEEE 33rd Int. Conf. Data Eng., 2017, pp. 535–546.

[19] Y. Wang et al., "DataShot: Automatic generation of fact sheets from tabular data," IEEE Trans. Vis. Comput. Graph., vol. 26, no. 1, pp. 895–905, Jan. 2020.

[20] G. Ko et al., "Closha: Bioinformatics workflow system for the analysis of massive sequencing data," BMC Bioinf., vol. 19, no. 1, pp. 97–104, 2018.

KADALI ROSHINI is currently pursuing the MCA (Master of Computer Applications) in Ideal college of Arts and science, Vidyuth Nagar, Kakinada. Her research interests include Big Data.

M. Kameswara Rao is currently serving as the Additional Head of the Department of Computer Science at Ideal College of Arts & Sciences(A). He possesses more than 20 years of academic and administrative experience in the field of Computer Science.

Dr. V. S. V. Deepak is currently serving as the Head of the Department of Computer Science at Ideal College of Arts & Sciences (A). He possesses more than 18 years of academic and administrative experience in the field of Computer Science and Engineering. His areas of interest include Medical Image Processing, Cyber Security, Artificial Intelligence, Software Testing and Networking. He completed his Ph.D. research in Medical Image Processing from Swami Vivekananda University.

He has actively contributed to curriculum development, academic planning, and student mentoring. He has served as Chairman of the Board of Studies (BOS) for BCA, B.Sc. (Computer Science), B.Sc. (Artificial Intelligence), and MCA programs.