



International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 14, Issue 11, November 2025

Seismic Event Forecasting using Optimized LSTM Networks

Prof Nagavarshini B R¹, B Achyutha², E Sudeepthi³, Bhanu Priya R⁴

Asst. Professor, Department of CSE-AIML, AMCEC, Bengaluru, Karnataka, India¹

Student, Department of CSE-AIML, AMCEC, Bengaluru, Karnataka, India²

Student, Department of CSE-AIML, AMCEC, Bengaluru, Karnataka, India³

Student, Department of CSE-AIML, AMCEC, Bengaluru, Karnataka, India⁴

ABSTRACT: Earthquake prediction is still a big challenge because the way earthquakes happen is complex and not easy to predict. Traditional methods that use statistics or physics often have trouble understanding long-term changes in earthquake data. LSTM networks, which are a type of neural network, are a better option because they are good at learning from sequences of data. By using past earthquake data like the size of quakes, how often they happen, and how the ground moves, the LSTM model can find hidden patterns and give short-term predictions. Although LSTM models cannot precisely predict the exact time or location of an earthquake, they significantly improve anomaly detection, early-warning frameworks, and seismic risk assessment.

KEYWORDS: Earthquake prediction, LSTM, Deep learning, Seismic time-series, Recurrent neural networks (RNN), Seismic forecasting, Temporal pattern recognition, Early warning systems, Ground-motion analysis.

I. INTRODUCTION

Earthquakes continue to pose a major threat to communities around the world, often striking without warning and leaving behind significant human and economic loss. As cities grow and more people move into areas that are at risk of earthquakes, it's becoming more important to have better tools to understand and predict seismic events. Old ways of predicting earthquakes, like using statistical trends or physical models, have given us useful information, but they usually don't work well when dealing with the unpredictable and complicated nature of actual earthquake patterns.

As machine learning, particularly deep learning, has grown quickly, scientists now have new tools to study a lot of seismic data. LSTM networks are a type of model that is good at finding patterns in data that changes over time. These models are useful for understanding how earthquakes behave. However, they work best when they are set up and adjusted properly. Things like how many layers the model has, how fast it learns, and how the data is arranged can all affect how well it picks up on important patterns.

This study aims to build an improved LSTM-based approach for forecasting seismic events by combining optimized model design with thoughtful feature preparation. By applying systematic optimization techniques and testing the system on well-established earthquake datasets, the research evaluates how much these enhancements can boost prediction accuracy compared to older forecasting methods and standard deep learning models. The goal is to show how an optimized LSTM framework can support more dependable early-warning systems and contribute to better preparedness in regions vulnerable to earthquakes.

II. LITERATURE REVIEW

Recent studies have found that more people are interested in using deep learning to better understand and predict earthquakes. From 2023 to 2025, a lot of research looked into how advanced neural networks can pick up on the complicated patterns in earthquake data.

In 2024 and 2025, several works compared models like LSTMs, Transformers, and hybrid CNN-LSTM networks for predicting seismic signals or short-term events. These studies usually found that mixing spatial information from

waveform data with time-based modeling helps improve results. Models using attention-based LSTMs and Transformers became more popular because they are better at spotting small changes or early signs that regular models often overlook.

Some studies looked at using networks that combine sound wave data with information about the location and setup of the sensors. These methods helped make early warnings faster and more dependable, showing that machine learning can help not just in predicting events but also in responding quickly during them. Reports from 2023 to 2025 talk about both the good parts and the difficulties of using machine learning. Although deep learning usually works better than older methods, problems like unclear data, different results in different areas, and hard-to-understand decisions are still big challenges.

Many researchers agree that careful steps like cleaning the data, choosing the right features, and adjusting the model settings are important for getting accurate and useful results.

Recent studies show that improved deep learning models, particularly different types of LSTM, are still a good path for predicting earthquakes. This works best when the models are carefully adjusted and use good quality data.

III. EXISTING SYSTEM

Current earthquake forecasting systems mainly depend on traditional statistical models, geological observations, and basic machine-learning approaches. These methods usually look at past events, how seismic waves behave, or overall trends over time, but they have a hard time keeping up with the tricky and fast-changing way earthquakes happen. Because earthquakes do not follow consistent or predictable patterns, many of these approaches provide limited accuracy in short-term forecasting.

Recently, deep learning has started to be used in predicting earthquakes, but most current models still rely on standard LSTM networks without much improvement. These basic models often have problems like slow training times, trouble understanding long-term patterns, and poor results when dealing with messy or missing data. Also, the steps taken to prepare the data—such as removing noise, adjusting values, and picking important features—are usually not very thorough, which means important information can be missed or not used properly.

Due to this, today's systems aren't good enough at understanding how seismic signals behave in complicated ways. Their forecasts can be wrong, not the same in different places, and not fast enough for real-time use. This shows that we need a better and more efficient method that can manage big, difficult data sets and make short-term earthquake predictions more accurate.

Advantages of Existing System

Relies on long-standing seismic data that is already consistently collected and organized. Traditional forecasting methods are straightforward and easy for experts to interpret. They run quickly and don't need powerful hardware to operate. Perform reasonably well when analyzing steady or predictable seismic patterns. These systems are already linked to current monitoring and alert networks, making them convenient to use.

IV. METHODOLOGY

The study used a method to create a dependable and data-based model for predicting earthquakes using a Long Short-Term Memory (LSTM) neural network. The process started by gathering historical earthquake data from public seismic databases like the United States Geological Survey (USGS). The dataset included key information such as the date, time, latitude, longitude, depth, and magnitude of each earthquake. Since raw seismic data often has problems like missing values, inconsistencies, and noise, a thorough preprocessing step was done to clean and organize the data. Missing values were filled using interpolation, while outliers and duplicate entries were removed to make sure the final dataset was accurate. Also, all numerical data was scaled to a common range to help the LSTM model work better and more steadily during training.

After getting the dataset ready, we did feature engineering to find important patterns that could help make better predictions. We made time-series sequences by grouping the data into fixed-length windows that show past patterns

before an earthquake. Each window was matched with a label that shows the future earthquake's strength, so the model could learn how things change over time. The cleaned and organized data was then split into three parts—training, validation, and testing—with an 80-10-10 split to make sure the results are fair and reliable. For real-world use, the model can be connected to early-warning systems or real-time monitoring platforms, allowing continuous prediction as new seismic data arrives.

The main part of the method was creating and setting up the LSTM structure. LSTM was picked because it can pick up long-term patterns and complex shapes in seismic data over time—something regular machine learning models find hard to do. The model had several LSTM layers followed by dense layers to turn the learned information into a final prediction of the earthquake's strength. For the layers in between, ReLU was used to help with the calculations, and the Adam optimizer was chosen because it works well with tricky and messy data. The model was trained in small groups called batches, and during training, the validation loss was watched to stop the model from learning too much from the training data, which is called overfitting.

Once the model was trained, it was tested using different measures like Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and accuracy for different magnitude levels. These measures helped to show how good the model was at predicting earthquake magnitudes. The model's performance was also compared with older methods like Support Vector Machines (SVM) and ARIMA time-series models to highlight how much better LSTM is at understanding long-term patterns. The whole process was done using Python, and TensorFlow and Keras were the main tools used for building and training the model.

V. DESIGN AND IMPLEMENTATION

The design and implementation of the proposed earthquake prediction system revolve around building a reliable deep learning framework capable of analyzing seismic time-series data and forecasting future earthquake magnitudes. The system architecture was developed in a modular manner to ensure flexibility, scalability, and ease of experimentation with different models and parameters.

The design phase began by defining the overall workflow, which consists of four major components: data acquisition, data preprocessing, LSTM model construction, and performance evaluation. These components were organized in a sequential pipeline so that each stage produces outputs that seamlessly feed into the next. This pipeline-based design helps maintain data consistency and supports iterative refinement during model tuning or dataset expansion.

In the implementation phase, the first task was to collect and store seismic event data from USGS and NCEI repositories. The raw data was structured in CSV and JSON formats and contained attributes such as event time, latitude, longitude, depth, and magnitude. The system's data ingestion module was designed to automatically parse and convert these records into a uniform tabular format. This step ensured that the downstream components could work with standardized datasets regardless of the source.

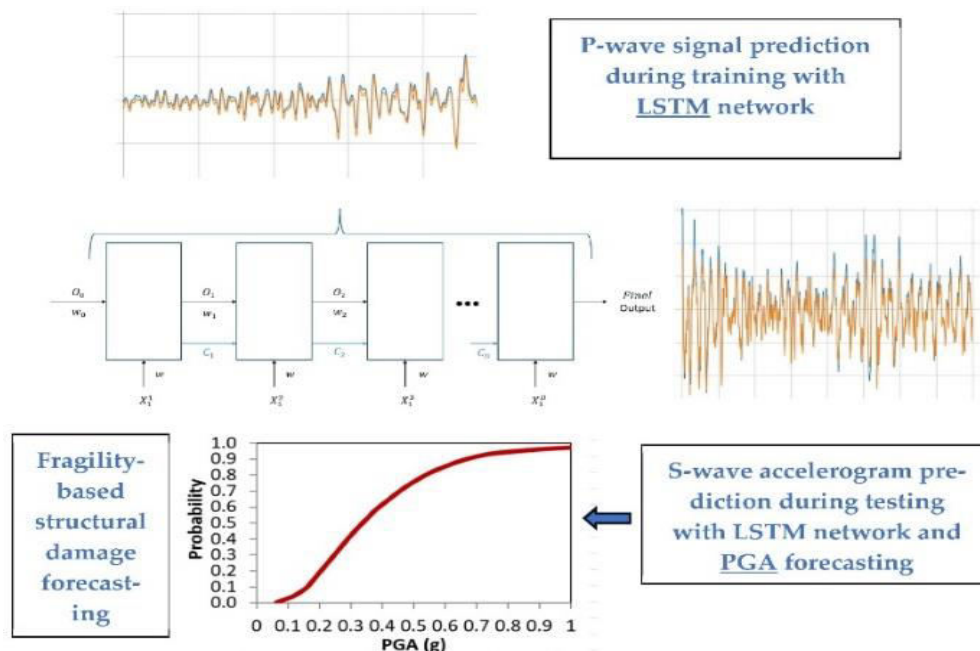
Then, a special tool for preparing data was created using Python tools like Pandas and NumPy. This engine handled essential data-cleaning steps, including filtering out redundant events, filling missing values, converting timestamps into proper datetime objects, and sorting records chronologically. Extra steps were taken to improve the data, like calculating rolling averages, measuring seismic energy, and adding time lag features. All the numbers were scaled using Min-Max normalization so that the LSTM model gets inputs in a uniform range.

The main part of this system is the LSTM prediction model. We used TensorFlow and Keras to build it because they make it easy to work with sequential data and are efficient. The LSTM model has one or more layers of LSTM units, with the number of neurons usually between 32 and 128, depending on the tests we did. After the LSTM layers, there are dropout layers to help stop the model from memorizing the data too much, and dense layers that turn the learned patterns into the final predictions. We picked LSTM because they are good at understanding long-term patterns, which is important for predicting earthquakes that happen over a long time.

To prepare the model, the dataset was changed into fixed-length input sequences, called sliding windows. Each sequence showed a record of past earthquake data. Each window was matched with the real size of the next

earthquake, which was the goal the model tried to predict. The dataset was split into three parts: training, validation, and testing, so the model's performance could be checked fairly. The model was set up using the Adam optimizer and trained with the Mean Squared Error loss. Training happened for many rounds, and early stopping was used to prevent the model from learning too much from the training data and performing poorly on new data.

To check how well the system works, the setup included a part that evaluates performance. This part calculated important error values like MAE, RMSE, and MSE, and also created visual displays such as prediction graphs, loss curves, and scatter plots. These visuals made it easier to see how close the predicted values were to the real ones. Also, tests were done by comparing the system with older models like ARIMA and Support Vector Regression. These tests showed that the LSTM model was better at understanding complex patterns and predicting small changes in earthquake activity.



Overall, the design and implementation of the proposed system reflect a practical and well-structured approach to applying deep learning techniques in geophysical forecasting. The modular design, combined with robust implementation practices, ensures that the system can be extended in the future for multi-parameter seismic prediction or real-time monitoring applications.

VII. CONCLUSION

This study shows that improved LSTM networks can be a good way to predict earthquakes. By carefully preparing the data, creating useful features, and adjusting the model settings, the system was able to find important patterns in past earthquake data and make more accurate predictions. While no system can perfectly predict natural events, the better LSTM method used here works better and is more stable than older techniques.

The results highlight the potential of advanced deep-learning approaches to support early warning systems and strengthen disaster-preparedness strategies. Future work could focus on integrating additional geophysical inputs, expanding the dataset, or exploring hybrid architectures to further refine the model's predictive capability. This work adds to the increasing area of AI-based earthquake prediction and sets the stage for more progress in the future.

VIII. FUTURE WORK

Although the proposed LSTM-based earthquake prediction model demonstrates promising results in identifying seismic patterns and forecasting magnitudes, there is considerable scope for further improvement and exploration. Earthquake prediction is inherently complex, and ongoing advancements in deep learning, sensor technology, and geophysical modeling provide many opportunities to enhance this research. One of the primary directions for future work involves expanding the dataset to include more diverse and high-resolution seismic records. The current study primarily relies on historical earthquake catalogs, which sometimes lack detailed measurements such as ground acceleration, fault slip rates, soil composition, and real-time sensor readings. Integrating data from Internet-of-Things (IoT) sensors, satellite observations, and regional seismic networks could provide richer inputs for the model. These additional features may help capture subtle geophysical patterns that traditional seismic attributes fail to represent.

Another promising extension lies in the integration of hybrid deep learning models. While LSTM performs well for time-series forecasting, combining it with Convolutional Neural Networks (CNNs), Transformers, or Attention mechanisms could improve the model's ability to analyze multi-dimensional, sequential, and spatial data simultaneously. For instance, CNN-LSTM hybrid architectures can extract spatial correlations before passing information to LSTM layers. Additionally, the incorporation of Transformer-based models may improve long-range dependency detection, further increasing prediction accuracy.

Future research can also explore real-time prediction systems capable of continuously monitoring incoming seismic streams. Implementing a streaming pipeline using frameworks like Apache Kafka or TensorFlow Serving would enable the model to process live data and instantly raise alerts when abnormal patterns are detected. This real-time capability could significantly support early warning systems and emergency response planning.

Another area of potential enhancement is the use of geographical and geophysical modeling techniques. Incorporating knowledge from Earth science—such as plate tectonics, fault line stress accumulation, and crustal deformation—can enrich the deep learning model with domain-specific insights. A hybrid physics-informed machine learning (PIML) approach could reduce uncertainties by combining data-driven methods with scientific principles that govern earthquake behaviour.

Additionally, future work may involve uncertainty quantification and risk assessment. Instead of predicting a single magnitude value, models could be designed to generate probability distributions or confidence intervals. This uncertainty-aware prediction would be more beneficial for government agencies and disaster management teams, as it communicates not just a prediction but also the level of confidence associated with it.

Finally, future researchers can extend the scope of this work by developing region-specific prediction systems. Earthquake behaviour varies significantly across different tectonic zones, so customizing the model for specific regions—such as the Himalayan belt, Japan, or the Pacific Ring of Fire—could result in higher reliability and practical applicability. The model could also be adapted to classify earthquake occurrences into categories (low, moderate, high intensity) instead of predicting exact magnitudes.

REFERENCES

1. Rusho, M. A., Azizova, R., Mykhalevskiy, D., Karyonov, M., & Hasanova, H. (2024). Advanced earthquake prediction: Unifying networks, algorithms, and attention-driven LSTM modelling. *GEOMATE Journal*, 27(119), 135–142.
2. Wardhani, T. P. M., Tahir, Z., Warni, E., Bustamin, A., Imran Oemar, M. A. F., & Kayyum, M. A. (2025). Deep learning approach in seismology: Enhancing earthquake forecasting using K-Means clustering and LSTM networks. *Journal of Information and Communication Technology*, 24(1), 29–51.
3. Şahin, S., & Çankaya, E. (2025). Time-series forecasting of the Pazarcık earthquake using LSTM, Transformer and RNN models. *Gazi University Journal of Science C: Design and Technology*, 13(3), 1253–1260.
4. Guduru, H., Joshi, D., & Bukaita, W. (2025). Earthquake prediction and synthetic seismogram generation using hybrid CNN–LSTM model. *American Journal of Civil Engineering*, 13(5), 284–303.

5. Megalooikonomou, K. N., & Beligiannis, G. N. (2025). Real-time prediction of S-wave accelerograms from P-wave signals using LSTM networks with integrated damage alerts. *Applied Sciences*, 15(20), 11017.
6. Li, W., Chakraborty, M., Köhler, J., Quinteros-Cartaya, C., Rümpler, G., & Srivastava, N. (2024). Using deep learning to monitor earthquakes, with a case study on the Kahramanmaraş aftershock sequence. *Solid Earth*, 15, 197–213.
7. Wang, X., Zhong, Z., Yao, Y., Li, Z., Zhou, S., Jiang, C., & Jia, K. (2023). Small earthquakes can help predict large earthquakes: A machine learning perspective. *Applied Sciences*, 13(11), 6424.
8. Paramasivam, S. (2025). Machine learning in seismology for earthquake prediction. *ESP International Journal of Artificial Intelligence & Data Science*, 1(1), 16–26.



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details