International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 06 | June - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

REAL-TIME EARTHQUAKE PREDICTION USING CONTINUOUS CONVOLUTIONAL NETWORKS

Dr.T.Prabakaran[1], Anjali K[2], Ruchitha K[3], Vipunsai K[4], Bhavani M[5]

[1]Professor, Computer Science & Engineering, Joginpally B.R.Engineering College,
[2]Computer Science & Engineering, Joginpally B.R.Engineering College,
[3]Computer Science & Engineering, Joginpally B.R.Engineering College,
[4]Computer Science & Engineering, Joginpally B.R.Engineering College,
[5]Computer Science & Engineering, Joginpally B.R.Engineering College

ABSTRACT

Traditional earthquake prediction methods were based on physical and geographical properties before sensor networks were invented. The oldest method was the observation of animal's behaviour to predict any unusual changes. Then it improved to measuring low-frequency electromagnetic waves that may be emitted due to stress accumulation in rocks. Seismographs and instrument-based monitoring were introduced in the early 1900s to measure the intensity of the earthquakes which quickly became the global standard and also helped in study of wave propagation, but they were just used for aftermath and could not help in predicting before the event occurred. As there was a constant increase in the population in seismic active areas the need for accuracy and timeliness increased. With the advent of sensor networks, real-time seismic data became available from many geographically distributed sources. This created an opportunity to move beyond vague forecasting and toward predictive modelling. In our project, we aim to leverage this shift by integrating modern sensor data with machine learning models capable of recognizing early seismic patterns and anomalies. Our system preprocesses raw seismic waveform data by merging multiple HDF5 chunks and extracting 35 critical features, including magnitude, depth, location, signal-to-noise ratio, and wave arrival times.

Keywords: Artificial Intelligence (AI), Deep Learning (DL), Convolutional Neural Networks(CNN), Machine Learning (ML), Primary Wave(P-wave), Secondary Wave (S-wave).

1. INTRODUCTION

Earthquakes are one of the most dangerous natural disasters, which often causes damage economically and takes many human and animal lives [1]. As urban areas continue to expand their size to fit the increasing human population, the ability to detect and respond to earthquakes in real time becomes increasingly vital for minimizing loss of life, infrastructure damage, and economic disruption [2]. While progress has been made in earthquake engineering, accurate and timely prediction remains a major scientific and technological challenge for people all over the world [3].

Various systems provide essential information for post-disaster monitoring and assessment, helping governments and response agencies evaluate damage and allocate resources [4].

With increase in popularity of artificial intelligence, especially in the field of machine learning (ML) and deep learning (DL), new techniques have been used for seismic event detection and even predicting earthquakes from raw waveform data [5][6]. These approaches make use of the large datasets and algorithms to identify subtle patterns in seismic and microseismical data which help in identifying earthquakes which traditional methods miss[7][8].

Traditional seismic monitoring systems rely heavily on seismograph data and electromagnetic waves [9]. Even though they help in understanding seismic behaviour and wave propagation[10], these methods lack in identifying the precision needed for effective early warning due to their late response time and incorrect prediction.

One of the most popular tools for post-earthquake analysis and damage is the Earthquake Damage Analysis and Processing System based on Remote Sensing (RS) and Geographic Information Systems (GIS), known as RSEDAPS [11]. However, RSEDAPS is reactive rather than predictive, designed for analysis after an event has occurred.

This project proposes a real-time earthquake prediction system built using Continuous Convolutional Neural Networks (CCNNs), for capturing temporal dependencies in time-series data. The CNN model, inspired by the VGG architecture, processes 3-channel input signals (P-wave, S-wave, and noise) with a sequence length of 6000 samples to capture temporal patterns and signal characteristics.

The aim is to not only reduce false positives and missed detections but also to enable early alerts before the most damaging waves arrive. Additionally, the system is integrated with an interactive dashboard, enhancing accessibility for both seismic professionals and non-experts. This user-friendly interface empowers users to visualize real-time predictions, monitor ongoing activity, and make informed decisions during critical time windows.

The project begins with comprehensive data preprocessing, where raw seismic waveform data from multiple HDF5 chunks are merged and converted into a standardized format. This process includes the extraction of 35 critical features such as 1 magnitude, depth, latitude, longitude, arrival times of seismic phases, signal-to-noise ratio, and event uncertainty metrics. These features form the basis for training the predictive model. The core of the system is a CCN model inspired by temporal convolutional architectures, designed to capture long-range dependencies and various variations in waveform signals across three channels: P-wave, S-wave, and noise. The model takes input sequences of 6000 samples and produces a classification output indicating the likelihood of an earthquake event. Beyond model development, the project also enables the creation of an interactive front-end dashboard using Streamlit.

2. MATERIALS AND METHODS

These materials determine the workflow of a system that is designed to process user input, process data and give predictions. It contains

- **User Input**: Users provide a location through a text box in the Streamlit UI (ex: “California”, “Delhi”).
- **Geolocation Module**: The system uses the Geopy package to convert the location into geographic coordinates (latitude and longitude).
- **Distance Calculator**: The Haversine formula calculates distances between the input location and various trace locations in the metadata to identify the closest seismic trace.
- **Trace Retrieval**: “merged.hdf5” file contains information about the seismic events. The closest trace is extracted based on the minimum distance to the user’s location.
- **CNN Classifier**: A trained 1D Convolutional Neural Network processes 3-channel seismic signals of 3000 samples each to predict the likelihood of an earthquake.
- **Metadata Source**: Additional event metadata like magnitude, time, and coordinates is retrieved from “merged.csv”.

- **Prediction Result**: A probability score between 0 and 1 is generated to label the event as either "likely earthquake" or "no earthquake".
- **Visualization Engine**: The dashboard includes:
  - Waveform plots for 3 seismic channels.
  - A correlation heatmap of channels.
  - A map view of the trace's epicentre.

2.1 REQUIREMENTS

- **System requirements:** Ram(8GB or higher), CPU(dual-core processor e.g., AMD Ryzen 9 5900HS, Intel Core i3-6100), GPU.
- **Operating System**: Windows/ Linux/macOS (any OS with Python 3.8+ support).
- **Python Libraries**: torch, h5py, pandas, matplotlib, seaborn, streamlit, geopy.

2.2 METHODOLOGY

The methodology used in this system combines geospatial computation, signal processing, and deep learning for real-time earthquake prediction. The model is based on a VGG inspired model and 1D Convolutional Neural Network (CNN) trained on pre-labeled seismic waveform data. The CNN works with three-channel seismic traces, each containing 3000 time samples. These signals are extracted from an HDF5- formatted dataset and the corresponding metadata(latitude, longitude, magnitude, and event time) is retrieved from a structured CSV file.

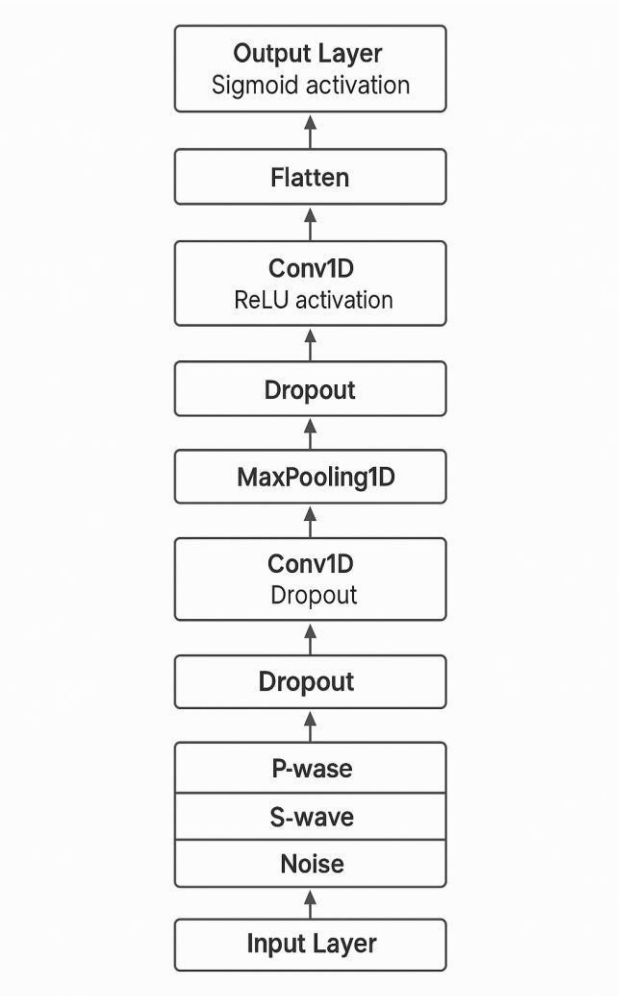

Fig- 2.2.1: Architecture of the CNN

The system architecture begins with user interaction via Streamlit interface, where users input a location. This location is geocoded into geographic coordinates using the Geopy library. To associate this location with the available seismic data, the Haversine distance formula is applied to compare the user's coordinates with all epicenters in the dataset. The trace with the minimum distance is then selected for analysis.

The corresponding waveform is retrieved from the HDF5 file and formatted into the required tensor structure before being passed through the CNN model. The network contains four convolutional-pooling blocks followed by three fully connected layers, using dropout for regularization and sigmoid activation at the output layer to generate a probability score.

Based on a threshold of 0.5, the output is calculated as either the likelihood of an earthquake or not an earthquake. Various visualisations such as waveform plots, correlation heatmaps and epicenter mapping are displayed on the dashboard to provide users with full details of the event.

3. PROCEDURE

Figure 3.1 shows a requirements.txt file, which we usually use in Python projects to list the libraries required for the project. This file defines the packages needed for installation using package manager called pip.

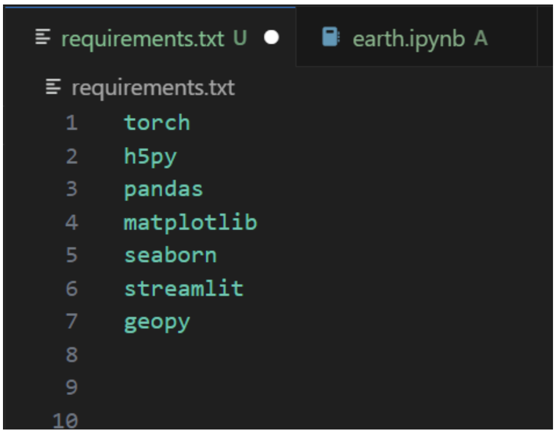

Fig 3.1: Listing Libraries required in a text file

3.1 Prerequisites

Before executing the application, ensure that Python 3.8 or above is installed on your system. Fig.3.1.1 illustrates the command prompt where the user installs all required packages using pip. The following command is executed:

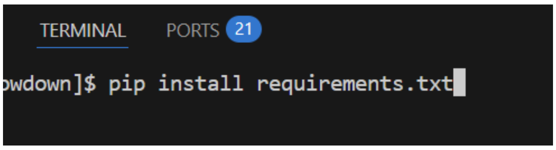

Fig-3.1.1: Installing the libraries

3.2 Setup and Installation

Fig .3.2 shows terminal in VS code where the user works with Python virtual environment. Here environment preparation is done, and the command follows:

Fig-3.2.1 Enabling a virtual environment on windows

In Fig.3.2.2, shows the Streamlit application to launch the earthquake prediction dashboard locally. The command below runs the app and opens a web interface:

Fig-3.2.2: Running the dashboard using Streamlit

4. OUTPUTS

Figure 4.1 shows the earthquake prediction dashboard running on port 8501. The dashboard also shows the geolocation, nearest trace and the probability of the occurrence of the earthquake.

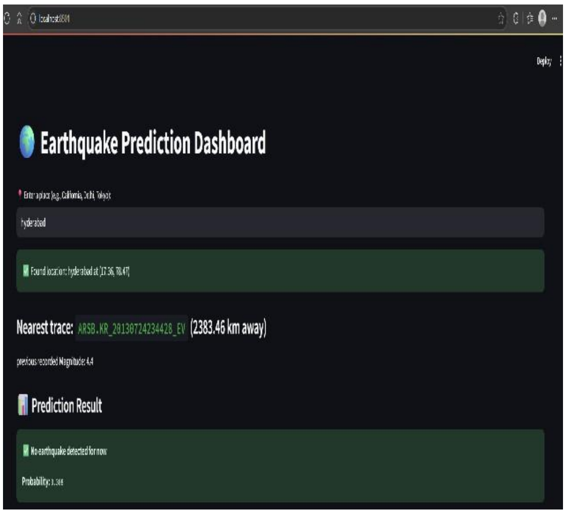

Fig 4.1: Dashboard prediction

The dashboard also displays the seismic waveform with 3000 samples x 3 channels along with a correlation map.

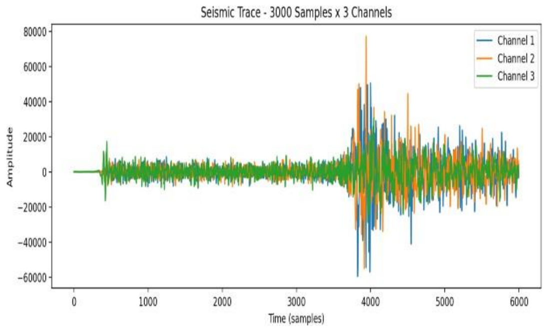

Fig 4.2: Seismic Waveform

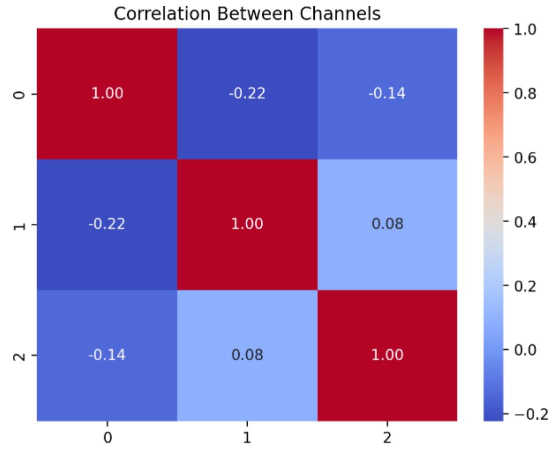

Fig 4.3: Correlation Map

The dashboard also provides an epicentre map to trace the given location to the nearest epicentre.

Fig 4.4: Epicentre Map

4. RESULTS

The system was tested through unit testing, integration testing, functional testing, and system testing to validate the model's accuracy, robustness and usability. The results presented below provide a summary of the performance of the system across multiple metrics including accuracy, precision, recall, F score, inference time and the size of the model.

The evaluation of different models for earthquake prediction is presented in Table 1. The Convolutional Neural Network (CNN) model demonstrated competitive performance with an accuracy of 95.37%, the highest precision at 96.69%, and strong recall of 94.95%, resulting in a balanced F1 score of 93.32%. These results confirm the CNN's superior capability to effectively capture complex seismic signal patterns, making it well-suited for this predictive task.

The Random Forest classifier achieved a slightly higher accuracy (95.50%) but exhibited lower precision (92.60%) and recall (92.40%) compared to the CNN. However, the Random Forest's inference time was significantly longer (30 ms), which may limit its applicability in real time scenarios.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	Inference Time (ms)	Model Size (MB)
CNN	95.37	**96.69**	**94.95**	**93.32**	12	**>30**
Random Forest	**95.50**	92.60	92.40	92.00	30	>70
Support Vector Machine	88.90	87.30	89.00	88.10	80	<50
Logistic Regression	85.60	84.10	86.20	85.10	**3**	>39
Threshold-Based Heuristic	67.20	72.00	54.80	62.10	1.5	<40

Table 1: Comparison with other models

5. CONCLUSION

In this project, we developed a real-time earthquake prediction system based on a VGG inspired continuous 1D convolutional neural networks (CNNs). The objective was to understand the seismic waveform data, detect early P-signals with minimal delay, meeting the goal of rapid earthquake alert generation.

The real-time earthquake prediction system developed in this project showcases a practical application of Convolutional Neural Networks (CNNs) to the domain of seismic data analysis. By leveraging a carefully designed 1D CNN architecture specially tailored for seismic waveform data, the system can trace the nearest epicentre based on the nearest trace recorded and can generate results based on the data and it also makes use of 3 channels namely S- waves, P-waves and noise for further classification. This approach allows the model to classify earthquake occurrences with a good probability, demonstrating the power of deep learning techniques in enhancing early warning capabilities.

6.REFERENCES

waves and S-waves and predict ground motion intensity. By implementing a dedicated 1D CNN model and data pipeline, the system continuously analyzes incoming seismic Arush Kaushal, Ashok Kumar Gupta, Vivek Kumar Sehgal, Earthquake prediction optimization using deep learning hybrid RNN- LSTM model for seismicity analysis, Soil Dynamics and Earthquake Engineering, Volume 195, 2025, 109432, ISSN 0267-7261, https://doi.org/10.1016/j.soildyn.2025.109432

[1] Bao W, An Q, Guo Y, Wang L, Gao J. Ambient noise levels of seismic stations located in urban agglomerations in central Inner Mongolia, China. PLoS One. 2024 Dec 5;19(12):e0315004.doi: 10.1371/journal.pone .0315004. PMID: 39636840; PMCID: PMC11620444.

[2] Bilal, M. A., Ji, Y., Wang, Y., Akhter, M. P., & Yaqub, M. (2022). Early Earthquake Detection Using Batch Normalization Graph Convolutional Neural Network (BNGCNN). Applied Sciences, 12(15), 7548. https://doi.org/10.3390/app12157548

C. Barnes, A. Siqueira and S. Labahn, "USGS CEOS Analysis Ready Data for Land Achievements and Future Plans," 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS,Brussels, Belgium, 2021, pp. 1785-1788, doi: 10.1109/IGARSS47720.2021.9554440.

[3] H. A. Nugroho, E. Y. Astuty, A. Subiantoro and B. Kusumoputro, "Ensemble Deep Learning NARX for Estimating Time Series of Earthquake Occurrence," 2023 3rd International Conference on Robotics, Automation and Artificial Intelligence (RAAI), Singapore, Singapore, 2023, pp. 301-305, doi: 10.1109/RAAI59955.2023.10601278

[4] Huang, Ym., Xie, Y., Miao, Fj. et al. 1D Convolutional Seismic Event Classification Method Based on Attention Mechanism and Light Inception Block. Appl. Geophys. (2024). https://doi.org/10.1007/s11770-024-1117-4

[5] Lei Liu, Weiqi Song, Chao Zeng, Xiaohui Yang,Microseismic event detection and classification based on convolutional neural network, Journal of Applied Geophysics, Volume192,2021,104380,ISSN0926-9851, https://doi.org/10.1016/j.jappgeo.2021.104380.

[6] M. Elbes, S. AlZu'bi and T. Kanan, "Deep Learning-Based Earthquake Prediction Technique Using Seismic Data," 2023 International Conference on Multimedia Computing, Networking and Applications (MCNA), Valencia, Spain, 2023, pp. 103-108, doi:10.1109/MCNA59361.2023.10185869. https://ieeexplore.ieee.org/document/10185869

[7] Moez Krichen, Mohamed S. Abdalzaher, Mohamed Elwekeil, Mostafa M. Fouda, Managing natural disasters: An analysis of technological advancements, opportunities, and challenges, Internet of Things and Cyber-Physical Systems, Volume 4, 2024, Pages 99-109, ISSN 2667-3452, https://doi.org/10.1016/j.iotcps.2023.09.002

[8] S. Azadi and A. A. Safavi, "S-transform based P-wave and S-wave arrival times measurements toward earthquake locating," The 2nd International Conference on Control, Instrumentation and Automation, Shiraz, Iran, 2011, pp. 241-246, doi: 10.1109/ICCIAutom.2011.6356663.

[9] Xiang Ding, Xiaoqing Wang, Aixia Dou and Long Wang, "Study on a practical earthquake damage analysis and processing system based on RS and GIS," 2009 IEEE International Geoscience and Remote Sensing Symposium, Cape Town, 2009, pp. IV-490-IV-493, doi: 10.1109/IGARSS.2009.5417420.