International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 04 | April - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

Earthquake Detection and Alerting system using ANN and IOT

K. Chethan[1], K.V.S.N.S. Swetha[2], M. Krishna Sri[3], M. Tejaswini[4]

Students of the Department of Electronics and Communication Engineering, R.V.R & J.C College of Engineering

---

Abstract - Earthquakes are one of the deadliest natural disasters. The damage caused by earthquakes is very destructive and the number of deaths are at a very large scale. The National Earthquake Information Centre now detects about 20,000 earthquakes around the globe each year, or approximately 55 per day. Many of the Earth's earthquake zones coincide with areas of high population density. When a large earthquake occurs in such areas, the results can be catastrophic.

The basic parameter to determine an earthquake before it was going to occur is the p wave or primary shock wave which is not destructive to infrastructure because it travels parallel to the propagation of seismic wave. A series of destructive waves that follows p waves are s waves or secondary waves which has a speed of about 60% that of p waves. The movement of the s wave is perpendicular to the direction of propagation of the seismic wave and hence it is extremely risky to infrastructures and cause a lot of damage.

This research aims to build an earthquake detection and alerting system using a low-cost accelerometer, ESP32 Wi-Fi module, and ANN algorithm. The input earthquake data is taken by the accelerometer and with the help of the ESP32 Wi-Fi module the data is sent to a system that processes the ANN algorithm with Multi layered perceptron trained with stochastic gradient descent learning algorithm and detects whether an earthquake is going to occur or not and alerts users with ESP32 Wi-Fi module.

Key Words: Artificial Neural Networks, ESP32, Accelerometer, Earthquake, Multi-layer perceptron, Sigmoid function, Stochastic Gradient Descent.

1. INTRODUCTION

The very nature of earthquakes is quick and destructive, this makes them very unpredictable mainly in the areas along tectonic boundaries where many numbers of earthquakes occur very frequently. So, we as humans have developed many types of Earthquake Early Warning (EEW) devices that gives minutes to seconds prediction before an earthquake is going to occur so that people in that area can evacuate to safety immediately.

Based on the available technologies we can divide IoT based earthquake detection devices into two parts. A mobile-based earthquake early warning system which uses low-cost MEMS sensors inside our smartphone device for checking of seismic activity in a dynamic environment, while the stationary sensor-based early warning system uses a dedicated device as a seismic sensor in static (i.e., fixed) environment. Each has its own advantages and disadvantages. For mobile based system it needs to constantly take data from the MEMS sensor and run an algorithm to predict an earthquake which takes up a lot of mobiles battery and also since mobiles are no stationary and always being moved, we are more prone to noise and false predictions which reduces our accuracy. Whereas with stationary systems they are prone to less noise when compared to mobile devices because they are stationary which improves our accuracy but it also requires extra hardware to be installed along the building.

In both cases we are using low-cost MEMS sensors which are very sensitive to noise this makes our model very less accurate if we use traditional algorithms to predict earthquake. So, to avoid this we use machine learning models to improve accuracy, in our case we are using ANN (Artificial Neural Networks) with Multi layered perceptron trained with stochastic gradient descent learning algorithm.

Our research is sort of a hybrid between both cases. We are using a standalone IoT device with ADXL345 accelerometer and a ESP32 WIFI module but all the main processing is done by another system which is capable of collecting the data from many devices and process it with our ANN to predict earthquake and based on its prediction ESP32 WIFI module sends notification to users through a telegram bot.

The advantage of this model is that it is very fast once it is setup and we can get data from multiple accelerometers across the building to get a more accurate prediction. Another advantage of this method is that this is very cheap and for processing unit we can simply use any old computer or mobile device which has enough specifications to constantly run ANN.

The contribution of this work are as follows

1.We have developed a low cost IoT device to accurately predict earthquakes and send notification to users almost immediately once setup.

2.We also implemented a model to collect data from a network of MEMS sensors and predict the earthquake occurrence.

This article is structured as follows section 2 explains about all the related works that inspired us and helped us to achieve our model, section 3 explains about all the details that went into our model, in section 4 we evaluate our results and discuss further options and section 5 contains our conclusion for this research.

2.RELATED WORK

For this project we have referenced many research papers without them we would not be able to do this research and these are the things we learned from some of those papers

In [1] "Earthquake Alert Device Using a Low-Cost Accelerometer and Its Services" which is the main base paper for our project they tested various machine learning models such as ANN with three and five features, CRNN. CRNN proved to be the most accurate but it requires a lot of computation power hence they declared that ANN model with 5 features is the most effective model.

In [2] "Earthquake Detection in a Static and Dynamic Environment Using Supervised Machine Learning and a Novel Feature Extraction Method" which is a predecessor to [1] they used normal ANN and k means clustering algorithm to balance non earthquake dataset. Its disadvantages are also that it requires a lot of computational power.

In [3] "Earthquake Monitoring and Warning System" they used a normal root mean square calculation with a basic threshold to determine earthquakes and alert nearby people. Although this does not have much accuracy to determine earthquakes this paper published a very advanced and sophisticated model to alert nearby people.

In [4] "Remote Real Time Monitoring and Safety System for Earthquake and Fire Detection Based on Internet of Things" this paper does not only focus on earthquakes but also in detecting fire and other accidents that may occur. Due to its focus on other aspects, it does not perform well as an earthquake early warning system.

3. PROPOSED METHODOLOGY

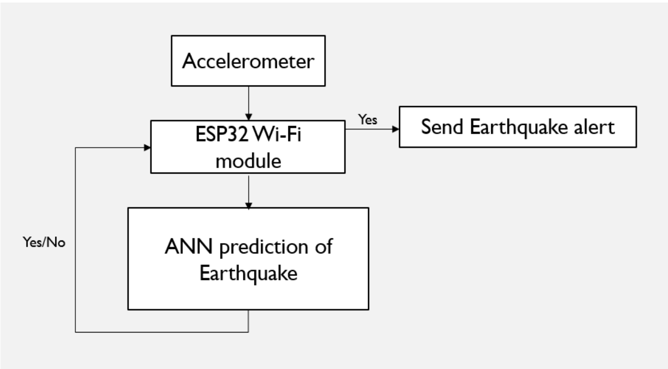

Fig1.Framework

This is the framework we designed to detect potential earthquakes using low-cost accelerometer sensor data and artificial neural networks (ANN). The system utilizes an accelerometer, specifically the ADXL345, to collect acceleration data in all 3 x, y, and z directions. This acceleration data is sent to an ESP32 Wi-Fi module, which acts both as a link between the accelerometer and the ANN and also to alert users in case of an earthquake predicted by our artificial neural networks. The collected accelerometer data is then forwarded to a computer, where an ANN uses a multi-layered perceptron trained with the stochastic gradient descent algorithm. This ANN predicts the likelihood of an earthquake and sends that prediction back to the ESP32 Wi-Fi module. Then, based on our ANNs predictions, our ESP32 Wi-Fi module alerts users through a Telegram bot, which it accesses through the internet.

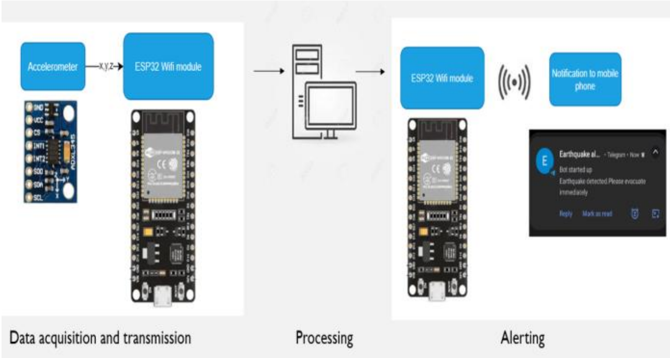

Fig2.Proposed Model

Our methodology consists of mainly three stages as shown in the above figure

1.Data acquisition and transmission

2.Processing

3.Alerting

3.1 Data acquisition and Transmission

In this stage accelerometer ADXL345 calculates the acceleration data in all three x, y, z directions and sends that data to ESP32 WIFI module. The connections for this are shown below

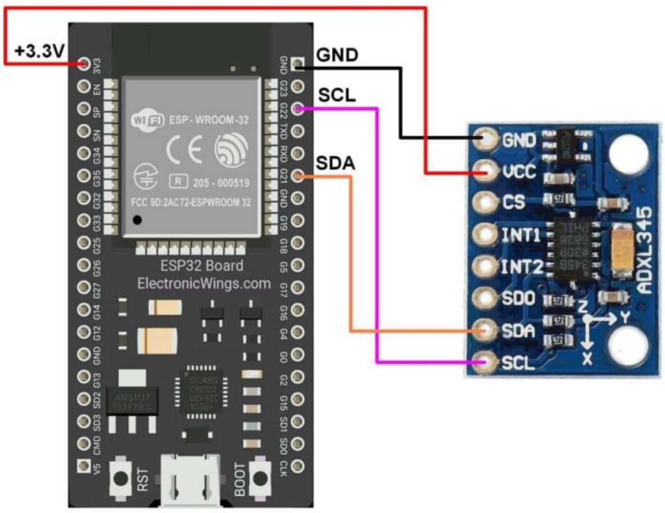

Fig3.Connections between ESP32 and ADXL345

Here SCL – Serial Communication Clock, GND – Ground and SDA – Serial Data. Vcc, GND pins are used to provide power to ADXL345 from ESP32, SCL pin is responsible for providing and synchronizing the clock between both devices and SDA pin is to send the acceleration data serially.

And to program our ESP32 WIFI module we have used Arduino platform with some libraries that help us with easy communication with ADXL345 such as Adafruit ADXL345 library. Then with the help of Arduino platform and this library we set the sampling rate of our accelerometer to 100Hz and acceleration range to +/- 2g since we do not require large changes in acceleration as we are only trying to detect p or primary wave.

After completing our program, we load our code into ESP32 and run it with accelerometer. Now the data from accelerometer is successfully being sent to ESP32 serially and that data needs to be transmitted to processing system, this transmission can be done through many ways such as on internet or through a cable, etc. We have chosen to transmit data through a USB cable to the system because it provides very low latency which is of utmost significance our scenario. Another advantage of transmitting data through USB is that we do not require any extra cables to provide power to either ESP32 or accelerometer and also we can easily make changes to our code in ESP32 which provides easy repairability and versatility.

3.2 Processing

In this stage we acquire the acceleration data sent by the ESP32 through USB and then we need to preprocess and extract features from this data.

But the acceleration data is being sent to the Arduino program in our system and we need that data in our python program which is responsible for preprocessing, feature extraction and also to predict our result through ANN. So, we need to establish a two-way communication between ESP32 and our python program. Again, there are many ways in which you can do this but we went with the one which has low latency that is to establish communication through port and serial. To do this we need to install serial library from python. After setting up we can read acceleration data from ESP32 WIFI module.

Then we need to choose our window size which essentially means after how many seconds we need to run our ANN and predict with our data. We have tested it with 2,4,6 and 8 seconds and selected the one with best results. After selecting our window size, we stored all the serial data from that time period into a excel sheet including time stamps. Then that file is brought back into our python program to calculate our required features.

The features are calculated through python program and that calculated features and sent to our trained ANN model more about the features and ANN model in section 4.

Then our prediction is sent back to ESP32 WIFI module through the same port communication. Then based on our prediction ESP32 decides whether or not to send alert to the users.

3.2. Alerting

Then based the on the decision from our ANN, ESP32 decides whether to send an alert to the users. To send the notification we are using telegram bot. There are many ways such in which we can alert our users such as through WhatsApp and SMS. But we went through telegram because it is easy to access and use. To send this message through telegram we are using universal telegram bot again for which we need to include required libraries.

But before accessing telegram bot we need to connect our ESP32 WIFI module to our internet to do this we need to include WIFI libraries to our program. After connecting our internet then we can program to send message based on our decision.

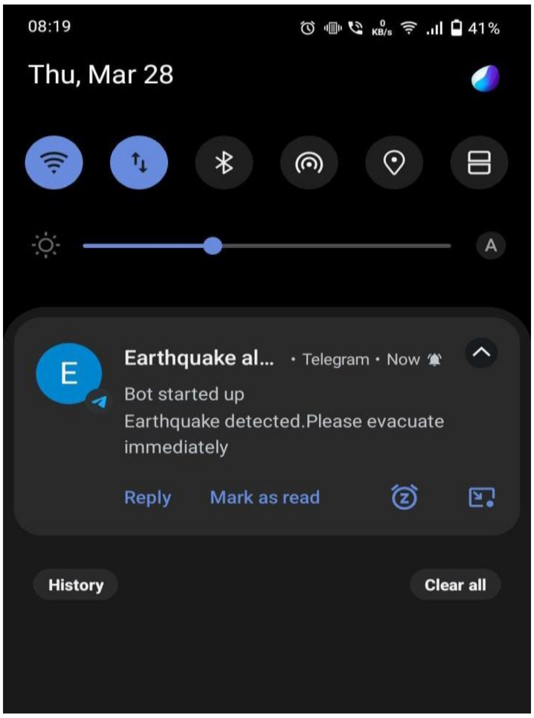

Fig4.Notification sent to mobile

4. DATASET AND ANN MODEL

In this section we discuss about all the datasets and ANN

4.1. Dataset

In this project we used two three types of datasets from NIED database [6],[7] An Open Dataset for Deep Learning-based Earthquake Detection using MEMS Sensors [8], random data from accelerometer.

In NIED dataset [6],[7] each earthquake contains 9 files acceleration, velocity and displacement for all three x, y, z directions. This contains the raw data for each direction so we need to process this data and extract features to automate this process we built a python framework. After calculating all the features, we store this data into a excel sheet and label this data as earthquake data. We are going with 100Hz sampling rate so we only need to select earthquakes with only 100Hz and we also upscaled some earthquakes from 50Hz.

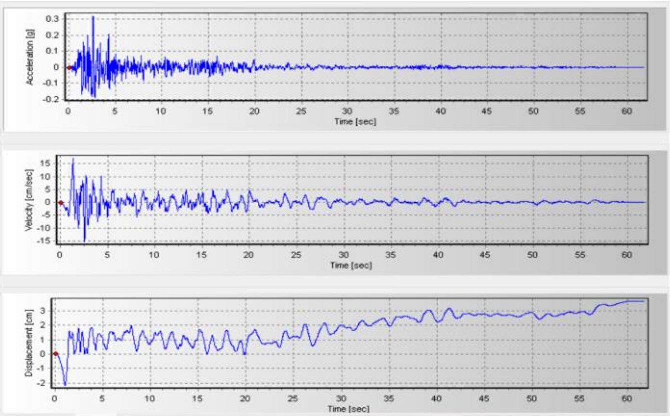

Fig5. Acceleration, velocity and displacement of a sample earthquake.

An Open Dataset for Deep Learning-based Earthquake Detection using MEMS Sensors [8] this dataset contains the earthquake data from MEMS sensors. This dataset is very important because we are also using MEMS sensors to implement our ANN. This dataset also needs to be labeled as an earthquake data.

Then for Non earthquake data we used some datasets from google as random noise and also, we created some of our own datasets for non-earthquake data.

After collecting all the data from these datasets, we processed and extracted required features for training our ANN model. Then we use this data to train our ANN.

4.2. Artificial Neural Network model

For our project we used ANN algorithm with Multi layered perceptron trained with stochastic gradient descent learning algorithm and we went with 5 features as shown in [1] as it showed the best result. Those features are

1.Inter Quartile Range (IQR) :

IQR is the interquartile range Q3 − Q1 of the 3 component vector sum VS, it is given by

VS=✓(x^2+ y^2 + z^2) and

IQR=ΣVS

2. CAV (Cumulative Absolute Velocity) : CAV feature is the cumulative measure of the VS in the time window and it is calculated as

$$CAV = \int_0^s |VS(t)|dt$$

3. Max ZC: Counts for that component whose maximum absolute amplitude value is greater than the other two components when there is more than one zero-crossings at a particular time t. Otherwise, it will behave like the ZC feature.

4.Min ZC: Counts for the minimum one, which has lowest absolute amplitude value among the three, if there are zero-crossings in more than one component.

5.Max Non ZC: This feature counts the maximum absolute amplitude component for non-zero-crossings when there is more than one non-zero-crossings simultaneously at a particular time.

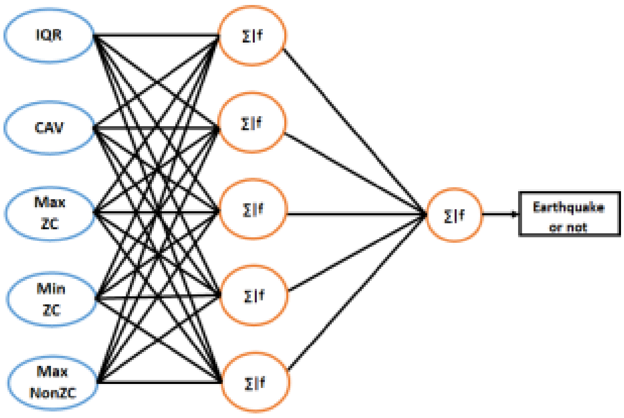

Fig6.ANN model

For training our model we used stochastic gradient descent (SGD) training algorithm and logistic sigmoid function as our activation function. It is a nonlinear activation function, it is given by

$$y = \phi(\sum_{i=1}^{n} w_i d_i + b) \quad \text{and} \quad \phi(d) = \frac{1}{1+e^{-d}}$$

Where y = Output of given node

Φ = Activation function

d = Input vector

b = bias

5. RESULTS AND DISCUSSIONS

After testing our ANN with multiple iterations such as testing with other activation functions and different training algorithms, optimizers, etc. We choose the one with the most accurate model and that model is very similar to the model used in [1]. The model with best results is shown below

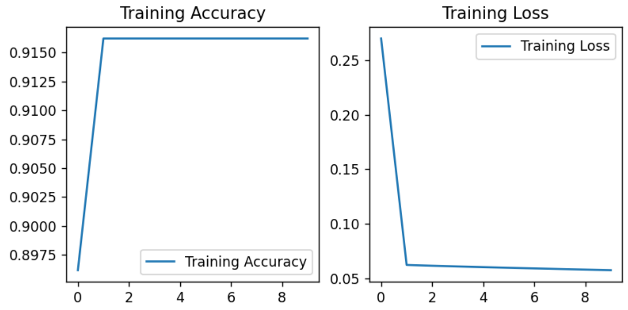

Fig7. Training Accuracy and loss of a model 1

Results for some other models are also shown below

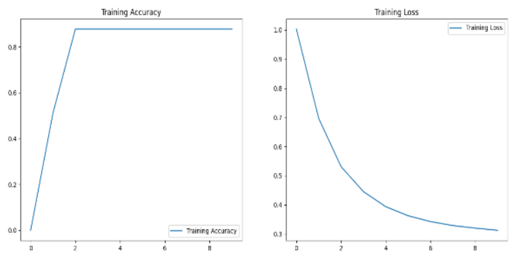

Fig8. Training Accuracy and loss of a model 2

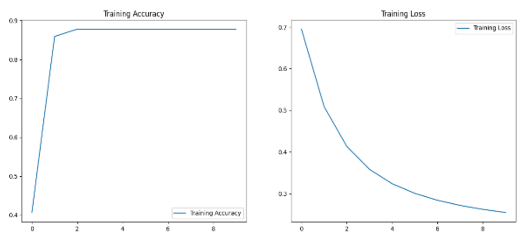

Fig9. Training Accuracy and loss of a model 3

For model 1, model 2 and model 3 we got an accuracy of 93.44% ,85.40% and 87.80% respectively. The difference between them is that model 1 used the same properties used in [1] whereas in model 2 we changed our activation function to relu and for model 3 we tried changed our learning algorithm to adam. We also tried with different number of features and ended up with 5 features as they have shown the best results.

6.CONCLUSION

In conclusion, this research presents an easy approach to earthquake detection and early warning system using IoT devices such as low-cost accelerometers, ESP32 Wi-Fi modules and also with the help of artificial neural networks (ANN). Through a thorough research on related works and different methodologies we have developed this model.

The proposed framework involves three key stages: data acquisition and transmission, processing, and alerting and the dataset used for training contains various earthquake scenarios combined with non-earthquake data to improve accuracy of our model. After exploring different methodologies, we have ended up with an accuracy of 93.44%. by employing the algorithm in section 4.2

Overall, this research contributes to the advancement of earthquake early warning systems by offering a low cost and accurate solution. While our model shows high accuracy there are many parameters that we did not consider and we hope o further develop this project in future.

REFERENCES

1. S. Kim, I. Khan, S. Choi and Y. -W. Kwon, "Earthquake Alert Device Using a Low-Cost Accelerometer and its Services," in IEEE Access, vol. 9, pp. 121964-121974, 2021, doi: 10.1109/ACCESS.2021.3103505.

2. Khan, I.; Choi, S.; Kwon, Y.-W. Earthquake Detection in a Static and Dynamic Environment Using Supervised Machine Learning and a Novel Feature Extraction Method. Sensors 2020, 20, 800. https://doi.org/10.3390/s20030800

3. R. Hoque, S. Hassan, M. A. Sadaf, A. Galib and T. F. Karim, "Earthquake monitoring and warning system,"2015 International Conference on Advances in Electrical Engineering (ICAEE), Dhaka,Bangladesh,2015,pp.109112,doi:10.1109/ICAEE.2015.750 6808.

4. M. R. Kibria Badhon, A. Ranjan Barai and F. Zhora, "Remote Real Time Monitoring and Safety System for Earthquake and Fire Detection Based on Internet of Things," 2019 3rd International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE), Rajshahi, Bangladesh, 2019, pp. 4144, doi: 10.1109/ICECTE48615.2019.9303575.

5. Kong, Q.; Allen, R.M.; Schreier, L.; Kwon, Y.W. MyShake: A smartphone seismic network for earthquake early warning and beyond. Sci. Adv. 2016, 2, e1501055.

6. National Research Institute for Earth Science and Disaster Prevention. Available online: http://www.kyoshin.bosai.go.jp

7. Peer Ground Motion Database, Pacific Earthquake Engineering Research Center. Available online: https://ngawest2.berkeley.edu/

8. J. Lee, J. -H. Sim, J. -K. Ahn and Y. -W. Kwon, "An Open Dataset for Deep Learning-based Earthquake Detection using MEMS Sensors," 2022 IEEE International Conference on Big Data (Big Data), Osaka, Japan, 2022, pp. 6755-6757, doi: 10.1109/BigData55660.2022.10020481.

9. ADXL345 Accelerometer Interfacing with ESP32 link: https://www.electronicwings.com/esp32/adxl345-accelerometer-interfacing-with-esp32

10. Artificial Neural Network in TensorFlow Link:https://www.geeksforgeeks.org/artificial-neural-network-in-tensorflow/

11. Communication Between Arduino & Python Link:https://www.learnrobotics.org/blog/communication-between-arduino-python/

12. Introduction to Keras for engineers Link: https://keras.io/getting_started/intro_to_keras_for_engineers/