Volcanic Eruption Prediction

Abdulrahman Huessein
E-JUST University
abdulrahman.hussien@ejust.edu.eg

Farah Mahmoud AbdelMoneim
E-JUST University
farah.abdelmoneim@ejust.edu.eg

Magy Gamal Matta
E-JUST University
magy.gamal@ejust.edu.eg

Toka Ossama Barghash
E-JUST University
toka.barghash@ejust.edu.eg

Walid Gomaa
E-JUST University
walid.gomaa@ejust.edu.eg

Abstract

Detecting volcanic eruptions before they occur is a significant issue that has traditionally proved to be a very challenging one.However, it is a very interesting concern because of its great potential to save millions of lives by accurately forecasting the eruption time of volcanoes in the proximity of the inhabited areas, which would provide additional time and encourage the evacuation process of people early enough before the catastrophe. In order to forecast the eruption of volcanoes, 10 sensors were mounted around 4431 volcanoes to monitor the eruption time. The goal is to estimate the time of the eruption of another 4520 volcanoes in the coming years. Now that we have established the problem, our approach is formulated using various regression models of machine learning to see the best solution for prediction; Random Forest, DNN, CNN, and LSTM. More details of the models and the data set are mentioned below. In result, we were more interested in the mean absolute error and since that the time to erupt values are large, the loss will be as well. The best loss is 6042665 belongs to CNN model. In addition, we also used techniques to optimize networks, acting as activation function, dropout and max pooling. Finally, we analyzed the result from 4 models to observe the relationship between them and loss produced.

1 Introduction

Every year about 60 volcanoes erupt and since the year 1500, about 280,000 people have been killed by volcanoes - 170,000 of those by just six eruptions [1].

With the growth of the machine learning field and its applications in every field one can imagine, in addition to the enormous amount of time series data collected from sensors applied on volcanoes, researchers were able to make use of that to predict volcanic eruption time to alert citizens living around a volcano before eruption to avoid death and property damage.

In 2019 paper titled "CRED: A Deep Residual Network of Convolutional and Recurrent Units for Earthquake Signal Detection" [2] Mostafa Mousavi et al used CNN and biLSTM for earthquake signal detection which is the most close to our topic. Each of the two architectures is good in a way different than the other.

The first successful applications of CNNs were developed by Yann LeCun in 1990's. Of these, the best known is the LeNet architecture that was used to read zip codes, digits, etc.[3]Day by day, CNNs has been developing by the scientist community. Specially, in computer vision, there are a lot of momentous works which use CNNs approach such as AlexNet[4], VGG-Net[5], GoogleNet[6], and ResNet[7]. Besides, there are some contribution from public challenges, typically INGV - Volcanic Eruption Prediction in Kaggle (2020).

LSTM was first proposed by Sepp Hochreiter and Jurgen Schmidhuber [8] to deal with the problem of the vanishing gradient In 1995-1997. The initial version included input and output gates. The forget gates were introduced in 1999.[9] The architecture of the LSTM was made to deal with time series data that depend on each other like words in a sentence or voice and audio.In 2015 Google used LSTM in voice recognition.[10]In 2017 Facebook made 4.5 million automatic translation using LSTM.[11]. In our paper we used bidirectional LSTM which is even better than unidirectional LSTM because the learning goes both ways at the same time and this makes it faster and better.

Random forests are also a good way to check that the model will work where its idea was first proposed by Ho [12] and the first proper introduction was in a paper by Leo Breiman [13]. Random forests are a part of ensemble learning which is the method of using multiple learning algorithms at the same time to get better prediction results which is represented by the multiple trees training at the same time used for classification and regression too.

2 Related Work

The problem of volcano forecasting is known to be a hard one however, the prediction of the eruption time is very struggling challenge for scientists. New technologies was developed to monitor volcanic activities, the broad-band seismology is the most important one. Seismic monitoring is very correlated with all different activities in the volcano that helps in recognizing the signature of each activity in order to forecast volcanoes[14]. A lot of forecasting is being practised on some active volcanoes and observations are taken. It appears that the key for forecasting is better understanding of volcanic processes not just manipulating raw data. The numbers only is not enough for volcanoes forecasting [15]. Four methods was used to predict the time to eruption on Kilauea earthquakes; Logistic regression, K-means, random forest and neural network. The best model of them which is the random forest gives a weak correlation between observed and predicted times to eruption.[16]. Figure 1 shows the results of random forest model on Kilauea earthquakes volcanoes to predict time-to-eruption.

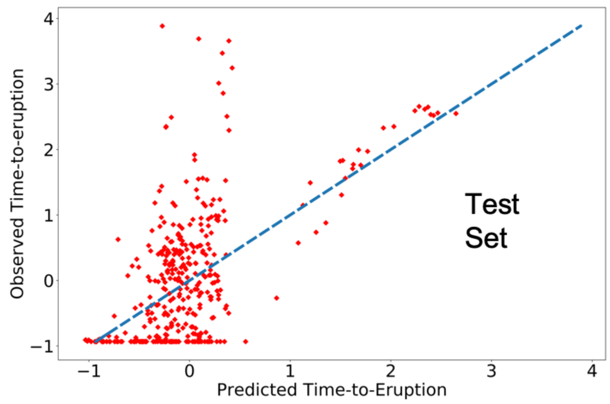

Figure 1: Prediction of time to eruption

3 Dataset

We trained and tested our model on Kaggle dataset from Volcanic Eruption Prediction Challenge [17]. Data is collected by researchers at INGV ,National Institute of Geophysics and Volcanology. The dataset consists of .csv files, divided into training files, testing files, a label file for the training data and another one for the testing data. The data files (both training and testing), each file contains ten minutes of logs from ten different seismic sensors arrayed around a volcano, the data represent a classic signal processing setup that has resisted traditional methods. The readings have been normalized within each segment, in part to ensure that the readings fall within the range of int16 values in addition to the presence of some nulls.

The labeled dataset contains 4,431 training files each named after the ID of a volcano, we used 75% of these files for training and 18.75% for validation and 6.25% for testing (apart from the original testing data we have so we will call it **Labeled Testing Data**), each file consists of 60,001 reading from the 10 different sensors for that specific volcano as shown in figure 2. An example of the readings of one of the sensors ,sensor_2 to a given volcano is showed in figure 3. As for the testing, there are 4520 files, also named after the IDs of testing volcanoes and consist of the same number of readings and sensors as the training files but their labels are not published as it is part of the Kaggle competition so we will call it the **Unlabeled Testing Data**.

Training data label file lists training volcanoes' IDs and the remaining time to their next eruption which ranges from 6250 to 49 Million time unit as shown in figure 4. Finally, the testing data labels file lists the testing volcanoes' IDs and the predicted remaining time to their next eruption produced by each model using the same time unit in the training data. Therefore, the testing data labels file represents the output of the models used.

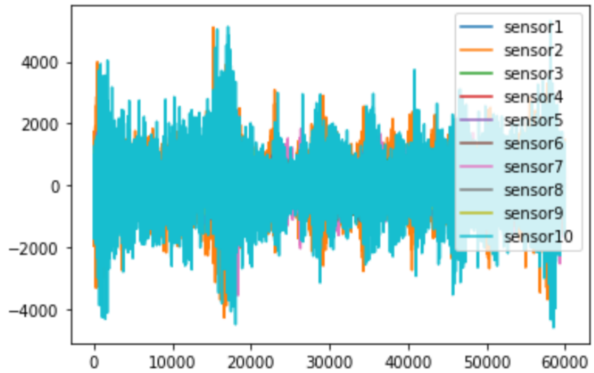

Figure 2: Readings of all Sensors of a Given Volcano

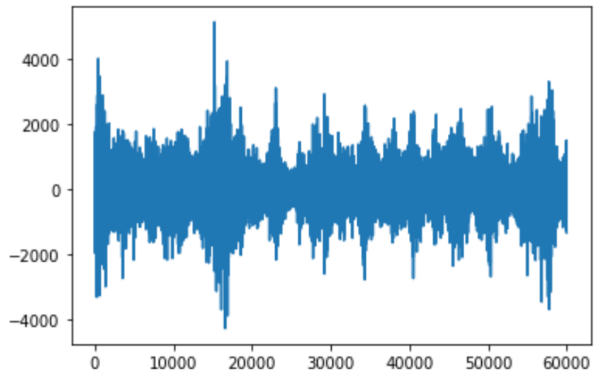

Figure 3: Readings of Sensor_2 of a Given Volcano

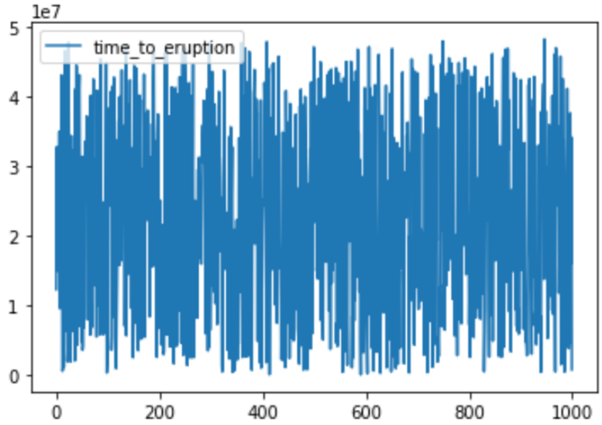

Figure 4: Time to Eruption

4 Method

4.1 Overview

We use 4 different models, Convolution Neural Networks (CNNs), Deep Neural Networks (DNNs), Long Short Term Memory (LSTM) and Random Forest (RF) to predict eruption time of 4520 volcanoes. Each model is briefly explained below.
In CNNs approach, the input data passes by Short Time Fourier Transform as a feature extraction to improve the performance while training and save time. The extracted features then passes by 3 convolution layers with tuned parameters including learning rate, regularization, max pooling and dropout.
In DNNs approach, preprocessed data using STFT as in CNN, passes through multiple fully connected layers in addition to a batch normalization layer to automatically standardize the inputs to the following FC layer using the mean and standard deviation of each batch in addition to accelerating the training process in our deep learning neural network, and it finally improves the performance of the model via modest regularization effect.
In LSTM approach preprocessed data by STFT passes through two bidirectional LSTM layers then a flatten layer for feature extraction then three dense layers for regression. There is batch normalization in between the biLSTM layers. The LSTM architecture was chosen because it was made for those kind of problems which have time series data where they memorize the sequence of data of length of the number of units of the layer.
To reach the motivation of this paper, we implement our four models from scratch and according to their results, we compare them to reach our final conclusion. Below is the explanation of data preprocessing and full explanation of each model.

4.2 Pre-Processing

This pre-processing of data is used in three of the four models we used, CNN, DNN, and LSTM. We used Short Time Fourier Transform (STFT) function [18] that is used to assess the sinusoidal frequency and phase content of the local signal parts as they shift over time. In fact, the method for calculating STFTs is to split the longer time signal into shorter segments of equal length and then calculate the Fourier transform separately for each shorter segment. This shows the Fourier spectrum for each shorter section. We set the number of segments to 400 and the overlapping factor to 0.25 that decreased the Time factor (reduction from 60001 to 202) while retaining an appropriate time resolution (3sec) with frequency factor 201 as shown in figure 5

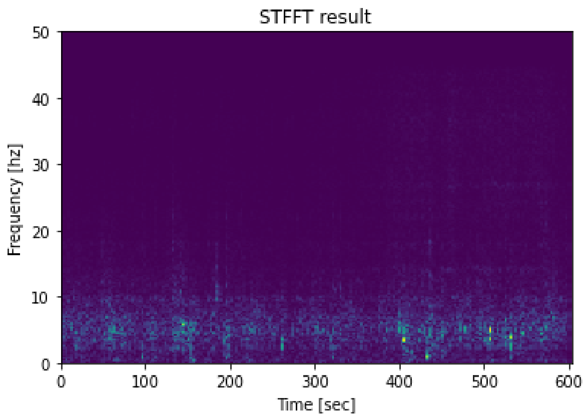

Figure 5: STFT results

As shown in figure 5, A new frequency range was created with range(0-50Hz) with coverage 0.25Hz resolution but that range is not practical. That's why redimensioning to frequency was done where the first row of 0HZ frequency was removed, the frequency is reduced by a factor of 4, by averaging the value of 4 continuous frequencies. Then the frequency from range (1-25Hz) was kept and the others were removed. The results are show in figure 6

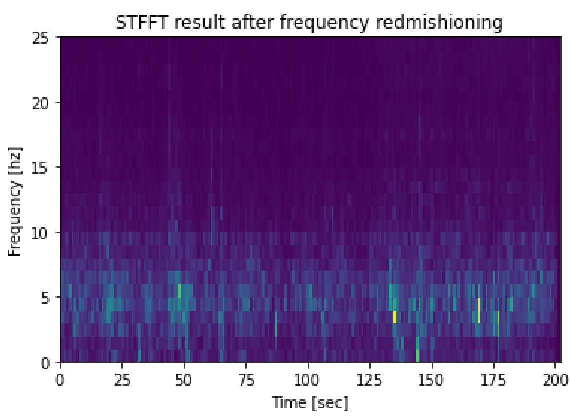

Figure 6: Redimentioned STFT Results

For each sensor of each instance (volcano), the above mentioned data processing steps were replicated. When done, we ended up with a dimensional matrix

of (4431, 25, 202, 10), (instance, frequency, time and sensor dimensions respectively)
We removed the nan, took the absolute, concatenated the data of sensors to give me matrix of (3323,202,250) after splitting the training data for validation, training and testing to 831,3323,277 each respectively.

4.3 CNN

This network has three convolution layers. In the first convolution layer, we had 64 filters with kernel size is 15x15, padding mode is ‘same’ and value of input shape is (202,10*25) since the signal has time dimension of 202 and a frequency dimension of 25 of each of the ten sensors. The second convolution is a bit different from first one. In this layer, we have 32 filters with kernel size is 10x10, padding mode is ‘same’. The first two convolution layers are followed by, max-pooling layer each and dropout of 0.2 after the second one.In the third convolution, we have 12 filters with kernel size is 7x7, padding mode is ‘same’ that is don’t change the size. It is followed by a max-pooling layer and dropout as well. Pooling setup is 2x2 with a stride of 1 to reduce the size of the receptive field and avoid over-fitting. In dropout layer, a fraction of 0.2 and 0.3 are used.
After 3 convolution layers, flattening takes place, then the compiling with loss function "mean absolute error" optimizer Adam, learning rate of 0.03, $beta_1 = 0.9, beta_2 = 0.999$, epsilon= $1e-07$. and finally fitting with parameters shown in figure 7. Noting that the first cell in X and x_valid is the cell of size of each of them
Also in all the layers, Rectified Linear Unit (ReLU) is used as the activation function to model all the convolution layers. ReLU is simple and make high performance for regression.

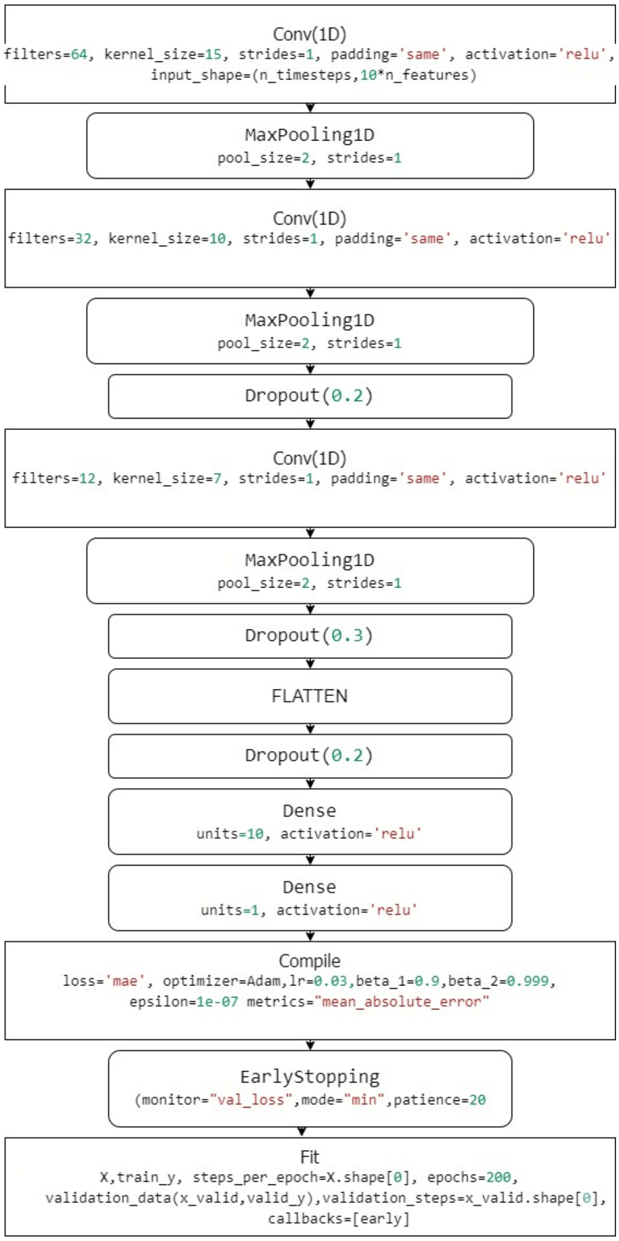

Figure 7: CNN Model

4.4 DNN

The second method we used was a deep neural network, this network mainly depends on multiple fully connected layers in addition to a batch normalization layer and a dropout layer built as following. In the first dense layer,we had 256 neurons (units) where units represent dimensionality of the output space and value of input shape is (202,250) for the reasons mentioned in the Pre-processing subsection, we use the input with batch size of 32, and finally the Rectified Linear Unit (ReLU) is issued as the activation function for its simplicity and high performance to model non-linearity. Followed by a batch normalization layer that normalizes its output using the mean and standard deviation of each batch of inputs with momentum for the moving average of 0.8.
The 3 successive fully connected layers have the same parameters as the first one excluding defining the

already stated input shape, then a flatten layer to convert the data into a 1-dimensional array for inputting it to the next layer creating a single long feature vector. Afterwards, another dense layer is added, however with 512 units and same activation function. Then a dropout layer that randomly sets input units to 0 with a frequency of a fraction set to 0.5. Finally, the last dense layer with 1 output neuron and linear activation function that returns the input, unmodified. The architecture of the model is shown in figure 8

The model is compiled using Nadam optimizer which is (Nesterov-accelerated Adaptive Moment Estimation) thus combines Adam and NAG with learning rate = 0.001, while setting our loss function to be mean absolute error.

4.5 LSTM

This network has three biLSTM layers with 64 units then a flatten layer to prepare the input to the fully connected layers (FCL) that come after where we used three FCLs one with 64 units, one with 32 units and the last one with 1 unit where all the three dense layers use ReLU activation function even the output layer because our output is a real number and normal kernel initializer which means that the weights were initialized with values from the normal distribution. We used dropout of value 0.2 which means that in every batch we deactivate 20 % of the neurons randomly in the successive epochs for the purpose of decreasing overfitting. The input shape is (4431,202,250) where the input to LSTM layer should be number of instances then number of samples in the sequence and last number is the number of features where here the 4431 is the number of volcanoes and 202 is the time samples and 250 is the 25 frequencies each for the 10 sensors which represent the features. We added batch normalization before and after each biLSTM layer which helps make training more efficient. We used adam optimizer with learning rate of 0.005 where adam depends on gradient descent for learning and mean absolute error to compute the validation loss where we took quarter of the training dataset and then we took 0.75 of that quarter for the validation dataset which resulted in a shape of (831, 202, 250) and the other 0.25 for the testing to be of shape (277, 202, 250). We also trained for 1000 epochs with a batch size of 128.

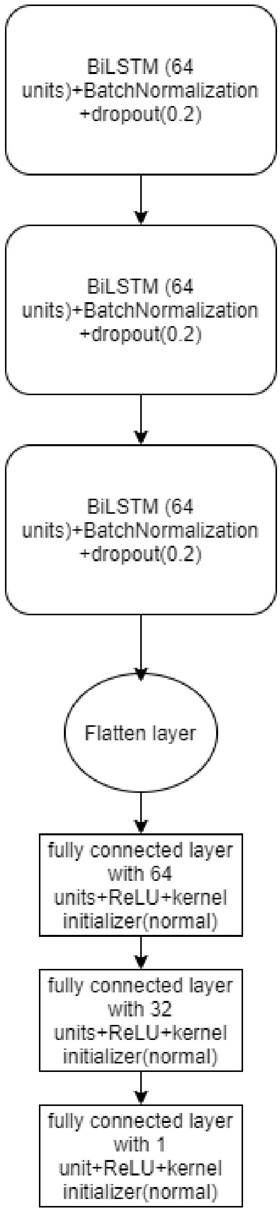

Figure 9: LSTM Model

4.6 Random Forest

4.6.1 Pre-Processing

Since we have 4431 training samples, each sample contains 6000 values for each of the 10 sensors for each volcano monitoring its seismic wave signals. We could extract features depending on time-series waves. Auto-correlation method between the wave and itself was applied to show how much the wave is similar to its time-shifted version. At zero time difference, the auto-correlation yields highest values which reveal trends and patterns in time series waves. Furthermore, it helped us to lower training data size by considering only the greatest points values resulted from auto-correlation by truncating the rest of the output. The 6000 values are lowered to only 200 values for each sensor that gives more variance due to the high values difference among data points generated from auto-correlation. The

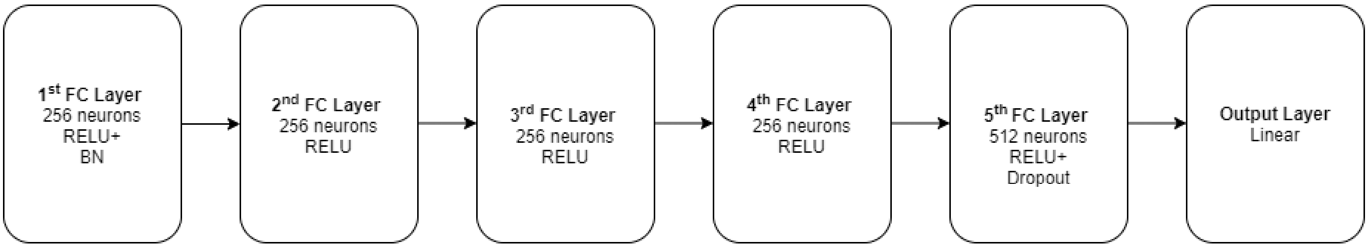

Figure 8: Architecture of DNN Model

following graphs compare two signals before and after auto-correlation.

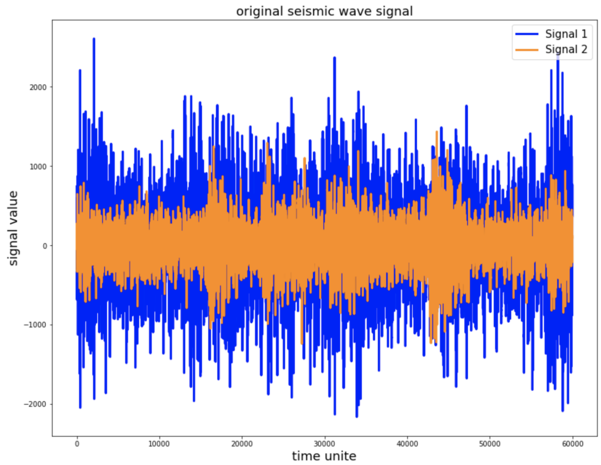

Figure 10: Original signal of sensor 1 on two different volcano

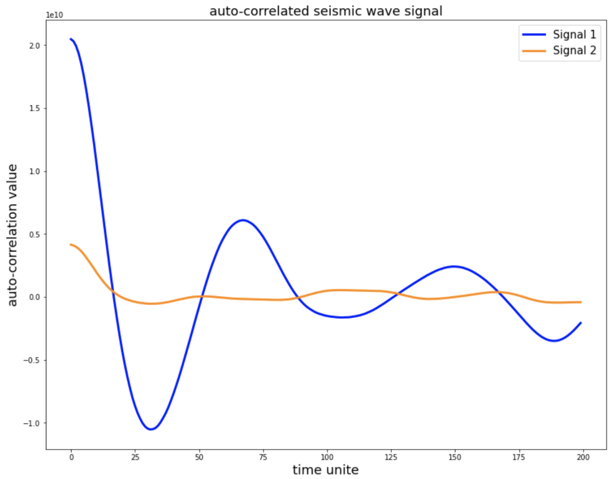

Figure 11: Auto-correlated signals of sensor 1 on two different volcano

4.6.2 Random-Forest Model

Random forest regression techniques, it creates number of decision trees depending on the input features, then evaluate an output depending on average of multiple decision trees and get their weighted predictions to provide a more stable output. After auto-correlation feature extraction, we demonstrated random forest regression technique to train our model with 100 estimators and tested on 20% of training data. The model take features in shape of (4431,10). The size reduction of input was very helpful to speed up training processes to 9 minutes training.

5 Result

5.1 CNN

The model was trained using optimizer Adam, learning rate 0.03 and compiled with Mean Absolute Error loss function as mentioned in figure 7 for 200 epochs with patience of 25 for the early stopping. The mean absolute error for training and validation are where the blue represents the validation mean absolute error and the red represents the training mean absolute error and the results are show in figure 12. model.evaluate was done to the testing data which is 6.25% of the data (277 volcanoes) and the result was 2907124. Also the comparison between predicted and true labels of testing data is graphed and showed in figure 13

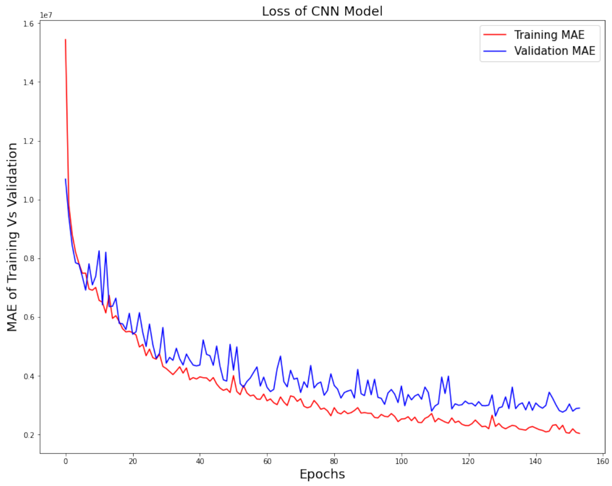

Figure 12: Loss of CNN Model

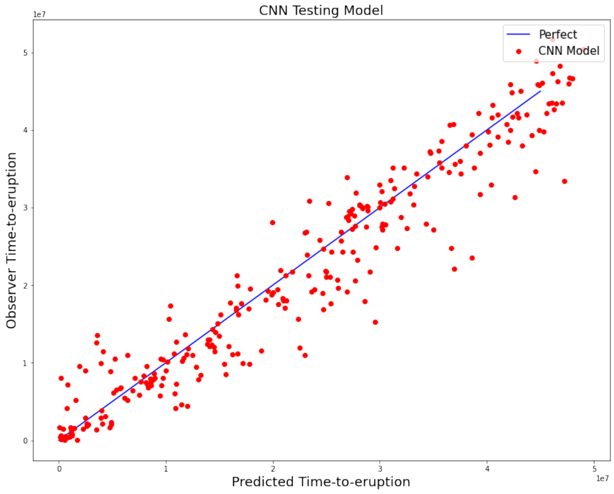

Figure 13: CNN Testing Model

5.2 DNN

The mean absolute error for training and validation over 300 epochs are shown in figure 14 where the blue represents the validation mean absolute error and the cyan represents the training mean absolute error. Evaluating the model over the validation and testing data resulted 2585667.75 and 2611425.5 respectively. Also the comparison between predicted and true labels of testing data is graphed and showed in figure 15

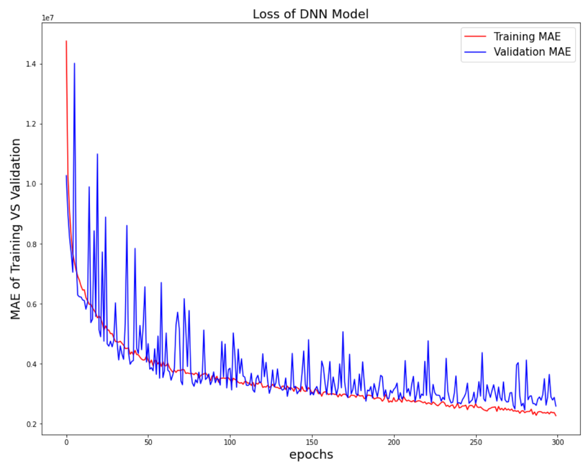

Figure 14: Loss of DNN Model

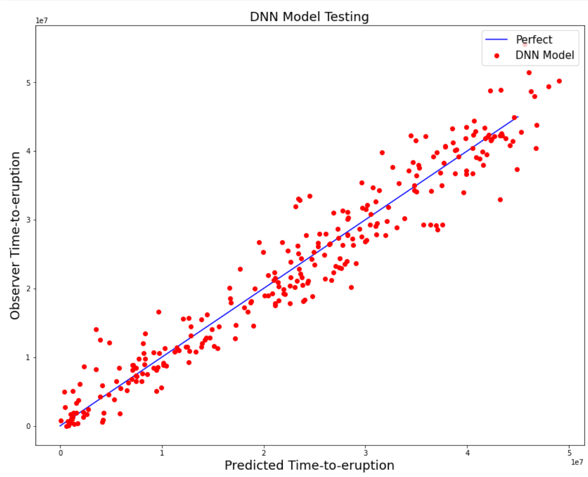

Figure 15: DNN Model Testing

5.3 LSTM

Below the first figure shows the loss during training which ended with a very small training loss of about 343475.3819 and validation loss of 2366989.0000.The evaluation of the model on the test data resulted with loss of 2460357.5. The second figure shows the plotting of the predicted values versus the true labels.

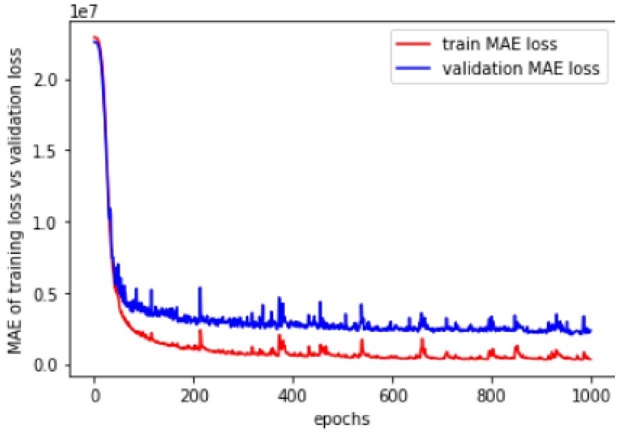

Figure 16: Loss of LSTM Model

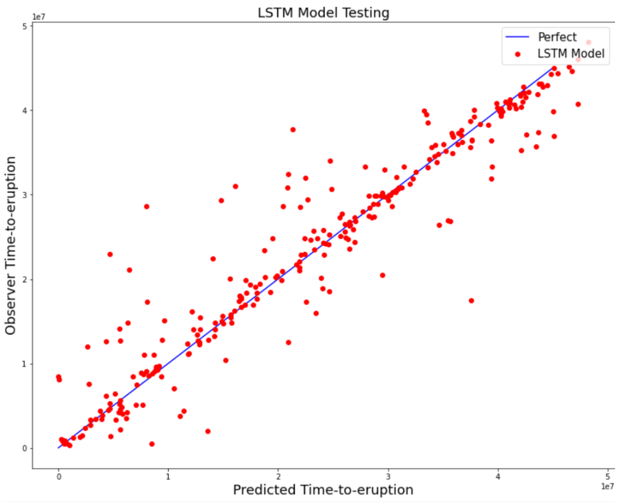

Figure 17: LSTM Model Testing

5.4 Random Forest

The trained model based on random forest regression technique gives us prediction accuracy 91% and mean absolute error 2705934 time unit. However, the model does not perform well on unseen test data. Random forest is reliable when the problem is a classification problem, but in our case the problem is a regression problem which random forest does not perform well beyond range of training data with $MAE = 13296316$ and over-fits on noisy data, which is the case here with seismic waves and missing testing data sensors. Random forest model is a back box technique that is not interpretable, so validation loss is not evaluated. 13296316

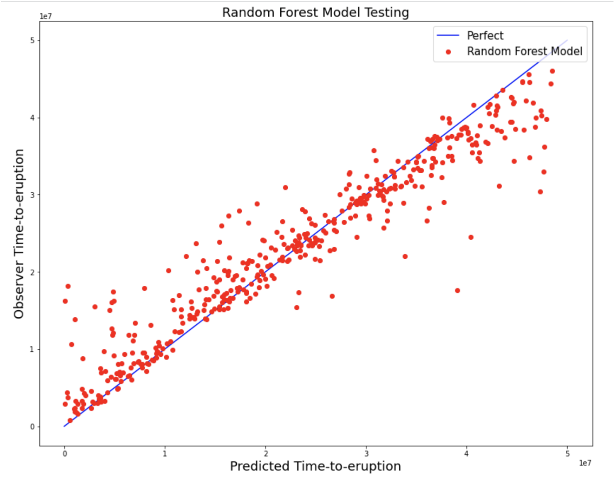

Figure 18: Original signal of sensor 1 on two different volcano

In table 1, we can see the RF model has the least loss in seen testing data, while CNN has the least loss in unseen data.

6 Conclusion

In this paper, we have explored CNN, DNN, LSTM and RF in our regression problem that to predict the time of eruption of Volcanoes. Firstly, we implemented a CNN, which have three convolution layers, and got good results for the unlabeled data of 6042665 and for the labeled we got 2907124. we demonstrated a DNN model by adding two convolution networks Five dense layers and got better results at Labeled Testing of 2611425.5 but slightly less ones at the labeled ones of 623026. The highest score we got at the unlabeled testing was of 6028014 by the LSTM model. However it has much more loss than CNN and DNN model of 3805102.5. The three mentioned models above took an input data that passes by feature extraction model (STFT) and then to the model. Our Final model was Random Forest, we used autocorrelation for pre-processing it gave best results for the labeled testing of 2705934 , on the other hand, for the unlabeled testing data the score was 12.9 million. That's because of model interpret-ability which is one of the main draw back of RF, they are not interpretable; they are like black box also the unlabeled data set contain some noise and some missing data, and Random Forest is sensitive to noise. As mentioned above the best model for the prediction of the unseen data was the LSTM

References

[1] D. S. Brown. Hawaii volcano: How many people do volcanoes kill? [Online]. Available: https://www.bbc.com/news/world-us-canada-44212666#:~:text=This%20could%20appear%20to%20suggest,are%20much%20deadlier%20than%20Kilauea.&text=Since%20the%20year%201500%2C%20about,those%20by%20just%20six%20eruptions.

[2] Y. S. G. C. S. Mostafa Mousavi, Weiqiang Zhu, "Cred: A deep residual network of convolutional and recurrent units for earthquake signal detection," *scientific reports*, no. 10267, 2019.

[3] P. H. Y. B. Yann Lecun, Leon Bottou, "Gradient based learning applied to document recognition," *The IEEE*, 1998.

[4] G. E. H. Alex Krizhevsky, Ilya Sutskever, "Imagenet classification with deep convolutional neural networks," 2012.

[5] A. Z. Karen Simonyan, "Very deep convolutional networks for large-scale visual recognition," 2014.

[6] Y. J. P. S. S. R. D. A. D. E. V. V. A. R. Christian Szegedy, Wei Liu, "Going deeper with convolutions," 2014.

[7] S. R. J. S. Kaiming He, Xiangyu Zhang, "Deep residual learning for image recognition," 2015.

[8] J. S. Sepp Hochreiter, "Long short term memory," 1995.

[9] J. K. B. R. S. J. S. Klauss Gref, Rupesh Kumar Srivastava, "Lstm: A search space odyssey," *arXiv*, vol. 28, no. 1503.04069, pp. 2222 – 2232, 2015.

[10] F. Beaufays, "The neural networks behind google voice transcription," *Google AI blog*, 2015.

[11] T. Ong, "Facebook's translations are now powered completely by ai," 2017.

[12] T. K. Ho, "Random decision forests," *Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal*, p. 278–282, 1995.

[13] L. Breiman, "Random forests," *Machine Learning*, vol. 45, 2001.

[14] R. S. J. Sparks, "Forecasting volcanic eruptions," *Earth and Planetary Science Letters*, vol. 210, no. 1-2, pp. 1–15, 2003.

[15] R. Decker, "State-of-the-art in volcano forecasting," *Bulletin Volcanologique*, vol. 37, no. 3, p. 372–393, 1973.

Table 1: Validation VS Testing Loss of each Model

Model	Validation Loss	Labeled Testing Loss	Unlabeled Testing Loss
CNN	2904451.25	2907124	6042665
DNN	2585667.75	2611425.5	6230261
LSTM	343475.3819	2460357.5	6125066
RF	-	2705934	13296316

[16] Y. Q. W. B Mullet, A Singh, "Predicting eruptive events at volcanoes from earthquake data," https://github.com/bmullet/PEEVED, 2019.

[17] INGV, "Kaggle volcanic eruption prediction challenge." [Online]. Available: https://www.https://www.kaggle.com/c/predict-volcanic-eruptions-ingv-oe/data

[18] scipy.signal.stft. [Online]. Available: https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.stft.html