

Article

Deep-Learning-Based Strong Ground Motion Signal Prediction in Real Time

Mohammad AlHamaydeh ^{1,*} , Sara Tellab ² and Usman Tariq ³ 

¹ Department of Civil Engineering, College of Engineering, American University of Sharjah, Sharjah P.O. Box 26666, United Arab Emirates

² Department of Mechatronics, College of Engineering, American University of Sharjah, Sharjah P.O. Box 26666, United Arab Emirates; saratellab@gmail.com

³ Department of Electrical Engineering, College of Engineering, American University of Sharjah, Sharjah P.O. Box 26666, United Arab Emirates; utariq@aus.edu

* Correspondence: malhamaydeh@aus.edu

Abstract: Processing ground motion signals at early stages can be advantageous for issuing public warnings, deploying first-responder teams, and other time-sensitive measures. Multiple Deep Learning (DL) models are presented herein, which can predict triaxial ground motion accelerations upon processing the first-arriving 0.5 s of recorded acceleration measurements. Principal Component Analysis (PCA) and the K-means clustering algorithm were utilized to cluster 17,602 accelerograms into 3 clusters using their metadata. The accelerograms were divided into 1 million input–output pairs for training, 100,000 for validation, and 420,000 for testing. Several non-overlapping forecast horizons were explored (1, 10, 50, 100, and 200 points). Various architectures of Artificial Neural Networks (ANNs) were trained and tested, such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM) networks, and CNN-LSTMs. The utilized training methodology applied different aspects of supervised and unsupervised learning. The LSTM model demonstrated superior performance in terms of short-term prediction. A prediction horizon of 10 timesteps in the future with a Root Mean Squared Error (RMSE) value of 8.43×10^{-6} g was achieved. In other words, the LSTM model exhibited a performance improvement of 95% compared to the baseline benchmark, i.e., ANN. It is worth noting that all the considered models exhibited acceptable real-time performance (0.01 s) when running in testing mode. The CNN model demonstrated the fastest computational performance among all models. It predicts ground accelerations under 0.5 ms on an Intel Core i9-10900X CPU (10 cores). The models allow for the implementation of real-time structural control responses via intelligent seismic protection systems (e.g., magneto-rheological (MR) dampers).

Keywords: Artificial Neural Networks (ANN); Convolutional Neural Networks (CNN); CNN-LSTMs; Deep Learning (DL); Long Short-Term Memory (LSTM) networks; Machine Learning (ML); Recurrent Neural Networks (RNN)



Citation: AlHamaydeh, M.; Tellab, S.; Tariq, U. Deep-Learning-Based Strong Ground Motion Signal Prediction in Real Time. *Buildings* **2024**, *14*, 1267. <https://doi.org/10.3390/buildings14051267>

Academic Editors: Minshui Huang and Jianfeng Gu

Received: 22 December 2023

Revised: 7 April 2024

Accepted: 10 April 2024

Published: 1 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Globally, around 20,000 tangible earthquakes, which do not require measuring tools to be perceived, happen every year [1]. If 100 earthquakes with sufficient magnitude occur in the countryside, considerable damage is expected. Earthquakes of significant magnitudes, which provoke tsunamis, fires, and landslides, occur on average once annually [2]. Immeasurable infrastructure destruction and millions of beings have been lost throughout the ages.

Predicting earthquake signals in real time is a critical challenge in earthquake engineering. The ability to accurately forecast ground motion can provide valuable seconds or even minutes of warning, allowing for the implementation of protective measures and the evacuation of at-risk areas. Traditional methods for predicting earthquake signals rely

on ground motion prediction equations (GMPEs), which are statistical models that estimate the intensity of ground shaking based on earthquake magnitude, distance, and local site conditions. However, GMPEs are limited in capturing earthquake ground motion's complex and often unpredictable nature.

Currently, the prediction of forthcoming earthquakes is not numerically possible. Nonetheless, ground motion can be discerned initially, allowing some seconds to notify the public before the big shake. An Early Earthquake Warning (EEW) system can diminish the resulting losses, demonstrating promising potential. The ShakeAlert system, by The U.S. Geological Survey, is one of the different EEW systems examined and implemented on the USA's West Coast and the Pacific Northwest [3]. The ShakeAlert system employs an extensive network of sensors that forward the data to the processing center and notify the users of the texts. Such a system needs around USD 16 million annually for maintenance and operation [4]. It is often unrealistic to expect timely implementation of safety measures, such as evacuations.

This research aims to eliminate the need for human intervention from the process by implementing intelligent seismic protective systems. Machine Learning (ML) and Deep Learning (DL) techniques are ideal for forecasting and continuously updating future segments of an arriving strong ground motion signal by merely identifying the patterns in the first 0.5 s of the event. New predictions are made through constantly feeding points as additional points come on.

The main challenge other researchers have not yet solved, which this work addresses, is the need for human intervention in the current EEW systems. As such, this research proposes using ML and DL techniques to automate the process of forecasting and updating future segments of an arriving strong ground motion signal, eliminating the need for human intervention. Moreover, the presented models can be implemented in real time, allowing for an immediate response to seismic events. Furthermore, the presented approach does not require the extensive network of sensors and processing centers used in current EEW systems, making it more cost-effective. Lastly, the presented models utilize advanced ML and DL techniques, which have been shown to achieve high accuracy in forecasting tasks.

Next, the existing literature on time series prediction is presented. In addition, the methods of predicting ground motion parameters and applications of ML/DL in seismology are discussed.

1.1. Time Series Prediction

A time series is a collection of numbers that are equally separated and arranged sequentially. As a numerical example, time series x is a collection of vectors: $x(t) = \{x_1, x_2, \dots, x_T\}$ [5,6]. A time series with a single value per timestep is described as univariate. Moreover, a time series with multiple values is described as multivariate. To produce future predictions in time series forecasting, the historical data of a time series are utilized—selecting the appropriate model is an important step for representing the underlying form of the series. The prediction models can be categorized as linear or nonlinear based on how the history data are integrated.

Babušiak and Mohylová [7] developed ANN-based models based on previously developed models to predict the electrocardiogram sample. One-layer and multi-layer backpropagation models were the two architectures examined. The multi-layer network showed better accuracy than the one-layer network; nonetheless, it required a more significant number of epochs for convergence purposes. Coelho et al. [8] forecasted the future time period in the electroencephalogram signal. Neighborhood structures were utilized to select and modify the input during the training stage. A maximum accuracy of about 70% was reached with 30 models per person. Moreover, there was no noticeable variation in accuracy once the maximum forecast horizon was predicted. The uncertainty in extremely nonlinear data challenges the modeling process. The stock market is a good example, as various prediction approaches can be followed to predict the future trend. Convolution Neural Net-

works (CNN), Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM) networks have been used by many researchers [9–11]. For example, Selvin et al. [12] used the sliding window approach to predict stock prices, employing the CNN, RNN, and LSTM networks. CNN could produce the best results in accuracy since it is independent of the history, far from RNN, which requires preceding sequences for generating a prediction.

RNN has been considered the most common for time series forecasting. Romazanov et al. [13] predicted the temperatures of buildings indoors using RNN. The input was multivariate, while the output was univariate. The wind and temperature outdoors are examples of the inputs. The number of observations, known as the sequence length, was altered to decide the optimum. A sequence length of 120 had the highest prediction accuracy. Furthermore, a similar accuracy was reached for a sequence length of 12 for short-term prediction.

The LSTM network is more robust than RNN. Qing and Niu [14] employed an LSTM network to perform solar irradiance prediction the next day (a day-ahead prediction). A multivariate input, made of 11 timesteps and nine features, was utilized. The LSTM architecture was composed of a single layer and a dense layer afterward, i.e., the output layer. A better performance of 18% was determined for the LSTM network compared to the ANN, maintaining an output-to-input ratio of 9%.

Recently, CNN has become a common state-of-the-art model in time series forecasting, computing vision techniques [15], and natural language processing [16]. Hussain et al. [17] proposed a one-directional CNN (1D-CNN) model for a step-ahead flow prediction for the river stream. Daily, weekly, and monthly time intervals were considered in designing the three 1D-CNN models. An input and an output-to-input ratio of 4 and 25%, respectively, generated superior performance. Moreover, the worst performance was observed for the model with monthly intervals because of the need for more capability to display the dynamic changes. Hence, the data sampling time is essential to the model's performance.

For an optimized network architecture that can reflect the time series' dynamic properties, the CNN's pattern extraction capability and LSTM's memory should be integrated. Livieris [18] operated a CNN-LSTM model for predicting the daily price of gold by using past and present prices to obtain the future price of gold. The best performance was implied when implementing two CNN layers, one LSTM layer, and a fully connected layer. Furthermore, six data points referred to the forecast horizon of optimal performance. Similarly, Qin et al. [19] operated a CNN-LSTM model to forecast the concentration of air pollutants. The multivariate input combines meteorological factors and historical concentration and was 72 h long. Additionally, the forecast horizon was 24 h; thus, the output-to-input ratio was 33%. For comparison purposes, a shallow ANN model was the baseline model. The proposed CNN-LSTM achieved a Root Mean Squared Error (RMSE) of 36% and 20% lower than the ANN and conventional LSTM models, respectively.

1.2. Ground Motion Estimation

Ground Motion Prediction Equations (GMPEs) evaluate the parameters following the classical method. The classical method depends on the earthquake's route, source, and area conditions, which are the fundamental aspects of the ground motion [20], for obtaining a definition in simplified linear regression form for the ground motion. Gülerce [21] proposed GMPEs that involve a more significant number of regression coefficients for better accuracy; nonetheless, the complexity was increased. A model's definition needs details about the rupture developments and the fault characteristics to be accurate and complete. These are initially unidentified and challenging to acquire. Moreover, the model's region-dependent parameters need to be estimated and tested. Thus, the parameters are appropriately defined where a dense network of seismographs is available [22]. However, that is not true for most seismically active regions due to the limited available measurements.

1.3. Applications of ML/DL in Seismology

ML has been broadly incorporated in the seismic fields of forecasting, feature extraction, and classification, as there has been a considerable expansion in the availability of seismic data. Ramirez and Francois [23] utilized a supervised ML and feature extraction technique for classifying the seismic waves flowing in three directions. P-phase implies compressional waves, and L-phase implies surface waves, the two output types. An accuracy of around 68% was observed in the study.

Li et al. [24] applied ML techniques to minimize the false EEW. The method was utilized to distinguish P-waves from noise with great accuracy. Feature extractors, Generative Adversarial Networks (GANs), and random forest algorithms were utilized for the input signal. An accuracy of 98% and 99% for the noise signals and P-waves, respectively, indicates excellent potential for the constructed discriminator in seismic applications. Böse et al. [25] estimated the hypocenter and magnitude using feed-forward ANNs of double hidden layers. The proposed system depends on the seismic signals from the various sensors in the Marmara zone to generate an EEW. Four stations and a 3.5 s window resulted in superior accuracy. Nevertheless, an improvement in accuracy comes at the expense of time efficiency for EEW.

Kuyuk and Susumu [26] proposed a more reliable technique for classifying the P-waves into near or far sources. The input had a 1 s length and was divided into 13 points. The system comprised a hidden layer of 100 neurons and a classification layer, i.e., the LSTM layer. For both classifications, an accuracy of more than 95% during the training phase and 66% during the testing phase was determined. Adeli and Panakkat [27] investigated the performance of probabilistic ANN for predicting the magnitude depending on eight seismic indicators. Based on data collected from the Southern California region, the magnitude was classified into seven ranges on the Richter scale, between less than 4.5 up to 7.5, with R^2 equal to 0.78. On the other hand, an R^2 of less than 0.5 was determined for significant, i.e., large-scale, earthquakes. The lack of available large-scale earthquakes in the dataset justified the classification error.

Rouet-Leduc [28] developed an algorithm for predicting fault failure in the laboratory. The remaining time preceding an artificial earthquake was ascertained by monitoring the acoustic signal and removing the noise. Chakraverty et al. [29] utilized ML for seismic response prediction of a two-floor shear building. Computed responses and actual acceleration signals were used to train a Feed-Forward Back-Propagation (FFBP) network with one hidden layer. The results displayed adequate accuracy for the two stories; however, sophisticated calculations are needed for feature extraction. Kerh and Ting [30] estimated the Peak Ground Acceleration (PGA) using a Multi-Layer Feed-Forward neural network. The input combined epicentral distance, magnitude, and focal depth in the three networks of one hidden layer. Based on 21 testing cases, the output was PGA in one axis. A low correlation level was demonstrated, as a correlation coefficient, R^2 , of less than 0.5 was determined for 86% of the cases. Arjun and Kumar [31] utilized a Multi-Layer Feed-Forward neural network for duration estimation of strong ground motion using the shear wave velocity, hypocentral distance, magnitude, and average soil properties. An accuracy of 55% and 61% was determined using six and three inputs, respectively, in the developed models. The same inputs were utilized by Günaydın and Günaydın [32], who executed three different ANN architectures: radial basis function, FFBP, and generalized regression. The ANNs were employed to predict the maximum PGA in three directions by feeding the direction of the maximum PGA. FFBP with one hidden layer exhibited the best performance in all three directions. Moreover, the radial basis function attained a Root Mean Square Error (RMSE) of 58.17 cm/s², which is considerably high.

Pozos-Estrada et al. [33] further proposed an ANN model to predict the PGA in one direction, employing the focal depth, hypocentral distance, and magnitude. One and two hidden layers were implemented in the FFBP algorithm. Using between 3 and 20 neurons with 1 hidden layer achieved optimal results. An RMSE of 1.1 cm/s² was attained for some PGA values. Similarly, Dhanya and Raghukanth [34] predicted the initial 26 points

of spectral acceleration from 13,552 shallow-type earthquakes. Furthermore, the ground motion parameters, such as PGA, were predicted. The ANN involved five neurons and one hidden layer. Additionally, the inputs were the shear velocity, distance to rupture, focal mechanism, magnitude, and logarithmic value. The focal mechanism was assigned a value between 1 and 3 depending on fault formation. A genetic algorithm was used for optimization; thus, exceptionally accurate results were obtained.

According to the literature, the predictions were presented for some earthquake-related parameters, not acceleration time series. This study explores a relatively new approach for real-time strong ground motion prediction. It uses DL models based on the first points to predict an earthquake signal's window. The DL algorithms are ANN, CNN, RNN, LSTM, and CNN-LSTM. As novel points are registered, the predictions are brought up-to-date for future signal windows. Metadata analysis, such as Principal Component Analysis (PCA) and K-means clustering, is also performed. The metadata identifies clusters in the metadata, followed by modeling using DL methods for the identified clusters.

Notably, obtaining helpful information requires an adequately long input window of reasonable length. That is because long sequences may experience gradient vanishing; hence, the expanding window is inappropriate in this study, as the window becomes excessively long after a while. Precisely, the points become greater than 100,000. Moreover, a sliding window approach with a size of 357 points (0.5 s) is used. The non-overlapping outputs are also obtained through the time shift/lag, equivalent to the forecast horizon. Additionally, as most forecasting problems are tackled through relatively few layers, it is advantageous to start with fewer layers initially. If the desired satisfactory performance still needs to be achieved, deeper networks may be employed.

2. Research Significance and Novelty

Predicting earthquakes ahead of time is currently a rather challenging feat. Instead, engineers aim to detect strong ground motions as they make time-sensitive decisions. For instance, The U.S. Geological Survey has developed Early Earthquake Warning (EEW) systems, such as ShakeAlert, in the Pacific Northwest and the U.S. West Coast [3]. However, these systems still require human supervision to achieve the desired optimal performance levels. Considering the short-notice nature of the early warnings, the utilization and usefulness of such systems still need to be improved. Moreover, the reviewed literature indicates a need for research to predict earthquake signals for real-time seismic protective applications, e.g., semi-active magneto-rheological (MR) dampers. It is well established that structural control devices, including MR dampers, suffer from performance degradation due to time-delay effects. This is attributed to the fact that the control force exerted by the damper requires time to develop through the different components and mechanisms of the system [35–37]. Some of the identified sources of time delays include (a) control electronics (power supply operation, data acquisition, and data processing of the sensor signal), (b) damper electrical or electromagnetic circuitry, and (c) damper mechanical components (electric current-driven modifications to the damping characteristics) [38]. As such, this research aims to employ DL to develop novel prediction models that facilitate high-performance real-time active or semi-active structural control implementations. It would be achieved via a continuously updated accurate forecast of the ground motion signals as they evolve in real time. Hence, the novelty of this work is developing Machine Learning models for ground motion prediction that can be employed in active or semi-active structures to minimize damage during an earthquake.

3. Methodology

This section describes the ground motion acceleration prediction methodology using DL algorithms. The database NGA-West2 has been introduced along with the data preprocessing process. Moreover, the details of the proposed models and their training have been explained.

3.1. Database

The NGA-West2 database generated by the Pacific Earthquake Engineering Research Center [39] is a collection of ground motion recordings from shallow crustal earthquakes in active tectonic regions of western North America. The Pacific Earthquake Engineering Research Center (PEER) compiled the dataset, which is widely used for developing and evaluating ground motion prediction models. The NGA-West2 dataset contains over 20,000 3-component accelerograms recorded at 154 stations during 68 earthquakes, with magnitudes ranging from 4.0 to 7.9. The earthquakes occurred in various tectonic settings, including strike-slip, reverse, and normal faults. The dataset also includes information on the earthquake source, site conditions, and ground motion intensity measures. The NGA-West2 dataset is a valuable resource for researchers and engineers working on earthquake ground motion prediction and seismic hazard assessment. The dataset has been used to develop a wide range of GMPs and has helped to improve our understanding of the factors that control ground motion variability.

The earthquake records in the NGA-West2 dataset exhibit a wide range of characteristics, including:

- Magnitude: The earthquakes in the dataset range in magnitude from 4.0 to 7.9.
- Distance: The recording stations are located at varying distances from the earthquake sources, ranging from a few kilometers to over 100 km.
- Tectonic setting: The earthquakes in the dataset occurred in various tectonic settings, including strike-slip, reverse, and normal faults.
- Site conditions: The recording stations are located on various sites, including rock, soil, and sediment.

The diversity of the earthquake records in the NGA-West2 dataset makes it a valuable resource for developing and evaluating ground motion prediction models. The dataset can be used to study the effects of magnitude, distance, tectonic setting, and site conditions on ground motion variability.

This study used the NGA-West2 dataset to train and evaluate our DL models for real-time earthquake signal prediction. We used the first 0.5 s of each accelerogram as input to our models and trained the models to predict the remaining ground motion. We believe that the NGA-West2 dataset is suitable for training and evaluating our models because it contains a large number of high-quality earthquake records from various tectonic settings and site conditions.

This study utilized the ground motion triaxial data for 599 earthquakes from the NGA-West2 database. Specifically, the data were for shallow crustal earthquakes, which have their hypocenter within the continental crust, coming from various locations. Moreover, 17,602 metadata records, which depict recording stations and events, were considered. The ranges of moment magnitude and closest distance were from 3.0 to 7.9 Mw and from 0.05 to 1533 km, respectively. A smaller amount of data was obtained for distances that exceeded 400 km. Additionally, a casual Butterworth filter was used to reduce the raw ground motion time series high- and low-frequency noise.

3.2. Data Preprocessing

The criteria for processing the metadata of 276 features were as follows: (a) station name and date and other unnecessary features were removed, (b) features that involved more than 50% missing data were banished, (c) numeric values to denote some properties, such as the co-seismic surface rupture: 0 = No, 1 = Yes, replace −999 with −1 (no measurement), were used, and (d) a mean of zero and variance were targeted through standardization.

The last dataset contained 17,602 accelerograms and 48 features for the metadata. Throughout this study, the measured unit for acceleration is the gravitational acceleration ($g = 9.81 \text{ m/s}^2$). A sampling frequency of 714.3 Hz, i.e., 1.4 ms, was maintained for each timestep through interpolation; thus, great precision was determined for the prediction.

The prediction performance was further improved via the additional training examples acquired per signal.

The time series was distributed/split into input and output pairs known as the window and forecast horizon, respectively, as displayed in Figure 1. After that, the sliding window approach was implemented, adding to the existing training examples of the models. Additionally, since the prediction horizons from several windows were not overlapping, the forecast horizon needed to shift the windows toward the right as time flows.

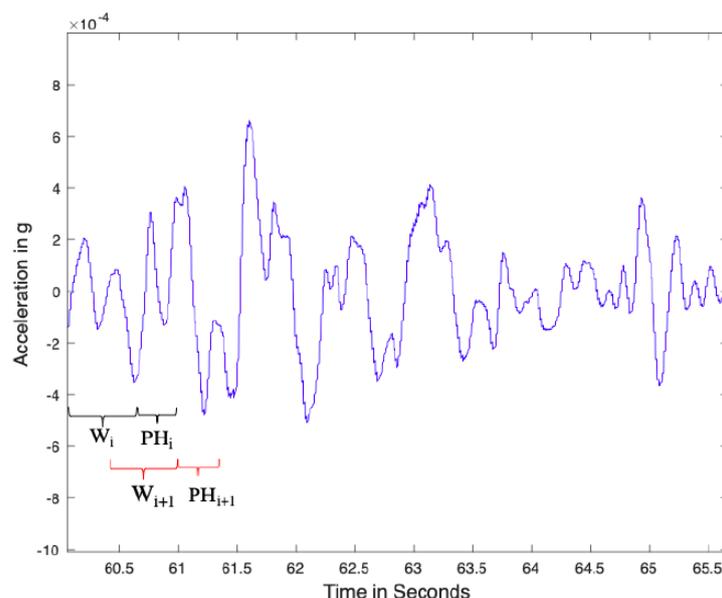


Figure 1. The sliding window approach.

In the following section, we will present the DL architectures and training details for the testing setups of the networks after presenting the metadata analysis.

3.3. Metadata Analysis

As stated in the former section, 48 features are involved in the record file for the NGA-West2 database. We decided to perform some preliminary analysis of the data. PCA was used to decrease the number of dimensions through a linear combination of the original parameters, followed by a K-means clustering algorithm to visualize if there were any clusters in the dataset. As shown in Figure 2, the dimensional space was decreased from 48 to 3 using the PCA algorithm for data visualization. Table 1 presents the most considerable coefficient for every linear combination of the variables with the most significant impact on the principal components. One of the most widely used parameters for earthquake description is the magnitude. Furthermore, the principal component illustrates other earthquake characteristics, such as the fault rupture width and the magnitude type. The magnitude type is a local, body wave, moment, or surface-wave magnitude [39]. Moreover, the second principal component, defining the strength of an earthquake at a particular location, would be its distance (epicentral distance, hypocentral distance, and Joyner–Boore distance). The epicentral distance is between the earthquake’s epicenter and the recording station. Additionally, the hypocentral distance is between the hypocenter or focus of the earthquake and the recording station. The Joyner–Boore distance is the distance through the recording station and the rupture surface’s surface projection. The third principal component expresses the time series of the ground motion, such as peak ground acceleration, velocity, and the lowest usable frequency.

As displayed in Figure 3, using the K-means clustering algorithm, the data were distributed into three discernible groups, consistent with the visual clustering. The groups applied to all 48 earthquake parameters, not particular ones, since PCA, a linear combination of all parameters, exhibited clustering. These clusters provided some sense of the

structure of the data. We performed this analysis to see if we needed to treat different classes of earthquakes for training and testing differently. However, since there were enough earthquakes in different groups, we let the supervised Machine Learning algorithms extract the information by themselves.

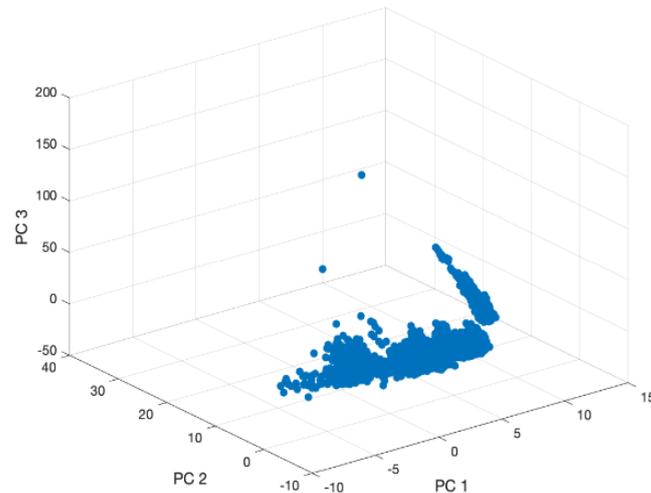


Figure 2. Dimensionality reduction to three dimensions.

Table 1. Variables with the greatest coefficients in each principal component.

PC1	PC2	PC3
Magnitude	Epicentral distance	Lowest usable frequency—V
Magnitude type	Joyner–Boore distance	PGV
Fault rupture width	Hypocentral distance	PGA

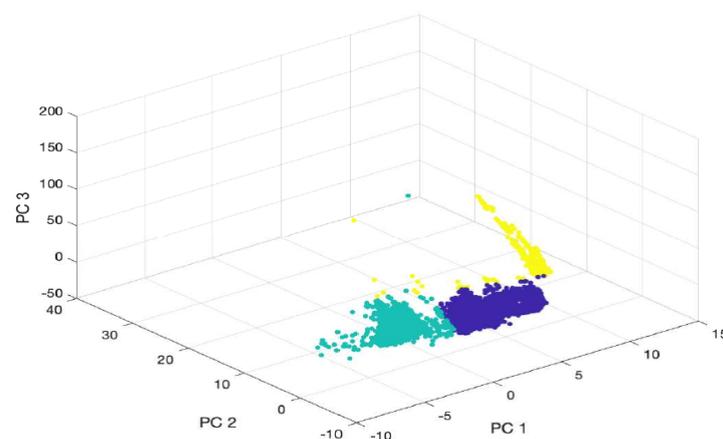


Figure 3. Three clusters in three dimensions from the PCA, shown in three different colors.

3.4. Supervised Machine Learning Algorithms

The following section presents the Machine Learning algorithms used in this work. We experimented with ANN, CNN, RNN, LSTM, and CNN-LSTM algorithms. We selected the ANN architecture as the baseline. We then compared its performance with CNN and other models that model the time axis.

3.4.1. ANN Architecture

In this study, ANN was considered the comparison baseline to compare the performance of the models. The ANN model comprises the input, hidden, and output layers. The first, second, and third axes from various axes were horizontal, horizontal, and vertical,

respectively, and were included in the input layer. The timesteps corresponding to the axes are the number of nodes within the input. Hence, the window size was set at 0.5 s or 357 points.

The hidden layer size for this study was the average of the sizes corresponding to the input and output. Every predicted timestep had an output neuron in the three axes. Moreover, several prediction horizons were tested.

Regularization was adopted because of the many parameters, and ANNs are susceptible to overfitting. A regularization value of 0.0001 was assigned to L2, while it was raised to 0.1 for prediction horizons of greater length, as the parameter number can be larger than the training examples. Additionally, if validation loss is not enhanced within five epochs, early stopping will cease the training. The Adaptive Gradient optimization algorithm (Adagrad) was used as the optimizer based on experiments. Figure 4 shows the network configuration.

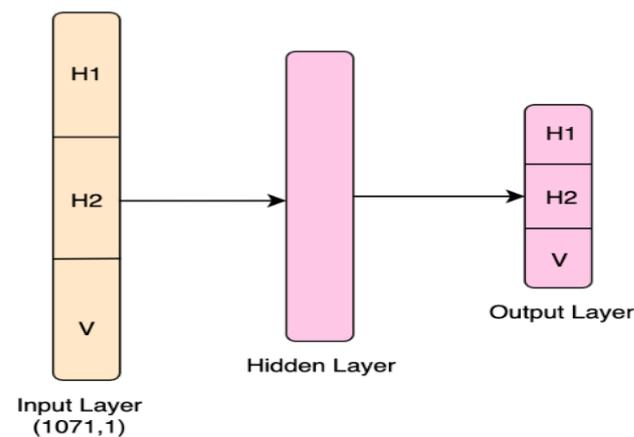


Figure 4. Illustration of the developed ANN model.

A learning rate scheduler was implemented to avoid excessive tuning efforts. As shown in Equation (1), the learning rate (lr) was set to be a significant value that decreased with each epoch. Fast learning occurred first; then, the weight optimization happened with training progress:

$$lr = \frac{lr_0}{1 + epoch * R_d} \quad (1)$$

where R_d is the rate of decay, set to 0.5, and lr_0 is the initial learning rate equal to 0.01, as determined by experimentation. Furthermore, a normalization layer and a batch size of 128 were implemented following the input and hidden layer for standardizing the batch. In other words, the batch becomes a mean and variance of zero and unit, respectively. That avoids the weight explosion, contributes to decreasing the number of epochs in training, and warrants training stability.

3.4.2. CNN Architecture

CNNs were employed for time series predictions due to their outstanding properties for extracting meaningful feature maps. The proposed model contains an input layer with three channels corresponding to three axes, as presented in Figure 5. A 1D-convolutional layer follows the input layer. The 1D-convolutional layer had a filter of size three. Additionally, a stride of 2 was used in the layer to reduce the output size; thus, the computations required were reduced. Additionally, a flattening layer changed the feature maps into one vector. It feeds it to a Fully Connected (FC) layer, allowing it to restrict the output size by varying the number of neurons.

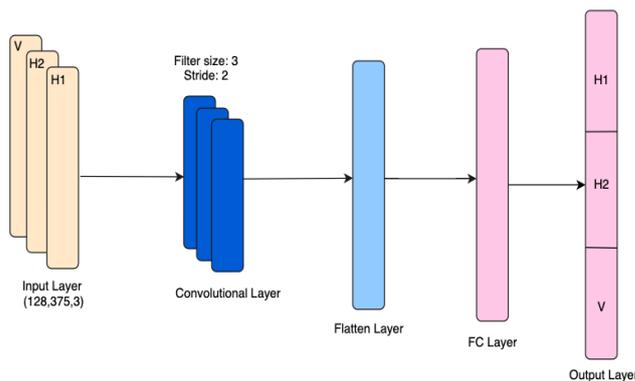


Figure 5. Illustration of the developed CNN model.

A single FC layer was provided ahead of the output layer to reduce the number of parameters encompassing 100 neurons. One timestep was produced for each output neuron for the three axes. The number of convolutional layers and the corresponding filters were chosen based on the losses during testing. Additionally, based on experimentation, the batch size was 128, and the initial learning rate was 0.01. Adaptive Moment Estimation (ADAM) was the chosen optimization algorithm, most commonly implemented for CNNs [18]. This algorithm banishes the vanishing learning rate and assists in advancing convergence. Nevertheless, it requires significant computational costs [40]. A five-epochs patience criteria for early stopping was utilized.

3.4.3. RNN Architecture

As illustrated in Figure 6, an input layer, an RNN layer of three units, and an output layer composed the RNN network. A channel with windows corresponding to every axis was included in the input. Channels 1 and 2 were for the horizontal axes, and channel 3 was for the vertical axis. This cut the computational cost and split up the axes, enabling the RNN to manage a multivariate instead of a univariate sequence input, similar to the ANN. Converting the layer back to a sequence is impossible, but it can be converted to a vector. The latter denotes the input summary fed to the FC layer.

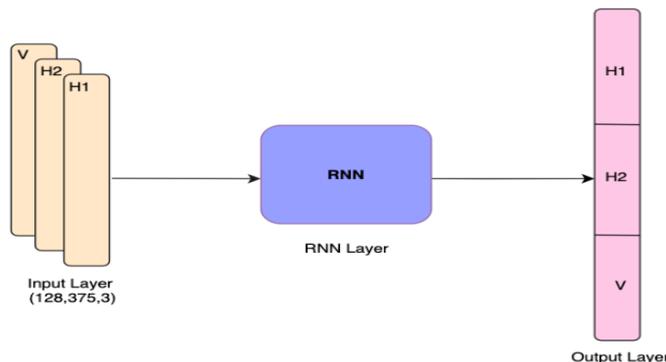


Figure 6. Illustration of the developed RNN model.

In addition, the FC layer represents the output of three partitions, i.e., forecast horizons, each on a distinct axis. Overfitting was avoided through early stopping with the five epochs. Along with the ADAM optimizer, a batch size of 128 and a learning rate of 0.005 were first assigned prior to the exponential decay.

3.4.4. LSTM Architecture

Figure 7 shows the architecture of the LSTM network, which was made up of an input, an LSTM with three units, and an output layer. All other settings were similar to the previous network.

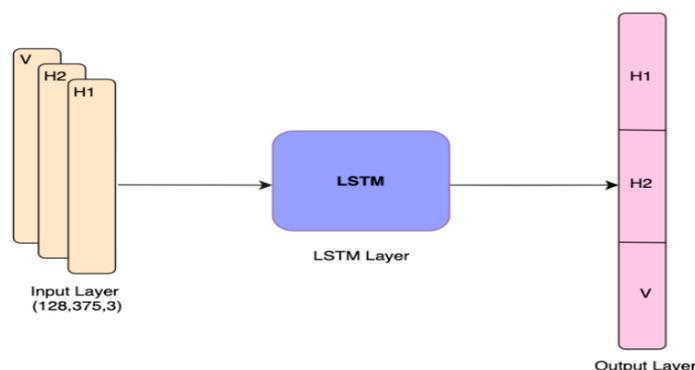


Figure 7. Developed LSTM model.

3.4.5. CNN-LSTM Architecture

Due to LSTM's ability to handle time dependency and recall information and CNN's ability to obtain information, CNN and LSTM were combined to bring advantages to predictions in the long term. Besides CNN's benefit of producing valuable input, it restricts input size. An input, 1D-convolutional, LSTM, and output layer form the CNN-LSTM model, as presented in Figure 8. After the 1D-convolutional layer, max-pooling reduces the feature's map size. Three and six are the filter size and number of units in the 1D-convolutional and LSTM layers, correspondingly. All other settings were similar to the ones in the previous network.

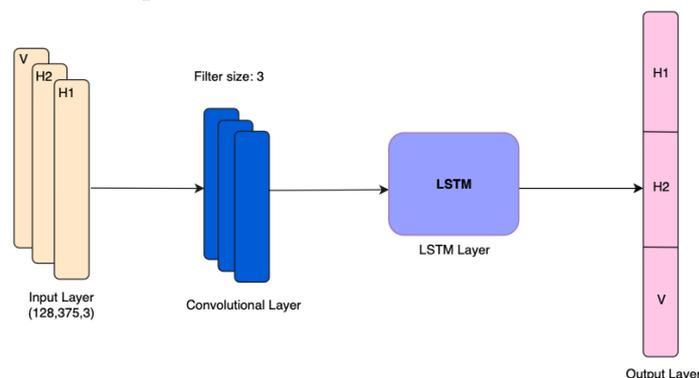


Figure 8. Illustration of the developed CNN-LSTM model.

3.5. Training Details

A Python 3.4 code was developed on an Ubuntu 18.04 operating system of Intel Core i9-10900X CPU (10 cores) with 62 GB of RAM. DL models were implemented based on the Keras library on GeForce GTX 1080 GPU and CUDA 10.2. One million, 100,000, and 420,000 samples were used in the training, validation, and testing, respectively. It was ensured that the earthquake data from this dataset were from different earthquakes in training, validation, and testing. A batch size of 128 and an epoch of 25 were used to train all the models, except CNN-LSTM. The details of training for all the models are presented in Table 2.

Table 2. Summary of the activation function and training, validation, and testing splits used for the models.

	ANN	CNN	RNN	LSTM	CNN-LSTM
Activation Function	ReLU-linear	Tanh-linear	Tanh-linear	Tanh-linear	Tanh-linear
No. of epochs	25	25	25	25	35
Training Data	1 M	1 M	1 M	1 M	1 M
Testing Data	420 k				
Validation Data	100 k				

Performance Evaluation Metrics

An appropriate loss function must be selected to evaluate the network's performance in predicting ground motion time series. Cosine similarity was the first method, documented in [41]. The method operates based on the measured similarity between the predicted output \hat{y} and actual output y vectors, as shown in Equation (2):

$$\text{sim}(y, \hat{y}) = \frac{y \cdot \hat{y}}{\|y\| \|\hat{y}\|} \quad (2)$$

where $\|y\|$ is the Euclidean norm of vector $y = (y_1, y_2, \dots, y_n)$, expressed as $\|y\| = \sqrt{\sum_{i=1}^n y_i^2}$. The cosine of the angle enclosed by the two vectors was estimated using the cosine similarity method. A zero value indicates dissimilarity or orthogonality. The angle becomes smaller as the value moves near one, making the vectors comparable and pointing to just the same. Nevertheless, the dot product equals zero if any vector is zero. Hence, the cosine similarity is also zero, which is tricky since the vectors might be co-equal. Because a significant part of an earthquake is zero, cosine similarity becomes inadequate for time series prediction.

Another method in the literature is the Root Mean Squared Error (RMSE), employed for performance evaluation by Cheng et al. [42] for a multi-step time series prediction. Geng [43] also applied RMSE to predict the seismic energy based on the added parameters and time series. Equation (3) conveys the computation for RMSE:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad (3)$$

where N is the total number of training examples, y_i is the ground-true points, and \hat{y}_i is the predicted points. RMSE was used to measure the predicted points' deviation from the actual points. Execution time is also an evaluation metric for a real-time system. It is vital to take less considerable time than the forecast horizon. This study aimed to ensure that the maximum delay was less than 10% of the forecast horizon.

4. Results and Discussion

This section illustrates and discusses the parameter selection for the models, prediction horizons, and superior-performing models. Following that, a summary of the results and execution time of the models are presented. Table 3 presents the RMSE values for all the DL algorithms at different prediction horizons.

Table 3. The RMSE (g) for the proposed models across different forecast horizons.

Forecast Horizon	ANN	CNN	RNN	LSTM	CNN-LSTM
1	1.35×10^{-3}	5.41×10^{-5}	1.96×10^{-5}	1.56×10^{-5}	7.21×10^{-5}
10	1.74×10^{-4}	3.32×10^{-5}	3.93×10^{-5}	8.43×10^{-6}	4.90×10^{-5}
50	4.71×10^{-4}	6.02×10^{-5}	4.52×10^{-5}	3.90×10^{-5}	4.65×10^{-5}
100	4.76×10^{-4}	4.75×10^{-5}	4.24×10^{-5}	4.75×10^{-5}	2.76×10^{-5}
200	7.01×10^{-3}	1.47×10^{-3}	1.51×10^{-3}	1.52×10^{-3}	1.53×10^{-3}

4.1. Forecast Horizon of Size 1

The RMSE test set acquired from 450,000 test examples was used to predict the results displayed at this point onwards in this study. The best-performing and baseline model, ANN, is also presented. With an input of 357 points (0.5 s), one point (1.4 ms) has been predicted in each axis for the ANN. The LSTM model performed the best among the models and had an RMSE of 1.56×10^{-5} g. Moreover, the RMSE was reduced by 99% compared to the baseline model of RMSE, which was 1.35×10^{-3} g. This is consistent with the expectation of LSTM being superb for short-term predictions. That is, since LSTM preserves the temporal nature of the data, i.e., memory. Furthermore, the RMSE was

reduced by 95% compared to the baseline for the CNN-LSTM model; thus, it was placed second after LSTM.

The maximum magnitude available in the database of NGA-West2 was 7.9 Mw for one point of a signal, and its baseline prediction is displayed in Figure 9a. Significant noise was observed in the baseline prediction. Moreover, as pointed out by the black box, a zoomed-in view of a 5 s interval, the baseline failed to keep up with the variations. That is, the baseline failed to learn or grasp the dynamic nature. Compared with the actual signal, the predictions had a higher magnitude in Figure 9b. However, the LSTM network prediction was nearly identical to the true acceleration in the three axes. Smooth and accurate modeling with slight variations was performed for the patterns.

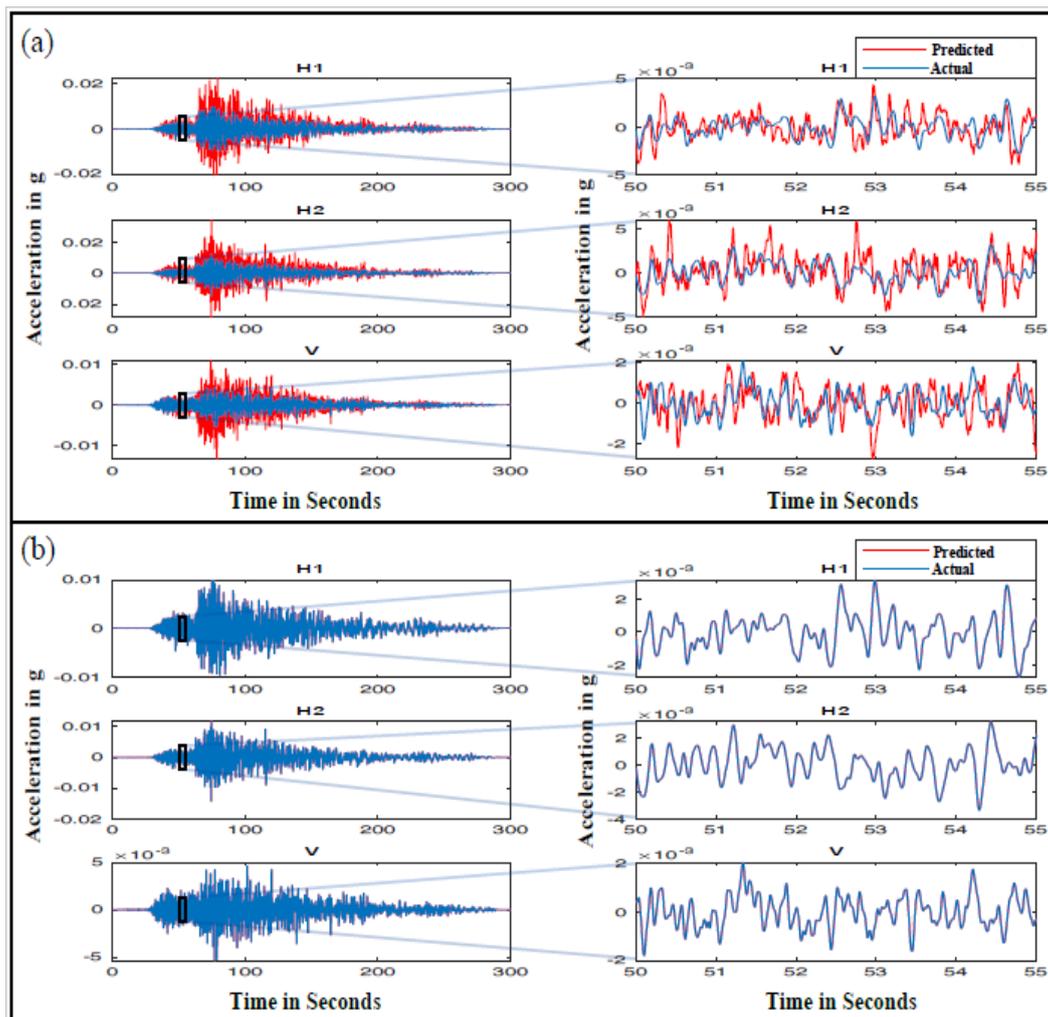


Figure 9. (a) ANN prediction vs. actual acceleration: the window size is 1, and the figures on the right side are a zoomed-in segment of the left figures. (b) LSTM network prediction vs. actual acceleration: the window size is 1, and the figures on the right side are a zoomed-in segment of the left figures.

4.2. Forecast Horizon of Size 10

For the forecast horizon of size 10, similar to that of size 1, the input window size was 357 points (0.5 s), and 10 points (0.014 s) were predicted in each axis for the forecast horizon of size 10. An RMSE of 8.43×10^{-6} g was determined for LSTM, making it superior to other models, with an improvement of 95% compared to the baseline with an RMSE of 1.74×10^{-4} g. The CNN-LSTM model achieved 72% less errors than the baseline, constituting the model with the second-highest RMSE.

The forecast horizon for 10 points of the 7.9 Mw signal was similar to that of 1 timestep, and its baseline prediction is displayed in Figure 10a. Baseline prediction was very noisy

because of the highly dynamic nature of the data. The prediction of the LSTM network shown in Figure 10b was almost the same as the actual acceleration in the three axes.

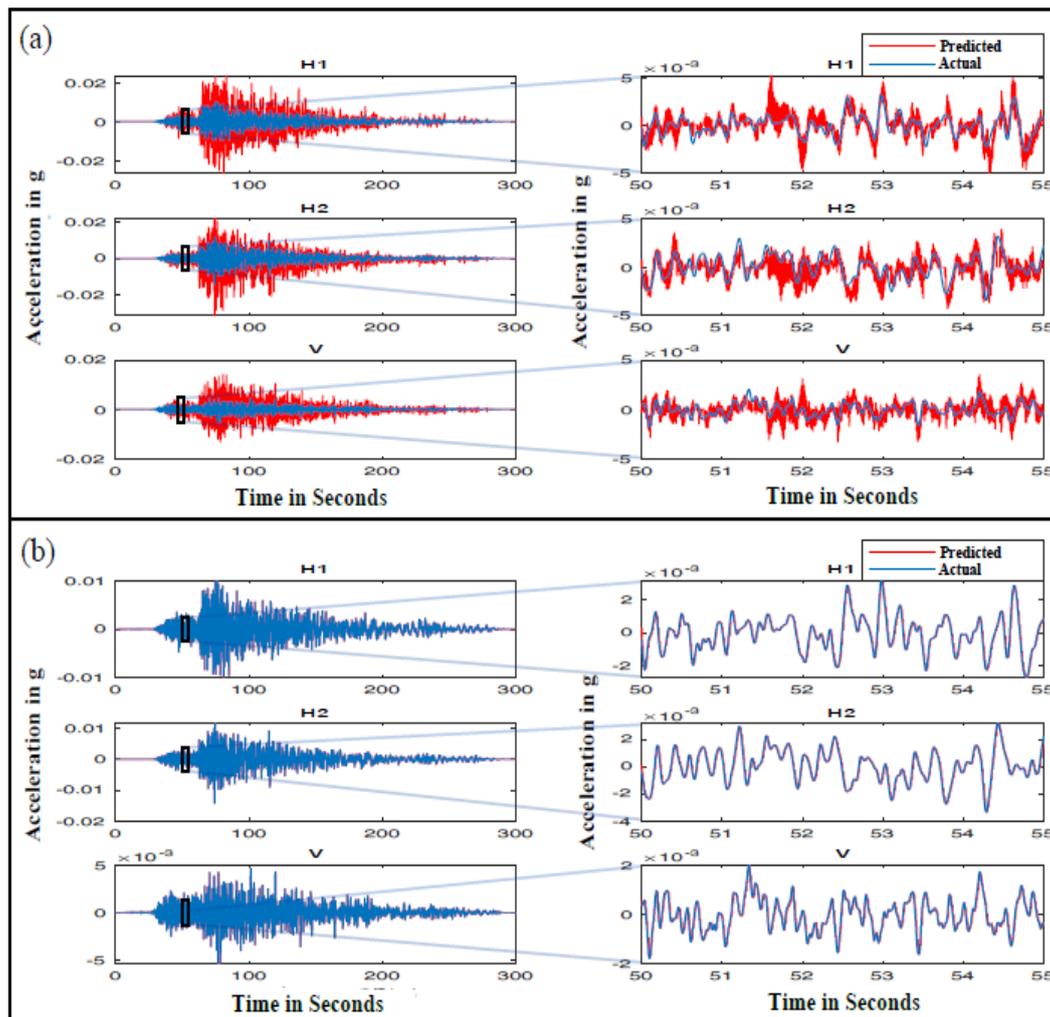


Figure 10. (a) ANN prediction vs. actual acceleration: the window size is 10, and the figures on the right side are a zoomed-in segment of the left figures. (b) LSTM network prediction vs. actual acceleration: the window size is 10, and the figures on the right side are a zoomed-in segment of the left figures.

4.3. Forecast Horizon of Size 50

For each axis, the input window was set at 357 points (0.5 s), and about 50 points (0.07 s) were predicted. An RMSE of 3.90×10^{-5} g was computed for the LSTM model, allowing it to perform best. Specifically, compared to the baseline model with an RMSE of 4.71×10^{-4} g, an improvement of 92% was determined. The model with the second highest RMSE was the CNN, which improved by 87% compared to the baseline. It is worth noting that performance improvement deteriorated with the growth of the forecast horizon.

For a magnitude 7.9 Mw signal, the baseline prediction is shown in Figure 11a for 50 points. Excessive noise was noticed in the prediction. Moreover, the predictions did not match the variations and resulted in a higher magnitude than the actual acceleration because of the highly dynamic nature of the data. Figure 11b illustrates prediction via the LSTM network. The prediction was about identical to the actual acceleration in the three axes. For 300 s of earthquake duration, which is the entire duration, LSTM well modeled the pattern. Nonetheless, since the prediction magnitude was somewhat lower than the signal of the actual earthquake, the predictions of the signal's minimums and maximums were not very accurate.

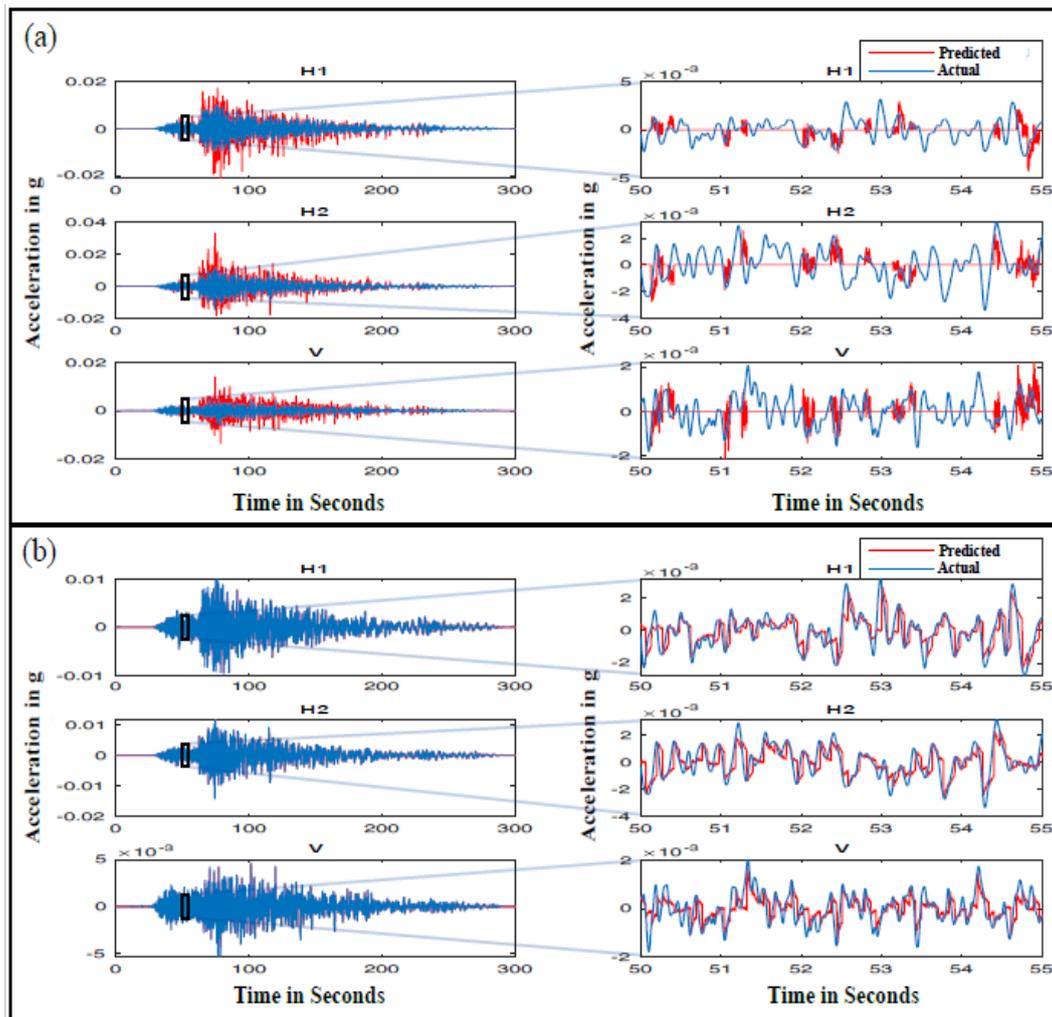


Figure 11. (a) ANN prediction vs. actual acceleration: the window size is 50, and the figures on the right side are a zoomed-in segment of the left figures. (b) LSTM network prediction vs. actual acceleration: the window size is 50, and the figures on the right side are a zoomed-in segment of the left figures.

4.4. Forecast Horizon of Size 100

The input window size was 357 points (0.5 s), and a prediction of 100 points (0.14 s) was performed on the triaxial data. The best performance was recognized for CNN-LSTM, which had an RMSE of 2.76×10^{-5} g. An RMSE of 4.76×10^{-4} g was disclosed for the baseline model, which improved by 94% for the CNN-LSTM model. The LSTM network's accuracy decreased with the increase in the forecast horizon because LSTM is ideal for shorter ranges. Furthermore, the LSTM and CNN exhibited the second-highest RMSE, with 90% less errors than the baseline.

The baseline prediction for the magnitude 7.9 Mw signal with 100 points is displayed in Figure 12a. Similar to earlier results, excessive noise existed in the baseline prediction. More extended zero regions of the baseline were noticed when zooming into the 5 s interval. Moreover, the predictions were determined to be more significant in magnitude than the actual acceleration. The predictions of the CNN-LSTM network were accurate with minimal variations, as shown in Figure 12b. Conversely, the minimums and maximums of the signal were not predicted accurately.

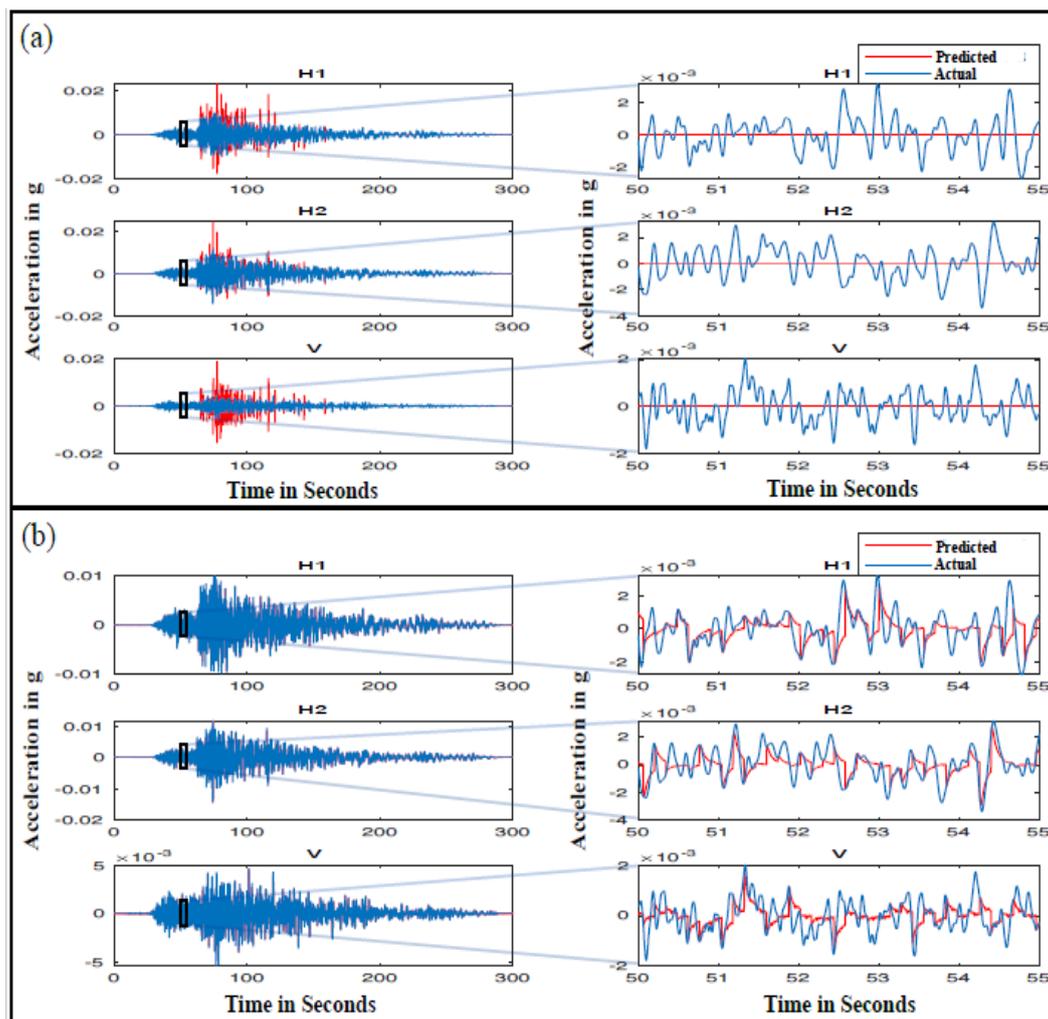


Figure 12. (a) ANN prediction vs. actual acceleration: the window size is 100, and the figures on the right side are a zoomed-in segment of the left figures. (b) CNN-LSTM prediction vs. actual acceleration: the window size is 100, and the figures on the right side are a zoomed-in segment of the left figures.

4.5. Forecast Horizon of Size 200

Again, 357 points (0.5 s) was the input window size, and 200 points (0.28 s) were predicted on each axis. The best performance was indicated for the CNN model, with an RMSE of 1.47×10^{-3} g. That indicated a performance improvement of 79% compared to the baseline, with an RMSE of 7.01×10^{-3} g. CNN-LSTM exhibited the second highest RMSE, with 78% less RMSE than the baseline. The baseline prediction for a signal with a 7.9 Mw magnitude for 200 points is shown in Figure 13a. Figure 13b illustrates excellent attenuation in magnitude for CNN predictions. The minimums and maximums of the signal lack accuracy, as fluctuations around zero were noticed, specifically on axis V.

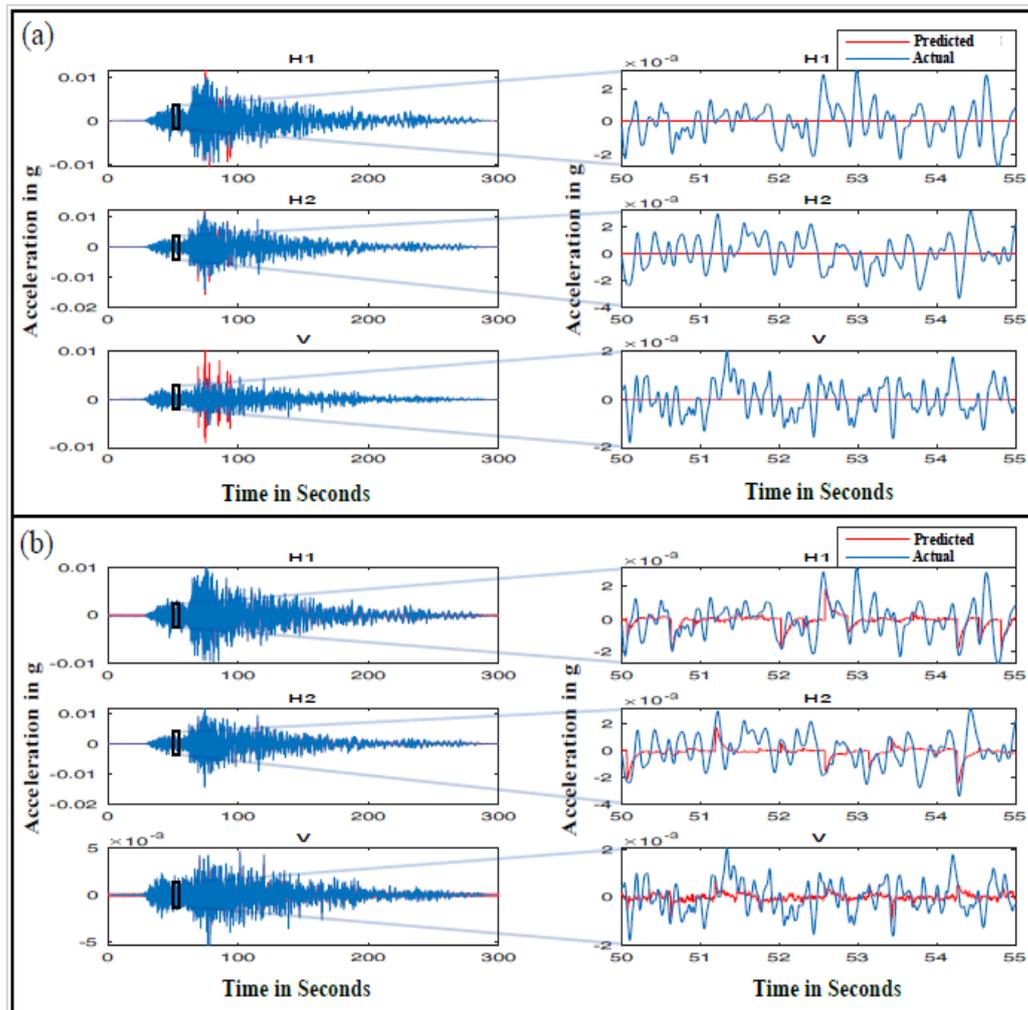


Figure 13. (a) ANN prediction vs. actual acceleration: the window size is 200, and the figures on the right side are a zoomed-in segment of the left figures. (b) CNN prediction vs. actual acceleration: the window size is 200, and the figures on the right side are a zoomed-in segment of the left figures.

4.6. Overall Performance

As a proposed solution, a sliding window algorithm was adopted to separate the input from the output earthquake acceleration pairs. Non-overlapping outputs from various windows were observed. The window size was determined as 357 points, and various predictions were made, from 1 to 200 points. The performance of the models over several forecast horizons is presented in Table 4. All models achieved acceptable accuracy (less than 0.01 g RMSE). However, the LSTM model attained excellent RMSE for a prediction horizon of 10 and had the best performance for short-term prediction from 1 to 50 points. This agrees with the expectation since LSTM is adequate for short-term prediction. A value of 14% of the input for 50 points as output was achieved, which is consistent with Qing and Niu's finding [14]. For an output-to-input ratio of 9%, the LSTM network was better than the shallow ANN by 18%. Compared to the baseline, LSTM achieved a 92% improvement. The LSTM's architecture was similar to that in [14]. The only difference is that 3 instead of 30 units that lasted for 100 epochs were used. The results validate that LSTM performs better than the shallow ANN for comparable output-to-input ratios.

Based on the output of 28% of the input for 100 points, the results were compared to that of Qin et al. [19], which had a forecast horizon of 24 h, an input of 72 h, and an output-to-input ratio of 33%. The RMSE was 36% and 20% lower for CNN-LSTM than ANN and LSTM, respectively. It was further evidenced that CNN-LSTM performed

better by 94% and 42% than ANN and LSTM, respectively. Additionally, the CNN-LSTM exhibited better performance than ANN when maintaining comparable output-to-input ratios. For a forecast horizon of 200 points, none of the models could obtain results of acceptable accuracy, and all the models had equivalent RMSE values, which became higher. Furthermore, the output-to-input ratio reached 56%. CNN outperformed the other models, whereas all the models performed better than the baseline model by at least 70%.

Table 4. Average time to produce forecast horizons for all the proposed models.

Model	Prediction Time (ms)
ANN	0.63
CNN	0.49
RNN	41.7
LSTM	3.12
CNN-LSTM	0.65

One other interesting comparison is to observe the results visually. One can see from Figures 9–13 that the predicted values stopped following the actual values as the prediction horizon increased. This effect was more pronounced in the case of ANN vs. the other models, where the results deviated significantly from the original values even when the prediction horizon was 1. However, as outlined earlier, the other models continued to show an acceptable performance for a much higher prediction horizon.

The reason for the markedly worse performance of ANNs is perhaps the lack of implicit dependence on time. This dependence on time was shown in the RNN, LSTM, and CNN-LSTM models. This was also shown in the performance of these models. Hence, for processes dependent on time, one must use the models that model the time axis. In addition, the best way to use hyperparameters is by using a validation set and experimenting with a range of values to discover the best ones.

The real-time aspect is a further vital metric for evaluating the performance of models. The average prediction time for the models is displayed in Table 4. Based on the results, CNN had the shortest time, requiring minimal computations and entailing 0.5 ms to generate a prediction. The CNN model provided superior real-time performance (less than 1 ms), as did the CNN-LSTM and ANN models. On the other hand, RNN required the longest time, about 41.7 ms, to make a prediction. Note that this time will be different for different pieces of hardware. Hence, though the precise values of time may not be important, it is insightful to compare different algorithms with each other on the same set of hardware. Advanced hardware, which efficiently parallelizes computations, can produce predictions more rapidly.

5. Conclusions

This work aimed at achieving accurate real-time predictions for triaxial strong ground motions via multiple DL algorithms. The earthquake records used were from the NGA-West2 dataset from the PEER center, which includes records from worldwide sources. Earthquakes with a magnitude (M_w) between 3.0 and 7.9 of shallow crustal type were considered. PCA was utilized as a first step to reduce the parameters in the metadata associated with the earthquake records. Subsequently, K-means clustering was performed based on the identified principal components. The 17,602 accelerograms used were designated to 3 clusters. The training and testing datasets appropriately represented signals based on three clusters. The training and testing datasets were disjointed so that no common earthquake records from the training set could be reused in the testing set. The accelerograms had different sampling times, so resampling to a uniform sampling rate (714 Hz for a 1.4 ms sampling time) was performed to facilitate appropriate processing. The input was 357 points long, using a sliding window approach. Multiple non-overlapping forecast horizons were investigated (1, 10, 50, 100, and 200 points). Different ANN architectures were trained and tested: CNNs, RNNs, LSTMs, and CNN-LSTMs. The utilized training

methodology applied various aspects of unsupervised and supervised learning. ANN served as the baseline benchmark for the prediction performance of the other networks. One million input/output sequence pairs were employed for training. A small subset of 50,000 sample points was used to optimize the parameters of each model. The input was fixed to 0.5 s. The models' performance was tested for different prediction horizons. More specifically, the horizons ranged from 1 to 200 points. Moreover, a sliding window approach was used to acquire non-overlapping prediction horizons.

For time series predictions, the general performance was related to similar studies. It was indicated that the best performance was exhibited by similar models in this and other studies. However, this research resulted in a better improvement than the ANN baseline. Moreover, the LSTM model had the best performance for short-range prediction, with a prediction horizon of 10 points. An RMSE of 8.43×10^{-6} g was determined, which signified a 95% improvement in performance compared to the baseline with an RMSE of 1.74×10^{-4} g. In agreement with experience and intuition, the LSTM network demonstrated superior predictions in the short term. This could be attributed to the fact that LSTM retains short-term and long-term memories that represent the temporal nature and features of the data series. In addition, the prediction time for the CNN model was 0.5 ms on an Intel Core i9-10900X CPU (10 cores), making it the fastest model. It is worth noting that the ANN, CNN, and CNN-LSTM models demonstrated real-time performance (0.01 s) during prediction. That said, the other models are believed to be capable of producing faster predictions using additional computational resources, such as CPU cores and GPUs. The models allow for the implementation of real-time structural control responses via intelligent seismic protection systems (e.g., magneto-rheological (MR) dampers).

Overall, the authors achieved accurate real-time predictions for triaxial ground motions by developing and testing multiple DL models, with the LSTM model exhibiting the best performance for short-term prediction and the CNN model demonstrating the fastest computational speed. These findings have implications for the implementation of intelligent seismic protection systems.

Author Contributions: Conceptualization, M.A. and U.T.; methodology, M.A. and U.T.; software, S.T.; validation, S.T., U.T. and M.A.; formal analysis, S.T., U.T. and M.A.; investigation, S.T., M.A. and U.T.; resources, M.A., S.T. and U.T.; data curation, S.T.; writing—original draft preparation, S.T.; writing—review and editing, M.A. and U.T.; visualization, S.T.; supervision, U.T. and M.A.; project administration, M.A. and U.T.; funding acquisition, U.T. and M.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was financially supported, in part, by the American University of Sharjah (AUS) through the Open-Access Program (OAP), with grant number OAP24-CEN-097 and the professional development grant from the College of Engineering at the AUS.

Data Availability Statement: All data, models, or code generated or used during the study can be made available on request with approval from the American University of Sharjah, provided the follow-up studies are completed.

Acknowledgments: The second author is grateful for the financial support received as a Graduate Teaching Assistant (GTA) from the Mechatronics Masters' program at the American University of Sharjah (AUS). This paper represents the opinions of the authors and does not mean to represent the position or opinions of AUS. Special thanks to Florence Wacheux for her input in the language review of an earlier manuscript draft. The data used in this work were from the NGA-West2 database. Most of the work was conducted during the second author's MS thesis research. For further details, the readers are referred to [44].

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Cassidy, J.F. Earthquake. In *Encyclopedia of Natural Hazards*; Bobrowsky, P.T., Ed.; Springer: Dordrecht, The Netherlands, 2013; p. 208.
2. Crone, A.J. The geology of earthquakes. *Seismol. Res. Lett.* **1997**, *68*, 778–779. [[CrossRef](#)]

3. Burkett, E.R.; Given, D.D.; Jones, L.M. *ShakeAlert—An Earthquake Early Warning System for the United States West Coast*; US Geological Survey: Reston, VA, USA, 2014.
4. Given, D.D.; Cochran, E.S.; Heaton, T.; Hauksson, E.; Allen, R.; Hellweg, P.; Vidale, J.; Bodin, P. *Technical Implementation Plan for the ShakeAlert Production System: An Earthquake Early Warning System for the West Coast of the United States*; US Geological Survey: Reston, VA, USA, 2014.
5. Hipel, K. *Time Series Modelling of Water Resources and Environmental Systems*; Elsevier: New York, NY, USA, 1994; Chapter 2; pp. 63–64.
6. Raicharoen, T.; Lursinsap, C.; Sanguanbhokai, P. Application of critical support vector machine to time series prediction. In Proceedings of the 2003 International Symposium on Circuits and Systems, Bangkok, Thailand, 25–28 May 2003; Volume 5, pp. 741–744.
7. Babusiak, B.; Mohylová, J. The EEG signal prediction by using neural network. *Adv. Electr. Electron. Eng.* **2008**, *7*, 342–345.
8. Coelho, V.; Coelho, I.; Coelho, B.; Souza, M.; Guimarães, F.; da Luz, S.E.; Barbosa, A.; Coelho, M.; Netto, G.; Costa, R.; et al. EEG time series learning and classification using a hybrid forecasting model calibrated with GVNS. *Electron. Notes Discret. Math.* **2017**, *58*, 79–86. [[CrossRef](#)]
9. Harrou, F.; Zeroual, A.; Hittawe, M.; Sun, Y. *Road Traffic Modeling and Management: Using Statistical Monitoring and Deep Learning*; Elsevier: Amsterdam, The Netherlands, 2022.
10. Hittawe, M.; Langodan, S.; Beya, O.; Hoteit, I.; Knio, O. Efficient SST prediction in the Red Sea using hybrid deep learning-based approach. In Proceedings of the IEEE International Conference on Industrial Informatics (INDIN), Perth, Australia, 25–28 July 2022; pp. 107–114.
11. Afzal, S.; Ghani, S.; Hittawe, M.; Rashid, S.; Knio, O.; Hadwiger, M.; Hoteit, I. Visualization and Visual Analytics Approaches for Image and Video Datasets: A Survey. *ACM Trans. Interact. Intell. Syst.* **2023**, *13*, 1–41. [[CrossRef](#)]
12. Selvin, S.; Vinayakumar, R.; Gopalakrishnan, E.A.; Menon, V.K.; Soman, K.P. Stock price prediction using LSTM, RNN and CNN-sliding window model. In Proceedings of the International Conference on Advances in Computing, Communications and Informatics (ICACCI), Bangalore, India, 19–22 September 2018; pp. 1643–1647.
13. Romazanov, A.; Zakharov, A.; Zakharova, I. Temperature prediction in a public building using artificial neural network. In Proceedings of the 8th Scientific Conference on Information Technologies for Intelligent Decision Making Support (ITIDS), Ufa, Russia, 6–9 October 2020; Atlantis Press: Amsterdam, The Netherlands, 2020; pp. 30–34.
14. Qing, X.; Niu, Y. Hourly day-ahead solar irradiance prediction using weather forecasts by LSTM. *Energy* **2018**, *148*, 461–468. [[CrossRef](#)]
15. Khan, S.; Rahmani, H.; Shah, S.A.A.; Bennamoun, M.; Medioni, G.; Dickinson, S. *A Guide to Convolutional Neural Networks for Computer Vision*; Synthesis Lectures on Computer Vision; Springer: Cham, Switzerland, 2018; Volume 8, pp. 1–207.
16. Young, T.; Hazarika, D.; Poria, S.; Cambria, E. Recent trends in deep learning based natural language processing. *IEEE Comput. Intell. Mag.* **2018**, *13*, 55–75. [[CrossRef](#)]
17. Hussain, D.; Hussain, T.; Khan, A.; Naqvi, S.; Jamil, A. A deep learning approach for hydrological time-series prediction: A case study of Gilgit river basin. *Earth Sci. Inform.* **2020**, *13*, 915–927. [[CrossRef](#)]
18. Livieris, I.E.; Pintelas, E.G.; Pintelas, P.E. A CNN–LSTM model for gold price time-series forecasting. *Neural Comput. Appl.* **2020**, *32*, 17351–17360. [[CrossRef](#)]
19. Qin, D.; Yu, J.; Zou, G.; Yong, R.; Zhao, Q.; Zhang, B. A novel combined prediction scheme based on CNN and LSTM for urban PM2.5 concentration. *IEEE Access* **2019**, *7*, 20050–20059. [[CrossRef](#)]
20. Boore, D. Stochastic simulation of high-frequency ground motions based on seismological models of the radiated spectra. *Bull. Seismol. Soc. Am.* **1983**, *73*, 1865–1894.
21. Gülerce, Z.; Kamai, R.; Abrahamson, N.A.; Silva, W.J. Ground motion prediction equations for the vertical ground motion component based on the NGA-W2 database. *Earthq. Spectra* **2017**, *33*, 499–528. [[CrossRef](#)]
22. Wyss, M. Ten years of real-time earthquake loss alerts. In *Earthquake Hazard, Risk and Disasters*; Academic Press: Boston, MA, USA, 2014; Chapter 9; pp. 143–165.
23. Ramirez, J.; Meyer, F. Machine learning for seismic signal processing: Phase classification on a manifold. In Proceedings of the 10th International Conference on Machine Learning and Applications and Workshops, Washington, DC, USA, 18–21 December 2011; Volume 1, pp. 382–388.
24. Li, Z.; Meier, M.A.; Hauksson, E.; Zhan, Z.; Andrews, J. Machine learning seismic wave discrimination: Application to earthquake early warning. *Geophys. Res. Lett.* **2018**, *45*, 4773–4779. [[CrossRef](#)]
25. Böse, M.; Wenzel, F.; Erdik, M. Preseis: A neural network-based approach to earthquake early warning for finite faults. *Bull. Seismol. Soc. Am.* **2008**, *98*, 366–382. [[CrossRef](#)]
26. Kuyuk, H.; Susumu, O. Real-time classification of earthquake using deep learning. *Procedia Comput. Sci.* **2018**, *140*, 298–305. [[CrossRef](#)]
27. Adeli, H.; Panakkat, A. A probabilistic neural network for earthquake magnitude prediction. *Neural Netw. Off. J. Int. Neural Netw. Soc.* **2009**, *22*, 1018–1024. [[CrossRef](#)]
28. Rouet-Leduc, B.; Hulbert, C.; Lubbers, N.; Barros, K.; Humphreys, C.; Johnson, P. Machine learning predicts laboratory earthquakes. *Geophys. Res. Lett.* **2017**, *44*, 9276–9282. [[CrossRef](#)]

29. Chakraverty, S.; Gupta, P.; Sharma, S. Neural network-based simulation for response identification of two-storey shear building subject to earthquake motion. *Neural Comput. Appl.* **2010**, *19*, 367–375. [[CrossRef](#)]
30. Kerh, T.; Ting, S. Neural network estimation of ground peak acceleration at stations along Taiwan high-speed rail system. *Eng. Appl. Artif. Intell.* **2005**, *18*, 857–866. [[CrossRef](#)]
31. Arjun, C.; Kumar, A. Neural network estimation of duration of strong ground motion using Japanese earthquake records. *Soil Dyn. Earthq. Eng.* **2011**, *31*, 866–872. [[CrossRef](#)]
32. Günaydın, K.; Günaydın, A. Peak ground acceleration prediction by artificial neural networks for northwestern Turkey. *Math. Probl. Eng.* **2008**, *2008*, 919420. [[CrossRef](#)]
33. Pozos-Estrada, A.; Gomez, R.; Hong, H. Use of neural network to predict the peak ground accelerations and pseudo spectral accelerations for Mexican inslab and interplate earthquakes. *Geofísica Int.* **2014**, *53*, 39–57. [[CrossRef](#)]
34. Dhanya, J.; Raghukanth, S. Ground motion prediction model using artificial neural network. *Pure Appl. Geophys.* **2017**, *175*, 1035–1064. [[CrossRef](#)]
35. Cha, Y.; Agrawal, A.; Dyke, S. Time delay effects on large-scale MR damper based semi-active control strategies. *Smart Mater. Struct.* **2013**, *22*, 015011. [[CrossRef](#)]
36. Zheng, J.; Li, Z.; Koo, J.; Wang, J. Analysis and compensation methods for time delays in an impact buffer system based on magneto-rheological dampers. *J. Intell. Mater. Syst. Struct.* **2015**, *26*, 690–700. [[CrossRef](#)]
37. Nagamatsu, S.; Shiraishi, T. A simple and novel control strategy for semi-active vibration suppression by a magneto-rheological damper. *J. Intell. Mater. Syst. Struct.* **2021**, *33*, 811–821. [[CrossRef](#)]
38. Occhiuzzi, A.; Spizzuoco, M.; Serino, G. Experimental analysis of magneto-rheological dampers for structural control. *Smart Mater. Struct.* **2003**, *12*, 703–711. [[CrossRef](#)]
39. Ancheta, T.D.; Darragh, R.B.; Stewart, J.P.; Silva, E.S.A.W.J.; Chiou, B.S.; Wooddell, K.E.; Graves, R.W.; Kottke, A.R.; Boore, D.M.; Kishida, T.; et al. *PEER NGA-West2 Database (04)*; Rep. no. 2013/03; Pacific Earthquake Engineering Research Center: Berkeley, CA, USA, 2013.
40. Kingma, D.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2017**, arXiv:1412.6980.
41. Han, J.; Kamber, M.; Pei, J. *Data Mining: Concepts and Techniques*, 3rd ed.; Morgan Kaufmann Publishers Inc.: Burlington, CA, USA, 2011; pp. 77–78.
42. Cheng, H.; Tan, P.-N.; Gao, J.; Scripps, J. Multistep-ahead time series prediction. In *Advances in Knowledge Discovery and Data Mining*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 765–774.
43. Geng, Y.; Su, L.; Jia, Y.; Han, C. Seismic events prediction using deep temporal convolution networks. *J. Electr. Comput. Eng.* **2019**, *2019*, 7343784. [[CrossRef](#)]
44. Tellab, S. Machine Learning Based Real-Time Earthquake Signal Prediction. Master's Thesis, Department of Mechatronics Engineering, American University of Sharjah, Sharjah, United Arab Emirates, 2021.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.