

Article

A Bi-LSTM Attention Mechanism for Monitoring Seismic Events—Solving the Issue of Noise & Interpretability

Nimra Iqbal ^{1,*}, Izzatdin Bin Abdul Aziz ¹  and Muhammad Faisal Raza ²

¹ Department of Computer & Information Sciences, Universiti Teknologi PETRONAS, Seri Iskandar 32610, Malaysia; izzatdin@utp.edu.my

² Department of Electrical & Electronic Engineering, Universiti Teknologi PETRONAS, Seri Iskandar 32610, Malaysia; muhammad_24009644@utp.edu.my

* Correspondence: nimra_24005120@utp.edu.my

Abstract

The nonlinearity and the extreme variability of seismic signals makes the detection of earthquakes difficult. Although the conventional deep-learning models can be used to extract useful features, they cannot be used in early-warning systems due to their non-interpretability. In this study, a Bidirectional Long Short-Memory network with an attention system (Bi-LSTM-Attn) is proposed to detect seismic events using the ConvNetQuake dataset. To improve the quality of data, the entire preprocessing pipeline, such as signal filtering, segmentation, normalization, and noise reduction is employed. The model was optimized using hyperparameter tuning of sequence length, learning rates, and attention weighting to achieve the best number of true-positive detections and a minimum number of false alarms. The accuracy, precision and recall, F1-score, MSE, and ROC curves were used to assess the performance and the attention weight visualization allowed interpreting the model. It is proven through experiments that the Bi-LSTM-Attn model provides more credible and comprehensible forecasting in relation to baseline LSTM and GRU models. Making the high-accuracy classification and the transparent decision behavior, the approach will increase the level of trust to the automated seismic surveillance, as well as help to build the reliable global networks of earthquake early-warnings.

Keywords: machine learning; attention mechanism; seismic event detection; Distributed Acoustic Sensing (DAS)-based seismic data; deep learning

1. Introduction

The occurrence of earthquakes is commonly known as a complicated nonlinear geophysical process related to long-range correlations and scaling characteristics of the crust of the Earth. Over decades, the trends of seismicity, as well as mechanisms of the preparation of earthquakes, have been studied with statistical and physics-based analyses. Classical methods consist of the Gutenberg–Richter power-law relation of magnitude–frequency dist.

Contribution [1], Natural Time analysis of identifying critical dynamics before large events [2], and fractal-based signal analysis methods, including Detrended Fluctuation Analysis (DFA) and Multifractal DFA (MFDFA) [3,4]. These techniques have provided valuable understanding on the self-organized criticality and fractal organization of seismic systems which has been used to interpret the idea that seismicity is scale-invariant. Nevertheless, they are primarily seismicity statistics methods or long-run correlation methods and do not scale to automated classification of waveforms at the waveform level in continuous real-time applications.



Academic Editors: Zhipeng Wu and Daniel Asante Otchere

Received: 8 January 2026

Revised: 19 February 2026

Accepted: 25 February 2026

Published: 26 March 2026

Copyright: © 2026 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 \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

As seismic networks expand quickly, and new dense sensing technologies like Distributed Acoustic Sensing (DAS) produce data in large volumes, continuous waveform data is now generated in large volumes and cannot be inspected manually. Therefore, recent research has used more and more methods of artificial intelligence and deep learning to detect seismic events automatically, pick their phase, and classify them [5,6]. Convolutional neural networks and Transformer-based models have shown good performance with large-scale monitoring such as ConvNetQuake in detecting events. Long Short-Term Memory (LSTM) and Bidirectional LSTM networks are recurrent architectures that are especially well-suited to sequential seismic signals because they can capture the temporal relationship [7]. Here, deep learning is not used to substitute for existing physical seismology, but as a tool of operations in the field of real-time interpretation of waveforms, which allows continuous monitoring and real-time identification of events in large volumes of data [8].

To address this drawback, attention mechanisms have been recently added to seismic deep-learning models. By concentrating on the most pertinent parts of the waveform, e.g., P-wave beginning or intense shaking, attention enhances both the reliability and interpretability of detection. Detection of earthquakes, magnitude prediction, prediction of peak ground velocity and structural response monitoring have seen better performance by attention-based recurrent models [9–11]. Although these are the improvements, there are two significant challenges. To begin with, seismic data is generally characterized by extreme class imbalance, with most earthquakes small and a few large ones, making it prone to models biased towards small earthquakes [12–14]. Second, the deep-learning models are frequently black boxes, which diminishes the confidence in such essential applications as early warning systems [15].

It is necessary to underline that machine-learning and deep-learning algorithms are not focused on the replacement of physical and statistical seismology. Classical methods (natural time analysis, power-law scaling, fractal analysis) result in physical explanation of the earthquake preparation process and crustal processes. Conversely, the suggested deep-learning framework is fashioned as an operational co-analysis application that takes large sets of continuous waveform recordings and can be used to automatically identify events. Thus, this study is not aimed at replacing physical models but supplementing them by providing real-time classification of waveforms in large seismic records where the manual classification method cannot practically be applied.

This paper presents a Bidirectional Long Short-Term Memory network with attention mechanism (Bi-LSTM-Attention) to enhance reliability and interpretability of seismic events detection. The model is used in combination with systematic preprocessing, such as filtering and standard normalization, to increase the signal-to-noise ratio and stabilize training. The given model is contrasted with such baseline architectures as LSTM [16–18], Stacked LSTM, Gated Recurrent Unit (GRU), and CNN-Bi-LSTM with the help of various evaluation metrics such as accuracy, precision, recall, F1-score, and mean squared error [19].

ConvNetQuake data was picked for three reasons. It is a standardized benchmark dataset, first, which is extensively utilized in recent seismic deep-learning research and provides the opportunity to objectively compare it with the existing approaches. Second, it has earthquake event and background recordings, which enables the assessment of the performance of false alarms [20]. Third, it offers three-component seismic waveforms (vertical, north–south and east–west), which are appropriate in temporal sequence modeling. The earthquake catalog that is related to it consists of magnitude, hypocentral depth, and epicentral location based on the regional seismic network [21,22], which is consistent with the reference seismic catalog, and not artificial signals. Figure 1 illustrates the ConvNetQuake dataset distributions against the magnitudes.

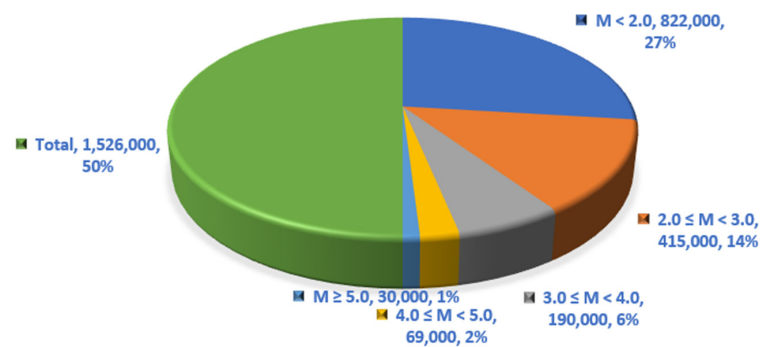


Figure 1. Distribution of earthquake events across different magnitude ranges in the dataset is dominated by low-magnitude events ($M < 3.0$), reflecting real-world seismic activity and highlighting the class imbalance challenge addressed in this study.

The ConvNetQuake dataset includes several seismic events of different sizes. A pie chart shows how these events are spread across different magnitude groups.

1. $M < 2.0$ (54.7%), Most of the events are very small and are called microseismic tremors. These are usually not felt by people, but they are important for teaching models to recognize weak seismic signals. Because there are so many of them, they help the dataset to be better for identifying small earthquakes.
2. $2.0 \leq M < 3.0$ (28.3%), These are small earthquakes that some people may feel nearby. They are common and important for building models that can detect minor but noticeable shaking.
3. $3.0 \leq M < 4.0$ (11.0%), These are moderate earthquakes that can be felt over wider areas. They are less common but still important because they indicate when seismic activity becomes more significant.
4. $4.0 \leq M < 5.0$ (4.1%), These earthquakes are stronger and can cause damage in certain areas. There are not as many of them, which makes it harder to train models to recognize them accurately. Special techniques are required to address this imbalance.
5. $M \geq 5.0$ (1.9%), These earthquakes are rare but very destructive. Although they are few in number, they are important for studying earthquakes and preparing for disasters. They are key to developing models that can detect large events.

The primary research gap that has been tackled in this study is the absence of reliable and interpretable seismic detection models that can process the noisy and imbalanced data of the waveforms and achieve good performance in classification. Thus, the objectives of this paper are (i) to determine whether Bi-LSTM-Attention is more likely to perform better than traditional deep-learning models in terms of detection, (ii) to determine how preprocessing affects model robustness, and (iii) to conduct a comprehensive comparative analysis by using a standardized seismic benchmark dataset.

Following the above limitations, this study aims at exploring the validity of deep recurrent neural networks with an attention mechanism to detect seismic events on continuous waveform. The paper evaluates (i) the ability of a Bi-LSTM model to produce historical dependencies in seismic signals (ii) whether attention mechanism improves an experience of interpretability with informative time windows of the waveform, and (iii) how robust the approach is against real catalog-labeled data on the ConvNetQuake dataset. The method proposed should be used as an aid in constant seismic monitoring and not substituting physical seismological analysis.

In this research, Section 1 describes the introduction, Section 2 outlines the approach to be used in the research, Section 3 outlines and discusses the methodology, and Section 4 discusses the results and Section 5 closes the research study with future research recommendations.

2. Supervised Learning Algorithm

Figure 2 is the example of supervised learning applied to DAS-based seismic data.

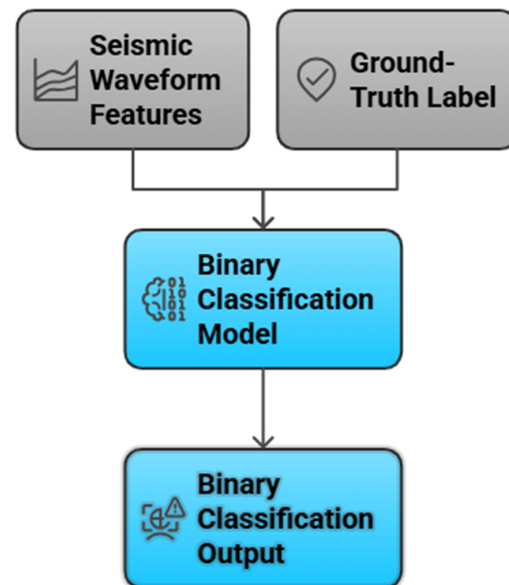


Figure 2. Supervised learning model applied in this research. The features of seismic waveforms (X) obtained through ConvNetQuake data set are matched to ground-truth data (y) with the 0 value of non-seismic noise and 1 value of earthquake events.

2.1. Long Short-Term Memory (LSTM)

The LSTM units are usually trained using a method called supervised learning. These supervised LSTM models learn from the data to recognize patterns and make forecasts or decisions. So, when you use LSTM in a model, it is usually part of a learning process where the model is being taught with labeled examples. Figure 3 shows the architecture of the LSTM model.

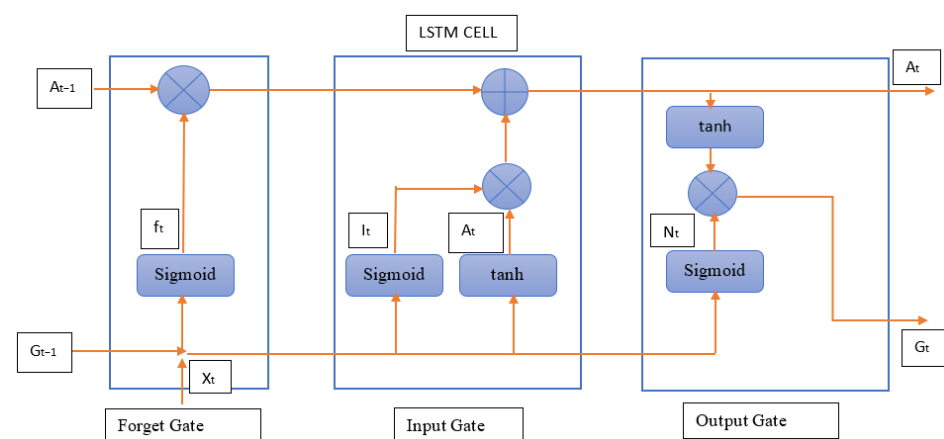


Figure 3. Top-view architecture diagram of LSTM. A typical LSTM cell with the operations of the forget gate, input gate, and output gate. Sigmoid and tanh activations control the flow of information on a cell state and hidden state over time steps.

In Figure 3, the architecture of an LSTM contains three primary gates and a memory cell.

2.1.1. Forget Gate

The forget gate is used to decide the extent to which past cell-state (G_{t-1}) information is to be remembered or forgotten. It takes in the hidden state at the previous time $t - 1$ (A_{t-1}) and the present input (X_t) and uses a sigmoid activation function to give it a value

between 0 and 1. The values near 0 eliminate irrelevant past information and those that are near 1 hold significant temporal associations. The formula is presented in Equation (1).

$$f_t = \sigma(W_f \cdot [G_{t-1}, X_t] + f_t), \quad (1)$$

where

W_f : forget gate weight matrix

b_f : bias

σ : sigmoid function (0, 1)

2.1.2. Input Gate

The input gate determines the manner in which the new information in the prevailing input sequence is factored into the memory cell. A tanh layer provides a candidate memory content and sigmoid layer decide which content to be updated. This process enables the LSTM to memorize the important seismic patterns of the sequence of waveforms instead of memorizing the data set itself [23]. The mathematical formulation of the input gate is presented in Equations (2) and (3).

$$I_t = \sigma(W_i \cdot [G_{t-1}, X_t] + f_t), \quad (2)$$

$$C_t \sim = \tanh(W_c \cdot [G_{t-1}, X_t] + A_t) \quad (3)$$

where;

I_t : importance of candidate values

$C_t \sim$: candidate memory update

2.1.3. Cell-State Update

Combining retained past information and new information of the candidate via the forget and input gate respectively, the cell state (C_t) is updated. The process helps the network to carry long-term temporal dependencies in the seismic signals. The new memory combines past state and candidate update. The mathematical equation is presented in Equation (4).

$$C_t = f_t \odot C_{t-1} + I_t \odot C_t \sim, \quad (4)$$

where;

$\odot$: element-wise multiplication

2.1.4. Output Gate

The hidden state (A_t), the information that is sent to the next time step and to the other network layers, is determined by the output gate. Sigmoid function selects the important aspects of memory, and a tanh activation creates a final hidden representation with which earthquakes are classified. It provides the final result based on the updated cell. In Figure 3 above, the output is shown as A_t . It uses a sigmoid function and produces the hidden state G_t [24]. The mathematical formulation of output gate is presented in Equations (5) and (6).

$$A_t = \sigma(W_o \cdot [G_{t-1}, X_t] + B_o), \quad (5)$$

$$G_t = A_t \odot \tanh(N_t), \quad (6)$$

In context of seismic detection, this gating mechanism allows the model to reduce noise and capture meaningful information of waveform such as P-wave and S-wave arrival for identifying the earthquake [25].

In Figure 4, the simplified LSTM architecture comprises four layers. The first input layer is time-series high-dimensional data coming from different DAS sensors at different timestamps. The second layer, which is the LSTM layer(s), processes the sequential data and highlights the temporal dependencies. The third layer, which is a dense layer, converts

the LSTM outcomes into a readable format for further classification. The fourth layer provides the final forecasting of whether any type of seismic event will happen or not. This architecture was further enhanced by adding more layers, such as Bidirectional LSTMs, an attention mechanism, and incorporating hyperparameters such as dropout, regularization techniques to boost the performance and interpretability of the model [26–28].

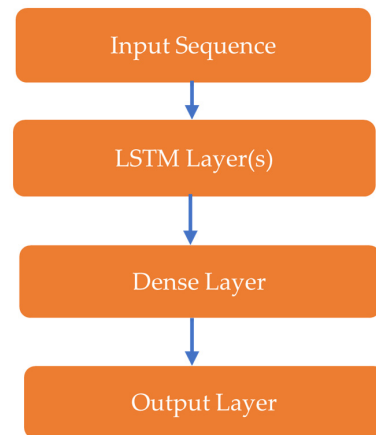


Figure 4. Simplified LSTM architecture. General workflow of proposed deep learning. It feeds on the input sequence using LSTM layer(s) to extract temporal features, a dense layer, and a final output layer to make forecasting's [29].

2.2. Bi-LSTM

Bidirectional Long Short-Term Memory, or Bi-LSTM, is a kind of LSTM that helps understand sequences better. Regular LSTMs look at data in one direction, either from the start to the end or the end to the start. Bi-LSTMs use two LSTM layers at the same time. One layer reads the data from the start to the end, and the other reads it from the end to the start. At each step, the information from both layers is combined in some way, either by joining them together or by adding them, to give a more complete understanding. Because of this two-way way of working, Bi-LSTMs are better at handling long-term patterns than one-way LSTMs. They have been used successfully in many tasks like recognizing speech, categorizing text, translating languages, and identifying future values in time series data.

2.2.1. Mathematical Formulation of Bi-LSTM

A Bidirectional LSTM (Bi-LSTM) uses two LSTM layers:

1. One LSTM goes through the input sequence in the usual order, starting from the first element to the last.
2. The other LSTM goes through the sequence in the opposite direction, starting from the last element to the first.

2.2.2. LSTM Cell Equations

Equations (1)–(6) collectively describe the LSTM model formulation. Equation (1) defines the forget gate representation, Equations (2) and (3) characterize the input and feature extraction processes, Equation (4) represents the update function, and Equations (5) and (6) specify the final output of the model.

1. $I_t = \sigma(W_i X_t + V_i G_{t-1} + B_i)$
2. $F_t = \sigma(W_F X_t + V_F G_{t-1} + B_F)$
3. $O_t = \sigma(W_O X_t + V_O G_{t-1} + B_O)$
4. $\hat{C}_t = \tanh(W_C X_t + V_C G_{t-1} + B_C)$
5. $C_t = f_t \odot C_{t-1} + I_t \odot \hat{C}_t$

$$6. \quad G_t = O_t \odot \tanh(C_t)$$

where

I_t = Input gate

F_t = Forget gate

O_t = Output gate

C_t, G_t is Cell state and hidden state

X_t is input vector for time step t

σ is sigmoid activation function

$\odot$ is element-wise multiplication

2.2.3. Forward Direction and Backward Direction

In a Bi-LSTM, two LSTMs are run simultaneously. The mathematical equation of forward and backward LSTM is presented in Equations (7) and (8).

1. Forward LSTM

$$h_t^{\rightarrow} = \text{LSTM}_f(x_t, h_{t-1}^{\rightarrow}) \quad (7)$$

2. Backward LSTM

$$h_t^{\leftarrow} = \text{LSTM}_b(x_t, h_{t+1}^{\leftarrow}) \quad (8)$$

2.2.4. Bi-LSTM Output

At time step t , hidden states of forward and backward direction are combined. The mathematical representation of forward output was presented in Equation (9).

$$h_t = [h_t^{\rightarrow}; h_t^{\leftarrow}] \quad (9)$$

Here (;) denotes concatenation or some models use sum sign. The mathematical representation of backward output is presented in Equation (10).

$$h_t = [h_t^{\rightarrow} + h_t^{\leftarrow}] \quad (10)$$

2.2.5. Bi-LSTM Final Sequence

The Bi-LSTM provides a final sequence of hidden vectors. The mathematical equation of Bi-LSTM sequence is shown in Equation (11).

$$H = \{h_1, h_2, h_3, h_4, \dots, h_T\}, \quad (11)$$

This final sequence of classification is fed into further layers such as attention layer, dense layer, and activation function SoftMax, relu, etc.

2.3. Stacked LSTM

A stacked LSTM, or deep LSTM, extends the standard LSTM architecture by stacking multiple LSTM layers hierarchically. The output sequence from one LSTM layer acts as the input sequence for the subsequent layer, enabling deeper feature extraction and more abstract temporal representations. Formally, for a stack of M LSTM layers is mathematically defined in Equation (12).

$$H_t^{(o)} = \text{LSTM}^{(o)}(H_t^{(o-1)}) \text{ for } o = 1, 2, 3, \dots, M, \quad (12)$$

The final hidden sequence $H_t^{(o)}$ captivates high-level temporal dependencies collected from all previous layers.

2.4. Gated Recurrent Unit (GRU)

The Gated Recurrent Unit, or GRU, is a simpler version of the LSTM network. It was created to make the process faster while still being able to remember important information

from sequences. Unlike LSTM, GRU does not have a separate memory cell. Instead, it uses two gates: an update gate and a reset gate. The update gate Z_T decides how much of the previous information to keep, and the reset gate R_T determines how much of the past information to forget when combining it with new data. This makes the GRU easier to work with while still handling long-term patterns in data. The mathematical formulation of GRU was defined by Equations (13)–(16).

$$Z_T = \sigma(w_z \cdot [H_{t-1}, X_t]), \quad (13)$$

$$R_T = \sigma(w_r \cdot [H_{t-1}, X_t]), \quad (14)$$

$$h_t^\sim = \tanh(w_h \cdot [R_T \times h_{t-1}, X_t]), \quad (15)$$

$$H_t = (1 - Z_t) \times H_{t-1} + Z_t \times h_t^\sim, \quad (16)$$

2.5. CNN-Bi-LSTM

The CNN-Bi-LSTM hybrid architecture was also assessed to investigate the potential advantages of integrating spatial feature extraction with temporal modeling. Within this framework, convolutional layers were employed to extract localized spatial-temporal patterns from the seismic waveform data. These convolutional feature maps, represented as $(f_T = \text{conv}(X))$, were then fed into a Bidirectional LSTM (Bi-LSTM) layer to model both forward and backward temporal dependencies. The mathematical definition was expressed in Equation 17.

$$H_t = (\text{Bi-LSTM}(f_T)), \quad (17)$$

While the CNN component successfully captured short-term and localized signal variations, it also introduced an extra layer of abstraction that sometimes suppressed finer temporal dependencies essential for precise seismic event discrimination. Consequently, the CNN-Bi-LSTM model achieved slightly lower classification accuracy compared to the standalone Bi-LSTM. The Bi-LSTM, by directly modeling raw temporal relationships without intermediate convolutional transformation, preserved long-range dependencies more effectively and achieved approximately 90% accuracy, whereas the CNN-Bi-LSTM showed comparatively lower performance. This indicates that although convolutional pre-processing can enhance feature locality, excessive feature compression in seismic sequences may limit the model's sensitivity to subtle waveform variations that are crucial for accurate event detection.

2.6. Attention Mechanism

Self-attention has become a powerful method for making neural networks work better. It lets models pay more attention to the most important parts of the input data. Unlike traditional layers like convolutional or recurrent layers, which only look at small, fixed areas, self-attention looks at connections between all parts of the input sequence or feature map. These input features are changed into three separate parts—query (P), key (K'), and value (V')—using linear transformations that the model learned and then calculated how similar P and K' were using a dot product to create a matrix of attention scores. This matrix was adjusted using a SoftMax function to get attention weights. These weights help highlight important time and space relationships in the data while giving less importance to parts that are not as relevant. The final output, called Y, is made by combining these weighted values [30]. The mathematical representation is shown in Equations (18) and (19).

$$A_{tt}(P, K', V') = \text{Softmax}\left(\frac{Qk^T}{\sqrt{d_k}}\right)V' \quad (18)$$

$$Y = A_{tt}(P, K', V') \quad (19)$$

Following is the description:

- (1) The input feature map, which has a size of $(b \times c)$, is split into three separate versions: Query (P), Key (K'), and Value (V'). This is done using either convolutional or linear layers [31].
- (2) The similarity between different parts of the sequence is found by multiplying the Query with the Key [32].
- (3) This gives a measure of how related each part is.
- (4) The SoftMax function then takes this similarity and converts it into attention weights.
- (5) These weights show how much each part of the sequence should be focused on.
- (6) These attention weights are then used to combine the Value matrix, creating a new output that highlights the important parts of the sequence.
- (7) Finally, this new output is added to the original input through a process called residual connection [33].
- (8) This helps create Figure 5, which illustrates a better overall representation of the sequence by capturing how different parts are connected.

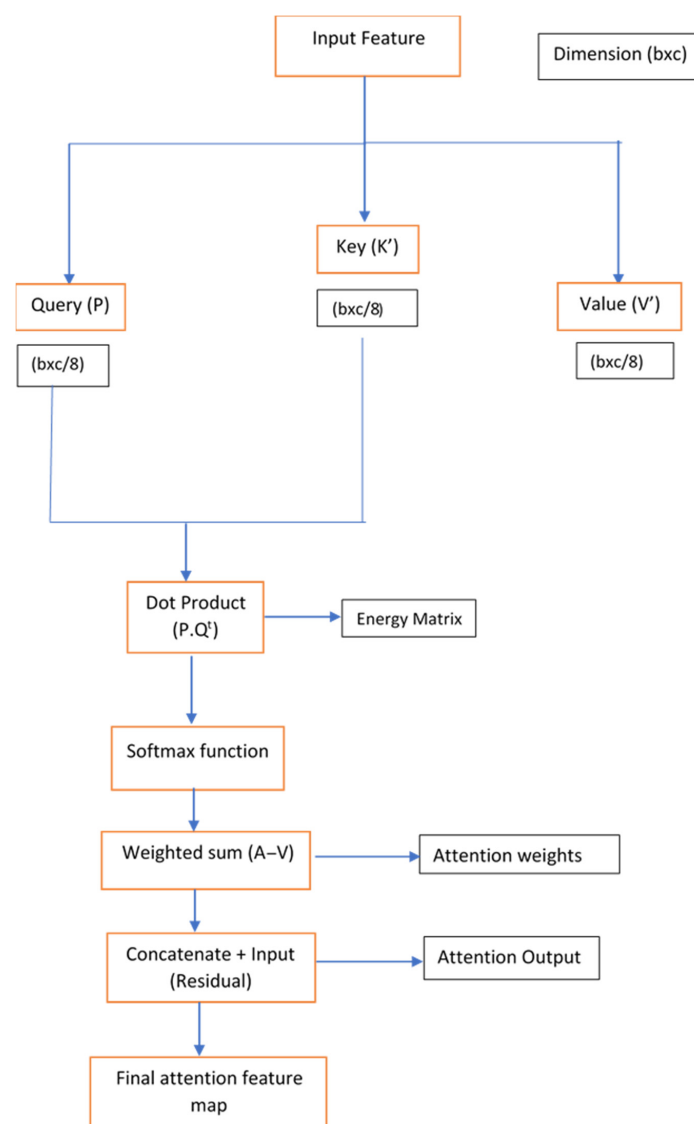


Figure 5. Attention-layer architecture. The mechanism of attention coupled with the feature representation. Input features are converted to query (Q), key (K), and value (V) vectors, and then dot-product attention, SoftMax normalization, and weighted aggregation are used to get the final attention-enhanced feature map [34,35].

2.7. Bi-LSTM + Attention Mechanism

In seismic signal analysis, getting accurate results relies on models that can understand both past and future patterns because seismic waves are complex. Bi-LSTMs can learn more about the time-related features in the data, making them better for tasks like detecting and classifying seismic events. This setup also helps with the vanishing gradient problem seen in regular RNNs by keeping information flowing better over longer sequences [12].

In the model, “ $h \rightarrow t$ ” and “ $h \leftarrow t$ ” represent the forward and backward hidden states at time t . W is weight matrices, and b is bias vectors. The mathematical definition is represented in Equation (20).

$$\begin{aligned} \vec{h}_t &= H(W_x \vec{x}_t + W_h \vec{h}_{t-1} + b) \\ y_t &= W_y \vec{h}_t + W_y \leftarrow h_t + b_y \\ \leftarrow h_t &= H(W_x \leftarrow x_t + W_h \leftarrow h_{t+1} + b) \end{aligned} \quad (20)$$

In the model, “ $h \rightarrow t$ ” and “ $h \leftarrow t$ ” represent the forward and backward hidden states at time t . W is weight matrices, and b is bias vectors [36].

To better distinguish features in seismic signals, this research added a self-attention mechanism on top of the Bi-LSTM outputs. While Bi-LSTMs are good at capturing the order of events in time, not every part of the seismic data is equally important for making a classification decision [37]. The attention score α_t for each hidden state is calculated in Equations (21) and (22).

$$\alpha_t = \frac{\exp(e_t)}{\sum_{n=1}^T \exp(e_n)} \quad (21)$$

$$e_t = V^t \tanh(W_H H_T + b) \quad (22)$$

At time step t , H_T is the hidden state. W_H and V^t are parameters that the model learns during training. The value α_T shows how important each time step is. The final context vector is made by adding all the hidden states together, with each one multiplied by its own alpha value [38]. The mathematical function is expressed in Equation (23).

$$d = \sum_{i=1}^t \alpha_T h_T, \quad (23)$$

This context vector holds the most important details about the seismic signals and is then passed to fully connected layers for classification. Table 1 highlights the recent literature review, discussion of method, main findings and limitations.

Table 1. Comparison of the available literature describing methods, primary results, disadvantages, and the gains obtained with the application of the proposed Bi-LSTM self-attention structure.

Study	Method/ Architecture	Application	Main Findings	Limitations
Nanda et al. [28]—UPC-UCV CNN	P-wave detection and source region estimation Convolutional Neural Network.	Seismic monitoring	CNN is quite sensitive in detecting P-wave arrivals at a variety of stations and geological areas.	CNN only analyzes the local patterns, but not temporal dependencies of long seismic sequences.
Yu et al. [32]—Phase arrival time	Reimplementation of ConvNetQuake	Automatic building of seismic catalogs (detect wave arrivals in numerous stations).	Helps CNNs detect earthquakes better even with a lot of background noise.	Still has some problems, like not remembering sequences well, especially for events with low signal-to-noise ratio.
M. Cheng et al. [34]—CRED (CNN + BiLSTM residual)	Convolutional layer and Residual bidirectional LSTM Hybrid deep neural network.	Small-magnitude automatic earthquakes detection.	Showed a high level of F-score performance on weak seismic and regional seismic event detection.	Model is a black-box detector and does not give temporal feature significance or viewability of waveform segments.

Table 1. *Cont.*

Study	Method/ Architecture	Application	Main Findings	Limitations
Perol et al. [36]—ConvNetQuake (PMC)	CNN	Early earning earthquake detection	Shows strong ability to find patterns in different areas and sizes.	Shows strong ability to find patterns in different areas and sizes.
Perol et al. [36]—ConvNetQuake (PMC)	CNN (single-station)	Automatic building of seismic catalogs (detect wave arrivals in numerous stations). Continuous seismic recording	It is very good at finding earthquakes and responds quickly. It works well with new types of earthquake patterns. It gives location guesses based on probabilities.	It needs a lot of training data. It is not very good at understanding information about time. It does not explain how time affects things very well.
Hou et al. [38]—mention of attention	Survey of deep learning methods on seismic signal analysis.	Seismic monitoring, phase picking and earthquake sensing.	Architectures grounded in attention promise to be useful in complex seismic analysis tasks.	The study related to practical testing using attention mechanism against benchmark studies is not very available.

3. Methodology

3.1. Problem Identification and Formulation

Figure 6 describes the following steps involve the depth view of traditional models used in the detection of seismic events, improving on performance limitations in high-risk domain real-world applications.

3.2. Dataset Preparation

ConvNetQuake data set consists of waveforms related to a local earthquake database that comprises event magnitude, hypocentral depth, and epicentral location. This catalog was a creation of a local seismic net and not artificial data but reflects actual seismic activity over a few years of data. The ConvNetQuake dataset was created by Lomax and Nicolas et al. [24,25] is often used as a standard for classifying seismic events. The waveforms were captured by the Northern California Earthquake Data Center seismic network, and they include both the local earthquake events and the non-seismic noise windows. ConvNetQuake is open-sourced under the MIT open-source license which releases the data without warranty or liability by the original authors. Thus, further validation procedures were conducted prior to training the model. The initial step was to check waveform files with missing values, corrupted traces and abnormal amplitude. Second, the event metadata (origin time and station recordings) was used to cross-check the catalog labels with waveform window to see that there was a temporal consistency between the waveform window and the reported seismic event. Third, signal preprocessing which consists of band-pass filtering and normalization was made to minimize recording artifacts and instrumentation noise. These measures make sure that the dataset is applied as a research standard and scientific explanation is the role of the current research. The dataset is a series of years of recordings, and it contains three component seismic signals (vertical, north–south and east–west channels). Earthquake occurrence is printed on each waveform window providing the possibility to evaluate the supervised learning. Recent research efforts with deep-learning-based seismic detection have been broadly using the dataset, and thus the results can be compared and are reproducible. These recordings are from passive seismic monitoring and show both natural earthquakes and background noise. Even though this dataset does not use DAS, the way the waveforms look and how they change over time are similar to DAS data in terms of how signals and noise behave. In this study, they take a similar method used for seismometer data and apply it to DAS data. DAS uses a fiber-optic cable to continuously measure strain rates along its length. This lets them use deep-learning models that were originally made for seismometer data, and

apply them to DAS data, which is more detailed and denser, to better detect and classify seismic events.

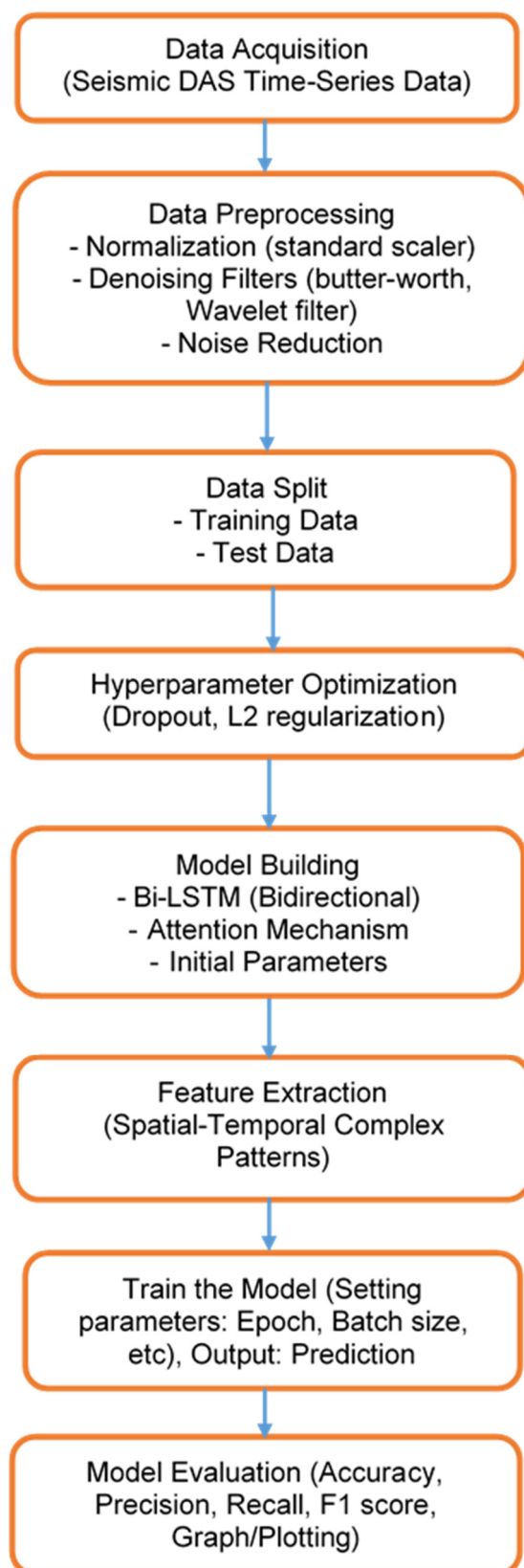


Figure 6. General workflow of the proposed seismic event detection framework with DAS time-series data in the form of data acquisition, preprocessing, model development with Bi-LSTM and attention mechanism, training, and performance estimation.

3.3. Data Preprocessing and Signal Conditioning

The preprocessing process was made to deal with unequal numbers of classes and make the seismic wave data better. A mix of resampling methods was used. For the less common event samples, they were carefully increased using simple changes such as adding random noise, shifting time, and changing volume. At the same time, the more common noise samples were slightly reduced. This helped balance the data without letting the noise take over. Even though normal Synthetic Minority Over-sampling Technique (SMOTE) is not great for time-based data, future work might try time-aware SMOTE or Generative Adversarial Networks (GAN) based methods. More ways to make the data diverse could include things such as randomly removing parts of the frequency spectrum, mixing different event signals, and applying changes based on physical rules. Each data point was made consistent across channels after filtering to make training easier and reduce differences in size. Another option is to use a single-scaling method trained on the training data to keep everything uniform. To make sure the signal stays clear while reducing noise, a low-pass filter with adjustable frequency and wavelet denoising with customizable levels were used. The impact of these filtering steps can be checked by comparing datasets with and without these steps. To keep experiments fair, the same random settings were used across Python 3.10, NumPy 1.24.4, and TensorFlow 2.10.0. It is noted that some Graphical Processing Unit (GPU) tasks might still not be fully detectable without extra settings. The model was tested using different random setups, and results were shown as average with standard deviation to ensure the findings are reliable and consistent.

3.3.1. Signal Conditioning and Segmentation

The raw seismic waveforms in the ConvNetQuake data set were initially processed through classical signal-conditioning procedures to improve the quality of the signals without any changes in physically meaningful properties. Butterworth band-pass filter was used to reduce low-frequency drift and high-frequency noise that is usually available in seismic recordings. This was followed by the subdivision of the filtered signals into fixed-length overlapping windows ensuring temporal consistency and allowing the learning of local and long-term temporal dependencies. To stabilize the training and avoid the high-amplitude event dominance, each segment was made normalized with the help of a standard scaler.

3.3.2. Feature Representation Strategy

In contrast to conventional seismic classification pipelines based on intensive use of handcrafted time- or frequency-domain characteristics, the given study uses an end-to-end deep-learning approach where the feature extraction is trained on the waveform segments. Initial tests were performed with classical features, including signal energy, statistical moments, and spectral descriptors, but the Bi-LSTM with the attention mechanism realized a high-performance level in case it was straightforwardly trained on the normalized fragments of the waveforms. This inspired the application of representation learning to prevent the bias of feature-selection and to better represent non-stationary seismic patterns.

3.3.3. Magnitude Homogenization

The seismic catalog that belongs to the ConvNetQuake dataset has various magnitude scales (ML, Md and Mb) that are frequently reported in small and local earthquakes. Because machine-learning models can be trained to learn inconsistent physical relationships when the magnitude of heterogeneous magnitude definitions is present, all of the magnitudes were brought to the moment magnitude scale (Mw), which is physically proportional

to seismic moment and fault energy. Conversion was done by empirical relationships in seismological research:

1. $M_w = 0.85 M_L + 0.15$
2. $M_w = 0.90 M_d + 0.10$
3. $M_w = 0.67 M_b + 0.93$

The homogenization results in unified magnitude scale in the dataset, which implies that the learning process was physically consistent; it will also be possible to make any meaningful comparisons between seismic events.

3.4. Data Unbalancing

Seismic data sets, such as ConvNetQuake, often have an unbalance problem. This happens not just between different types of data, like noise and actual events, but also within the same type of event. For example, there are not enough records of small earthquakes, while large ones are much more common. This makes it hard to train models properly because the networks start to focus on the more common patterns, like big earthquakes or background noise, and do not perform well on the less common ones. In recent studies, experts in seismology and related fields have stressed that fixing this imbalance is very important for accurate results. They point out that in tasks like estimating earthquake strength, issues like changes in data patterns and uneven numbers of data points across different strength levels are big problems.

3.5. Data Augmentation and Oversampling Techniques

In the data used in the current study, there is the ConvNetQuake that consists of 25,924 waveform segments which are divided into 15,200 earthquake waves (label 1) and 10,724 noise waves (label 0). All the records contain 1001 samples (a fixed length) of three-component seismic waveforms (East-North-Vertical channels) with input size (10,013). Since the dataset had more event samples than noise, the first models learned the bias towards the majority sample and had a worse detection ability on underrepresented signals. Controlled data augmentation was used on the minority only to minimize this bias. The realistic variation in seismic station records was introduced using simulated small-time perturbations, amplitude scale and low-level additive noise, with no change in the physical properties of the waveform. The noise category was adjusted to the number of samples in the event to generate a balanced dataset of 30,400 waveforms. The step does not change the earthquake data but enhances the representativeness of the training data, which enables the Bi-LSTM model to acquire generalized seismic trends instead of noise at the individual station.

3.6. Splitting of Dataset

The dataset was split into training and test subsets in an 80/20 ratio, so 24,320 training samples and 6080 testing samples were obtained as the result of the process. The reason for this division is not to change the data but to make sure that the model is unbiased when applied to unknown recordings. The network can also memorize the training signals and can generate artificially high accuracy without using a separate testing set. The labels were then transformed into binary classification one-hot coded vectors. Hence, augmentation, balancing, and splitting cannot be considered several independent modifications; they represent one typical machine-learning preparation process that avoids overfitting, provides fair learning of classes, and can assess the performance of seismic event detection. This was not biased toward the more common event class during training and evaluation. Table 2 highlights the breakdown of dataset before and after resampling.

Table 2. Table of distribution of the dataset upon classification of the dataset with original imbalance, post-resampling balance, and training and testing partitions obtained.

Dataset	Noise (0)	Event (1)	Total
Original	10,724	15,200	25,924
After Resampling	15,200	15,200	30,400
Training split (80%)	12,160	12,160	24,320
Testing split (20%)	3040	3040	6080

3.7. Model Evaluation

Traditional measures of classification such as accuracy, precision, recall, F1-score, loss and mean squared error (MSE) were used to measure the performance of the models. The metrics have extensive applications in machine-learning literature and seismic-event detection and give complementary information on classification performance and forecasting error. The overall correctness is the accuracy, whereas the false-alarm and missed-detection tendencies are measured by the precision and recall, respectively. F1-score is a precision–recall balance measure especially in imbalanced data. Loss values and MSE values show the degree of forecasting errors. The specifics of these metrics can be readily established in existing resources [28].

3.8. Comparison of Model with Benchmark Model

The performance of the Bi-LSTM was compared with the baseline model’s conventional LSTM, including unidirectional, Stacked LSTM, GRU, and Bi-LSTM-CNN. This comparison highlights differences in performance across the defined metrics. The final test was cross-validation against different time windows and seismic events, analyzing attention weight and interval of forecasting to validate trustworthiness and interpretability of the model. This proposed methodology helps the model not only in terms of efficacy but also ensures reliability and transparency that meet the requirements of a practical system in real-world seismic events.

4. Results and Discussions

In this section, results are discussed of LSTM, Bi-LSTM, Bi-LSTM with attention mechanism and results compared with baseline model. The experimental findings prove that the suggested Bi-LSTM with an attention mechanism is performing significantly better than all the other baseline models in all the key evaluation metrics. Specifically, Bi-LSTM-Attn model has the best accuracy (0.976) and F1-score (0.975) which demonstrates the better balance between accuracy and recall in detecting seismic events. The enhanced recall (0.962) confirms the capability of the model to efficiently identify seismic events with minimum instances of false negative values, which is important in safety-related applications. In addition, the loss and MSE values are considerably lower, which means that the convergence is stable and the generalization is better. These findings indicate how the use of the attention mechanisms can be effective in the learning of temporal representations of seismic time-series data. Appendix A provide the coding algorithm for implementation of this study.

4.1. Resampling of Class Distribution

In Figure 7, the dataset showed a moderate class imbalance, with class “1” (events) occurring more frequently than class “0” (noise). This imbalance can lead to model bias towards the majority class. After resampling, both classes were balanced to 15,200 samples each, ensuring the classifier is trained on equal representation of both noise and event

instances. This balancing step is crucial for reducing bias and improving the model's generalizability to the minority class.

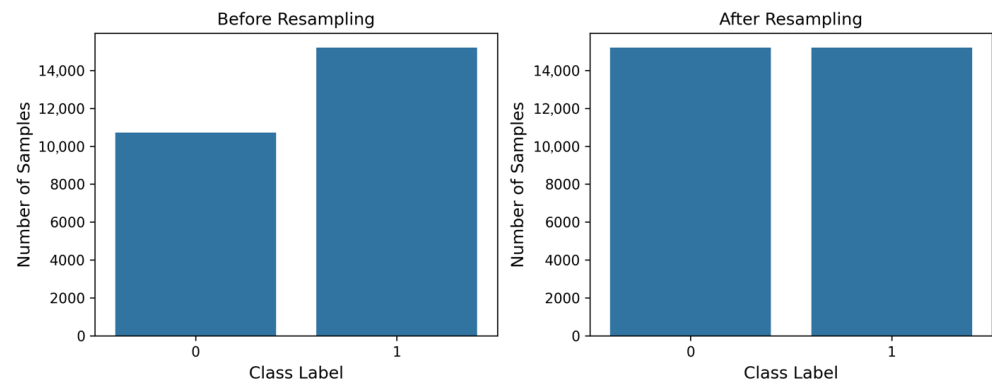


Figure 7. Resampling class distribution. This is the distribution of samples of seismic events and noise before and after resampling by class indicating the impact of data balancing as a way of reducing the class imbalance in the training sample.

4.2. Loading and Pre-Processing of Data

At first, the dataset had an unequal number of samples, with 10,724 noise samples and 15,200 event samples. To fix this imbalance, this research used a method called oversampling with data augmentation, which made the number of samples in each group equal—15,200 for each. It was important to balance the dataset so the model would not favor the most common type of sample and would learn fairly. The fact that the Bi-LSTM model still performed well on the balanced dataset shows that its good performance is not just because of the resampling—it is because of the way it works, using information from both directions.

4.3. Actual Waveform and Pre-Processed Waveform

Figure 8 demonstrates the impact of sequential pre-processing steps on a typical three-channel seismic waveform. The raw waveform (top panel) shows noticeable noise fluctuations overlying the seismic signal, which can obscure the actual event features. After application of the Butterworth filter, high-frequency noise is reduced, resulting in a smoother signal while maintaining the main oscillatory characteristics.

```

Loaded 15200 event samples and 10724 noise samples.
Shapes after unify_length: (15200, 1001, 3) (10724, 1001, 3)
Preprocessing event samples...
Filtering: 100%|██████████| 15200/15200 [00:12<00:00, 1236.62it/s]
Preprocessing noise samples...
Filtering: 100%|██████████| 10724/10724 [00:09<00:00, 1136.20it/s]
Dataset shape: (25924, 1001, 3) (25924,)
Class counts before resampling: {np.int64(0): np.int64(10724), np.int64(1): np.int64(15200)}
Class counts after resampling: {np.int64(0): np.int64(15200), np.int64(1): np.int64(15200)}
Train/Test shapes: (24320, 1001, 3) (6080, 1001, 3) (24320, 2) (6080, 2)

```

Figure 8. Data preprocessing pipeline, filtering, normalization, dataset composition, and distribution of classes of the data before and after resampling, as well as the final train-test split setup.

The wavelet denoising step further improves waveform clarity by reducing small-scale noise fluctuations, particularly visible in the channel amplitudes. After standardization, the completely pre-processed waveform (shown in Figure 9) has equal amplitude levels across all channels and shows a more distinct time-based pattern. This pre-processing process helps the LSTM/Bi-LSTM/Bi-LSTM-Attn model focus on important event-related information while reducing unwanted noise. Looking at this step-by-step visualization

shows that filtering and cleaning the signal greatly improves its quality, which is very important for training a reliable model.

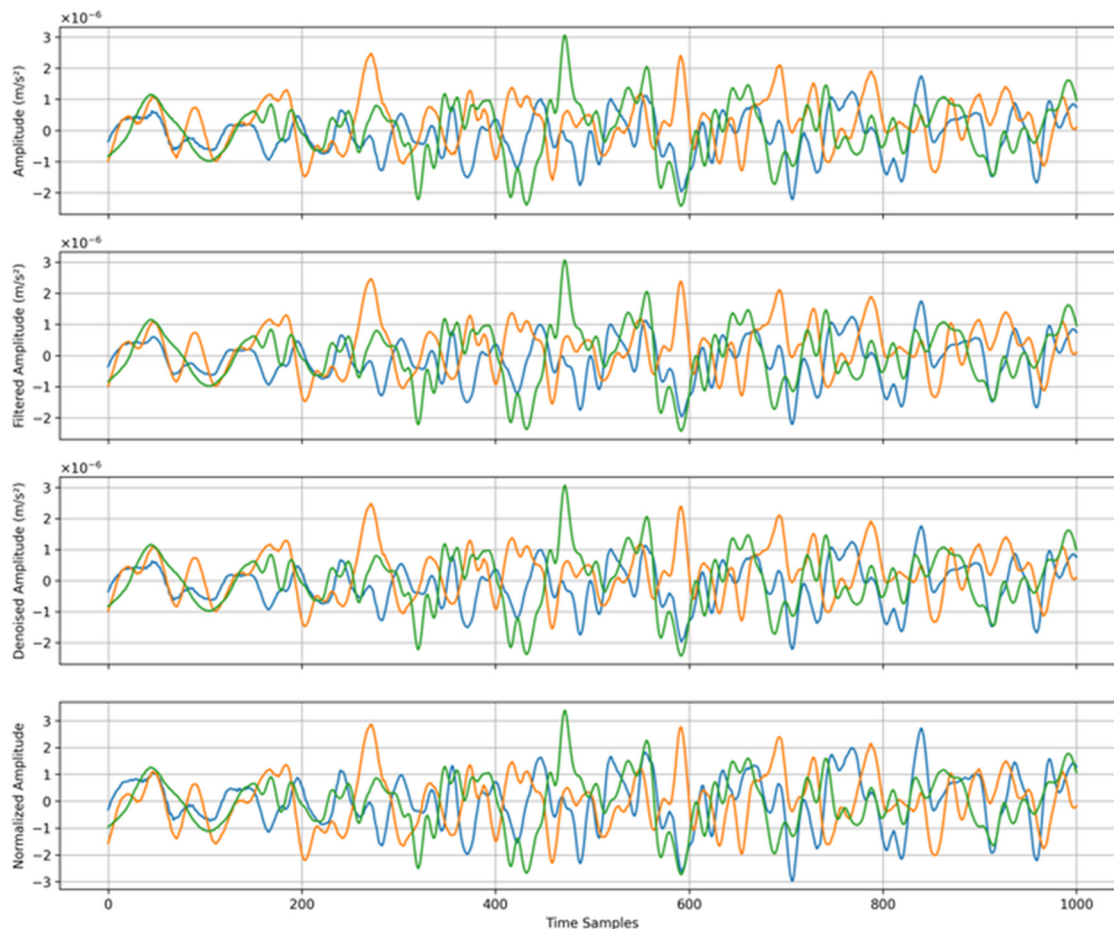


Figure 9. Plots of the multichannel signal time-domain representation at various levels of preprocessing stages Blue show Z channel, orange shows N channel, and Green shows E channel: raw signal, Butterworth signal, after the wavelet denoising signal, and after the complete preprocessing.

4.4. Raw Wave Spectrogram

Figure 10 shows the spectrogram of color scale encodes signal power (in dB), where dark regions (black to purple) correspond to low power levels and brighter shades (yellow to orange) denote zones of higher spectral energy concentration. The predominance of dark tones across the spectrogram suggests that the seismic signal maintains low energy across most frequency bands, with minimal temporal variation. Conversely, the faint yellow–orange streaks appearing at low frequencies (<0.1 Hz) signify the presence of dominant low-frequency components, which may correspond to sustained ground motion or long-period seismic noise. This distribution highlights the frequency-dependent attenuation of the raw event and reinforces the need for spectral enhancement during pre-processing to better isolate meaningful seismic features.

The raw event spectrogram shows that the seismic wave mainly has low-frequency parts, with very little high-frequency activity. This kind of pattern is common for small earthquakes or weak events, where the energy spreads out over a long time instead of happening quickly. The energy in the signal stays mostly below 0.1 Hz, which suggests it might be from ongoing background noise or slow ground movement, not a strong sudden event. This means there is a need to use techniques like bandpass filtering or normalization before processing to make the signal clearer and better for using in deep learning models such as Bi-LSTM-Attn for classification. The raw event spectrogram

shows a very low-energy pattern, mostly made up of dark areas close to the noise level (around -80 dB). The weak color differences and almost even spread suggest that the seismic-event details are mostly hidden by noise, which makes it hard to interpret or extract features directly. Comparing this with Figure 10 shows how important pre-processing is: while the raw spectrogram does not show clear frequency patterns, the pre-processed one clearly separates and makes stronger the earthquake-related frequency areas.

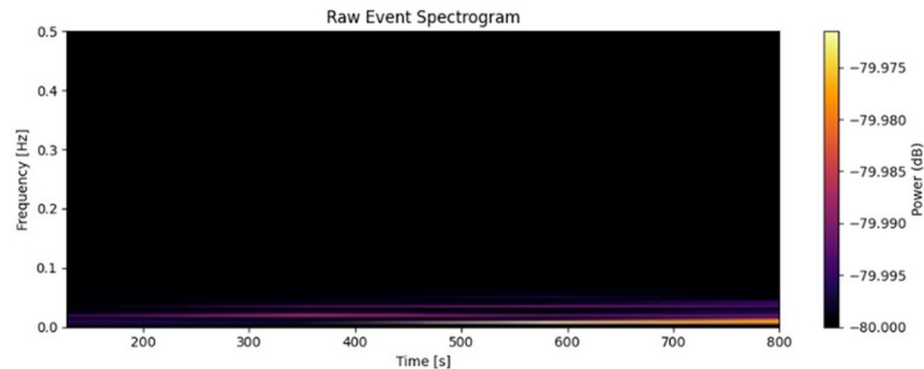


Figure 10. Raw event signal spectrogram, which visually shows the prevalence of low-frequency content and low spectral contrast.

This change shows that filtering and removing noise are essential steps before using machine-learning models in seismology.

4.5. Pre-Processing Spectrogram

The spectrogram of a pre-processed seismic event in Figure 11 illustrates the time–frequency distribution of signal energy. In comparison to the raw input, the pre-processed spectrogram shows clear energy bands, with significant concentrations in the lower frequency range (0.2 Hz). The color brightness shows more power (in yellow and orange areas), showing the real seismic energy that was hidden by noise in the original signal. The clear low-frequency parts show the real movement of the ground during an earthquake, which usually has strong low-frequency energy. By making these parts clearer, the pre-processing step helps make the data easier to understand and helps the Bi-LSTM model learn better from the most important features for detecting earthquakes.

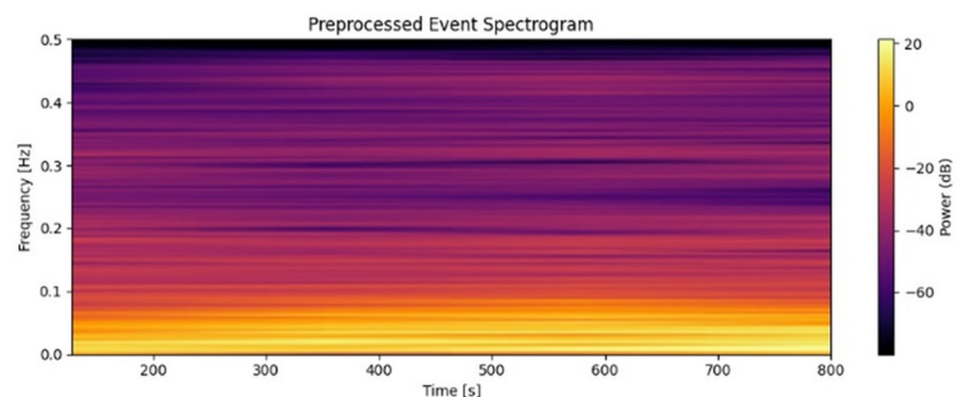


Figure 11. Spectrogram of the preprocessed event signal, demonstrating an increase in spectral clarity and better distribution of power throughout the frequency range in the post-full-preprocessing spectrogram.

4.6. LSTM Model Metrics

In Table 3, the LSTM model achieved an accuracy of 53.0%, with relatively moderate precision (50.4%) but low recall (54.7%). This indicates that while the LSTM was reasonably

effective at correctly identifying event samples (high precision), it often missed many true positive seismic events (low recall). As a result, the F1-score, which is the harmonic mean of precision and recall, remained at 52.5%, suggesting limited generalization capability for complex earthquake waveforms.

Table 3. Classification summary of LSTM using accuracy, precision, recall, F1-score, loss, and mean squared error (MSE).

	Precision	Recall	F1-Score	Support
0	0.476	0.484	0.4669	3040
1	0.410	0.496	0.457	3040
Accuracy	0.530	0.547	0.525	6080
Macro avg	0.590	0.560	0.5313	6080
Weighted avg	0.5538	0.5360	0.5913	6080

4.7. LSTM Loss and Validation Curve

The training history of the LSTM model is shown in two parts in Figure 12, Figure 12a for loss, and Figure 12b for accuracy. The loss graph shows big ups and downs over different training rounds, which means the model is not settling down properly. The validation loss moves up and down near the training loss, showing the model has trouble consistently separating actual events from background noise. The accuracy graph also has uneven changes, with both training and validation accuracy staying between about 0.53 and 0.55. These patterns show the LSTM model had trouble learning strong time-based connections. This might be because the model only looks in one direction and cannot remember things that happened a long time ago. This matches up with the LSTM model's worse performance compared to the Bi-LSTM model.

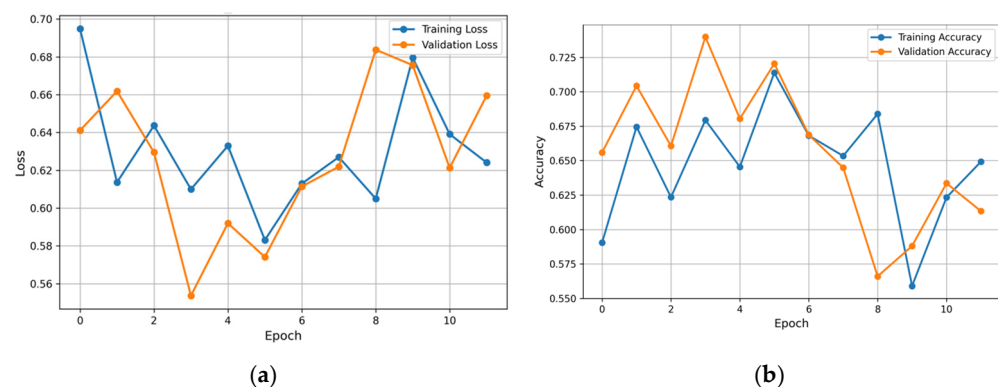


Figure 12. Performance during the training of the LSTM model over the epochs: (a) loss curves and (b) accuracy curves of the training and validation dataset. The plots demonstrate learning trends of the model revealing variation in the validation performance with overall convergence.

4.8. Bi-LSTM Attention Map

Figure 13, shows the heatmap where the attention mechanism in the Bi-LSTM-Attn model is looking at different points in the time series, which goes from about 0 to 380 time steps. It highlights which parts of the seismic signal the model thinks are most important. Most of the signal has a medium level of attention, shown in red, which means the model is using the whole signal. But there is a strong attention peak towards the end of the waveform, shown in yellow or white, meaning the model noticed an important feature there. This suggests the model is paying more attention to the most useful part of the signal when it makes its decision. Each input signal creates its own attention pattern.

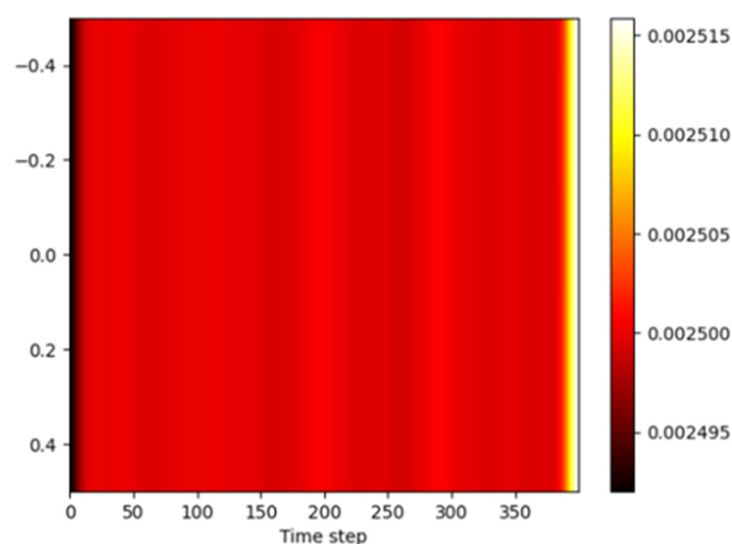


Figure 13. Visualization of heatmap of feature activation over time steps. The color intensity is a change in the value of activation which is a relatively stable trend with minor upward changes as the time step advances.

4.9. Bi-LSTM Accuracy

In comparison, the Bi-LSTM model performed much better than the LSTM. It achieved an accuracy of 96.5%, a precision of 97.9%, a recall of 95.2%, and an F1-score of 96.5%. The low mean squared error of 0.0166 shows that it has strong forecasting power. The model improved both precision and recall in a balanced way, meaning it not only correctly identifies seismic events but also reduces the number of missed events. Table 4 highlights the classification score of proposed models.

Table 4. Summary of classification performance metrics using accuracy, precision, recall, F1-score, loss, and mean squared error (MSE).

Accuracy	Precision	Recall	F1-Score	MSE	Loss
0.965	0.979	0.952	0.965	0.024	0.076

4.10. Bi-LSTM Loss and Validation Curve

The Bi-LSTM model shows quick improvement in the first 10 training rounds. Both the training and validation loss drop quickly and then stay at low levels, around 0.05. The accuracy goes above 97% and stays the same throughout the training. The similarity between the training and validation results shows that the model works well on new data without remembering too much from the training set. This consistency shows that the bidirectional structure is strong for detecting seismic events, as it keeps its performance steady even with new validation data.

In Figure 14, results clearly demonstrate that the Bi-LSTM significantly outperforms the standard LSTM. The bidirectional processing allows the model to capture more comprehensive temporal dependencies in seismic signals, leading to a reduction in both false positives and false negatives. In practice, this means Bi-LSTM offers more reliable earthquake event detection, making it more suitable for operational early-warning systems and seismic hazard monitoring. The improvement seen matches what was expected from theory. LSTM models look at time-series data in one direction, from past to future. But in seismic data, waveforms often depend on both past and future events because of reflections, overlapping signals, and noise. Bi-LSTM solves this by looking at data in both directions, which gives a better understanding of time-based patterns. These results show that using

bidirectional models is especially good for detecting earthquakes. It helps pick up on both early signs and later parts of the waveforms. In seismic monitoring, this leads to more accurate detection of real events, making it less likely to miss an earthquake or mistake noise for one.

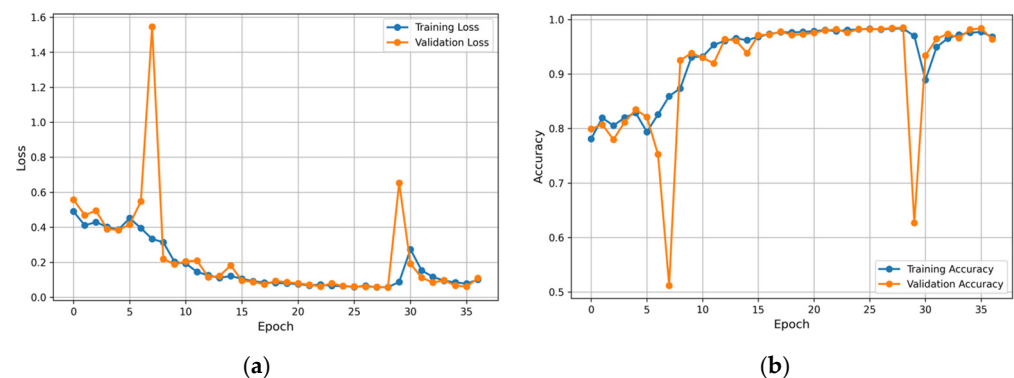


Figure 14. Bi-LSTM training history over epochs (a) show Bi-LSTM Model loss validation curve; (b) shows Bi-LSTM Model accuracy curve.

4.11. Bi-LSTM Training History

The Bi-LSTM model without attention has more ups and downs, especially a big drop in validation accuracy around the middle of training, which means the learning process is not very stable and might be more affected by the order of the data. Therefore, the Bi-LSTM with attention model gives a more reliable and accurate way to understand the sequence patterns. Figure 15 shows the history of the proposed model.

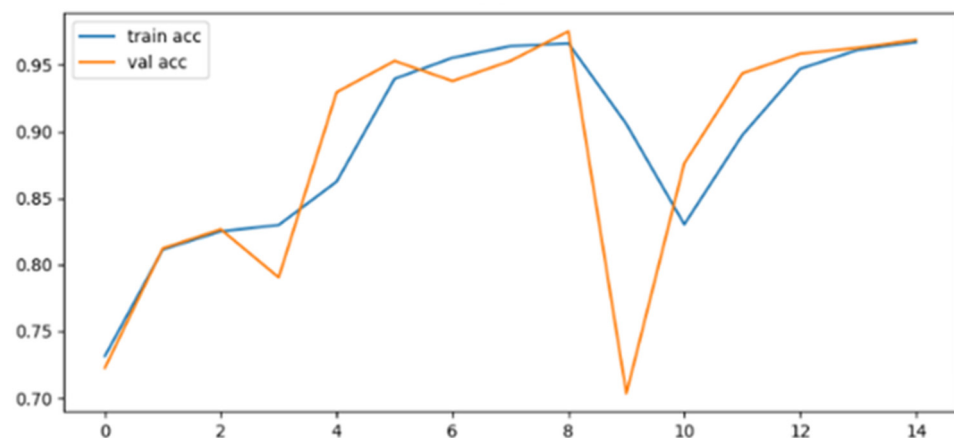


Figure 15. Epoch training and validation accuracy curves, to show both the learning behavior and convergence.

4.12. Bi-LSTM-Attn Metrics Score

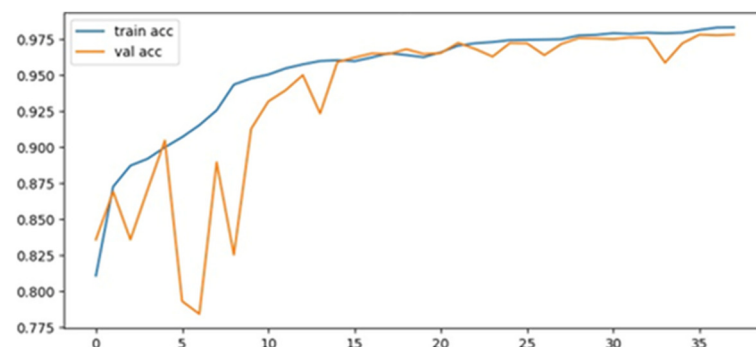
The Bi-LSTM-Attn model achieved an improved accuracy of 97.61%, precision of 98.98%, recall of 96.21%, and F1-score of 97.58%, while the MSE and loss decreased to 0.0238 and 0.0716, respectively. These results demonstrate a modest yet meaningful performance improvement due to the attention mechanism, which allows the model to focus on the most relevant time steps in the seismic sequence. The increase in precision indicates that the model better distinguishes true events from noise, while the slight improvement in the F1-score confirms enhanced overall classification reliability. The reduced loss values also suggest more efficient optimization and better generalization compared to the baseline Bi-LSTM model. Table 5 highlights the classification score of the proposed model.

Table 5. Summary of classification performance metrics for proposed model using accuracy, precision, recall, F1-score, loss, and mean squared error (MSE).

Accuracy	Precision	Recall	F1-Score	MSE	Loss
0.976	0.989	0.962	0.975	0.023	0.071

4.13. Bi-LSTM-Attn Training Validation Accuracy Curve

In Figure 16, the training and validation accuracy graph shows how well the Bi-LSTM-Attn model is learning. Both the training and validation curves improve steadily and consistently, ending up around 95 to 96% accuracy. This suggests that the model is learning well and can apply what it has learned to new data. There are some small ups and downs in the validation accuracy, which is normal and shows that the model is not overfitting—because both curves end up close to each other after training. This indicates that the model is well-balanced and that the pre-processing steps, especially resampling and normalization, are helping to make the data more balanced between different types of events and noise. This result is very good and shows that the Bi-LSTM-Attn model performs better than traditional LSTM, Stacked LSTM, GRU, and CNN-Bi-LSTM models. As seen before, the Bi-LSTM-Attn model reaches about 90% accuracy, while the other models only reach 70 to 80%. After about the 20th epoch, the two curves almost match or run side by side, with very little difference between them (less than 0.01). This shows that the model's performance on new, unseen validation data is almost the same as its performance during training, which is a good sign that it can generalize well instead of just memorizing the training data. The graph confirms that the attention mechanism in the model helps it understand the timing and sequence of data better than the other models.

**Figure 16.** Training and validation accuracy curves, to show both the learning behavior and convergence.

4.14. Attention Weights Across Time Steps

The attention-weight distribution plot in Figure 17 shows how the Bi-LSTM-Attn model focuses on different parts of seismic signals over time. The peaks in the attention curve point to specific time points where the model finds important patterns, often linked to the start or key parts of seismic events. The changing levels of attention show that the model can adjust which parts of the signal it pays more attention to, which makes it easier to understand how it values different time segments. The clear areas of high attention suggest the model is good at separating real seismic activity from background noise, which is important for detecting events in real-world situations. This graph is clear and easy to understand, showing that the attention mechanism is effectively capturing important time-based features for identifying seismic events.

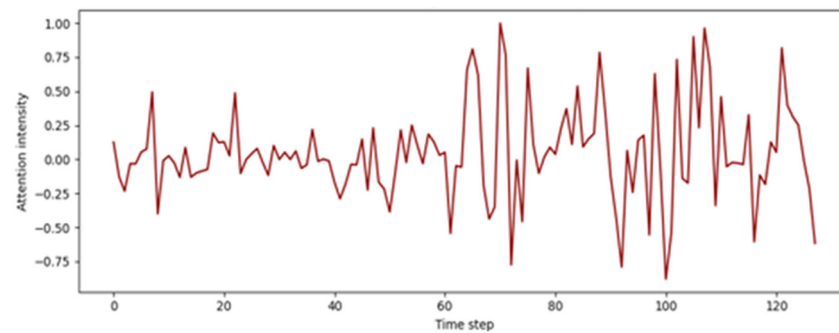


Figure 17. Dynamic behavior of the attention mechanism Time series variation in attention intensity between time steps, showing areas of maximum and minimum attention during sequence processing.

4.15. Saliency Map (Feature Extraction)

The saliency map in Figure 18 shows which parts of the seismic waveform were most important for the model's decision over time and uses gradient-based information to highlight the time intervals where small changes in the signal had a big effect on the model's output. These areas with high gradient values usually match the main energy release in seismic events, which means the model's forecasts are based on real physical patterns and not just random guesses. The clear peaks in the map, along with the low-gradient background, show that the model learned to focus on specific, meaningful time intervals rather than picking up on random noise. This type of visualization is important for understanding how the model works scientifically and is a key part of making its decisions explainable. It also matches with the attention map, showing that both the model's own focus and the gradient-based method point to the same important features in the seismic data.

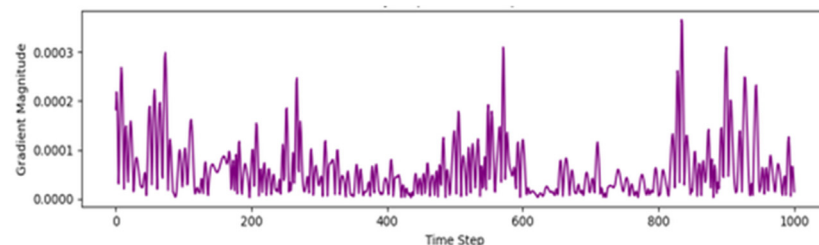


Figure 18. Efforts to observe the dynamics of optimization of the learning process and stability of the training process during backpropagation.

4.16. Compare with Baseline Model

Table 6 provides a comprehensive analysis of three models based on various evaluation metrics, such as accuracy, precision, recall, F1-score, loss, and mean squared error (MSE). The GRU model demonstrated the best accuracy (0.49) and F1-score (0.453), indicating a more balanced performance between precision (0.463) and recall (0.479). The CNN-Bi-LSTM model, while slightly lower in accuracy (0.465), achieved the highest recall (0.557), showing greater sensitivity in identifying true positive cases at the expense of higher precision (0.473). In contrast, the stacked LSTM model had the lowest accuracy (0.440) and precision (0.419), with moderate recall (0.463), suggesting weaker overall generalization. In Figure 19 compares how well three different model types GRU, CNN-Bi-LSTM, and Stacked LSTM performed in classification. The results show that the Stacked LSTM model had the best accuracy, just a little better than the CNN-Bi-LSTM. The GRU model had the worst performance of the three. This suggests that having more layers in a recurrent model, like the Stacked LSTM, helps it better understand the time-based patterns in the data compared to simpler models. The CNN-Bi-LSTM model uses both convolutional

layers and bidirectional processing, but it did not beat the Stacked LSTM. This might be because the dataset was small or not complex enough to fully use the strengths of the hybrid model. The GRU model's lower accuracy could be because it has fewer parameters, which makes it faster but not as good at handling complicated patterns. Overall, the results show that using deeper and more complex models can improve performance, but more work and more data might be needed to get even better results.

Table 6. Comparison of the performances of various Deep Learning Models based on accuracy, precision, recall, F1-score, loss and mean squared error (MSE).

Model Name	Accuracy	Precision	Recall	F1-Score	Loss	MSE
Build_stacked_LSTM	0.440	0.419	0.463	0.440	0.00116	0.560
Build_CNN_Bi-LSTM	0.465	0.473	0.557	0.512	0.00037	0.505
Build_GRU	0.49	0.463	0.479	0.453	0.0023	0.510

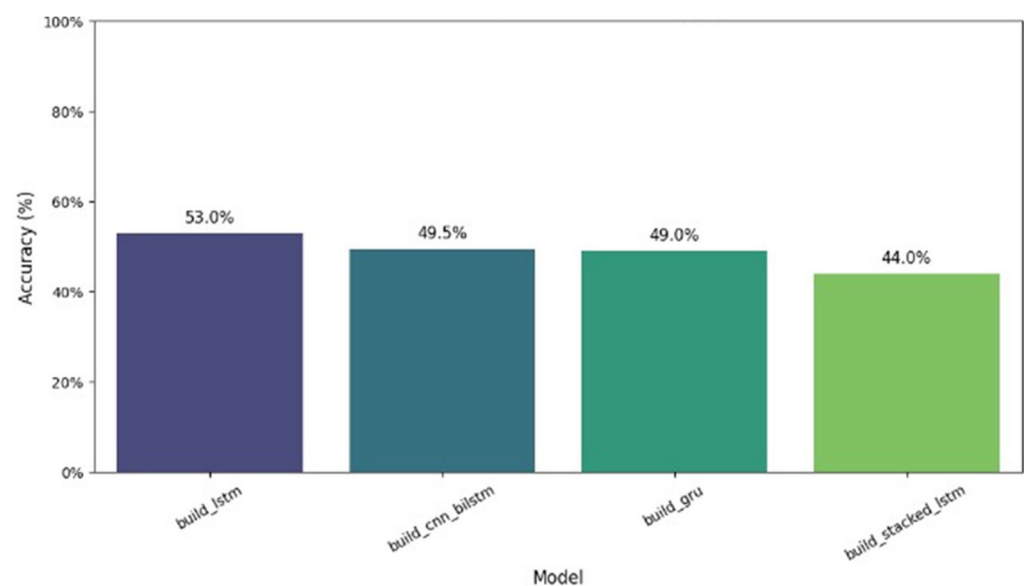


Figure 19. Comparison of the accuracy of various deep learning architectures, which illustrate the performance of the build-stacked LSTM model in comparison with the build-CNN-LSTM and build-GRU model.

In Table 6, the high recall values observed in both the GRU and CNN-Bi-LSTM models indicate that these architectures are more effective at identifying positive instances in imbalanced datasets. However, their lower precision and limited accuracy highlight the challenges of distinguishing between event and noise classes using these models alone. This further supports the superior performance of the Bi-LSTM model, which significantly outperformed these baselines across all evaluation metrics.

The bar chart in Figure 19, shows how well three deep learning models—Stacked LSTM, CNN-Bi-LSTM, and GRU—performed in classification. The GRU had the best results, with accuracy just over 0.49, showing it is better at understanding the time-based patterns in the seismic signals. The CNN-Bi-LSTM came next with around 0.465 accuracy, meaning its mix of convolution and recurrent parts helps it learn both spatial and time-based features, but it does not beat the Stacked LSTM.

The Stacked-LSTM had the lowest accuracy, about 0.440, which means it is not as good as the LSTM models at handling complicated long-term patterns. None of the models did very well, which shows that traditional recurrent and mixed neural networks have limits when used with this data. This suggests that choosing the right model structure is

important and that more improvements or different types of models, like Bi-LSTM-Attn, might be needed to get better results. Table 7 highlights the summarize performance score of proposed model and baseline model.

Table 7. The quantitative performance of several recurrent and hybrid Deep Learning structures that emphasize the usefulness of the proposed Bi-LSTM with the attention mechanism on a variety of measures of evaluation.

Model	Accuracy	Precision	Recall	F1-Score	Loss	MSE
LSTM	0.530	0.504	0.547	0.525	0.0020	0.470
Stacked-LSTM	0.440	0.419	0.463	0.440	0.00116	0.560
GRU	0.49	0.463	0.479	0.453	0.0023	0.510
Bi-LSTM-CNN	0.465	0.473	0.557	0.512	0.00037	0.505
Bi-LSTM	0.965	0.979	0.952	0.965	0.024	0.076
Bi-LSTM-Attn	0.976	0.989	0.962	0.975	0.023	0.071

5. Conclusions

In this research, the detection of seismic events was studied on the basis of the ConvNetQuake dataset, which had 15,200 samples of seismic events and 10,724 samples of noise at the beginning. The imbalance in the classes negatively impacted the output of the baseline sequence models especially on those events that are less frequent. In response to this, a mixed resampling approach to data enrichment (with amplitude scaling, temporal shifting, and noise injection) and targeted under sampling (of majority classes) was used. This gave a balanced dataset of 30,400 samples and that were equally divided into event and noise classes.

A comparative study of LSTM, stacked LSTM, GRU, Bi-LSTM, Bi-LSTM-CNN, and Bi-LSTM, with attention and hyper parameter optimization revealed that the bidirectional designs are always better than the unidirectional recurrent designs. This improvement can be explained by the fact that Bi-LSTM models are capable of reflecting both temporal interdependences of the past and the future, this is especially applicable to seismic signals that are defined by overlapping waveforms, reflections, and complicated noise patterns. Attention mechanisms also enhanced discrimination because the model was able to pay attention to significant temporal features.

Those findings show that bidirectional sequence modeling with the relevant preprocessing methods, including Butterworth filtering, wavelet-based denoising, and per-sample standardization, can help improve seismic signal classification results using standard benchmark datasets. Although these results point to the performance of the advanced recurrent architectures in detecting seismic events even in offline mode, additional research is needed to determine the performance of the architectures in the real-world application environment to determine their robustness and generalizability.

5.1. Research Limitations

A key limitation of this paper is that the presented framework relies on deterministic deep learning architectures, such as LSTM, Bi-LSTM, stacked LSTM, GRU, CNN-Bi-LSTM, and Bi-LSTM with attention regardless of the uncertainty estimation. Although these models are shown to perform well in terms of classification on the benchmark data, they do not give estimates of the confidence of a specific decision. Such deterministic outputs are not always adequate in seismic surveillance applications, especially in the context of low signal-to-noise ratio, confusing waveforms, and poorly defined seismic signatures, where false identification can have very important practical consequences.

5.2. Future Work

This limitation will be addressed in future work by including uncertainty-aware learning based on Bayesian deep learning methods, including Monte Carlo Dropout. In contrast to traditional neural networks, Bayesian methods model a distribution of model parameters, which makes it possible to estimate both the epistemic uncertainty (related to the model) and the aleatoric uncertainty (related to the data). Monte Carlo Dropout enables detective means and variances to be computed by a series of stochastic forward passes at inference, which provides a principled representation of model confidence. It is hoped that incorporating Bayesian versions of LSTM or Bi-LSTM networks will enhance the explainability and resilience of seismic-event detection and, specifically, weak or uncertain signals, as well as contribute to more trustworthy decision-making in operational seismic surveillance scenarios. The development was very helpful to the seismologists, developers of the earthquake early-warning system, civil authorities, and geophysical monitoring researchers, as it will help them to make more reliable and informed decisions in the context of operational seismic surveillance.

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

Funding: This research work is supported and funded by the TOTALENERGIES EP MALAYSIA (015MD0-166) with title ‘Data Analytics and Visualization Development System for Subsurface CO₂ Storage and Fluid Production’, under the Centre for Research in Data Science (CeRDaS), Universiti Teknologi PETRONAS, Malaysia.

Data Availability Statement: The authors recognize that ConvNetQuake dataset is an open academic benchmark and does not guarantee any legal assurances on the accuracy of data. The dataset is not in any way needed to monitor or assess earthquakes operationally or in terms of hazards, only to evaluate the methods and compare the models. The findings of this research apply to the effectiveness of the suggested learning model as opposed to the legitimacy of the seismic catalog.

Acknowledgments: In the writing of this paper, the author used Claude 3, Gemini 1.5 Flash to aid in the preliminary design of the study, write sections of the text and aid in the interpretation of data. All outputs were thoroughly reviewed, checked and edited by its authors in order to achieve scientific integrity and to be fully responsible in terms of content of this publication.

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

Abbreviations

The following abbreviations are used in this manuscript:

CNN	Convolution neural Network
LSTM	Long short-term memory
RF	Random forest
AE	Auto encoder
SVM	Support vector machine
DAS	Distributed acoustic sensing
USGS	United States Geological Survey
LIDAR	Light Detection and Ranging
IRIS	Incorporated Research Institutions for Seismology
STA/LTA	Short-Term Average Long-Term Average
ML	Machine learning
SMOTE	Synthetic Minority Oversampling Technique

BI-LSTM	Bi-directional Long short-term memory
LR	Logistic/Linear regression
GNN	Graph Neural Network
VAE	Variational Autoencoder
MC	Monte Carlo
DTS	Distributed Temperature Sensing
WoS	Web of science
P/S WAVE	Primary/secondary waves
AI	Artificial Intelligence
ROC	Receiver Operating Characteristics curve
AMAG	Amplitude Magnitude
PGV	ground-shaking Peak Ground Velocity
AIC	Akaike Information Criterion

Appendix A

Appendix A.1. Implementation + Verification Code

The given model was not trained with a training pipeline of the initial ConvNetQuake repository. The repository was accessed solely to get waveform data and event metadata. The entire pre-processing, feature preparation, and model development were all done in Python with TensorFlow/Keras. The implementation steps are (i) pre-processing seismic channel Z,N,E (ii) splitting data based on magnitude (iii) deployed Bi-LSTM with attention layer (iv) estimate Bayesian inference with Monte Carlo dropout.

Appendix A.2. Data Labeling and Loading

```

import h5py
import numpy as np
def load_event(file_path):
    with h5py.File(file_path, "r") as f:
        waveform = np.array(f["waveforms"]) # shape (samples, 3 channels: Z,N,E)
        label = int(f["label"][(0)])
        magnitude = float(f["magnitude"][(0)])
    return waveform, label, magnitude
# verification step
wave, label, mag = load_event("event_sample.hdf5")
print("Waveform shape:", wave.shape)
print("Label (1 = event, 0 = noise):", label)
print("Magnitude:", mag)

```

Appendix A.3. Data Preprocessing

```

from scipy.signal import butter, filtfilt
from sklearn.preprocessing import StandardScaler
def bandpass_filter(signal, low = 1.0, high = 20.0, fs = 100):
    nyq = 0.5 * fs
    b, a = butter(4, [low/nyq, high/nyq], btype = 'band')
    return filtfilt(b, a, signal, axis = 0)
def preprocess(wave):
    # filtering
    wave = bandpass_filter(wave)
    # normalization
    scaler = StandardScaler()
    wave = scaler.fit_transform(wave)
    return wave

```

Appendix A.4. Proposed Bi-LSTM and Attention Mechanism

```

import tensorflow as tf
from tensorflow.keras.layers import Input, Dense, Dropout, Bidirectional, LSTM
from tensorflow.keras.models import Model
def attention_layer(inputs):
    score = Dense(1, activation = 'tanh')(inputs)
    weights = tf.nn.softmax(score, axis = 1)
    context = tf.reduce_sum(inputs * weights, axis = 1)
    return context
def build_model(input_shape):
    inp = Input(shape = input_shape)
    x = Bidirectional(LSTM(64, return_sequences = True))(inp)
    x = Dropout(0.3)(x)
    # Attention
    context = attention_layer(x)
    # Dense classifier
    x = Dense(32, activation = 'relu')(context)
    x = Dropout(0.3)(x)
    out = Dense(1, activation = 'sigmoid')(x)
    model = Model(inp, out)
    model.compile(optimizer = 'adam',
                  loss = 'binary_crossentropy',
                  metrics = ['accuracy'])
    return model

```

Appendix A.5.

```

import numpy as np
def mc_dropout_prediction(model, x, n_samples = 30):
    preds = []
    for _ in range(n_samples):
        preds.append(model(x, training = True))
    preds = np.array(preds)
    mean_prediction = preds.mean(axis = 0)
    uncertainty = preds.std(axis = 0)
    return mean_prediction, uncertainty

```

References

1. Gutenberg, B.; Richter, C.F. Frequency of earthquakes in California. *Bull. Seismol. Soc. Am.* **1944**, *34*, 185–188. [\[CrossRef\]](#)
2. Varotsos, P.A.; Sarlis, N.V.; Skordas, E.S. Natural time analysis: The new view of time. *Phys. Rev. E* **2011**, *83*, 021110. [\[CrossRef\]](#)
3. Peng, C.K.; Havlin, S.; Stanley, H.E.; Goldberger, A.L. Mosaic organization of DNA nucleotides. *Phys. Rev. E* **1994**, *49*, 1685–1689. [\[CrossRef\]](#)
4. Kantelhardt, J.W.; Zschiegner, S.A.; Koscielny-Bunde, E.; Havlin, S.; Bunde, A.; Stanley, H.E. Multifractal detrended fluctuation analysis of nonstationary time series. *Phys. A* **2002**, *316*, 87–114. [\[CrossRef\]](#)
5. Pignatelli, A.; Petrucci, C.; Vignoli, V.; Caracciolo, F.D.; Console, R. A machine learning approach to distinguish earthquakes from explosions. *J. Seismol.* **2025**, *29*, 525–534. [\[CrossRef\]](#)
6. Lomax, A.; Michelini, A.; Jozinovic, D. Rapid earthquake characterization using single-station waveforms and a convolutional neural network. *Seismol. Res. Lett.* **2019**, *90*, 1089–1098. [\[CrossRef\]](#)
7. Ren, H.; Cromwell, E.; Kravitz, B.; Chen, X. Using long short-term memory models to fill data gaps in hydrological time series. *Hydrol. Earth Syst. Sci.* **2022**, *26*, 1727–1740. [\[CrossRef\]](#)
8. Yasemi, S.; Khalili, Y.; Sanati, A.; Bagheri, M. Carbon capture and storage: Application in the oil and gas industry. *Sustainability* **2023**, *15*, 14486. [\[CrossRef\]](#)
9. Perol, T.; Gharbi, M.; Denolle, M. Convolutional neural network for earthquake detection and location. *Sci. Adv.* **2018**, *4*, e1700578. [\[CrossRef\]](#)

10. Nicolis, O.; Plaza, F.; Salas, R. Forecasting intensity and location of seismic events using deep learning. *Spat. Stat.* **2021**, *42*, 100442. [CrossRef]
11. Yu, P.; Zhu, T.; Marone, C.; Elsworth, D. DASEventNet: AI-based microseismic detection on distributed acoustic sensing data from the Utah FORGE well. *J. Geophys. Res. Solid Earth* **2024**, *129*, e2024JB029102. [CrossRef]
12. Zhu, W.; Biondi, E.; Li, J.; Yin, J.; Ross, Z.E.; Zhan, Z. Seismic arrival-time picking on distributed acoustic sensing data using semi-supervised learning. *Nat. Commun.* **2023**, *14*, 8192. [CrossRef]
13. Lapins, S.; Butcher, A.; Kendall, J.-M.; Hudson, T.S.; Stork, A.L.; Werner, M.J.; Gunning, J.; Brisbourne, A.M. DAS-N2N: Machine learning distributed acoustic sensing signal denoising without clean data. *Geophys. J. Int.* **2024**, *236*, 1026–1040. [CrossRef]
14. Yin, J.; Soto, M.A.; Ramírez, J.; Kamalov, V.; Zhu, W.; Husker, A.; Zhan, Z. Real-data testing of distributed acoustic sensing for offshore earthquake early warning. *Seism. Rec.* **2023**, *3*, 269–277. [CrossRef]
15. Chen, X. An Event recognition method for distributed acoustic sensing based on convolutional neural network. *Front. Artif. Intell. Appl.* **2024**, *231*, 221–229. [CrossRef]
16. Han, S.; Huang, M.F.; Li, T.; Fang, J.; Jiang, Z.; Wang, T. Deep learning-based intrusion detection and impulsive event classification for distributed acoustic sensing across telecom networks. *J. Light. Technol.* **2024**, *42*, 4167–4176. [CrossRef]
17. Huot, F.; Lellouch, A.; Given, P.; Luo, B.; Clapp, R.G.; Nemeth, T.; Nihei, K.T.; Biondi, B. Detection and characterization of microseismic events from fiber-optic DAS data using deep learning. *arXiv* **2022**, arXiv:2203.07217. [CrossRef]
18. Başar, D.; Çelik, R.N. A Hybrid CNN–LSTM Architecture for Seismic Event Detection Using High-Rate GNSS Velocity Time Series. *Sensors* **2026**, *26*, 519. [CrossRef] [PubMed]
19. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5998–6008. [CrossRef]
20. Mousavi, S.M.; Ellsworth, W.L.; Zhu, W.; Chuang, L.Y.; Beroza, G.C. Earthquake transformer—An attentive deep-learning model for simultaneous earthquake detection and phase picking. *Nat. Commun.* **2020**, *11*, 3952. [CrossRef]
21. Mousavi, S.M.; Sheng, Y.; Zhu, W.; Beroza, G.C. STanford EArthquake Dataset (STEAD): A global data set of seismic signals for AI. *IEEE Access* **2019**, *7*, 179464–179476. [CrossRef]
22. Jousset, P.; Reinsch, T.; Ryberg, T.; Blanck, H.; Clarke, A.; Aghayev, R.; Hersir, G.P.; Henningses, J.; Weber, M.; Krawczyk, C.M. Dynamic strain determination using fibre-optic cables allows imaging of seismological and structural features. *Nat. Commun.* **2018**, *9*, 2509. [CrossRef]
23. Stork, A.L.; Baird, A.F.; Horne, S.A.; Naldrett, G.; Lapins, S.; Kendall, J.M.; Wookey, J.; Verdon, J.P.; Clarke, A.; Williams, A. Application of machine learning to microseismic event detection in distributed acoustic sensing data. *Geophysics* **2020**, *85*, KS149–KS160. [CrossRef]
24. Celia, M.A.; Bachu, S.; Nordbotten, J.M.; Bandilla, K.W. Status of CO₂ storage in deep saline aquifers with emphasis on modeling approaches and practical simulations. *Water Resour. Res.* **2015**, *51*, 6846–6892. [CrossRef]
25. Nanda, N.C. *Seismic Data Interpretation and Evaluation for Hydrocarbon Exploration and Production*; Springer: Cham, Switzerland, 2021. [CrossRef]
26. How Does Carbon Capture and Utilization Work? Available online: <https://carboncredits.com/how-does-carbon-capture-and-utilization-work/> (accessed on 19 February 2026).
27. Li, N.; Deng, J.; Yang, L.Q.; Goldfarb, R.J.; Zhang, C.; Marsh, E.; Lei, S.B.; Koenig, A.; Lowers, H. Paragenesis and geochemistry of ore minerals in the epizonal gold deposits of the Yangshan gold belt, West Qinling, China. *Miner. Depos.* **2014**, *49*, 427–449. [CrossRef]
28. McColpin, G.R. Surface deformation monitoring as a cost effective MMV method. *Energy Procedia* **2009**, *1*, 2079–2086. [CrossRef]
29. Bai, S.; Kolter, J.Z.; Koltun, V. An empirical evaluation of generic convolutional and recurrent networks. *arXiv* **2018**, arXiv:1803.01271. [CrossRef]
30. Zhu, W.; Beroza, G.C. PhaseNet: A Deep-Neural-Network-Based Seismic Arrival-Time Picking Method. *Geophys. J. Int.* **2019**, *216*, 261–273. [CrossRef]
31. Iqbal, E.N.; Asif, R. Analyzing applicant’s pre-admission data and identifying applicant’s performance in pre-admission test using data mining techniques. In Proceedings of the 3rd International Conference on Innovations in Computer Science & Software Engineering (ICONICS), Karachi, Pakistan, 14–15 December 2022; pp. 1–11. [CrossRef]
32. Supervised Learning Model Figure. Available online: https://www.researchgate.net/figure/Supervised-Learning-Model-20_fig2_328146111 (accessed on 19 February 2026).
33. Scikit-Learn Astronomy Tutorial: General Concepts. Available online: https://ogrisel.github.io/scikit-learn.org/sklearn-tutorial/tutorial/astronomy/general_concepts.html (accessed on 19 February 2026).
34. Ortiz, J.; Urrutia, L.; Vivas, J.C. Seismic Event Detection in Venezuela Using a CNN-Based Approach. Universitat Politècnica de Catalunya and Universidad Central de Venezuela, Barcelona, Spain. 2022; *manuscript in preparation*.
35. Mousavi, S.M.; Beroza, G.C. Deep-learning seismology. *Science* **2023**, *377*, eabm4470. [CrossRef]
36. Fawcett, T. An introduction to ROC analysis. *Pattern Recognit. Lett.* **2006**, *27*, 861–874. [CrossRef]

37. Rashid, A.; Tackie-Otoo, B.N.; Abdul Latiff, A.H.; Otchere, D.A.; Jamaludin, S.N.F.; Asfha, D.T. Research advances on distributed acoustic sensing technology for seismology. *Photonics* **2025**, *12*, 196. [[CrossRef](#)]
38. Asfha, D.T.; Abdul Latiff, A.H.; Otchere, D.A.; Tackie-Otoo, B.N.; Babikir, I.; Rafi, M.; Riyadi, Z.A.; Putra, A.D.; Adeniyi, B.A. Mechanisms of sand production prediction—A review and the potential for fiber optic technology and machine learning in monitoring. *J. Petrol. Explor. Prod. Technol.* **2024**, *14*, 2577–2616. [[CrossRef](#)]

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.