A Transformer-Based Classification System for Volcanic Seismic Signals

A Preprint

Anthony Rinaldi
Department of Statistical and Actuarial Sciences
Western University
London, Canada
arinald4@uwo.ca

Cindy Mora Stock
Department of Earth Sciences
Western University
London, Canada
cmorasto@uwo.ca

Cristián Bravo Roman
Department of Statistical and Actuarial Sciences
Western University
London, Canada
cbravoro@uwo.ca

March 11, 2022

Abstract

Monitoring volcanic events as they occur is a task that, to this day, requires significant human capital. The current process requires individuals to monitor seismographs around the clock, making it extremely labour-intensive and inefficient. The ability to automatically classify volcanic events as they happen in real-time would allow for quicker responses to these events by the surrounding communities. Timely knowledge of the type of event that is occurring can allow these surrounding communities to prepare or evacuate sooner depending on the magnitude of the event. Up until recently, not much research has been conducted regarding the potential for machine learning (ML) models to supplement or substitute human monitoring of volcanoes. Recent initiatives in this field have demonstrated that it is possible to classify volcanic events using ML techniques. Additionally, recent research in general signal processing has shown that the novel technique of multi-headed self-attention (MHSA), used in natural language processing (NLP), is also useful in signal analysis. In this report, we seek to apply MHSA to create a deep neural network (DNN) that can automatically classify volcanic events. Our proposed model architecture provides minor improvements over existing approaches on pre-processed data. When considering raw signals coming directly from monitoring stations, our model outperforms existing approaches by a much greater margin.

Keywords Volcanic seismicity · Deep neural networks · Time series analysis

1 Introduction

Volcanoes pose a serious threat to the communities surrounding them and being able to monitor the volcanic seismic signals more efficiently and timely would provide safety benefits to these communities. Prompt identification of the volcanic event allows for quicker responses by the local communities; thereby reducing the economic consequences and risk to human life. The four types of volcanic events are Long-period Earthquake (LP), Tremor (TR), Tectonic (TC), and Volcano-tectonic (VT).

LP events are caused by cracks resonating as liquids and gases move towards the surface. They occur as part of the normal background seismicity at some volcanoes and they do not necessarily indicate that an eruption is imminent. TR events are characterized by high-amplitude seismic signals caused by the extended flow of magma movement through cracks, the occurrence of VT and LP events so close together that they can not be visually separated, and eruptions. TC events are not related to volcanic activity, rather are a result of the dynamics of geological faults (Chouet, 1996). Lastly, VT events are brittle failures of rock. VTs occur due to "normal" tectonic forces, stresses caused by moving magma, and the movement of fluids through pre-existing cracks.[1] Examples of these four different events can be seen in Figure 1.

Traditional methods of signal processing and analysis are not powerful enough to fully automate the classification process, rather they serve as supplemental information for the individuals analyzing the signals. However, these humans still have errors in their classification efforts due to the similarities between the signals of the different events (as seen in Figure 1). The ideas proposed in this report have the potential to replace the individuals that currently analyze volcanic seismic signals for potential events. The proposed model could also be used to assist geologists that monitor volcanoes during eruptions; when even a large team lacks the ability and time to classify all transpiring events. This is especially important for lower-budget monitoring stations that lack human capital and would benefit from a powerful model to assist in event detection. Deploying a model that could classify the events transpiring inside a volcano would provide real-time information to the neighbouring cities and would eliminate the human bias that is currently inherent in the process.

One of the major difficulties that makes automatic monitoring even more challenging is the *noise* inherent in the seismic signals. This seismic noise is all of the unwanted recorded energy that contaminates seismic signals. The sources of this noise include ambient sources, wave propagation related noise, data acquisition related noise, and data processing artifacts (Kumar and Ahmed, 2011).[2] The noise is not related to the activity of the volcano and can cause difficulty when modelling volcanic events since the true patterns are more difficult to identify. For example, while most of the signals may be in the range [-100,100], some observations can be as large in magnitude as 30,000. This may provide reasoning to pre-process the data before passing it to a machine learning model; however, this is something we hope to avoid. Pre-processing the data would mean that the model would not work for real-time analysis, which is the goal of this research. The ML model should rely completely on raw data fed from the seismic stations to successfully classify volcanic events in real time.

The model we propose will work with raw seismic signals, allowing for volcanic monitoring responses on-the-fly and directly from the monitoring stations to support the work of the analyst to monitor the large number of existing events.

Recent advances in both volcanic event classification and signal processing have motivated the efforts of this report. First, it was recently shown that DNNs, more specifically convolutional neural networks (CNN), have high accuracy in classifying volcanic seismic events (Canário et al., 2020). The approach used an architecture based on inputs of size 20x20x3 and a two-layered convolutional-only model, achieving good results. The researchers demonstrated that it is possible to classify volcanic seismic signals using ML techniques that have not been considered for the problem before. It is now understood that ML techniques are successful in classifying volcanic events, but it is still unknown what an optimal model architecture would look like considering a wider range of deep learning tools that are available today.

Additionally, it is not known if models trained on one volcano can be applied to classify events on different volcanoes. Each volcano has a slightly different "personality" represented by its seismic signal and it is not known how significant these differences are when considering event classification. Further, each volcano has multiple stations surrounding it collecting these seismic signals. It is unknown how signals originating from these different stations could be used in the prediction task. For example, the soil that a station is built on is one of the main factors affecting the signals at that station. Two stations at the same volcano can have very different seismic signals depending on where the two stations are built. To simplify the task, our work will only consider a single volcanic station. We may also explore the transferability of ML models between different stations at the same volcano.

[1] More details can be found at USGS: https://volcanoes.usgs.gov/vhp/earthquakes.html

[2] For more information on seismic noise see https://doi.org/10.1007/978-90-481-8702-7_146

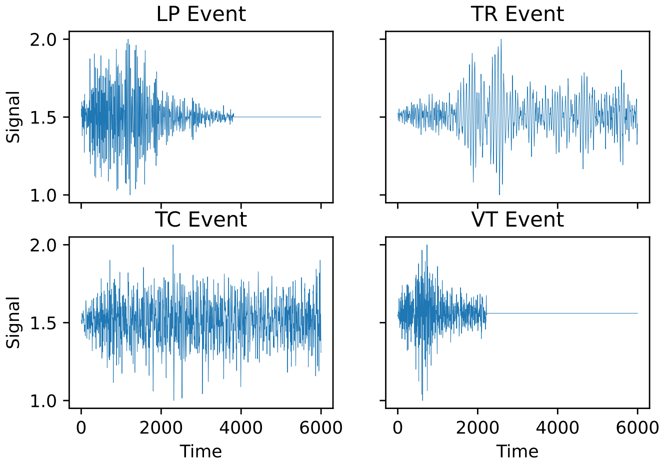

Figure 1: Signal Examples of the Four Events

Next, in a recent paper on earthquake detection, it was shown that the NLP technique of self-attention provides a substantial improvement over traditional approaches (Mousavi et al., 2020). The idea of self-attention was initially introduced by the Google Brain Team in 2017 for the use of NLP in translating sentences (Vaswani et al., 2017). Since sentences are sequences where the order is relevant, it is a natural extension that this novel idea of self-attention would be applied to other sequential data such as signals and time series. Mousavi et al. (2020) demonstrated that the above hypothesis is in fact true, and self-attention models perform well on other types of sequences. This motivated our ambition to apply self-attention to volcanic seismic signals since it performed so well on earthquake seismic signals. Furthermore, is has also been shown that the more complex *multi-headed self-attention* works well with time series data (Korangi et al., 2021). The research in this paper shows that MHSA works well with financial time series data and provides further evidence to support the work in this paper.

Given the numerous advances in the field of deep learning, we believe that we can address the automatic volcano monitoring problem that has proven difficult thus far. The objective of this paper is to create a multi-headed self-attention-based deep learning model to automatically classify volcanic seismic signals. After successfully developing the model we will ask the following questions:

- Does the model outperform the best standing CNN approach to volcanic seismic signal classification?
- What information can we extract from the fitted model to determine which sections of the input sequence are the most important?
- Can we apply this fitted model to other volcanoes and realize similar model performance?

In this report, we propose a DNN that depends solely on raw seismic data and outperforms all existing model architectures aimed at automatic volcanic monitoring.

The paper is organized as follows: Section 2 will introduce existing literature on the topic, Section 3 will talk about the data used in the report, Section 4 will introduce the methodology used in the report, Section 5 will talk about the model training specifications, Section 6 will display the results of the paper and Section 7 will talk about the plan for completing the report.

2 Literature Review

Volcanic seismic monitoring is a difficult task and there have been many attempts to identify the best analysis system. First, many statistical techniques have been applied to the problem. Some examples of existing approaches include feature extraction and modelling using Gaussian Mixture Models (Cortés et al., 2014), Bayesian inference (Bartolini et al., 2016), autocorrelation functions (Langer et al., 2006), wavelet transforms (Erlebacher and Yuen, 2004), and both amplitude and phase statistics (Joevivek et al., 2010). The main fallback of statistical techniques is that feature extraction is typically separated from statistical modelling. These models do not provide a complete pipeline for automatic volcanic monitoring. Further, the statistical models typically require significant data pre-processing which limits the real-time use of these models. These drawbacks have given rise to the research of machine learning techniques that have the capability to provide a complete pipeline.

Research in machine learning methods to automatically monitor volcanic activity is much less developed than the use of statistical techniques. Some of the techniques researched thus far include Support Vector Machines (Masotti et al., 2006), Hidden Markov Models (Ibáñez et al., 2009), multi-channel correlation Hidden Markov Models (Quang et al., 2015), extreme learning machines (Zhang et al., 2019), and neural networks (Esposito et al., 2013). These machine learning models typically provide increased performance above the basic statistical techniques. However, they too have only been considered with significant pre-processing, which limits the ability for their use in real-time monitoring.

Until recently, deep neural networks have never been considered for the task. The first consideration of deep neural networks for volcanic monitoring was Canário et al. (2020) using convolutional neural networks on spectrograms. The researchers' inspiration led them to create a model comprised of entirely CNN blocks (along with the necessary pooling and dropout layers in between). Their model performed exceptionally well with an accuracy of 97.52%. However, this model requires significant pre-processing. Rather than passing raw seismic signals to the model, the signals would go through a process called *wavelet transformation* to be turned into spectrograms. These two-dimensional images were then passed to the model for classification. This pre-processing step makes the model unusable for real-world volcanic monitoring. It would be inefficient to constantly perform these wavelet transformations on the seismic signals before attempting to classify the signals. Secondly, the data used in the report have been cleaned and does not seem as "noisy" as seismic signals are when captured directly from monitoring stations. The appropriate filter, as well, is a changing decision that is taken given the specific behaviour of the volcano, so an automated method that can filter the signal accordingly would provide a significant amount of value. Lastly, the model only accepts fixed length inputs. Since volcanic event durations have a wide range (from seconds to hours) it would be beneficial to allow for variable-length input when classifying the event. We will aim to improve on these shortcomings by feeding the seismic signals directly to our model (i.e. no pre-processing), by training our model on a second dataset that is visually more noisy and realistic, and allowing for variable input size by use of CNN blocks.

One of the most recent advances in deep learning is the transformer architecture proposed by Google (Vaswani et al., 2017). Extending the use of this model from NLP to signal processing is natural since both domains work with sequential data. The transformer model has yet to be used in volcanic signal monitoring, which is the main goal of our work, but has been shown to work well with earthquake signals. Mousavi et al. (2020) implemented a combined earthquake detection and phase picking model (phase picking refers to the measurement of arrival times of distinct seismic phases). This report was the first in the geology domain to apply the idea of self-attention to signal processing. It successfully demonstrated that the idea of self-attention from the domain of NLP is applicable to other sequential data. This resulted in a model that outperformed all existing models in terms of earthquake detection ability. This paper justifies that the addition of self-attention is a worthy step up from current techniques of seismic signal analysis and signal processing in general. The natural extension we would like to make from this report is to confirm that self-attention is as successful on volcanic-seismic signals as it is with earthquake signals. Further, the paper considered only single-headed self-attention, whereas we will consider multi-headed self-attention; which we believe would improve our results given the different nature of the volcanic signal origin. Lastly, the paper considered a sequence-to-sequence model, but we will be considering a sequence-to-class model; a classification model.

Another successful implementation of a transformer model using self-attention is in the finance domain. Here too the input data is sequential (a time series) and this novel application provided benefits in predicting company bankruptcies. Korangi et al. (2021) gives further inspiration for our objective and justification for the expected results. The paper considers the use of an MHSA model for predicting the default of mid-cap companies. This model is applied to time-series data that is comparable to seismic signals in that they are both time-dependent. The paper shows that the MHSA model outperforms existing DNN model architectures and other forms of ML. We expect that, similar to this paper, we should see improved classification performance over traditional approaches.

3 Data

There will be two different datasets used for this task, both having been collected from the LAV station (see Figure 2) at the Llaima volcano in Chile. The Llaima volcano is considered one of the most dangerous volcanoes in South America and has been used in the past for studying automatic classifiers (Bhatti et al., 2016; Canário et al., 2020; Curilem et al., 2016). Both datasets will be using the Z-vertical component of the seismic signals captured from the LAV station, as suggested by the Southern Andean Volcano Observatory.

The first dataset is much *cleaner* and should give us a measure of the model performance when used over data that has already been processed by a human user, while the second contains much more *noise* and will give us a measure of the model performance over raw data.

3.1 Filtered Data

The first dataset contains 3592 signals collected from 2010 to 2016 sampled at 100 Hz. The signals are classified as one of four event types: TC, LP, TR, and VT. Each instance in the dataset represents a one-minute signal sampled at 100 Hz (10 milliseconds), giving 6000 values per instance. There are 1488, 1310, 490, and 304 observations of the TC, LP, TR, and VT classes respectively. The volcanic signals have also been slightly processed before their analysis, which is not entirely desired for this report, as mentioned previously. The signals have been filtered using a 10th-order Butterworth bandpass filter in range [1, 10] Hz and normalized by their maximum value (Canário et al., 2020). The goal of this paper is to produce a model that can classify volcanic events in real-time, thus the model must accept raw signals as input only, not filtered signals.

An initial model will be trained and optimized for this filtered data to ensure that the addition of self-attention provides improvements before attempting to train a transformer model using raw volcanic signals. If the experiments with the filtered data prove successful, the second dataset comprised of raw signals will be used to form a new model.

3.2 Raw Data

The second dataset contains 1033 LP events and 41 TR events, once again coming from the LAV station at the Llaima volcano in Chile. It contains observations from November 2009 until January 2010, each representing a signal sampled at 100 Hz. To create instances from the data, we will extract one-minute intervals of each event until the entire event has been captured. For example, an event that lasts for 1.5 minutes will be turned into two instances, one containing the first minute of the event and another containing the last 30 seconds of the event, plus the 30-second interval after the event, totalling the one-minute length needed for an instance.

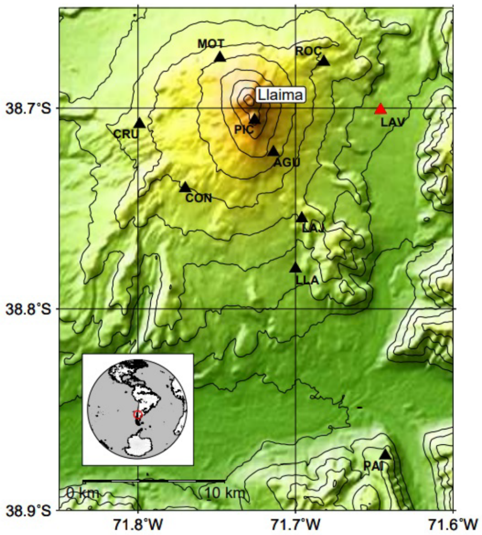

Figure 2: Location of LAV Station at Llaima

4 Methodology

In this section, we describe the DNN architecture that will be used to classify volcanic seismic signals. The transformer model architecture will be comprised of a combination of convolutional blocks, residual convolutional blocks, LSTM blocks and multi-headed self-attention blocks. An example of a model architecture that contains one of each of these blocks can be seen in Figure 3.

The model input is a vector of size N where N is the number of time-stamps for a single observation (i.e. 6000) and the output is a vector of size four, representing probabilities of the four event types LP, TR, TC and VT.

4.1 Convolution Layer

Convolutional neural networks have long been the golden standard for image processing, being used in the most powerful image recognition algorithms to date (Ciresan et al., 2012). Further, CNNs have also been used quite successfully for time-series data (Zhao et al., 2017).

Convolutional layers seek to find local patterns in the volcanic signals, only looking at small groups of the input sequence at a time. This is a typical first step in DNNs, extracting local features to be used in further processing.

The convolution layer also removes some of the temporal significance of the signals. After repetitive convolutions, features from the beginning of a signal can be carried to the end of the signal. However, given the size of our signals (6,000) and the low number of convolutions used, the temporal significance of signals is somewhat preserved.

4.2 Residual Convolutional Block

Next, residual neural networks (Res. CNN) have been shown to outperform the more traditional deep CNNs (He et al., 2016), inspiring their inclusion in our research. This model architecture was proposed to overcome the degradation in accuracy that arises when increasing the depth of deep CNNs. This is a problem because the model optimizers that are currently available do not easily converge for these deep CNNs. He et al. (2016) showed that it is easier to optimize the Res. CNNs than deep CNNs. These Res CNN. blocks do not increase model complexity and lead to improved performance in deep networks than stacked convolutional layers do.

Similar to the convolutional layer, the Res. CNN block looks for local patterns in the input sequence. After applying multiple convolutions, the Res. CNN adds the original input sequence to the output. Adding the input sequence after finding many locally important features ensures that the overall pattern of the sequence is not forgotten and local features are not over-emphasized.

4.3 LSTM

The choice to include LSTMs in our model is because of their ability to capture long-term dependencies in data. It has been shown that these models work well with sequential data and overcome the shortcomings of vanishing and exploding gradients in traditional recurrent neural network frameworks (Sak et al., 2014).

LSTMs are a type of recurrent neural network (RNN). This means that information in them persists over time, unlike traditional neural networks where there is no time dependence preserved. This layer accounts for the order in which data is passed, and uses all previous points in a sequence in computation for the subsequent data point in a sequence. It does so through the use of a *memory cell*.

The memory cell returns two values: (1) the current output and (2) the current memory. For the purposes of the network, only the output value is passed to the next layer in the neural network. The cell takes as inputs (1) the input covariates (i.e. seismic signal) of the current timestamp, (2) the memory from the previous timestamp, and (3) the output of the previous timestamp. The two methods that control how the memory is controlled in the cell are the *forget gate* and the *input gate*. The forget gate determines how much of the previous memory the cell should keep and the input gate determines what memory the cell should keep from the current input. Finally, the *output gate* determines how the new cell memory, the previous output, and the current input should combine to create the new output of the cell (Gers et al., 2000).

The LSTM layer focuses on the positional encoding of the sequence. It determines where the important information is in the input sequence by passing through values sequentially as they appear in the sequence. This is done so that the model learns the positional encodings of the input sequence, rather than using a fixed formula for positional encoding (Cho et al., 2014).

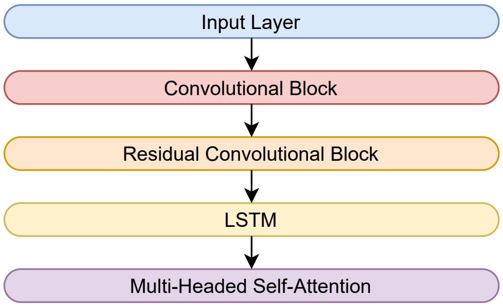

Figure 3: Example model architecture

4.4 Multi-Headed Self-Attention

Multi-Headed Self-Attention (MHSA) has been mainly used for natural language processing (NLP) thus far. These models find relationships between the input sequence and itself. This block will keep the positional encoding of the sequence intact, while adding the *attention scores* that are calculated based on these inter-sequence relationships.

Each *head* of this block will focus on different patterns in the input sequence, allowing the model to learn very complex non-linear relationships.

4.5 Model Architecture

The original model architecture included one of each of the above blocks/layers. This simple model architecture was implemented and subsequently improved by adding more of the above layers. No cross-validation was used for tuning model parameters since the model performed exceptionally already. The final model architecture can be seen in Figrue 4. This final model has 992,560 trainable parameters. Additionally, a decomposition of the Residual CNN block can be found in Figure 5.

The model and Res. CNN architecture introduces a few new blocks that we will now describe. The max-pooling layer reduces the dimensionality of the data to help reduce over-fitting. It does so by retaining only the most important information available in each local section of the sequence (the local maxima). Next, the spatial dropout layer sets entire feature maps from the convolutional layers to zero. Similarly, the dropout layer sets nodes to zero in dense layers. Both of these layers help remove the dependence between feature maps (or dense layer neurons) and force the model to learn by condensing information through only specific pathways. Lastly, the batch normalization layer normalizes (subtract mean and divide by standard deviation) each feature across the entire training batch. Neural networks work better with normalized values and this form of normalization has been shown to work exceptionally well (Ioffe and Szegedy, 2015).

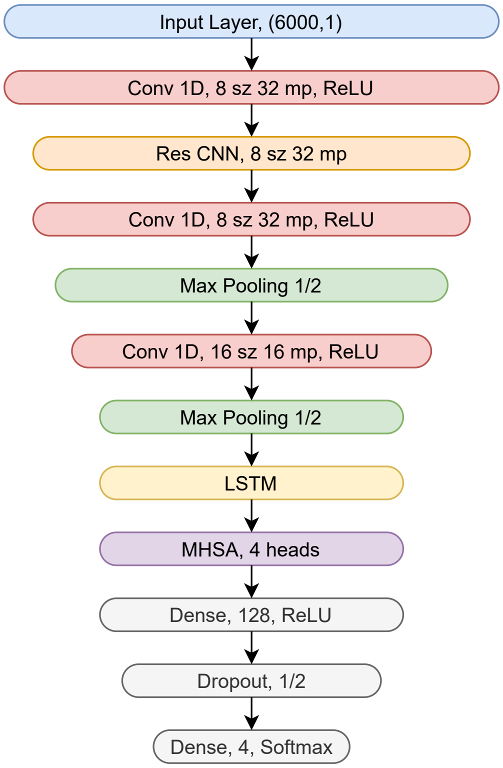

Figure 4: Model Architecture[3]

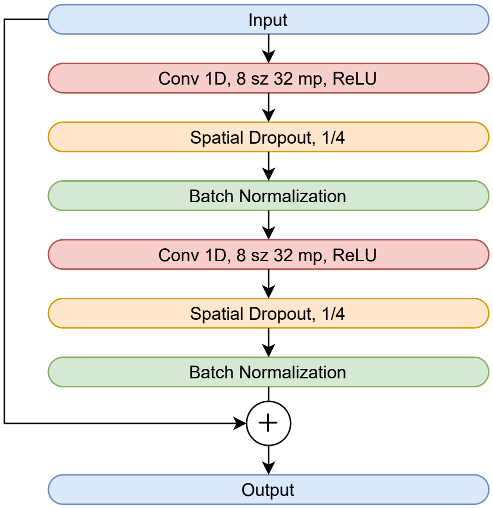

Figure 5: Residual Convolutional Block Architecture

[3] X sz Y mp denotes a layer with filter size of X and Y filter maps.

5 Model Training

5.1 Performance Metric

The metric we considered to evaluate the model performance was **classification accuracy**. Since no parameter tuning was necessary, the metric is not vitally important because it is used solely to assess model performance during training, rather than used to select the best set of parameters. If parameter tuning is required in the future, however, there are many other metrics to consider such as F1 Score, AUROC (Area Under Receiver Operating Characteristic Curve), Cross-Entropy Loss, and Cohen's Kappa Score (Cohen, 1960).

The second dataset contains very unbalanced classes, so we use the AUROC as the metric of choice. The chosen metric will be used to identify the best cross-validation results, thus the best set of parameters for the model.

5.2 Loss Function

Here we specify the loss function we use to train our models and find the optimal weights of the parameters. We use the *categorical cross-entropy* loss function. With y_i representing the true binary variable for class i, $\hat{y}_i$ representing the predicted probability for class i, and $f()$ representing the *softmax* function in the last dense layer, the loss function is

$$loss = -\sum_{i=1}^{4} y_i log(f(\hat{y}_i)) \tag{1}$$

This is the typical loss function for multi-class classification models.

5.3 Training on processed data

The model architecture described above has been implemented in `Python` using the `Tensorflow` library. The model was trained on the *Graham supercomputer* using two Tesla V100 GPUs. Each training epoch takes between one and three seconds, making the overall training time for 500 epochs under 30 minutes. This makes extensive parameter tuning possible in the future since training time is not a major constraint. The current model achieves a validation accuracy of 96.4%, outperforming all existing approaches. The model training history can be seen in Figure 6.

From Figure 6 we can see that the model architecture we proposed works very well for the task. The validation accuracy tends to be greater than the training accuracy, indicating that the model has not been overfitted. The convergence of the model parameters also happens quite smoothly and quickly in just under 200 epochs.

When comparing our proposed model to the model proposed by Canário et al. (2020), it initially seems that we have not achieved the same success that they had at 97.52% accuracy. However, to make the results comparable, we applied the same train-test split that was used on our model to the model that Canário et al. (2020) proposed. This allows for a more equal comparison since both models are trained and tested on identical data. With this approach, the Canário et al. (2020) model only achieves 94.5% accuracy. A summary of the training and testing results of both models can be seen in Table 1.

Table 1: Clean Model Comparison

Model	Training Accuracy	Testing Accuracy
Proposed Transformer Architecture	95.5%	96.4%
Canário et al. (2020) Model	93.3%	94.5%

5.3.1 Training Settings

To prevent overfitting the model on the clean dataset, we used early-stopping, where the model will stop fitting if the validation accuracy does not increase by 0.001 in 50 epochs. This ensures that the model will not solely increase training accuracy while validation accuracy suffers.

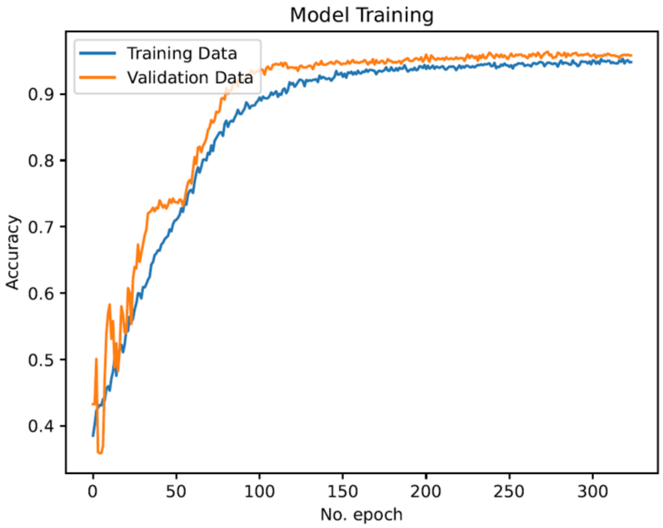

Figure 6: Clean Model Training and Validation Performance

5.4 Training on raw data

As mentioned in Section 1, there is a second dataset that is made up of raw seismic signals. All aforementioned steps have been performed on the filtered signals (first dataset). We will use a similar model architecture to identify if the model can fit the raw data well.

The model training process was more important for the raw data and we needed to carefully determine the number of epochs to use for both the Canário et al. (2020) model and our proposed model. To solve this problem, we performed 3-fold stratified cross-validation to identify the best number of epochs. We split the data into a training set (75%) and a testing set (25%). Within the train set, we performed stratified cross-validation, which ensures that each fold contains the same portion of true and negative cases as the original data. The average cross-validation score for the Canário et al. (2020) model and our proposed model are provided in Figure 7 and Figure 8 respectively. The model proposed by Canário et al. (2020) has difficulty learning on the raw data, as seen by its average cross-validation score. The longer we train the model, the worse the performance gets. With the transformer architecture we propose, we see that the model is learning since the average cross-validation score increases as we train the model. Using the cross-validation results, we find the optimal number of epochs for Canário et al. (2020) and our proposed model to be 8 and 1826, respectively. We then refit both models on the entire training set using these optimal epoch values and evaluated the fitted model on the test set.

The training process for each cross-validation fold is the same as it was on the first dataset, with some minor changes. First, we need to include sample weights in our training since we have very unbalanced data. This means that the model will be penalized more for miss-classifying the minority class, rather than miss-classifying the majority class. This ensures that the model does not only predict the majority class, which would result in a model with high accuracy, but a useless model nonetheless. Another difference from the original training procedure is that the raw dataset only has two classes. The loss function is now summed for two classes rather than four, but this is a minor technicality. Finally, the second dataset trained at a slower pace so we allowed for a maximum of 3000 epochs until its validation performance was no longer increasing.

We show the learning curve of both models in Figure 9 and Figure 10. The model proposed by Canário et al. (2020) only has 8 epochs to learn, thus it does not learn much and the performance is poor. We can see that our proposed model trains well, but not as well as it does on the clean data. The model has a greater struggle learning the correct weights, as seen by the volatility in the curve. However, it still achieves respectable performance in terms of AUC. Overfitting is not a concern since we used a cross-validation approach for selecting the number of training epochs.

Using raw data, we now see a large performance difference between the two models. Table 2 shows the drastic difference between the two. Our model achieves a test set AUROC of 62.4%, while the Canário et al. (2020) model only achieves 50.0% AUROC. This indicates that the latter model is predicting classes randomly. Although their model performed quite well on the clean data, when faced with raw data the performance decreases drastically.

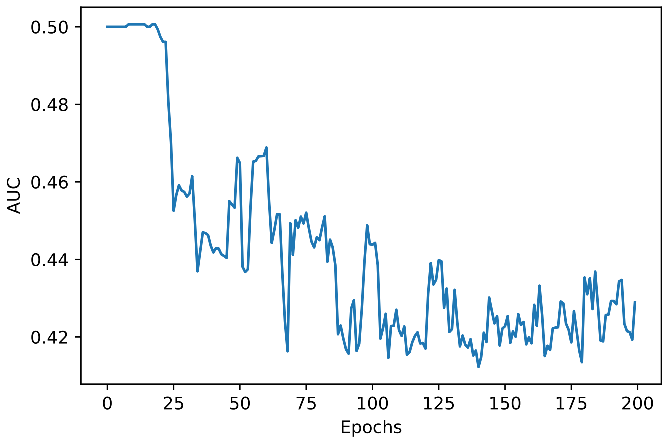

Figure 7: Canário et al. (2020) Model CV Performance

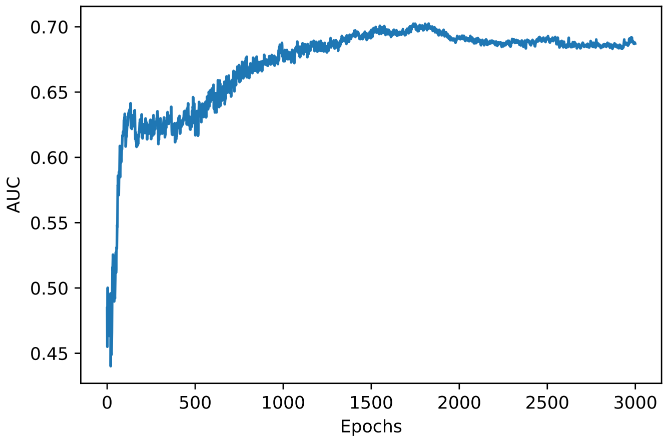

Figure 8: Proposed Model CV Performance

Table 2: Raw Model Comparison

Model	Training AUROC	Testing AUROC
Proposed Transformer Architecture	61.9%	62.4%
Canário et al. (2020) Model	48.7%	50.0%

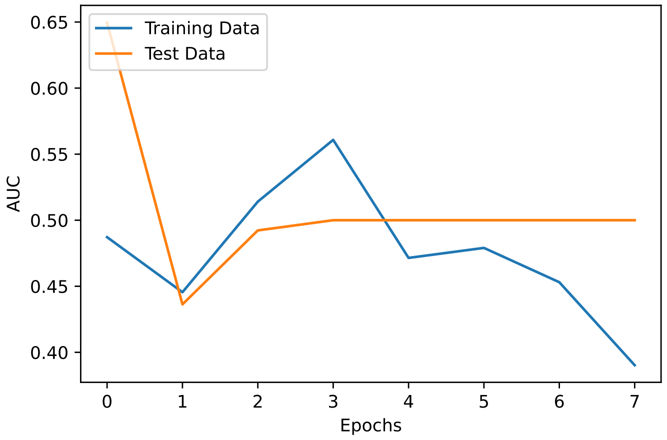

Figure 9: Canário et al. (2020) Raw Model Training and Validation Performance

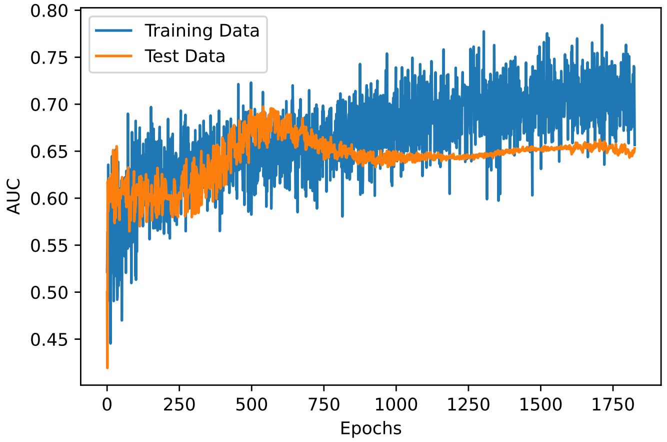

Figure 10: Raw Model Training and Test Performance

6 Preliminary Results

6.1 Processed Data

As mentioned in Section 5, the model trained on filtered data achieved an accuracy of 96.4%. The attention maps of the model are included in Figures 7 through 10. The plots are generated by calculating the average attention scores for each of the four attention heads when considering the four different event classes. Yellow colours in the plots indicate greater attention values and indigo colours in the plot indicate lower attention values. LP and VT events have similar attention plots, which makes sense since these two classes of events share similar characteristics. The TR and TC events have similar attention plots and these events share similar characteristics as well.

For LP and VT events, the first head places great emphasis on the beginning of the sequence. The two events diverge in that the remaining three heads have slightly different attention scores for the latter part of the sequence. For TR and TC events, the first head places great emphasis on the middle part of the sequence. These two events differ in the remaining three heads as well, where the attention scores for the middle part of the sequence are different.

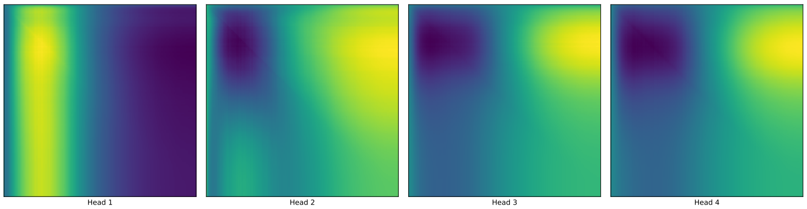

Figure 11: LP Events Attention Plot

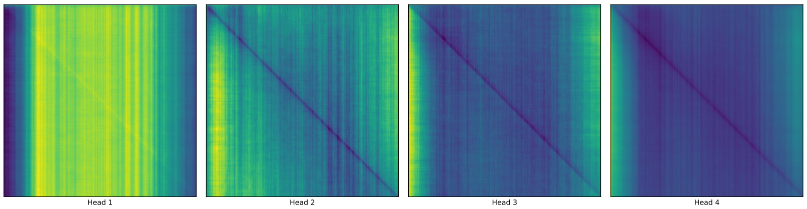

Figure 12: TR Events Attention Plot

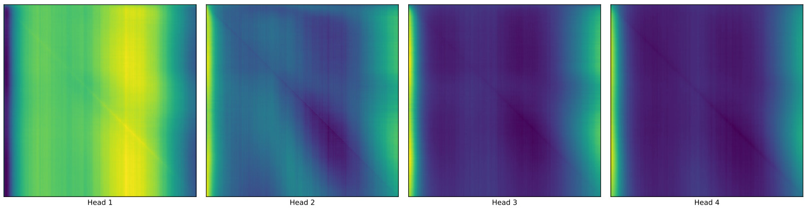

Figure 13: TC Events Attention Plot

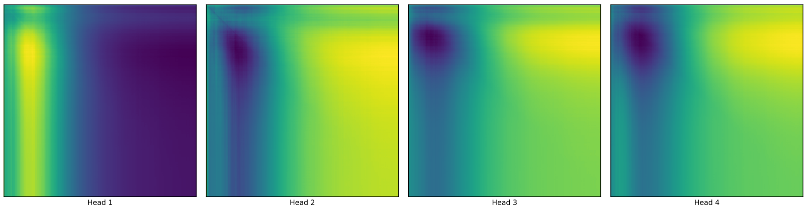

Figure 14: VT Events Attention Plot

6.2 Raw Data

Similarly, we calculate the average attention scores for the model trained on the raw data. This model only contains two attention heads and the response variable only contains two classes, so there are fewer plots.

We can see similar results in the attention scores from the clean and raw data. For the LP events, the heads place the most importance on the very beginning of the sequence. However, comparing the heads for the clean and raw data we see a big difference. One possible explanation for this is that with the clean data, the first head can pick up all relevant information at the beginning of the sequence, so the remaining heads focus on the other parts of the sequence. With the raw data, more heads are needed to pick up the important information at the beginning of the sequence. If we think of adding the plots for the first and second LP heads for the raw data, they seem to sum closely to the first head for the clean data. This supports the conclusion that the heads can individually extract more information from the clean data than the raw data. For the TR events, the heads place importance on small intervals of the sequence across the entire length of the sequence. The intervals alternate between significant and insignificant. This is similar to what we saw with the clean data, except the first head with the clean data did not pay much attention to the very beginning of the sequence, whereas the first head for the raw data does. The second heads are more similar, with the second head of the clean data paying a bit more attention to the beginning of the sequence than the second head of the raw data.

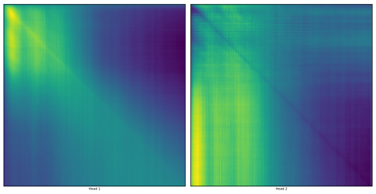

Figure 15: LP Raw Events Attention Plot

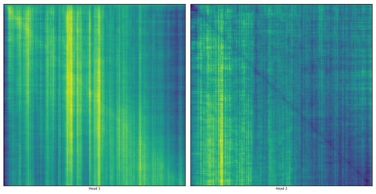

Figure 16: TR Raw Events Attention Plot

The raw training set suffered a severe class imbalance, so to get more informing results we performed a bootstrap on the test set. We evaluated the models on the entire test set to get the predicted probabilities for each class. Using these predicted probabilities, we performed a bootstrap, calculating the AUROC for every bootstrap iteration. This gives us a better idea of the distribution of the model performance. Figure 17 and Figure 18 show the bootstrap results for the Canário et al. (2020) model and our proposed model, respectively. These results confirm that our model performance is superior since the distribution is centred at a higher AUROC value. This means that in most situations, our proposed model performs better.

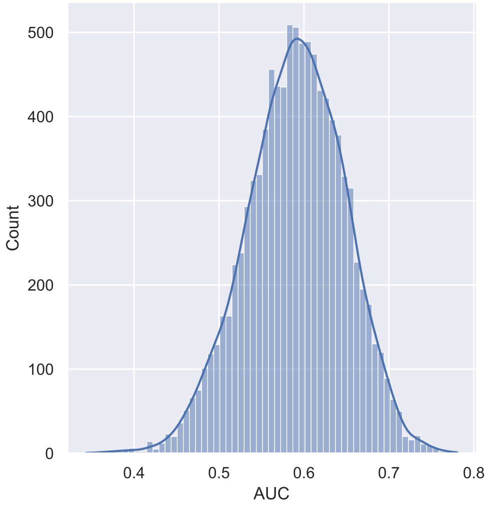

Figure 17: Canário et al. (2020) Test Set Bootstrap

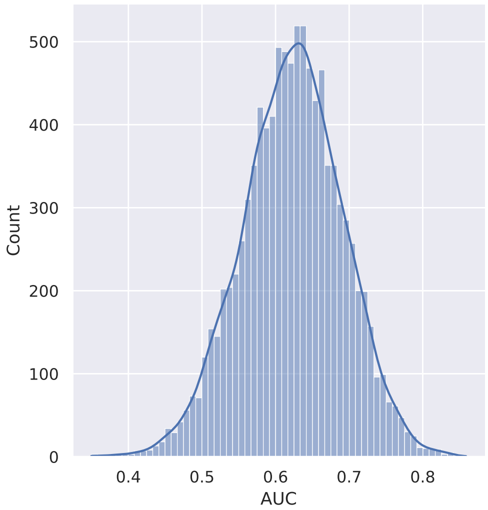

Figure 18: Proposed Model Test Set Bootstrap

7 Conclusion

The model we propose provides slight improvements over previous methods on pre-cleaned data and provides larger improvements on raw data coming directly from seismic monitoring stations. Where our model will excel is in stations where human capital is limited and there is difficulty in identifying all volcanic events. Of course, in a station with sufficient resources and the ability to clean all signals recorded, there is only a slight difference between our proposed model and the model proposed by Canário et al. (2020). In this case, there is not a material difference between the two models and either will provide highly accurate results. However, where the models differ is in stations where clean data is not readily available and the humans currently monitoring are struggling to identify all events. As mentioned previously, the inability to identify all events is exacerbated during volcanic eruptions since the number of events increases exponentially. In these such stations, our model will provide substantial improvements over existing models.

Even though the model does not boast 90%+ accuracy on raw data, it is still able to supplement the existing geologists working in monitoring stations. For example, the model could be loaded onto a chip that would be installed at one of these monitoring stations. With direct access to the raw data coming from the station, the model would be able to classify events as they occur. The human monitor would then simply need to confirm or reject the predictions of the model, rather than sifting through all the raw signals by hand. Since the model boasts around 62% AUROC, one would only sometimes need to intervene with the predictions that the model makes.

8 Plan of Work

The work plan for this report is summarized in Figure X.

Firstly, the first model is already finalized. This model included one of each of the blocks of interest. I.e. it contained one CNN block, one residual CNN block, one LSTM block, and one MHSA block. As mentioned in Section 4, this model has been completed already.

Throughout November and December, the model was iterated until its final state. During this period the correct number of repetitions of each block was selected, along with the parameters of the model such as CNN filter sizes and maps, dropout and max-pooling rates, choice of uni-directional or bi-directional LSTM, and the number of self-attention heads. As mentioned in the Section 4, this is also already completed.

During January, we pre-processed the new data that has been received and performed parameter tuning on a model trained on this new data.

Next, during January and February (and potentially March) we analyzed the final model to understand its inner workings. This included looking at the attention maps to discern where the model is "looking" for patterns.

Time was not permitting, so we did not have the chance to investigate the transferability of the model.

Finally, over the entire course of January to April, we will also be focusing our efforts on producing the finalized report, with the ultimate goal being a presentable and publishable paper.

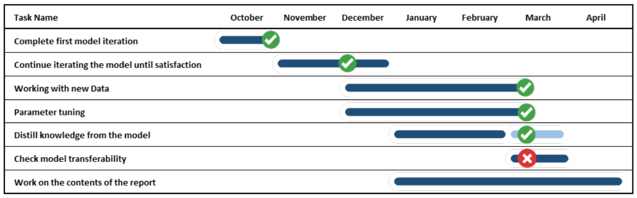

Figure 19: Work Plan Gantt Chart

References

Bernard A. Chouet. Long-period volcano seismicity: Its source and use in eruption forecasting. *Nature*, 380(6572): 309–316, 1996. doi:10.1038/380309a0.

Dhananjay Kumar and Imtiaz Ahmed. *Seismic Noise*, pages 1157–1161. Springer Netherlands, Dordrecht, 2011. ISBN 978-90-481-8702-7. doi:10.1007/978-90-481-8702-7_146. URL https://doi.org/10.1007/978-90-481-8702-7_146.

João Paulo Canário, Rodrigo Mello, Millaray Curilem, Fernando Huenupan, and Ricardo Rios. In-depth comparison of deep artificial neural network architectures on seismic events classification. *Journal of Volcanology and Geothermal Research*, 401:106881, 2020. doi:10.1016/j.jvolgeores.2020.106881.

S. Mostafa Mousavi, William L. Ellsworth, Weiqiang Zhu, Lindsay Y. Chuang, and Gregory C. Beroza. Earthquake transformer—an attentive deep-learning model for simultaneous earthquake detection and phase picking. *Nature Communications*, 11(1), 2020. doi:10.1038/s41467-020-17591-w.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Conference on Neural Information Processing Systems*, 31, 2017.

Kamesh Korangi, Christophe Mues, and Cristián Bravo. A transformer-based model for default prediction in mid-cap corporate markets. *European Journal of Operational Research*, 2021.

Guillermo Cortés, Luz García, Isaac Álvarez, Carmen Benítez, Ángel de la Torre, and Jesús Ibáñez. Parallel system architecture (psa): An efficient approach for automatic recognition of volcano-seismic events. *Journal of Volcanology and Geothermal Research*, 271:1–10, 2014. doi:10.1016/j.jvolgeores.2013.07.004.

Stefania Bartolini, Rosa Sobradelo, and Joan Martí. St-hasset for volcanic hazard assessment: A python tool for evaluating the evolution of unrest indicators. *Computers amp; Geosciences*, 93:77–87, 2016. doi:10.1016/j.cageo.2016.05.002.

H. Langer, S. Falsaperla, T. Powell, and G. Thompson. Automatic classification and a-posteriori analysis of seismic event identification at soufrière hills volcano, montserrat. *Journal of Volcanology and Geothermal Research*, 153 (1-2):1–10, 2006. doi:10.1016/j.jvolgeores.2005.08.012.

G. Erlebacher and D. A. Yuen. A wavelet toolkit for visualization and analysis of large data sets in earthquake research. *Computational Earthquake Science Part II*, page 2215–2229, 2004. doi:10.1007/978-3-0348-7875-3_8.

V. Joevivek, N. Chandrasekar, and Srinivas Yasala. Improving seismic monitoring system for small to intermediate earthquake detection. *International Journal of Computer Science and Security*, 4:308–315, 07 2010.

M. Masotti, S. Falsaperla, H. Langer, S. Spampinato, and R. Campanini. Application of support vector machine to the classification of volcanic tremor at etna, italy. *Geophysical Research Letters*, 33(20), 2006. doi:10.1029/2006gl027441.

Jesús M. Ibáñez, Carmen Benítez, Ligdamis A. Gutiérrez, Guillermo Cortés, Araceli García-Yeguas, and Gerardo Alguacil. The classification of seismo-volcanic signals using hidden markov models as applied to the stromboli and etna volcanoes. *Journal of Volcanology and Geothermal Research*, 187(3-4):218–226, 2009. doi:10.1016/j.jvolgeores.2009.09.002.

Paul Bui Quang, Pierre Gaillard, Yoann Cano, and Munkhuu Ulzibat. Detection and classification of seismic events with progressive multi-channel correlation and hidden markov models. *Computers Geosciences*, 83:110–119, 2015. doi:10.1016/j.cageo.2015.07.002.

Jinyong Zhang, Ruochen Jiang, Biao Li, and Nuwen Xu. An automatic recognition method of microseismic signals based on eemd-svd and elm. *Computers amp; Geosciences*, 133:104318, 2019. doi:10.1016/j.cageo.2019.104318.

Antonietta M. Esposito, Luca D'Auria, Flora Giudicepietro, Teresa Caputo, and Marcello Martini. Neural analysis of seismic data: Applications to the monitoring of mt. vesuvius. *Annals of Geophysics*, 56(4), 2013. doi:10.4401/ag-6452.

Sohail Masood Bhatti, Muhammad Salman Khan, Jorge Wuth, Fernando Huenupan, Millaray Curilem, Luis Franco, and Nestor Becerra Yoma. Automatic detection of volcano-seismic events by modeling state and event duration in hidden markov models. *Journal of Volcanology and Geothermal Research*, 324:134–143, 2016. doi:10.1016/j.jvolgeores.2016.05.015.

Millaray Curilem, Fernando Huenupan, Daniel Beltrán, Cesar San Martin, Gustavo Fuentealba, Luis Franco, Carlos Cardona, Gonzalo Acuña, Max Chacón, and M. Salman Khan. Pattern recognition applied to seismic signals of llaima volcano (chile): An evaluation of station-dependent classifiers. *Journal of Volcanology and Geothermal Research*, 315:15–27, 2016. doi:10.1016/j.jvolgeores.2016.02.006.

D. Ciresan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012. doi:10.1109/cvpr.2012.6248110.

Bendong Zhao, Huanzhang Lu, Shangfeng Chen, Junliang Liu, and Dongya Wu. Convolutional neural networks for time series classification. *Journal of Systems Engineering and Electronics*, 28(1):162–169, Feb 2017. doi:10.21629/JSEE.2017.01.18.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. doi:10.1109/cvpr.2016.90.

Haşim Sak, Andrew Senior, and Françoise Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. *Interspeech 2014*, 2014. doi:10.21437/interspeech.2014-80.

Felix A. Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. *Neural Computation*, 12(10):2451–2471, 2000. doi:10.1162/089976600300015015.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics. doi:10.3115/v1/D14-1179. URL `https://aclanthology.org/D14-1179`.

Sergey Ioffe and Christian Szegedy. Accelerating deep network training b y reducing internal covariate shift. *Google Research*, Mar 2015. URL `https://arxiv.org/pdf/1502.03167.pdf`.

Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1): 37–46, 1960. doi:10.1177/001316446002000104.