Universiteit Utrecht

Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Infrastructuur en Waterstaat

Internship and MSc. thesis research

Utrecht University

Faculty of Geosciences
Earth Structure and Dynamics
Physics of the Solid Earth and Planets

In partnership with
Koninklijk Nederlands Meteorologisch Instituut

Optimisation and enhancement of deep learning model for seismo-acoustic event detection

and framework development for future implementation of characteristics recognition

---

Camille G. M. Chapeland
5712432

August 2020

Abstract

The seismic network in the Netherlands is densely distributed over the Groningen gas field and close to populated areas due to the restricted landmass of the country. Combined with the history of natural gas extraction, the detection system is affected by man-made noise and struggles to pick up smaller seismo-acoustic events. This thesis describes the approach to optimise and enhance a convolutional neural network (CNN) event recognition model which can detect seismo-acoustic events from a single-station input time-series. Input types combinations are compared and the best performing CNN model is trained using low-processed time-series and spectrogram inputs. The model learns to recognise whether the input contains noise, a seismic event or an event of a different type (acoustic event, explosion, etc.) with a 98.8% accuracy, comparable to state of the art models in literature. Despite the high classification accuracy, the layer-specific training behaviours of the model are explored to find that deep hidden layers may be under-trained. The model's detection rates over randomly selected full days is compared to the current SeisComP3 detection system, showing a strong reduction in false positive hits but an inability to detect events with a magnitude of 0.8 or below. Recurrent neural networks architectures are also explored and preliminary results show a lower accuracy than the best CNN model. Finally, a proposal for the expansion of the project is discussed to create a workflow method composed of independently trained networks to work together and detect a seismo-acoustic event, identify its type and provide some predictions on major characteristics to aid in seismologists analysis.

Contents

List of Figures

1 Introduction

1.1 Seismo-acoustic landscape in the Netherlands

With the discovery of the Groningen Gas field in the Northeastern region of the Netherlands generating a strong interest in understanding subsurface structures, a dense network of seismograms was installed and came online in 1993 [8], [26]. This network is used to monitor seismo-acoustic events which include all events recorded seismically, irrespective of their medium of origin. They include seismic events like tectonic earthquakes, induced seismic events, explosions and mine collapse but also acoustic events like sonic booms in the atmosphere, oceanic explosions and underwater volcano eruptions. The acoustic events create a pressure wave which can seismically couple with the earth's surface to be recorded seismically by seismographs.

The Groningen area has been monitored for smaller seismic events, provoked by the nature of natural gas extraction from a porous solid medium, as well as global teleseismic events. While the latter are easily identifiable, it can be speculated that not all small seismic events, microseisms and even microseismic events are recognised by the current monitoring system.[1] These types of events are relevant as they can be used in regions of gas extraction to monitor reservoir changes, geochemical processes in the subsurface and monitoring CO_2 sequestration [24]. They also play an important safety monitoring role in regions like Groningen where the resource exploitation occurs close to infrastructure susceptible to seismic movements.

As the seismogram network is also used to record event of different nature including triggered seismic events like explosions and mine collapses as well as acoustic events, whether atmospheric or oceanic, smaller seismo-acoustic events must often be identified manually by correlating them to reported logs. The induced seismic events are treated similarly to small seismic events, however, sub- and supersonic acoustic events are used to model infrasound propagation in the atmosphere and ocean to explore weather dependence of sound propagation and the reconstruction of atmospheric and oceanic conditions from acoustic data [4], [20]. Oceanic acoustic data is also used to monitor submarine surface sources like volcanoes [5], [14] and atmospheric acoustic data is used as an additional resource to detect and monitor nuclear explosion in cooperation with the Comprehensive Nuclear-Test-Ban Treaty by the Comprehensive Nuclear-Test-Ban Treaty Organization (CTBTO) [1] of which the Netherlands is an active contributor.

1.2 Constraints of detection systems

Currently, the seismo-acoustic network in the Netherlands is primarily managed and monitored by the Koninklijk Nederlands Meteorologisch Instituut (KNMI, Dutch royal institute of meteorology). Continuous time-series from each seismic station is stored in different databases depending on their respective features, often verified by experts. Based on the open-source software package SeisComP3 [25], the data treatment algorithm

[1] Here microseisms are defined as the low-amplitude continuous motion caused by man-made and atmospheric disturbances and microseismic events are any events with a magnitude smaller than 0, as described by Vaezi et al. [24].

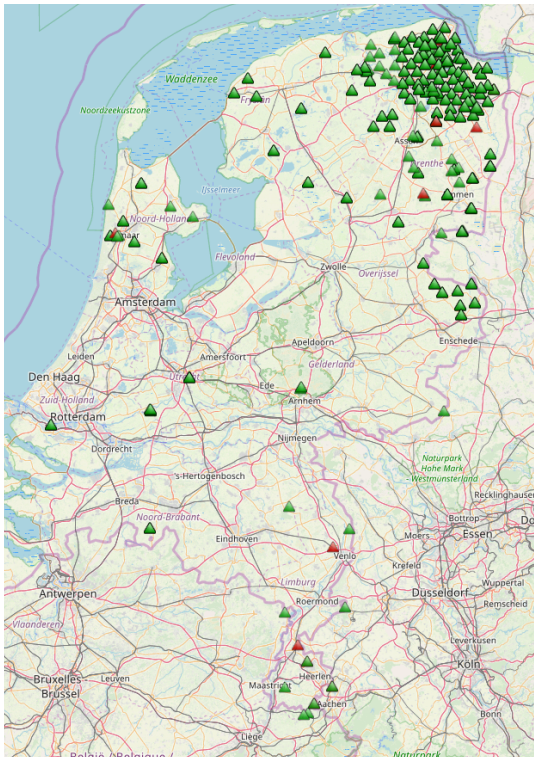


Figure 2: *Location of stations in the NL network, in green active and in red closed as of 02/2020 [8], shows the density of deployed stations.*

allows for potential seismic events from the continuous time-series to be automatically detected, stored and manually reviewed. The algorithm is triggered when a Signal-to-Noise Ratio threshold (SNR) is surpassed in at least 6 seismic stations within a given time frame and distance, allowing for seismic waves propagation between stations. Although this method is recognised to be sensitive to small events and less likely to be falsely triggered than others (e.g. amplitude or root-mean-square threshold triggers [22]), it is still challenged by the specifications of the Dutch seismic network and nature of the local seismicity.

The seismo-acoustic network in the Netherlands (shown in figure 2) is dense and exists in close proximity to populated area, especially in the Northeastern region over the Groningen Gas Field, leading to many stations suffering from both continuous (natural) and irregular (man-made) noise. High amplitude irregular noise, in particular, is more likely to trigger a false event using the SNR method than continuous noise [22] and leads to the generation of a trigger record riddled with false hits. Moreover, it is likely that increased noise 'drowns' signals of smaller events, especially for near-surface stations, most crucial in seismo-acoustic detection. Naturally, an event, whether seismic or acoustic, too low in energy to trigger the minimum of 6 stations, the event will most-likely be missed entirely.

Finally, the Netherlands subsurface's complexity is increased by the number of near-surface reflectors [3] rendering differentiation and allocation of small events' triggering more difficult. The number of valid arrivals available for analysis is thus significantly reduced.

1.3 Neural network solutions in literature

Recent rise in popularity of machine learning application to different big-data problems has led to widespread use of Convolutional and Recurrent Neural Networks (CNNs and RNNs) for event classification and data processing in seismology. Trained neural network models can theoretically be applied to a finite seismic time-series and recognise features not necessarily perceived by human analysts without heavy processing to return predictions on the nature of the series and whether a (part of) an event is contained within. The lofty goal within the seismic community of rendering other detection and processing methods obsolete is distant but interest in the integration of machine learning models in contemporary tools is burgeoning, although it currently mostly remains in the theoretical experimentation phase.

Deep learning models have been used (to varying degrees of success) to augment analyst decisions in event discrimination [12], phase detection [2] and magnitude [17] within others. Although the input is a segment of a continuous time-series, some authors have chosen to train on their model using time-frequency representations as the spectral information of seismic records has been more extensively used to improve the accuracy of small event detection, as stated by Dokht et al. [12]. Regardless of the input type, low-level processing of time-series is a running theme throughout the literature.

The practicality of recurrent neural networks to deal with time dependent data is profoundly explored by Mousavi et al. for magnitude estimation [15] and signal detection [16]. The magnitude estimation is further helped by the discretionary normalisation of the time sequence inputs for RNNs, retaining amplitude information otherwise lost. These models combining convolutional and recurrent neural network architectures are representative of the most complex models found in literature. A model geared towards real-world application of RNNs to earthquake detection system is produced by Kuyuk et al. [9], an earthquake early-warning system for far-sources based on low amplitude early P-wave arrivals. This model remains theoretical but is geared and tested for real-world implementation.

It must be noted that several common difficulties are described in the literature when attempting to apply deep learning methods to seismic recognition problems. Firstly, in order for a network to correctly recognise and classify an event, a broad and generalised dataset must be defined for training, with reasonably strict limits on classification labels. Not all regions benefit from a long-standing seismic network and, regarding detection of small events, if the existing infrastructure fails to recognise and archive smaller seismic events, the dataset can easily be biased towards events that would otherwise trigger existing event recognition systems. Moreover, although not explicitly verified in the literature, it can be hypothesised that dataset for small event recognition and feature evaluation is very region specific due to the dependence on local near-surface reflectors. Another factor in effective recognition is the depth of the training model. The deeper a model, the more features it will generally be able to classify, but, naturally, the effective training time is proportionally related. The trade-off therefore revolves around the agreed-upon acceptable threshold of accuracy, the breadth of the dataset relative to its size and the depth of a training network relative to computing time. Finally although characteristics classification is represented, no approach has aimed to differentiate between seismic and acoustic events.

2 Research objectives

Based on the state of the art in the field of neural networks applied to seismic detection and classification, the KNMI launched the DeepQuake project to explore the application opportunities. The aim is to create the framework for the development of a tool to be used in conjunction with the current SeisComP3 detection system to accurately detect a seismo-acoustic event, and classify its type and relevant features. This research specifically was initiated within 6 months of the DeepQuake launch, aimed at improving and optimising the rate and quality of seismo-acoustic event detection from the current model. The model at the start of this thesis research was largely based on a convolutional neural network (CNN) by Lomax et al. and showed low accuracy and precision and it and the training dataset needed to be adapted to the conditions of seismicity in the Netherlands. The final model should classify the series containing as noise, a tectonic earthquake or any other type of seismo-acoustic event such as induced seismic events, nuclear explosions and acoustic events. These three categories will be referred to throughout the report as "noise", "earthquake" and "others" respectivelly. The detection is aimed at single-station streams (time-series segments) which have been minimally processed. The accuracy and reliability of these results will be tested through different data types and neural network architecture types and the triggering system will be evaluated through comparison with SeisComP3 over randomly selected days. The event detection and classification will be compared to the publicly available database of events reviewed by experts. Finally, a framework of expansion for future research, working towards a practical integration into the current detection chain, must be considered and evaluated.

The contrasting aspects to the state of the art found in literature concern the detection seismic landscape in the Netherlands, the breadth of the performance objectives and the operationally oriented approach. As formerly mentioned, seismicity in the Netherlands contains a large proportion small seismo-acoustic events linked to subsurface exploitation in a noisy and dense seismic network. These often produce cluttered seismic time-series, especially at near-surface stations, more difficult to detect and classify than those from the clear event signals used in the literature. Detection models in literature did not attempt to differentiate between earthquake and other seismo-acoustic events, a necessary first step into being able to classify events based on their nature (earthquake, induced, acoustic, nuclear test etc.). All developments are approached with the future goals in mind, thus, the final product should be able to accurately detect any seismo-acoustic event, differentiate between seismic and acoustic events, characterise them by nature and identify features of the event to support seismologists' analysis. Secondly, the models created must meet the imperative quality, accuracy and reliability and criteria linked to the implementation alongside the current SeisComP3 detection system using single-station time-series (unlike the 3-6 station requirements for the earthquake warning in Japan by Kuyuk et al. [9]). This includes high detection accuracy and precision but also low rates of false positives over daily applications.

3 Dataset

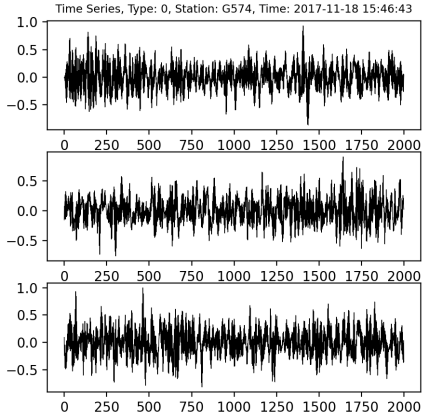
Events and noise for training are extracted using FDSN services from the open library run by the KNMI. The network chosen covers the Netherlands, especially the Groningen Gas Field region (figure 2). The NL network metadata is provided in the Table 1, most stations have a sampling frequency of 200Hz but not all. No station types or depth are excluded from the dataset, thus the 'noisy' surface stations are included in alongside borehole stations. Based on the distance of the event from each individual station and the strength of the event (if the magnitude of a seismic event is below 2.5, the radius which restrict the stations from which streams are called is 25km and it is shifted to 50km for events with a magnitude above 2.5). The streams are called using the fdsn KNMI client and tested such that all events with a SNR above 4 (tested using the STA/LTA method) are selected to be used in the dataset. The stream is called such that is event is randomly placed within the 20sec window and not necessarily centered to create similar effect that would be achieved if daily single-station seismic data was randomly segmented. The SNR threshold was selected based on manual reviews of different dataset catalogs with lower and higher SNR thresholds; the aim being to create dataset with events which can be visually detected, even in a noisy network while including as much individual streams as possible. Events include earthquakes, induced seismic events like explosions and mine collapses, and acoustic events. The three channels are rotated if necessary to ensure N-S/E-W homogeneity and resampled at 200Hz; filters applied include a linear detrending and a bandpass filter between 0.5-22Hz. The streams are re-normalised after filtering to insure the normalised inputs promote network stability during training.

Table 1: *NL seismic network metadata*

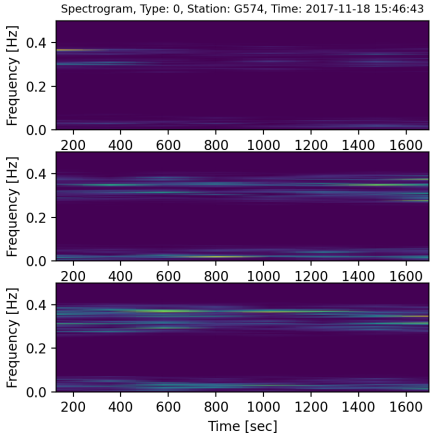
Network code	NL
Region	The Netherlands
Archive	KNMI
Start date	1993-01-01
Class	Permanent
Shared	True
Restricted	False
Type	VBB, BB

The events are stored as Tensorflow records, also called TFRecords, a type of binary information storage developed to be used alongside the Tensorflow machine learning modelling package. The records were created with a number of features: event type, which serves as label for the detection classes (earthquake event, other event and noise), distance, magnitude, depth, azimuth, start time, end time and time-series components. The noise is extracted semi-randomly between the events, assuming no event occurs within the time-window. SNR is used to verify the validity of the noise sample, however, the remaining uncertainty between noise and possible small events is a major pitfall of the dataset but cannot be avoided. The samples are treated like the events, and stored in tensorflow record files with the same features.

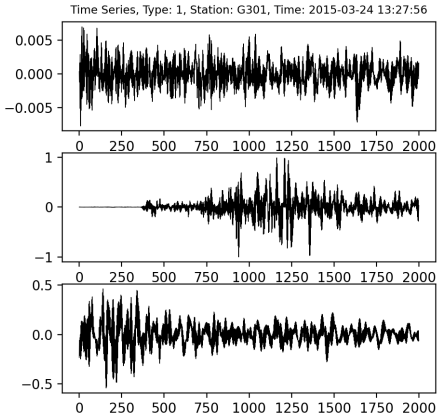
The data library for each class (earthquake events, other events and noise) is split into 3 sets: events from 2016-2017 are used for the training, events from 2014-2015 for



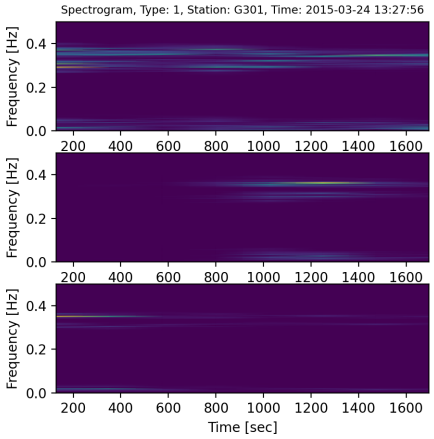
(a) *Noise time-series*



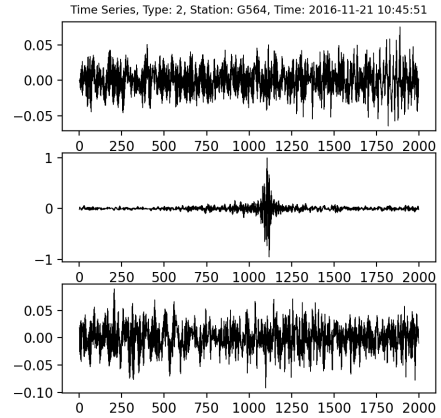
(b) *Noise spectrogram*



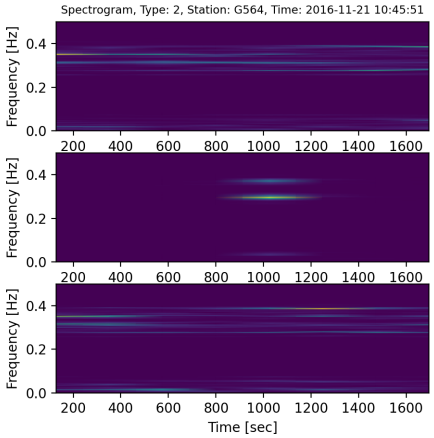
(c) *Earthquake time-series*



(d) *Earthquake spectrogram*



(e) *Other event time-series*



(f) *Other event spectrogram*

Figure 3: *Example of time-series and spectrograms used to create the dataset for the event class (3.a, 3.b) and the noise class (3.c, 3.d) and the other class (3.e, 3.f) in this case an explosion. The channel component order for each subplot from the top to bottom is vertical, horizontal N-S and horizontal E-W respectively. The spectrograms are smoothed using Goureaux shading and thus not representative of the shape of the spectrogram matrices.*

testing and events from 2018 to evaluate the trained model. The events are separated in time to ensure they do not interfere with the necessity for unbiased datasets during testing and evaluation. An event recorded at a particular station may look similar to that recorded at a nearby station, especially in such a dense network. If a network were to train on the former and perform tests on the latter it would skew the results towards a higher accuracy than may be observed with unrelated events. This effect was observed in preliminary tests whereby the network trained quickly with high accuracy but failed to achieve similar success when dealing with new data.

To increase the amount and quality of information available for training, the spectrograms of all three channels are computed independently using the Short-Term Fourier Transform (STFT) with a standard Hanning window 120 samples long. Because the STFT is performed on pre-normalised streams, the resulting spectrograms are independent of magnitude. This method is previously explored in literature with successful results [2], [12], [16]. Spectro-temporal analysis is advantageous to analyse complex amplitude to time and frequency signals, a definite feature of seismic time-series. They provide a tool of analysis which is sensitive to microseismic events ([24], [23]) and does not rely on the availability of wave phases, which is particularly useful in high-noise environments ([12]). Figure 3 shows an example of the 3 channels 20 seconds time-series input and their respective spectro-temporal representations for an earthquake event, an other event and a noise sample class.

The use of synthetic data, although strongly considered to deepen the dataset using Generative Adversarial Networks (GAN), was finally avoided. The dataset was also not augmented with Gaussian noise. These measures are taken to create a highly realistic model, looking towards integration into detection tools. The final size of the dataset is $\approx$30,000 event and noise streams respectively which is appropriate for this type of classification training.

4 SeisComP3 methods

In the Netherlands, the KNMI hosts the largest share of seismic stations. The seismic monitoring chain of the KNMI is detailed in their online documentation [19], but, as mentioned in section 1.2, current seismo-acoustic detection system is rooted in the SeisComP3 seismological software [25]. The primary detection algorithm used by SeisComP3 is the Short-Term Average / Long-Term Average (STA/LTA) to detect P-phase arrivals for local and regional earthquakes. The events are all manually reviewed by officials at the KNMI before being published openly in their directory, available online.

Different methods of automatic 'triggered' event recording from continuous time-series have been commonly used since the early 1990s, the most prevalent being the Short-Term Average / Long-Term Average (STA/LTA), a type of SNR method used as the primary trigger for event detection by the SeisComP3 software. STA/LTA, famously sensitive to smaller seismic events, has the advantage of being dependent on the single station continuous streams, 'triggering' an event recording when a pre-set amplitude threshold is passed. With adjustable parameters in different noise environments and continuous tracking of base noise levels used to adjust detection sensitivity, this method decreases the number of false triggers caused by natural noise (for example continuous marine source) and continuous man-made noise while allowing for some event type discrimination [22].

STA/LTA algorithms continuously monitor seismic data by segmentation and (almost) instantaneous computation of short window amplitude average (STA) to long window amplitude average (LTA) ratio. The short window will pick-up spikes in amplitude representing events while the long window will represent the basal noise level. A user selected threshold acts to trigger recording when exceeded by the ratio. Voting schemes parameters can be defined by which/how many channels must be triggered to manage how multi-channel data loggers record events. 'Detriggering' then occurs using a second pre-set threshold, generally lower than the triggering level. Settings can also be adjusted to include pre-event time (PEM) and post-event time (PET). The process is exemplified in figure 4.

Although effective and a standard for seismic detection worldwide, the STA/LTA methods may fall short when applied to noisy environments where it may easily be triggered by irregular high amplitude man-made noise (burst or spike type). The results and discussion will highlight this issue with many triggered hits unrelated to any events in figure 11. The goal of the neural network detection model is to detect events with a precision rivalling SeisComP3 while reducing the rates of false hits from irregular noise from single-station data.

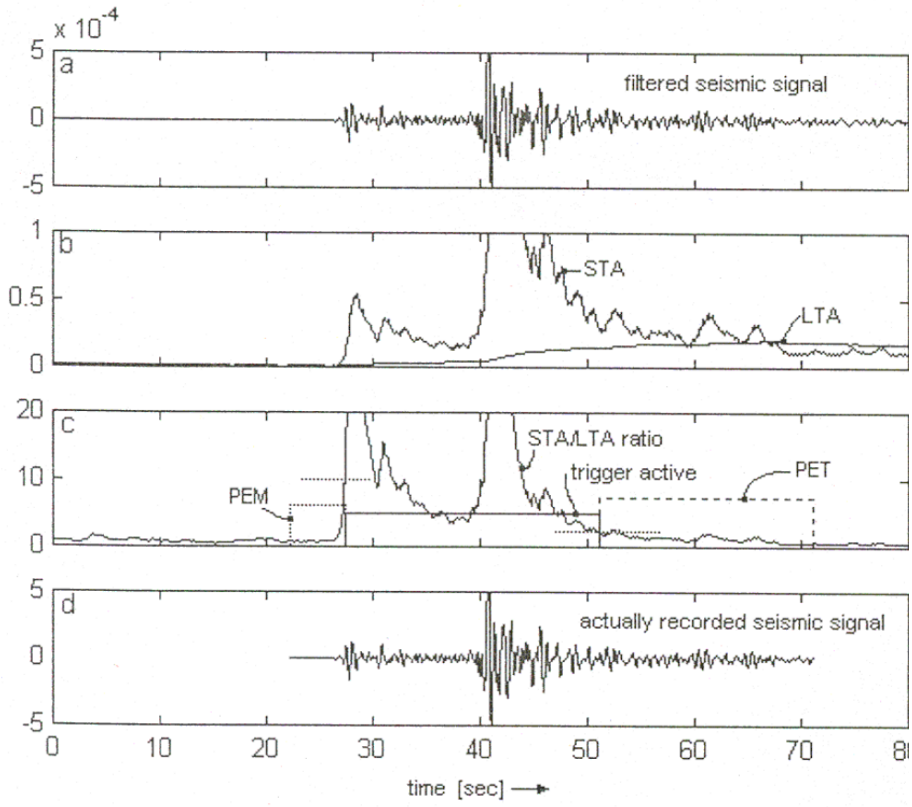


Figure 4: *Example of the STA/LTA trigger algorithm from : a) continuous seismic signal recorded at a station; b) the absolute values of the STA and LTA; c) the STA/LTA ratio with the added pre-event time and post-event time windows, the straight line denoting the trigger threshold; d) the final recorded event time-series. Figure obtained from [22].*

5 Research Methods

Supervised learning is performed using two deep neural networks, containing convolutional and fully connected layers. Both networks are trained using the same dataset, split identically, using either the time or frequency inputs independently or simultaneously. Adding the spectrograms as a separate input to the model is not providing new information (Parseval's theorem) but rather is used to investigate whether the visualisation of the same data in a new format improves the decision of the model just as it would help seismologists in their analysis.

All architectures are created and trained using the Tensorflow and Keras combined APIs. These allow for a faster implementation and provide sufficient control and insights over the training process through the Tensorboard module permits a deeper understanding on the training mechanics and performance of individual layers. All architectures make use of batch training and sparse categorical crossentropy loss function (for mutually exclusive classification classes) to save computing time and memory during training. Two deep convolutional neural network (CNN) architectures are given in figure 6. The convolutional method is selected on the basis that the filters created by the convolutional kernel operations will update to recognise features specific to an earthquake event, other seismo-acoustic events or noise, regardless of its position within the input (time-series and spectrogram). These features could possibly be imperceptible to an analyst due to noise or low amplitude of the sample.

5.1 Time-series inputs architecture

The first architecture (shown in figure 6.a) trains to detect an event using only time-series input. The input is thus composed of three 1D vectors representing the three seismic channels. They are processed through 1D convolutional layers applying L2 regression (lasso regression) convolutional kernel regularization to reduce overfitting. No other regularization method is employed due to in-necessity. The ReLU (Rectified Linear Unit) activation function is preferred to avoid vanishing gradients and further improve computing time. After working through six 1D convolutional layers each composed of 64 filters, the three channel outputs are flattened to work through two fully connected layers which perform the classification task. Based on probability, the input is classified as an earthquake event, other event or noise.
This model architecture will be referred to as CNN time model.

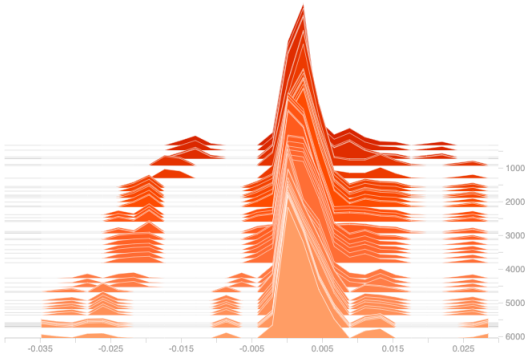
5.2 Time-series and spectrogram inputs architecture

The second architecture (shown in figure 6) trains to detect an event using both time-series and spectrogram inputs. The spectogram is added to improve the classification result on a validation dataset using the CNN method. Each input type is treated separately with respective dimensionality (2D for the spectrogram images), then flattened, concatenated and passed through fully-connected layers. Early in the development of this architecture, the training ran smoothly but the accuracy was below expectation on a validation dataset. After inspection of the weight and kernel values during training, it was noted that "dying"

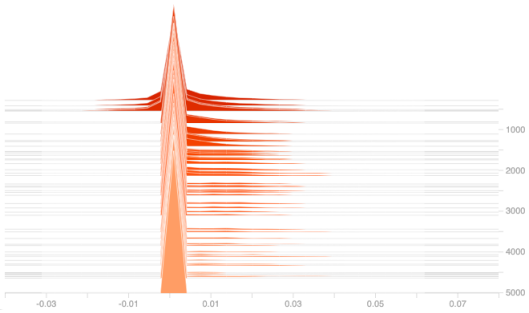
ReLU prevented many 2D convolution layers from truly learning the features of noise or an event. As seen in figure 5, the kernels and activation function outputs quickly go to zero and do not evolve with training. These are strong signs of the "dying" ReLU problem, inherently caused by the shape of the function which goes if pushed to return zero due to large update gradients, will not activate again for any input, leading the node to "die". This issue was tackled by converting the model to adopt the Scaled Exponential Linear Unit (SELU) activation function, including the necessary Lecun kernel and bias values initialization and α-dropout [7], used to reduce overfitting. The SELU activation function is given by:

$$SELU(x) = \lambda \begin{cases} x & \text{if} \quad x > 0 \\ \alpha e^x - \alpha & \text{if} \quad x \leq 0 \end{cases}$$

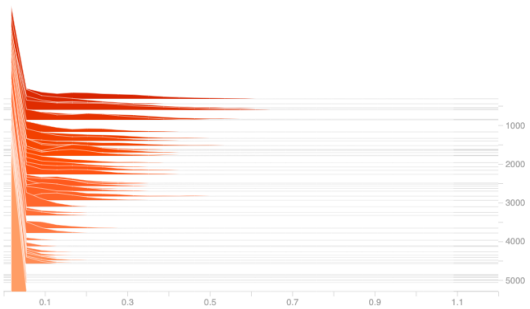
This function cannot stall at zero and has the benefit of being self-normalising, helping to prevent exploding gradients which may occur despite pre-processing normalisation of the inputs.
This model architecture will be referred to as spectro-temporal CNN model



(a) *Biases values*

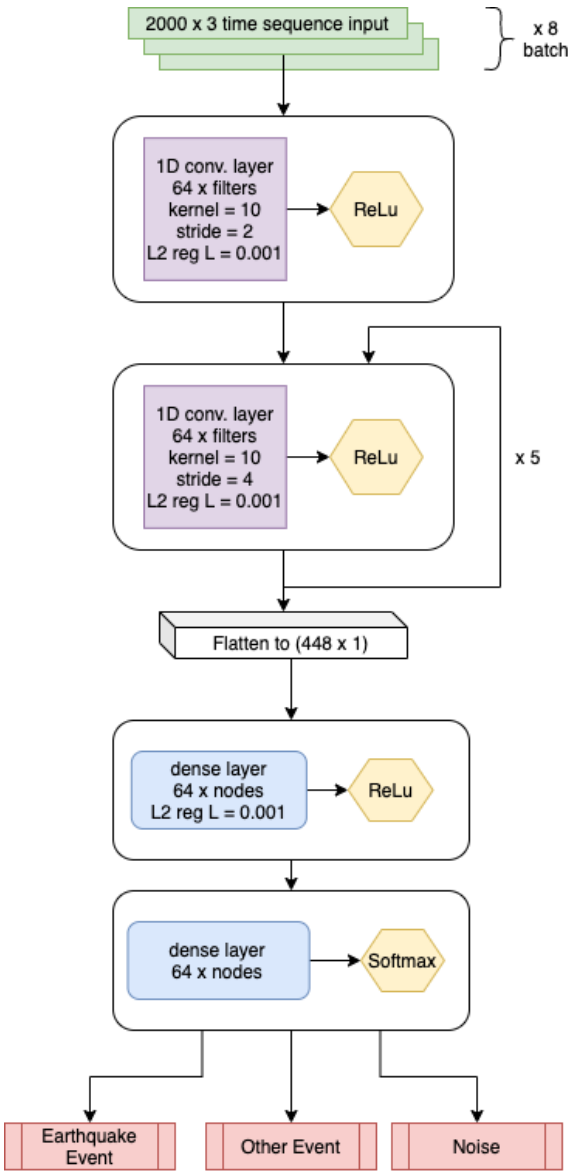


(b) *Kernels values*

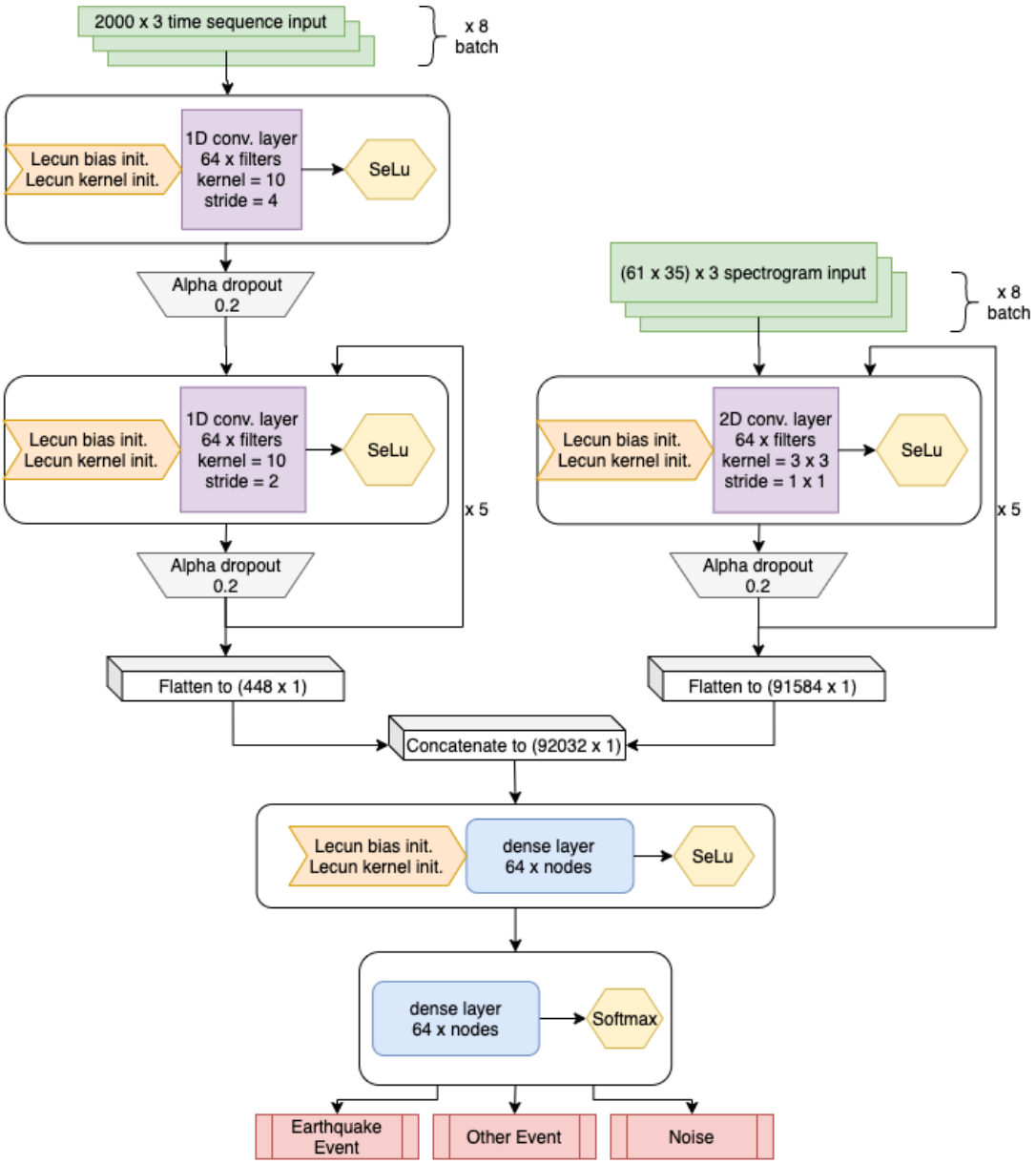


(c) *Activation function output values*

Figure 5: *The kernel, bias and output values and evolution over the duration of the training for a single 1D convolutional layer exhibiting signs of "dying" activation function ReLU. The x and y-axes give a distribution of kernel, bias and output values and the z-axis the batch-wise evolution during training (although not all batches are represented). The biases values (6.a) show some variation as the training evolves although values remain rather static, indicative of a limited learning behaviour. The convolutional kernels (6.b) and the activation function output values (6.c), however, quickly center around zero and stop evolving as training continues, a strong indication that the ReLU activation function has experienced large gradients, returned zeros and "killed" the nodes which did not reactivate.*



(a) *CNN time model:*
A single time-series input is composed of 3 × 2000 values long vectors (20 seconds of signal samples at 100Hz at each seismograph orientation). The input is first placed through a 1D convolutional layer with 64 filters with initial kernel stride of 4 and the through five other 1D-convolutional layers with kernel stride of 2 to attempt to pick up on finer features. ReLU (Rectified Linear Unit) activation function is used after all convolutional layer, in an attempt to avoid vanishing gradients. The output is then flattened to be processed through two fully connected layer, the first with a ReLU activation and the latter with a softmax activation function to output classification probabilities. L2 regularisation is used throughout to intrinsically handle possible overfitting.



(b) *Spectro-temporal CNN model:*
The time-series inputs are identical to those used (a). They initially independently go through six 1D convolutional operation cells similar to those seen in (a) apart for the kernel and biases Lecun initialization which is used throughout this model and the SELU (Scaled Exponential Linear Unit) activation function. An alpha dropout is used for regularisation rather than the L2 regularization to avoid over-regularization (measures discussed in V-B.2). The spectrogram input correspondent to the time-series of each seismograph channel as 2D image input. They thus go through five 2D convolutional cells each with a Lecun kernel and bias initialisation and SELU activation. Dropout is applied to both inputs and they are respectively flattened before concatenation. Now working as a single input, two fully connected layers are applied producing the classification probabilities.

Figure 6: *CNN architectures for time-series only input (a) and time-series and spectrogram inputs (b). The inputs shape and batch sizes are given in green, arrows represent the flow of the input through the network. White blocks represent cells of operation with a 1D or 2D convolutional layer (purple) or a fully connected layer (blue), an activation function (yellow) and an initialization state of bias and weights if applicable (orange). Arrows linking back to an operation cell shows how many times it is repeated. The white rectangular blocks represent a computation to the output of an operation cell the input and the output classes are given in red.*

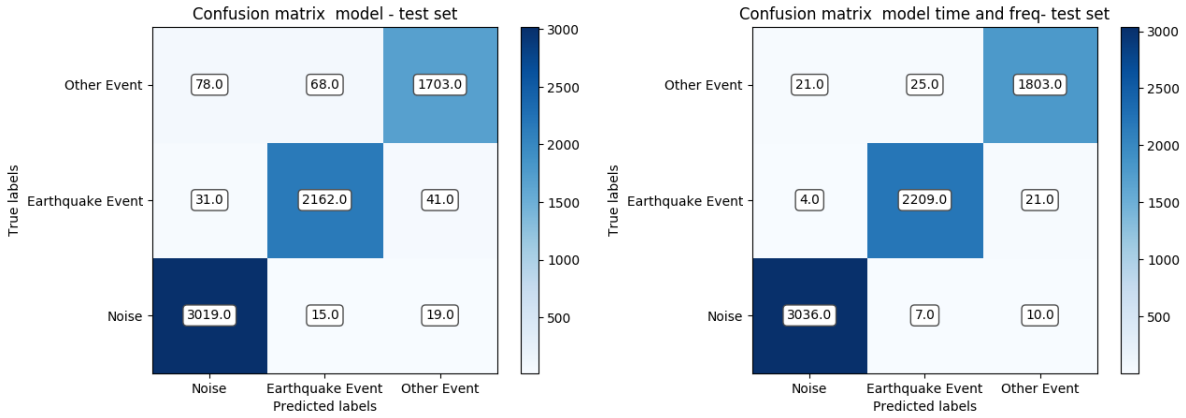
6 Results

The models are trained individually with the same dataset. The parameters are adjusted until the highest accuracy and testing values are achieved to results in the values described in the section 5. Due to the stance of integration into current detection system, special care is awarded to producing a model which is consistent, accurate and efficient.

The model training with both time-series and spectrogram (CNN time model) performed marginally better than the time-series only model (spectro-temporal CNN model) and offered a high level of accuracy and reliability. The CNN time model trained within 36 epochs while the spectro-temporal CNN model within 20 epochs. This is extremely fast training for a model dealing with complex natural data. As seen in figure 7, the CNN time model reached 96.5% accuracy at the end of training and the spectro-temporal CNN model 98.8% accuracy. The validation set accuracy is not as smooth as the training accuracy which may point to discrepancies between the training and validating datasets, possibly due to stations going online and offline between 2016-2017 and 2018-2019 as well as possible changes in the seismic landscape leading to the shut down of the Groningen Gas Fields (announced by the Dutch government in March 2018).

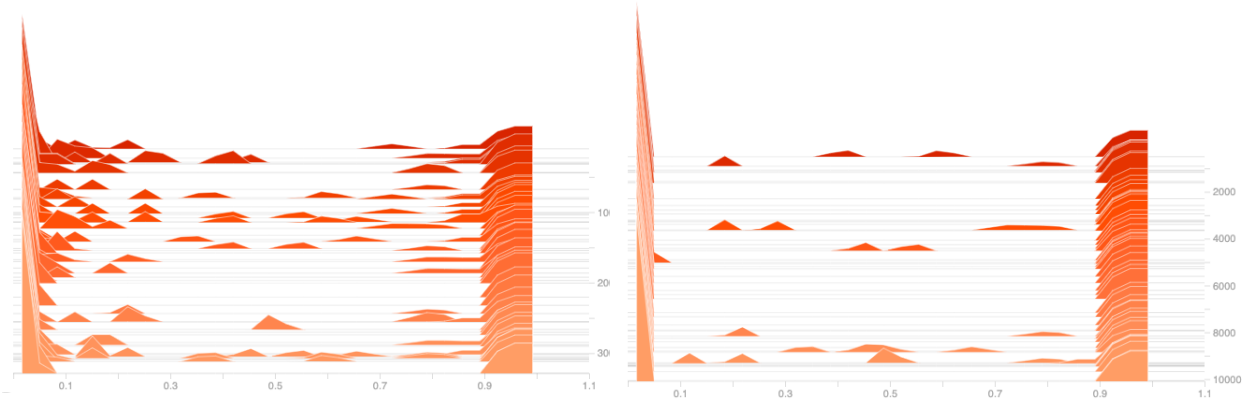
Figure 7 shows the softmax activation function outputs for both models. The x-axis the probability that an input belongs to an output class. From the distributions, both models have a stable certainty of classification throughout the training but the CNN model with spectro-temporal inputs has an overall higher certainty as very few inputs fall between the 90% and 5%. Because there are three output classes and the sum of the softmax output values must add to one to ensure the mutual exclusivity of the classes, the distribution histogram imply that for most inputs the output classification probability is given by approximately 90%, 5% and 5% for each class. This evidence further supports the advantage of including spectro-temporal inputs for better performance.

The signs of dying ReLU which had hindered training for the spectro-temporal input model is removed with the measures taken (SELU, bias and kernel initialization and alpha dropout). This is reflected in the improved classification of the validation dataset in the confusion matrix seen in figure 7 and the batch-wise evolution of biases, kernels and activation function outputs given in figure 9. The activation function outputs of the last layer (figure 9.f) does show low learning behaviour but this is unrelated to the activation function going to zero, as this is impossible with the SELU function.



(a) *CNN time model* (b) *spectro-temporal CNN model*

Figure 7: *Confusion matrices of both CNN models on the validation dataset. On the x-axis the true label of an input from the validation dataset and on the y-axis the predicted label from the trained model. The first with time-series only (a) corresponds to an accuracy of 96.5% and the latter with time-series and spectrogram inputs (b) to an accuracy of 98.8%.*



(a) *CNN time model* (b) *spectro-temporal CNN model*

Figure 8: *Softmax activation function outputs for both CNN models, the time-series and spectrogram inputs model (b) showing a higher level of classification certainty throughout training. The x and y-axes give a distribution of the softmax activation function output and the z-axis the batch-wise evolution during training.*

(a) *First layer biases values*

(b) *Last layer biases values*

(c) *first layer kernels values*

(d) *Last layer kernels values*

(e) *First layer activation function values*

(f) *Last layer activation function values*

Figure 9: *Bias, kernel and activation function output values for the first (a,c,e) and last (b,d,f) 1D convolutional layers of the spectro-temporal CNN model. The x and y-axes give a distribution of kernel, bias and output values and the z-axis the batch-wise evolution during training. The first layer shows healthy training trends with an evolution of all values, settling to optimal values before the end of the training. Last layer showing signs of under-training with unsettled bias values and slowed evolution of kernel and activation function output values.*

7 Discussion

7.1 Performance analysis

From the confusion matrix analysis (figure 7) and accuracy performance, the model training using the time-series and spectrograms is concluded to perform better. This result concurs with evidence seen in literature pointing towards the advantage of the inclusion of spectro-temporal data ([16], [15], [8]) and multi-input models ([2]). However, an important note remains: whether the spectro-temporal CNN model outperforms the other model because of the inclusion of the new data type or whether it is only caused by the increased model complexity. As previously mentioned, the addition of the spectrogram as a separate input does not provide new information to the model but rather the same information visualised differently. It can be argued that with no new information, the two models should perform identically, however the increased complexity of the spectro-temporal model leads to the improved performance.

The accuracy observed in the spectro-temporal CNN model is comparable to that of state of the art in literature for detection systems, however, it can be argued that 98.8% does not yet reach the standards of accuracy necessary for possible integration into a detection system through which would handle at least 4320 samples per day, excluding the necessity for overlap in continuous seismograph segmentation.

Looking at the earthquake detection only in the confusion matrix for this model (7.b), excluding the Other Event class, the accuracy for earthquake detection to noise increases to 99.3%. This is comparable to the most accurate models including the near 99% detection accuracy from Dokht et al. 2019 [2].

7.2 Heatmaps analysis

Although the accuracy is relatively high, fast training of the model may indicate that it only learns to recognise low-level features such as obvious amplitude peaks or chaotic noise. Andrea Pagani, researcher at the KNMI, sought to investigate the which parts of the input the model's nodes where most activated for, creating activation heatmaps. Here I propose a further analysis of this data using the average of the 64 filters for convolutional layer to elucidate on the recognition behaviour of each globally. It is important to note that the validity of the analysis can be disputed by the original input shape being used to analyse each layer. As the input travels through the network it is deprecated by the convolutional layers and, thus the recognition does not correspond to the input seen in figure 15. It is possible to transform the input to adopt the shape it would have at the each convolutional stage, however, due to the nature of the data, it loses all meaning to us. Nonetheless, the analysis is applied in a broad manner to attempt to understand the trained network's performance.

The individual filter heatmaps are averaged for each convolutional layer for a 1D time-series input of an explosion type event results in Appendix A, accompanied by examples of the individual filter maps. In these figure, areas of high activation are areas that the model relied on to make the correct prediction. In common CNN methods, shallow convolutional layers manage low-level feature recognition and increased layer depth promote

higher-level features recognition, this behaviour can be observed by investigating the activation heatmaps of the first, third and last convolutional layer.

The first convolutional layers (15) show low-level feature recognition behaviour with activations of small segments in the individual filter heatmaps (15.a, 15.b and 15.c). The 64 filters average close to 0.5 homogeneously in the part of the input which contains the event. Some parts do show spikes in activation from these layers but, notably, these do not seem to be related to obvious amplitude increases. We can conclude from these shallow layers that the model has trained well to recognise low-level features within the input which are unrelated to the jump in amplitude from the background noise, but rather are related to patterns within the event itself.

The third convolutional layer (16) recognise mid-level features seen with activations of longer segments of the input in the individual filter heatmaps ((16).a, (16).b and (16).c). Again, the model does not seem to focus on obvious features but rather event specific pattern within the time-series. Looking at the averaged activation (16.d) there is an overall higher rates of activations of specific areas compared to the first convolutional layer. The activations seem to be concentrated on the event, however there is some indication of under-training does appear in the averaged filter heatmap. The rainbow-like pattern before the P-wave arrival seems to show some indecision from the convolutional filters on handling this specific region.

The last convolutional layer activation heatmaps (17) are very different from those seen of earlier layers. Looking at the individual filter activations ((17).a, (17).b and (17).c) confirms the suspicion of under-training. These layers should handle more complex high-level feature recognition, however, increasing parts of the layer show the rainbow-like pattern which do not focus on particular particular features. It does not seem to focus on specific parts of the event, especially looking at the averaged activation heatmap for this layer it appear that it completely lacks purposeful activation. This layer seems to serve no purpose in the prediction as it may be completely untrained.

These results would imply, with reservation, that the model seem to correctly recognise the expected primary features of seismo-acoustic events, however, these features remain mostly low- to mid-level.

This conclusion can also be drawn from the batch evolution of the activation functions. As seen in figure 9(a, c, and e), the first convolutional layers which manages the low-level features recognition performs well and displays learning behaviours immediately within the training and continues throughout and settle on optimal values before the end of the training. The last layer, however, exhibits slower learning behaviours with unsettled bias values and low evolution from kernel and activation outputs. The first convolutional layers will typically learn to recognise pixels and simple shape, the complexity of learning is increased with the depth of the neural network. Improving learning behaviours from deeper layers would improve the high-level feature recognition and most probably classification accuracy.

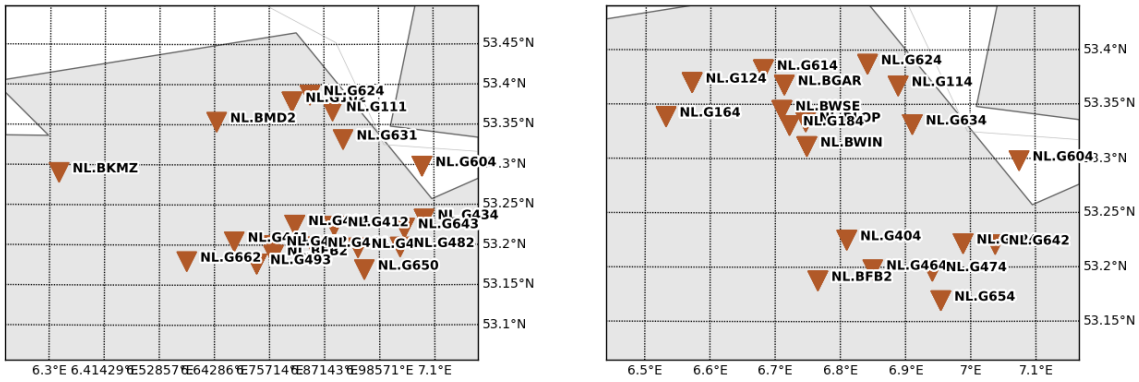
7.3 Comparison to SeisComP3

The evaluation of accuracy and efficiency against the current detection system is only carried out with the spectro-temporal CNN model which showed higher performance and accuracy. In section 4, the discussion on the methods of SeisComP3 event triggering mechanism, it was mentioned that many hits unrelated to events were triggered by the system. Figure 11 shows a comparison of the triggered hits from the best CNN model and SeisComP3 for randomly selected days in 2019 (excluded in training dataset) including: a day containing no event (11.a, 11.b), a day containing a low magnitude event (11.c, 11.d), a day containing a high magnitude event (11.e, 11.f) and a day of assumed lower background noise level during the Covid-19 lockdown period [11] containing events of varying magnitude (11.g, 11.h).

Overall, the SeisComP3 detection system triggers many more hits per day than the spectro-temporal CNN model. Looking at the figures with no seismo-acoustic event (11.a, 11.b), it must be noted that the location of the triggered stations is not accounted for. The number of hits from the SeisComP3 system is thus not indicative of a triggered event recording as the system is dependent on multi-station and distance conditions as described in section 4. In both the CNN model and SeisComP3, there is a dependence of number of hits on a daily cycle of human activity. This is a further indication that the NL seismic network is strongly polluted by man-made noise.

The spectro-temporal CNN model out-performs SeisComP3 reducing the number of false positives but seems to fall short detecting smaller seismic events. The Kanten event of magnitude 0.8 is seen in (11.b, 11.d), although the peak in hits is small in scale, it remains the highest of the day with 20 hits whereas the CNN model only registers 3 hits, dwarved by two other peaks within the day. This trend holds for another similar sized event, Appingedam of magnitude 0.8 seen in figure (11.g, 11.h) which is not picked up by the CNN model while it shows a significant peak in the SeisComP3 model. In the same day were two more events of slightly larger magnitude (GaretHuizen with magnitude 1.4 and Froomboach with magnitude 1.8) which were clearly registered by the CNN model. It could be hypothesised that the SNR and distance from source restrictions placed on the data set hindered the acceptance of smaller event time-series for training; which rendered the model unable to distinguish them from noise, even on a day with lowered base noise levels.

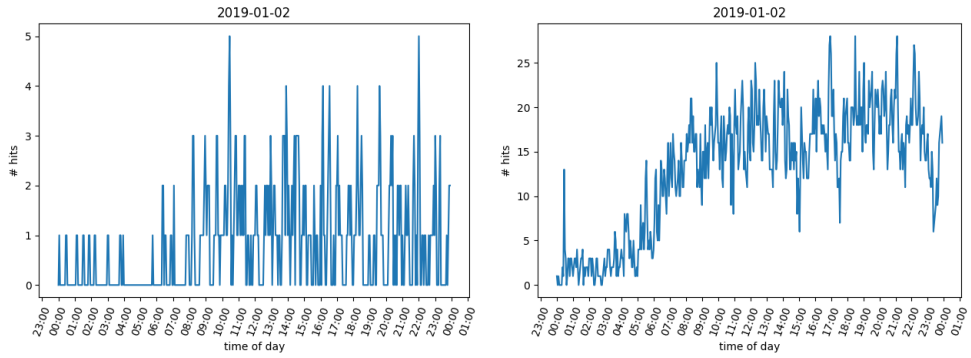
The detection of larger events like Westerwijtwerd with a magnitude of 3.4 ((11.e, 11.f)) are clearly picked-up by both detection systems. Interestingly, the event triggered many more hits for the SeisComP3 system, even after adjusting for base false-positive rate. Looking to investigate this further, the triggered stations by each system were plotted for this event in figure 10. Interestingly, although some stations are triggered by both detection systems, several were unique for each. The stations triggered by the CNN model are clustered around the source while those triggered by SeisComP3 are more spread. Moreover, the event was better detected by the stations Southwards of its origin by the CNN model and by stations Northwards of the origin by the SeisComp3 model. Overall, more individual stations were triggered by the SeisComP3 model than the CNN model, which correlates to the cumulative number of hits at this time.
This could be indicative of two assertions: firstly, it could imply that the CNN model has



(a) *CNN model* (b) *SeisComP3*

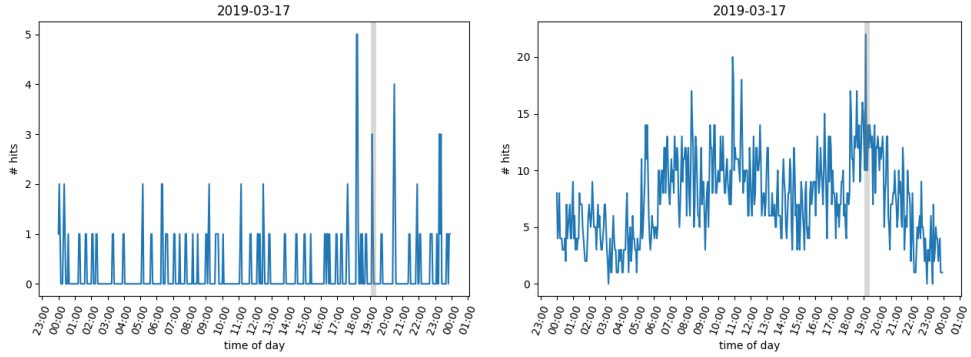
Figure 10: *Triggered stations from SeisComP3 and the spectro-temporal CNN model during the Westerwijtwerd seismic event (magnitude 3.4)*

a high detection rate indiscriminate of the condition of the station, and thus an event is recognised by most stations around it, represented by the high density of CNN triggered stations; secondly, SeisComP3 still outperforms the model at further stations which may again be linked to the loss of energy of the event seismic signal over larger distances not being recognised by the CNN model based on the restrictions on the training dataset.



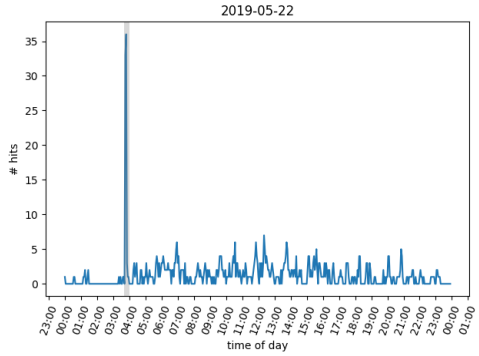
(a) *CNN, no event*

(b) *SeisComP3, no event*

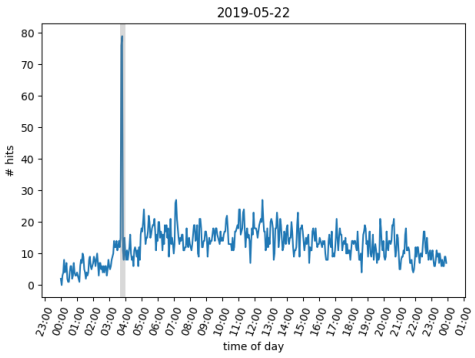


(c) *CNN, Kantens mag. 0.8*

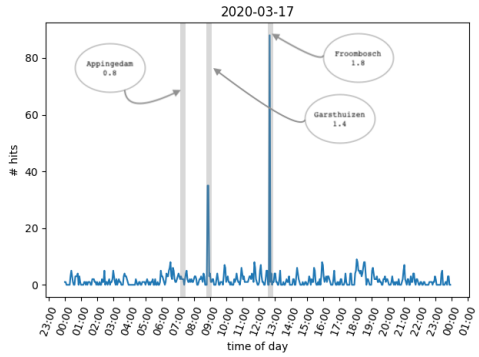
(d) *SeisComP3, Kantens mag. 0.8*



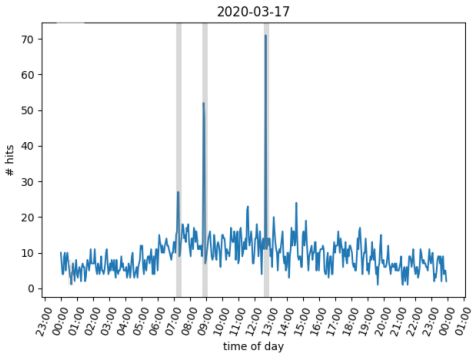
(e) *CNN, Westerwijtwerd (mag. 3.4)*



(f) *SeisComP3 picks, Westerwijtwerd (mag. 3.4)*



(g) *CNN model picks, low noise*



(h) *SeisComP3 picks, low noise*

Figure 11: *SeisComP3 and CNN model event hits for different days in 2019 and 2020. These figures are borrowed from the DeepQuake project paper that is currently in preparation for publication [21].*

8 Propositions for immediate improvements

Although the spectro-temporal CNN model compares well to the SeisComP3 detection method, its accuracy remains too low for implementation of single-station event detection. Analysing the results discussed in section 7, easily implementable methods come to mind which could improve the accuracy and precision of the model.

Firstly, high-level feature recognition could be greatly improved by increased training of the last layers which currently serve no purpose in classifying high-level features. Since the shallow layers seem to have trained appropriately within the 13 epochs, a per-layer patience function should be established such that the training of each layer can be "frozen" when the gradient descent has reached a minimum. This method would allow for more efficient training for a model with significant computing time.

Secondly, it would be especially important to implement a thorough hyperparameter search. Although the performance of the models was greatly increased by manual testing of the hyperparameters, their effect and co-dependence is extremely complex and variable. It would be greatly beneficial to automate a search through a different combinations hyperparameter including filters per layer, model depth, regularization methods, batch size and learning rate. All these factors have already shown to greatly improve accuracy, reduce training loss and computation time. Other factors which have not been explored in depth include layer types, kernel size and stride variation per layer, optimizer and loss function. In the layer type category, deconvolution layers could be beneficial to upsample the convolution output and improve deep layer training, especially if implemented with shallow layer freezing.

The dataset should also be inspected once again, with lowered restrictions on the SNR and distance from source of accepted event time-series, the model may be able learn to recognise smaller events with magnitudes below 1. Finally, due to the time-dependent nature of the data, further exploration of the recurrent network methods with LTSM and recursive loops should be considered. Preliminary exploration of the method was performed and is described in the following section.

9 Residual neural networks exploration

The recurrent and residual network solution was shallowly explored as time-permit. The model implemented was directly reproduced from the Cnn-Rnn Earthquake Detector (CRED) by Mousavi et al. [16], a deep neural network composed of convolutional layers, uni- and bi-directional Long Term Short Memory (LTSM) and residual loops. This model was published recently with promising results with both event detection and magnitude prediction [15]. Training with microseismic events down to -1.3ML, they perform event detection from single station time-series. They also perform a background noise analysis using semi-synthetic data from which they conclude that CRED outperforms STA/LTA and template matching methods for detection. For these reasons the model architecture was replicated as a starting point for recurrent neural networks applications to the NL dataset.

9.1 Theory

Due to the time sensitive nature of seismic data, recurrent neural network architectures, which were developed to deal with sequential data logically appear more adapted to perform the classification task. Recurrent layers treats each filter state at a given time as a function of its past state, in other words, each second in the time-series input is treated as a function of the one preceding and sometimes of the one following it. This allows RNNs to view the input sequentially and understand the time-dependent relationship of seismic data.

The current state of a recurrent layer is given by:

$$a^{(t)} = SELU(W_{aa}a^{(t-1)} + W_{ax}x^{(t)} + b_a) \tag{1}$$

where $a^{(t)}$ is the current state, $a^{(t-1)}$ the previous (recurrent) state with its set of weights W_{aa}; $x^{(t)}$ is the input with its own set of weights and W_{ax} and a bias b_a. The output of the layer is then given by:

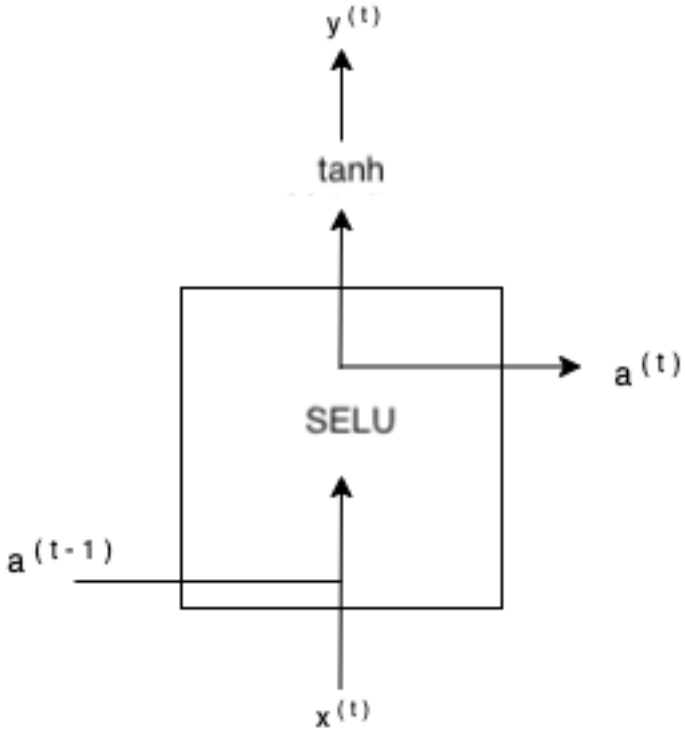
$$y^{(t)} = tanh(W_{ya}a^{(t)} + b_y) \tag{2}$$

where W_{ya} and a bias b_y are the weights specific to the current state. These equations can be visualised as done in figure 12.a. However, this structure, while although able to learn from the immediate previous input, struggles to capture long-term sequential dependencies and commonly suffer from vanishing and exploding gradients.

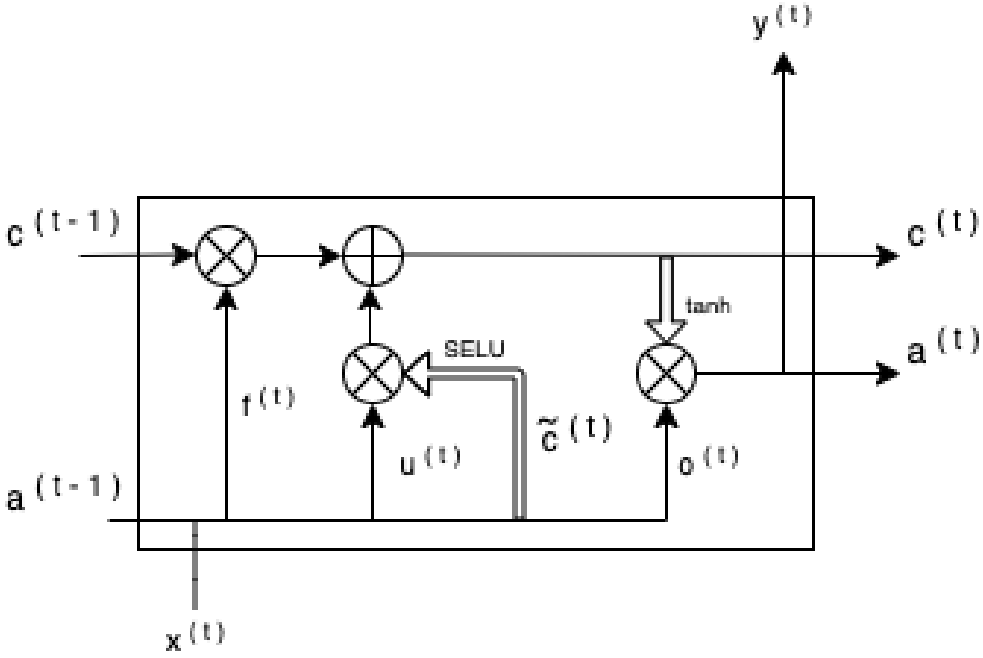
Long Term Short Memory (LTSM) layer were developed by Hochreiter and Schmidhuber (1997) [6] to tackle these issues while keeping the basic recurrent element. The LTSM layer is composed of four gates through which the previous state and input are processed: a forget gate $f^{(t)}$, an update gate $u^{(t)}$, an internal memory cell $\tilde{c}^{(t)}$ and an output gate $o^{(t)}$, respectively given by:

$$f^{(t)} = Sigmoid(W_f a^{(t-1)} + W_f x^{(t)} + b_f) \tag{3}$$

$$u^{(t)} = Sigmoid(W_u a^{(t-1)} + W_u x^{(t)} + b_u) \tag{4}$$



(a) *Basic flow of a recurrent layer cell, the SELU activation function is applied in each cell to the previous cell ($a^{(t-1)}$) and the input ($x^{(t)}$). After the SELU activation is applied the current state will be passed on to become the previous state of the next input and the tanh activation function is applied to predict the output of the current state ($y^{(t)}$).*



(b) *Basic flow of a LTSM layer cell, only the SELU and tanh activation functions are represented, the sigmoid activations of each gate is implied. The output of the layer is dependent on the past states $c^{(t-1)}$ and past outputs $a^{(t-1)}$ controlled by different gates which allow to forget unecessary information $f^{(t)}$, memorise information $\tilde{c}^{(t)}$ and update the state $u^{(t-1)}$ and outputs $o^{(t-1)}$. The special activation functions (SELU and tanh) are denoted by larger arrows, other sigmoid activations are implied for each transformer gate.*

Figure 12: *Recurrent and LTSM layer flow charts*

$$\tilde{c}^{(t)} = SELU(W_c a^{(t-1)} + W_c x^{(t)} + b_c) \tag{5}$$

$$o^{(t)} = Sigmoid(W_o a^{(t-1)} + W_o x^{(t)} + b_o) \tag{6}$$

Here, $a^{(t)}$ is the output of the layer which is fed to the next state as $a^{(t-1)}$. The forget gate handles the relative time dependence and influence of previous inputs by selecting states to be remembered or forgotten. It is activated by a simple sigmoid function which returns 0 to forget and 1 to remember.

The update gate and internal memory gate work together to determine the new cell state. The update gate is sigmoid activated and determines which input values are updated, it multiplied with the internal memory cell which is SELU activated and creates the which creates the values to for the current state.

The current state of an LTSM cell $c^{\{t\}}$ is updated from previous as follows:

$$c^{(t)} = u^{(t)} * \tilde{c}^{(t)} + f^{(t)} * c^{(t-1)} \tag{7}$$

The output of the cell involves the current state being passed through the output gate which is tanh activated such that:

$$a^{(t)} = o^{(t)} * tanh(c^{(t)}) \tag{8}$$

The flow of the LTSM cell is shown in figure 12.b, it works essentially like the recurrent cell with an additional memory of the state of the cell before the SELU activation function and with gates which allow for nonessential information to be discarded in order to free up computational memory. Concatenating two LTSM cells with opposite directionality can create a bi-directional LTSM layers, these layers are able to consider past and future inputs to process the current state.

Another method used in this model is the application of residual loops for the convolutional layers which act as a recurrent cell by linearly combining a previous state with a current state after a number of convolutional layers are applied.

9.2 Models and preliminary results

The starting architecture is given in figure 13, taken directly from the CRED publication [16]. After experiencing strong issues with exploding or vanishing gradients as training would not continue past ≈1000 batches and the loss would go NaN.

To ensure the gradients were stabilised, the ReLU activation function was once again replaced with SELU, accompanied by Lecun initialisation of weights and biases and α-dropout regularisation. The Root-Mean-Square backpropagation optimiser is implemented, as is common in RNNs, to further curb the exploding and vanishing gradients. It uses the moving average of the gradients squared which serves to normalise the gradients themselves; normalised gradients are balanced which helps curb exceedingly large or small gradients. This is also implemented with a decaying learning rate starting at

0.0001 to ensure the true global minima will be reached, with momentum of 0.05 to prevent the model from being stuck at local minima. The batch size was also decreased and the regularisation is performed on all layers. Finally, gradient clipping of values at 0.3, 0.5 and 0.8 is implemented separately as a final security cap again exploding gradients. The end of training patience was increased from the previous CNN models to avoid the under-training tendencies described previously, however, the efficacy could not be verified because the model never handled proper training.

The model was tested for event detection and magnitude detection with two bi-directional LTSM and one uni-directional LTSM layer, with one bi-directional LTSM and one uni-directional LTSM layer and with only a single uni-directional LTSM with no success in training, with the loss quickly going to NaN and accuracy hovering around 50% despite the measures discussed being implemented first individually and then concurrently.
The architecture and parameters were also tested for 2 inputs with concatenation before the dense layers which perform the classification task, but also with the convolutional layers treating only the spectrogram images input and the LTSM layers treating only the time-series inputs. Again, the progress

Unfortunately, since the results of Mousavi et al. could not be replicated or improved on to fit this particular dataset, the insights currently available to explain this phenomenon are currently limited. It can be suggested that the RNN model should be build from the ground up verifying it's training ability at each point in the process.

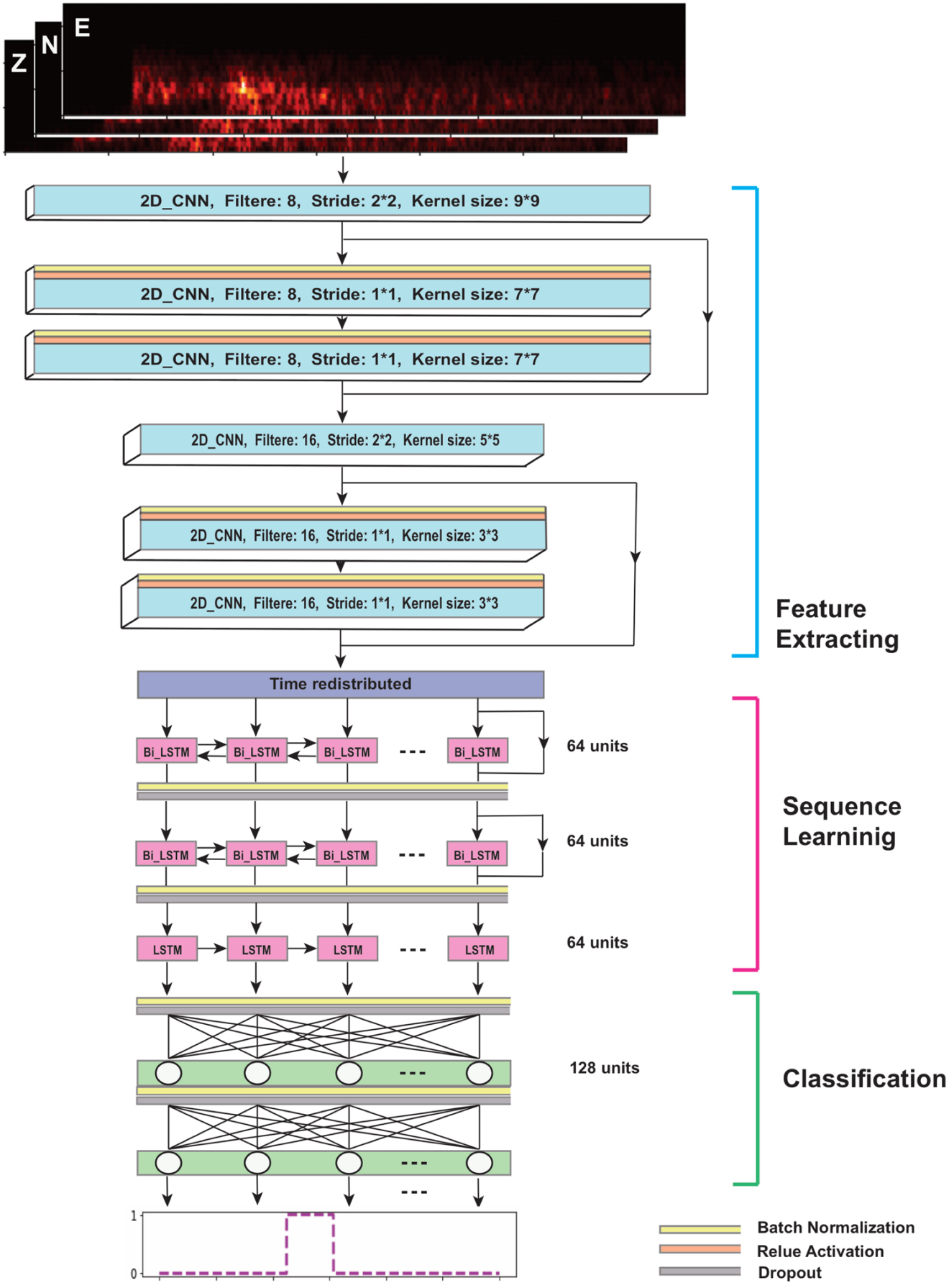


Figure 13: *The architecture used to train the models, this figure is taken directly from the CRED publication by Mousavi et al. [16].*

10 Conclusions

The spectro-temporal model with the final parameters shown in figure 6.b is the best performing model developed during this research. The accuracy of 98.8% from the validation dataset (figure 7.b) which rises to 99.3% when non-earthquake events are excluded outperforms the CNN time model and rivals state of the art models described in literature. The certainty seen in figure 8.b ascertains the reliability of the model which is later confirmed when applying it to segmented full-day seismic data (figure 11). The issues encountered with dying ReLU were appropriately resolved, however, further investigation of training behaviour of the deeper layers through kernels, bias' and activation values (figure 9) as well as heatmaps analysis (figure 15) indicate that increased training of these deep convolutional layers could further improve the model performance. The necessity of the spectro-temporal input should be investigated by testing the performance of a model of equivalent complexity using only time-series input. On the scale of the deployment of this detection model for daily seismic event detection, the computation of spectrograms for every 20 second segment, for every station data is computationally expensive. If the model could be train to perform as well only with the time-series input, it would be advantageous to implementation into detection system.

The spectro-temporal model performs well against the current SeisComP3 detection system in terms of event detection with false positive triggers reduction (figure 11). A positive results which suggests that the model truly learns to recognise event-specific features in the inputs rather than basic amplitude increase from background noise levels. However, the false positive hits are not reduced enough for the model to perform single-station event detection, which is the final goal of the DeepQuake project.

Despite good results in false positive hits reduction, the detection of smaller events (0.8 ML) is completely ineffective, as seen in figure 11.c and 11.g. As it stands, it is hypothesised that this deficiency stems from the constraints put in place during the dataset creation process. As described in section 3, a minimum SNR threshold of 4 was implemented to keep noisy time-series out of the dataset, however, this measure may have prevented time series from smaller events from being included in the dataset, thus making the network unable to learn to recognise their specific features. Rather than remake the entire dataset, it should be augmented to include time-series specific to smaller events. It must be noted that this measure may in turn increase the rate of false positive triggers from irregular man-made noise although this cannot be anticipated with certainty. Moreover, the comparison in triggered station locations from SeisComP3 and the spectro-temporal model (figure 10) suggests that it does not recognise event signals at farther stations, probably due energy loss during travel also being selected against during dataset creation through SNR threshold restrictions.

Interestingly, looking at the results for earthquake detection to noise only, the accuracy jumps to 99.3%. This effect may be caused by the heterogeneity of the "other" class which is composed of seismic and acoustic events. The results may be improved by splitting the training into two separate model, the first which would detect the presence of an event and the latter whether it is an earthquake or another type of event. This recognition of an event based on its type has not yet been explored in literature.

Finally, the recurrent neural network method is explored with no success despite parametrisation to adapt a successful model to the dataset. This technique is promising based on the time-dependent nature of seismic data and must thus be further explored. It is nonetheless surprising that a model which showed very conclusive results for one dataset was completely unusable for another.

11 Framework for future research

Attention was given to the possibility of extension of the project through different avenues. A proposal was created under the supervision of Prof. Dr. L.G. (Lslo) Evers and L. (Luca) Trani to be submitted to the Nederlandse Organisatie voor Wetenschappelijk Onderzoek (NWO) yearly grant program (ENW - KLEIN-1). This document describes the practicalities of the future research goals and methods described in this section.

11.1 Overarching goals

Building on the hypothesis that neural network techniques can expand the toolkit available to detect seismic events and improve the accuracy small event detection and reduce rate of false positives in noisy seismic networks, many further application opportunities arise. Specifically, whether a detection system could further classify an event based on the geophysical medium from which it originated without additional external information (i.e. seismo-acoustic events); and whether a detection system could advise analysts on event-specific features such as location or strength based on a single seismic trace.
Based on the research breadth at the KNMIs Research Department of Seismology and Acoustics (R&DSA) and their collaboration with the CTBTO, specific goals include:

Origin determination:

Seismically recorded events do not uniquely originate from a movement of the Earth's crust. The R&DSA department dedicates resources to furthering the quality and quantity of information to be inferred from acoustic wave coupling and arrival at a seismic station. These acoustic waves can be generated in the ocean and in the atmosphere. Upon contact with the surface, they can couple and travel to a seismograph, which can also detect the pressure wave (although the travel times and amplitude will be very different between the two). In order for the correct seismic records to be used in analysis, analysts must often retro-actively search the seismic archives based on known event logs (acoustic events often do not trigger enough stations and have lower amplitudes). Due to the high variability of the oceanic and atmospheric media, this process is made more complex by travel-time model estimations. A model able to recognise that a stream contains acoustic event and flag it immediately would be time-saving and possibly broaden the available datasets with unlogged events.

Nature determination

For each origin category (seismic or acoustic), an event can be of a different nature. A seismic event can be an earthquake, but can also be induced, whether by a nuclear explosion, a mine collapse or a quarry blast. An atmospheric acoustic event can be triggered by a sonic boom or an explosion; and an oceanic acoustic event may be caused by an explosion or an underwater volcano. The importance of nature determination of seismic event is more self-evident, subsurface seismicity monitoring depends on earthquake analysis while different types of tomography and subsurface mapping can make use of different kinds of seismic events. Flagging explosion from atmospheric and oceanic acoustic events

can broaden catalogs and play an important role in nuclear tests monitoring.

Features evaluation

Finally, each event whether seismic or acoustic, can be classified for specific features using neural networks. Notably, for seismic events, magnitude [17], azimuth and distance [9], [13] and for acoustic events yield and distance. This evaluation will not return pinpoint values but rather a classification range to guide analysts.

Additional goals

With the completion of the aforementioned objectives, an exploration of noise sample classification for noise tomography viability may be explored, as demonstrated by Paitz et al. [18]. The dependence of the models on regionality will also be considered, possibly to expand smaller training datasets.

11.2 Workflow method

Due to the nature of neural networks, in order to carry out the correct experiments to answer these fulfil these goals with the highest degree of success, it is crucial to create an adapted framework of consecutive models. Each model can perform a simple classification task and be trained with a distinct dataset and parameters. Using individual datasets for each classification can improve the accuracy and reliability, especially for natural, complex data. Moreover, splitting the categorisation into smaller subsets allows the integration of checks and balances within the decision-making.

Going more in-depth, a prototype of the workflow is given in figure 14. Based on the processing chain of seismic segments performed by SeisComP3 [19], a single single stream input is minimally processed (identically to those used for the training dataset) and fed through the detection model described in this paper. Based on the classification probably and algorithmic testing (i.e. SNR) the stream is placed within one of three categories: "event", "noise" and "uncertain". The uncertain streams are stored aside for manual review and noise streams as noise, possibly fed through an ambient noise classification model. The events streams are re-processed, by including more precise data types or adjusting the processing used for the the detection model, as well as re-segmenting to include a possibly a full seismic event and a longer portion of an acoustic event (lower frequency signals). Fed through the detection model, the acoustic and seismic events can then be handled separately to create feature maps describing the features picked up by respective models networks, their probability and their compatibility. It is expected that the nature and feature extraction models will likely perform at lower accuracy and will only be used as a consultative resource for seismologists.

The augmentation of datasets which may be too limited for as the models becomes more specific can be reconsidered, using GANs. GAN work by setting up two neural networks model which work against one another to create synthetic data which could pass for real data. One of the models (the generator) creates the synthetic data and the

other (the discriminator) evaluates the validity and both networks are adjusted based on this decision. New artificial data is created by the generator until it can pass as real data. Basing the discriminator on the neural network detection model in development can also offer further insights on its performance. This method is commonly employed in image and video processing. Due to the nature of seismic data, it becomes more difficult to judge whether the data created by the GAN is truly viable and representative of real-life data. Since the detection model can be used as the base model for the discriminator, this could lead to biases towards artificial data detection which may not improve detection rates of real-events. For this reason the GAN dataset enhancement phase of this project was terminated to give way to other testing, including data types (like spectrograms and maximum amplitudes for example) in the detection model. However, it may become an important part of developing specific models to recognise precise features.

This workflow-type framework can be adapted as the training progresses and new knowledge is acquired. This method of application of machine learning models to seismo-acoustic data detection and classification is not found elsewhere in literature.

11.3 Limitations

Limitation which will affect the implementation of this plan have been experienced throughout the completion of the research developing the detection model. The first, and possibly most intricate, is dealing with the dataset, for several reasons. Lots of thought was awarded to ensure the fidelity and correctness of the labelled training dataset, however, when dealing with larger datasets that are computationally compiled, manual verification is near impossible. With the number of independent datasets, special care must be attributed to the process. Finding enough events to populate a class for a training dataset is also limiting. During the process of this research, attempts were made to create the origin differentiation model, however, obtaining enough acoustic events required extensive work dealing with logs and travel-times which could not be fully explored due to time restrictions. Training a deep neural network model to classify complex natural data requires large datasets. In the case of the detection model, artificial (GAN generated) data was finally discarded to promote the fidelity of the dataset, but the method can be applied to increase the volume of specific event types available for training.
Limitations will also include time-management with model development and training, testing for optimal network type and parameters

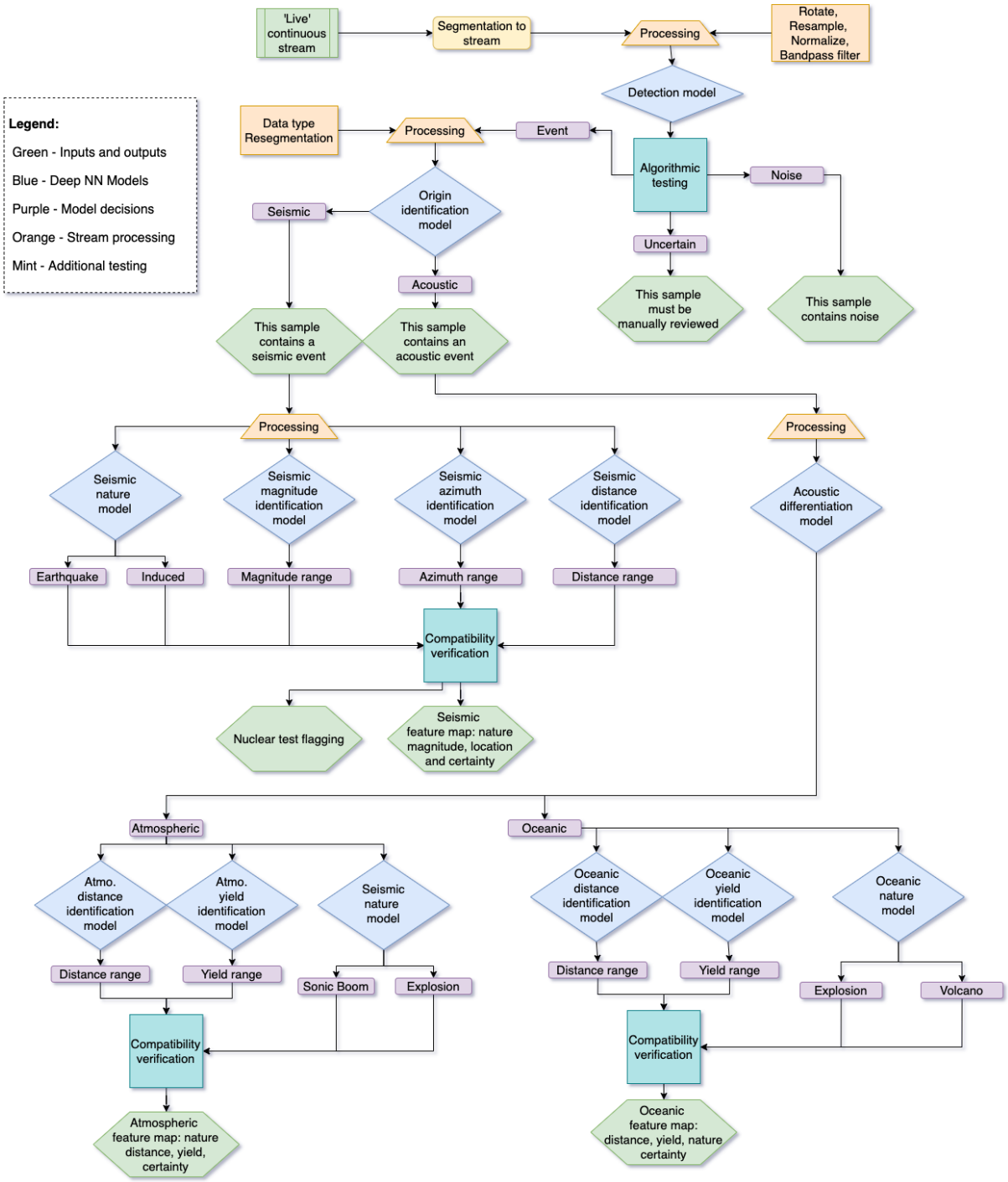


Figure 14: *Prototype of workflow method for seismically detected event detection and classification.*

12 References

[1] Assink J. D., Averbuch G., Smets P. S. M. Smets and Evers L. G., On the Infrasound Detected from the 2013 and 2016 DPRK's Underground Nuclear Tests. Geophysical Research Letters 43, no. 7 (2016): 352633. https://doi.org/10.1002/2016gl068497.

[2] Dokht R. M. H., Honn K., Visser R., and Smith B., Seismic Event and Phase Detection Using TimeFrequency Representation and Convolutional Neural Networks. Seismological Research Letters 90, no. 2A (2019): 48190. https://doi.org/10.1785/0220180308.

[3] Duin E.J.T., Doornenbal J.C., Rijkers R.H.B., Verbeek J.W., Wong T., "Subsurface structure of the Netherlands - Results of recent onshore and offshore mapping." Netherlands Journal of Geoscience no. 85 (2006). https://doi.org/10.1017/S0016774600023064.

[4] Evers L. G., A. R. J. Van Geyt, P. Smets, and J. T. Fricke. Anomalous Infrasound Propagation in a Hot Stratosphere and the Existence of Extremely Small Shadow Zones. Journal of Geophysical Research: Atmospheres 117, no. D6 (2012). https://doi.org/10.1029/2011jd017014.

[5] Green D. N., Evers L. G., Fee D.,Matoza R. S., Snellen M., Smets P., and Simons D., Hydroacoustic, Infrasonic and Seismic Monitoring of the Submarine Eruptive Activity and Sub-Aerial Plume Generation at South Sarigan, May 2010. Journal of Volcanology and Geothermal Research 257 (2013): 3143. https://doi.org/10.1016/j.jvolgeores.2013.03.006.

[6] Hochreiter S, and Schmidhuber J., Long Short-Term Memory. Neural Computation 9, no. 8 (1997): 173580. https://doi.org/10.1162/neco.1997.9.8.1735.

[7] Klambauer G., Unterthiner T., Mayr A., and Hochreiter S., Self-Normalizing Neural Networks. In 31st Conference on Neural Information Processing Systems (NIPS 2017). California, 2017.

[8] KNMI (1993): Netherlands Seismic and Acoustic Network. Royal Netherlands Meteorological Institute (KNMI). Other/Seismic Network. https://doi.org/10.21944/e970fd34-23b9-3411-b366-e4f72877d2c5

[9] Kuyuk S. H., and Susumu O., Real-Time Classification of Earthquake Using Deep Learning. Procedia Computer Science 140 (2018): 298305. https://doi.org/10.1016/j.procs.2018.10.316.

[10] Le Bras R., Nimar A., Noriyuki K., et al., NET-VISA from Cradle to Adulthood. A Machine-Learning Tool for Seismo-Acoustic Automatic Association. Pure and Applied Geophysics, (2020). https://doi.org/10.1007/s00024-020-02508-x.

[11] Lecocq T. et al. "Global quieting of high-frequency seismic noise due to COVID-19 pandemic lockdown measures." Science 23 (2020). https://doi.org/10.1126/science.abd2438.

[12] Linville L., Pankow K., and Draelos T., Deep Learning Models Augment Analyst Decisions for Event Discrimination. Geophysical Research Letters 46, no. 7 (2019): 364351. https://doi.org/10.1029/2018gl081119.

[13] Lomax A., Michelini A., and Jozinovi D., An Investigation of Rapid Earthquake Characterization Using SingleStation Waveforms and a Convolutional Neural Network. Seismological Research Letters 90, no. 2A (2019): 51729. https://doi.org/10.1785/0220180311.

[14] Metz D., Watts A. B., Grevemeyer I., Rodgers M., and Paulatto M., Ultra-Long-Range Hydroacoustic Observations of Submarine Volcanic Activity at Monowai, Kermadec Arc. Geophysical Research Letters 43, no. 4 (2016): 152936. https://doi.org/10.1002/2015gl067259.

[15] Mousavi S. M., and Beroza G. C., A MachineLearning Approach for Earthquake Magnitude Estimation. Geophysical Research Letters 47, no. 1 (2020). https://doi.org/10.1029/2019gl085976.

[16] Mousavi S. M., Zhu W., Sheng Y., and Beroza G. C., CRED: A Deep Residual Network of Convolutional and Recurrent Units for Earthquake Signal Detection. Scientific Reports 9, no. 1 (2019). https://doi.org/10.1038/s41598-019-45748-1.

[17] Ochoa L. H., Nio L. F., and Vargas C. A., Fast Magnitude Determination Using a Single Seismological Station Record Implementing Machine Learning Techniques. Geodesy and Geodynamics 9, no. 1 (2018): 3441. https://doi.org/10.1016/j.geog.2017.03.010.

[18] Paitz P., Gokhberg A., and Fichtner A."A neural network for noise correlation classification". Geophysical Journal International, 212:2 (2017): 14681474. https://doi.org/10.1093/gji/ggx495

[19] RDSA/Seismo - Operational Chain Documentation. Royal Netherlands Meteorological Institute (KNMI). Web: http://oper.knmi.nl/seismologie/OperationalChainDoc/index.html

[20] Shani-Kadmiel S., Assink J. D., Smets P. S. M., and Evers L. G., Seismoacoustic Coupled Signals From Earthquakes in Central Italy: Epicentral and Secondary Sources of Infrasound. Geophysical Research Letters 45, no. 1 (2018): 42735. https://doi.org/10.1002/2017gl076125.

[21] Trani L., Pagani G. A., Perreira Zanetti J. P., Chapeland C. G. M., Evers L. "DeepQuake An application of CNN for seismo-acoustic event classification in The Netherlands". (in preparation)

[22] Trnkoczy A., Understanding and Parameter Setting of STA/LTA Trigger Algorithm. GFZpublic. Deutsches GeoForschungsZentrum GFZ, (1970). https://gfzpublic.gfz-potsdam.de/pubman/faces/ViewItemOverviewPage.jsp?itemId=item_43337.

[23] Vaezi Y., and Van Der Baan M., Analysis of Instrument Self-Noise and Microseismic Event Detection Using Power Spectral Density Estimates. Geophysical Journal International 197, no. 2 (2014): 107689. https://doi.org/10.1093/gji/ggu036.

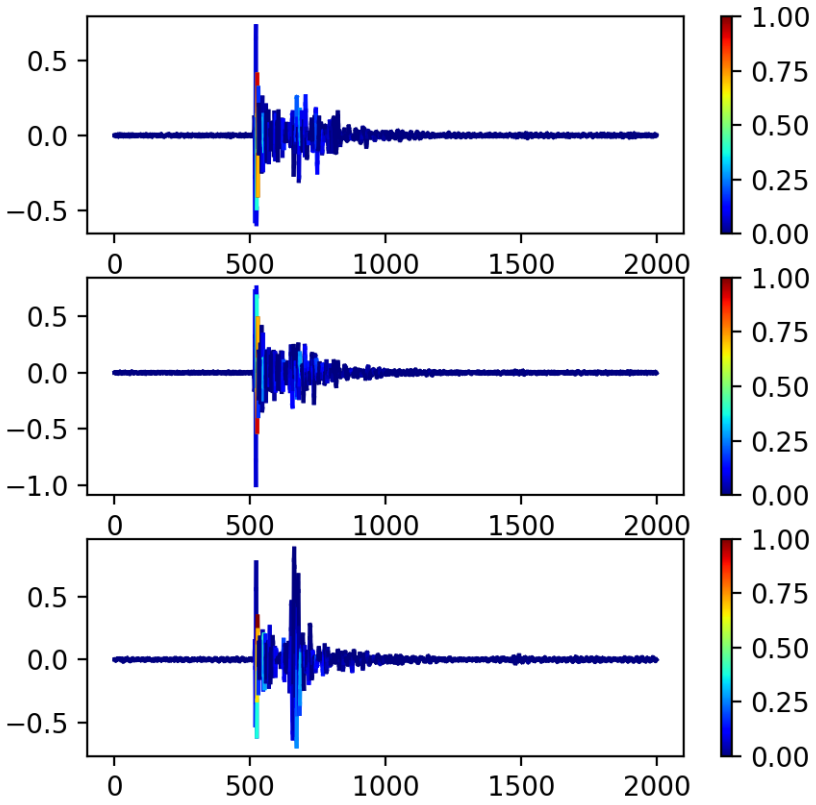
[24] Vaezi Y., and Van Der Baan M., Comparison of the STA/LTA and Power Spectral Density Methods for Microseismic Event Detection. Geophysical Journal International 203, no. 3 (2015): 18961908. https://doi.org/10.1093/gji/ggv419.

[25] Welcome to the Project Site of SeisComP3. Seiscomp3 Development. Helmholtz Cent. Potsdam, GFZ German Res. Cent. Geosc. and gempa GmbH. (2014). https://www.seiscomp3.org/.

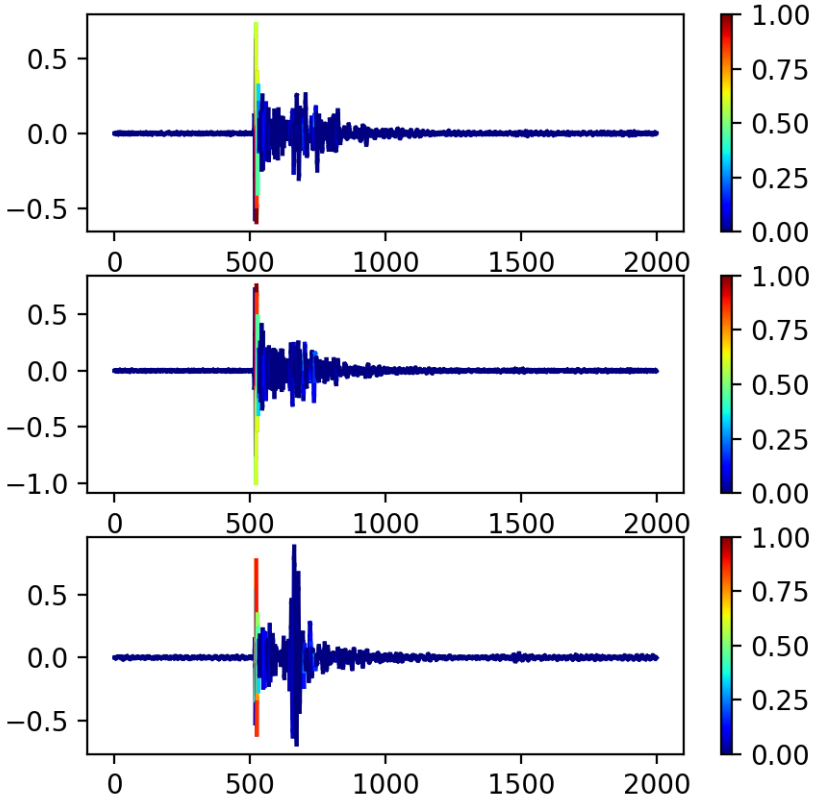
[26] Whaley, J., and Sorkhabi, R., The Groningen Gas Field. GEO EXPro. (2014) Web: www.geoexpro.com/articles/2009/04/the-groningen-gas-field.

13 Appendix

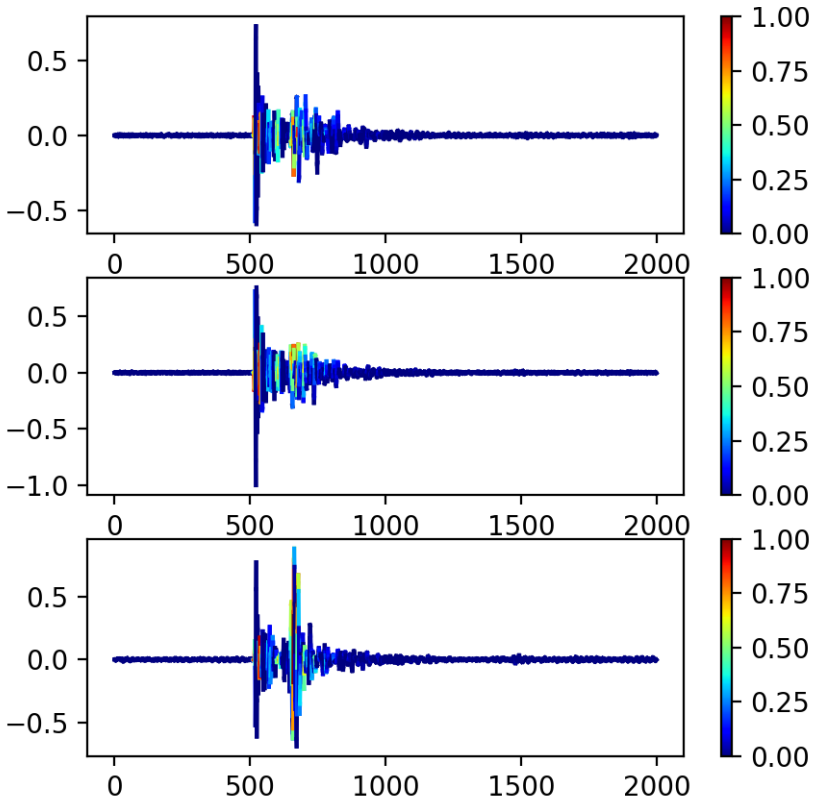
In this appendix are given the heatmap plots related to the analysis in section 7.2. The plots are of the activation functions response to a sample input: an explosion time-series. The activation is given between 0 and 1 (colour scale) and areas of high activation are areas the model relied on to make the correct prediction.



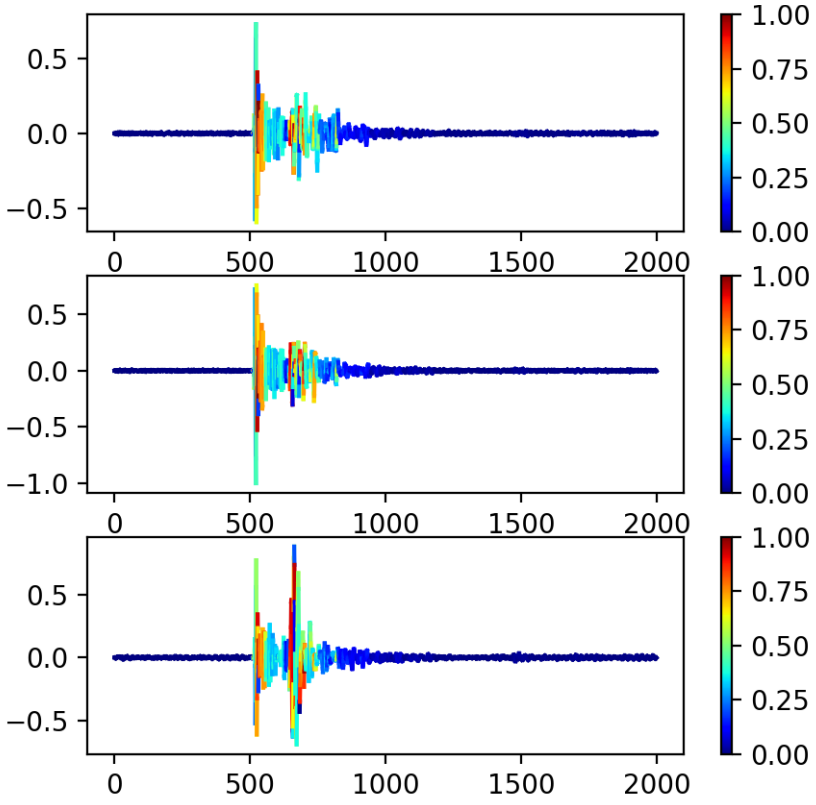
(a) *Filter 1 of convolutional layer 1*



(b) *Filter 7 of convolutional layer 1*

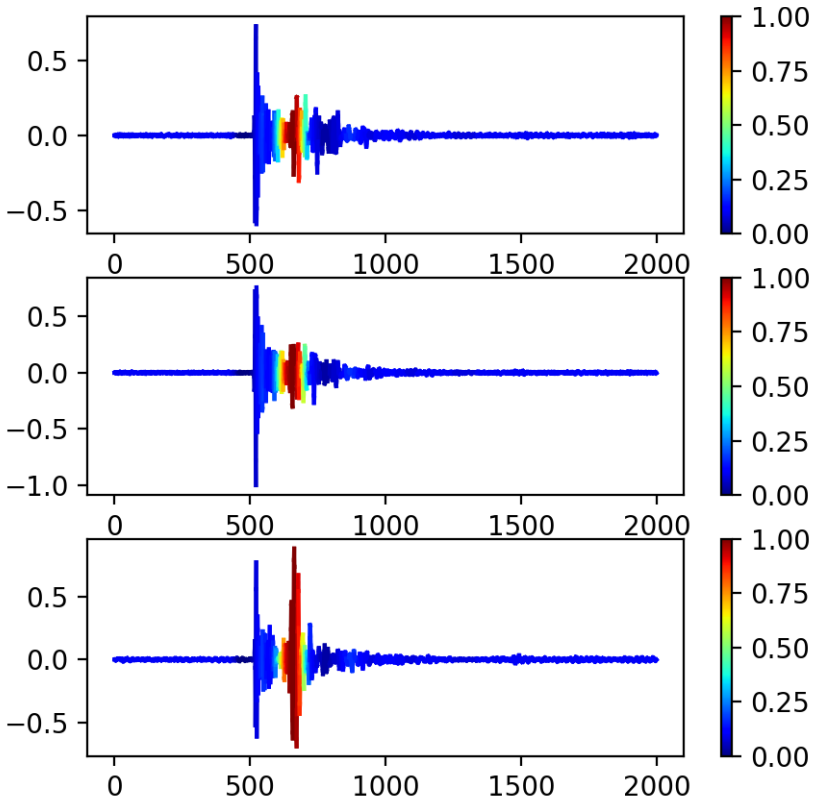


(c) *Filter 15 of convolutional layer 1*

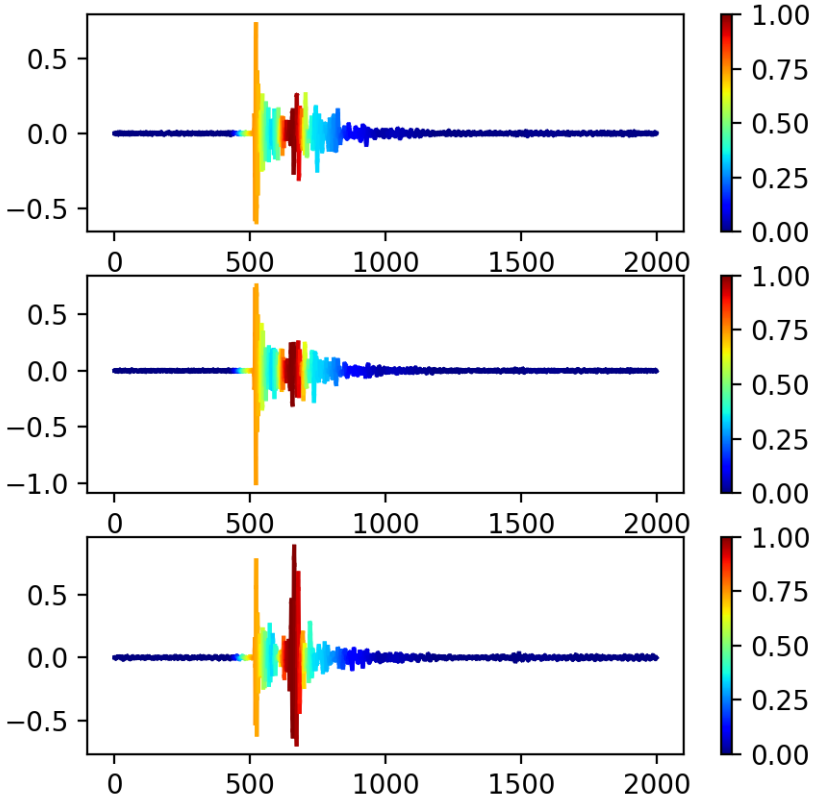


(d) *Averaged filter activations for convolutional layer 1*

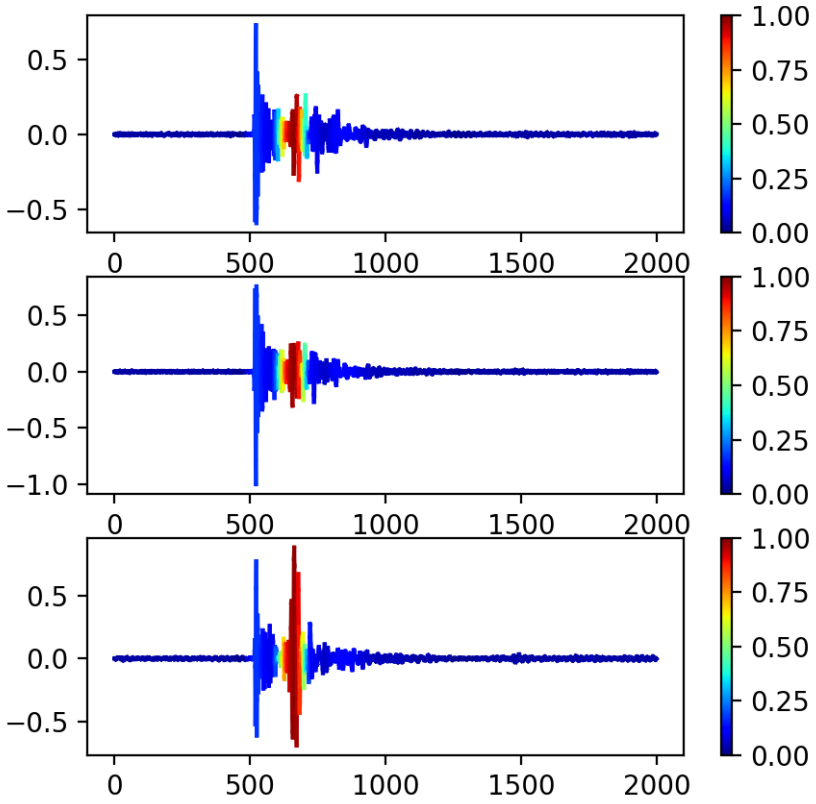
Figure 15: *Heatmaps of specific filter activations (a, b and c) and averaged activation function outputs for all 64 filters (d) in the first 1D convolutional layers of the spectro-temporal model. In this layer the model recognises low-level (small) features of the input.*



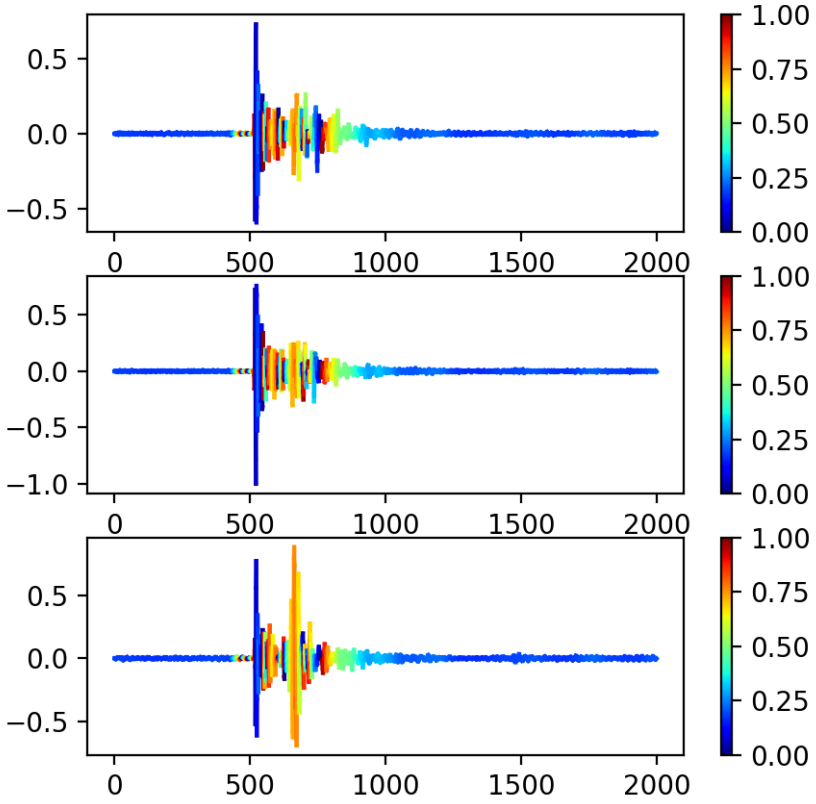
(a) *Filter 12 of convolutional layer 3*



(b) *Filter 19 of convolutional layer 3*

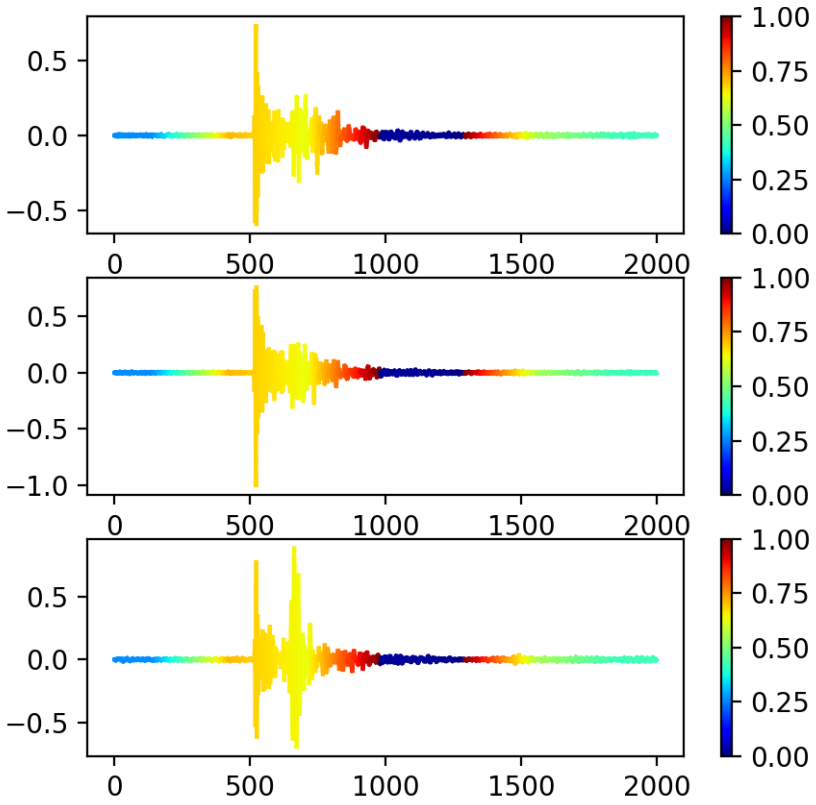


(c) *Filter 27 of convolutional layer 3*

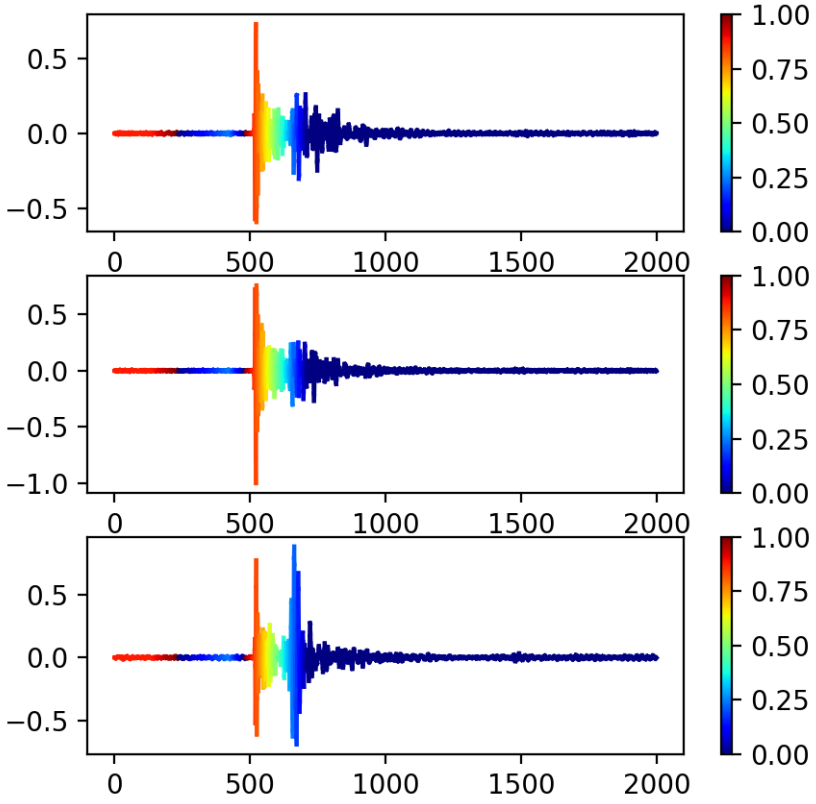


(d) *Averaged filter activations for convolutional layer 3*

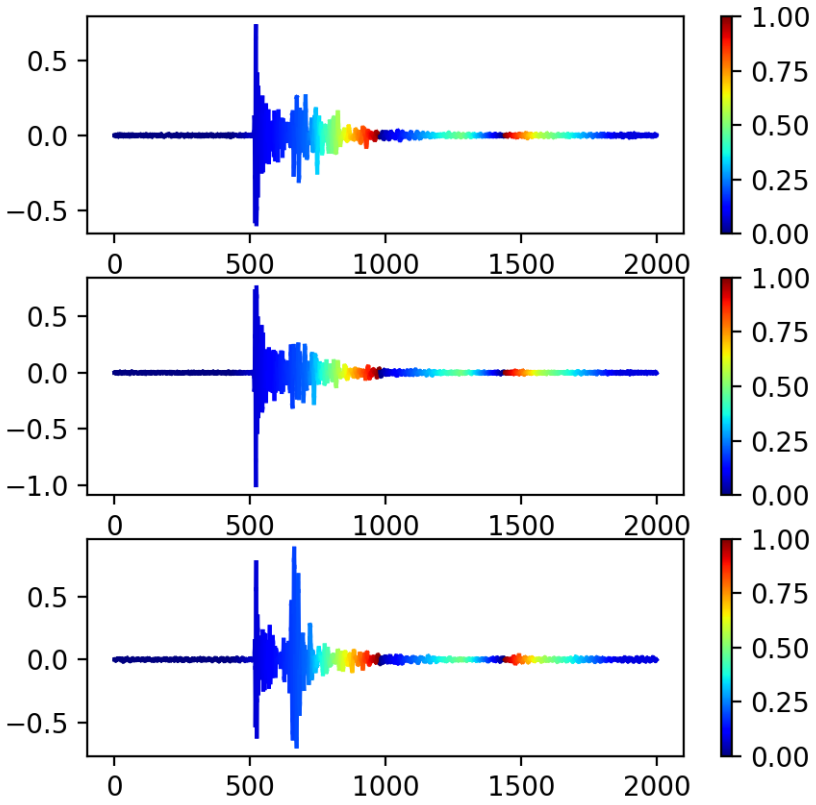
Figure 16: *Heatmaps of specific filter activations (a, b and c) and averaged activation function outputs for all 64 filters (d) in the third 1D convolutional layers of the spectro-temporal model. In this layer the model recognises mid-level (medium-sized) features of the input.*



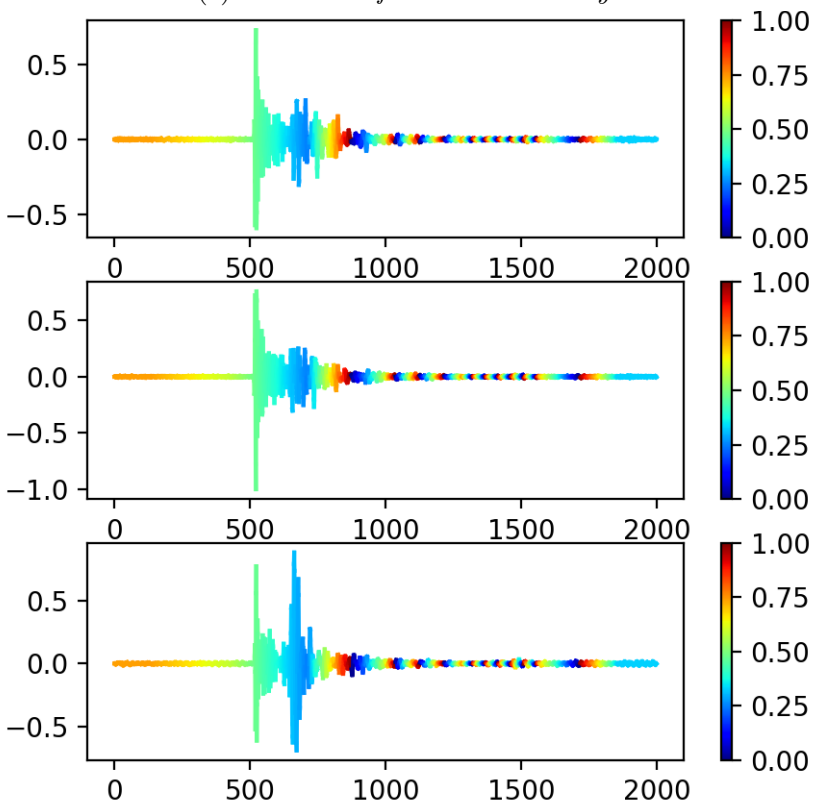
(a) *Filter 1 of convolutional layer 6*



(b) *Filter 4 of convolutional layer 6*



(c) *Filter 16 of convolutional layer 6*



(d) *Averaged filter activations for convolutional layer 6*

Figure 17: *Heatmaps of specific filter activations (a, b and c) and averaged activation function outputs for all 64 filters (d) in the last 1D convolutional layers of the spectro-temporal model. In this layer the model recognises high-level (large) features of the input.*