

Deep Neural Networks for Earthquake Detection and Source Region Estimation in North Central Venezuela

by Ruben Tous¹, Leonardo Alvarado^{2,4},
Beatriz Otero¹, Leonel Cruz¹ and Otilio Rojas^{2,3}

¹Universitat Politècnica de Catalunya (UPC). Barcelona, Spain.

²Universidad Central de Venezuela, Facultad de Ciencias, Caracas, Venezuela.

³Barcelona Supercomputing Center (BSC). Barcelona, Spain.

⁴Venezuelan Foundation for Seismological Research, FUNVISIS, Caracas, Venezuela.

Abstract

Reliable earthquake detection algorithms are necessary to properly analyze and catalog the continuously growing seismic records. We report the results of applying a deep convolutional neural network, called UPC-UCV, over single-station three-channel signal windows for P-wave earthquake detection and source region estimation in north central Venezuela. The analysis is performed on a new dataset of hand-picked arrivals of P-waves from local events, named CARABOBO, built and made public for reproducibility and benchmarking purposes. The CARABOBO dataset consists of three-channel continuous data recorded by the broadband stations of the Venezuelan Foundation of Seismological Research (FUNVISIS) in the region of 9.5 to 11.5°N and 67.0 to 69.0°W during the time period from April 2018 to April 2019. During this period, 949 earthquakes were recorded in that area, corresponding to earthquakes with magnitudes in the range from 1.1 to 5.2 Mw. To estimate the epicentral source region of a detected event, the proposed network employs geographical distribution of the CARABOBO dataset into K clusters as a basis. This geographical partitioning is automatically performed by the k-means algorithm, and the optimality of the K values for our dataset has been assessed using the Elbow ($K = 5$) and Silhouette ($K = 3$) methods. For target seismicity, the proposed network achieves 95.27% detection accuracy and 93.36% source region estimation accuracy when using $K = 5$ geographic clusters. The location accuracy slightly increases to 95.68% in the case of $K = 3$ geographic partitions. The detection capability of this network has also been tested on the OKLAHOMA dataset, which compiles over two thousand local earthquakes that occurred in this US state. Without any modification, the proposed network yields excellent detection results when trained and evaluated on that dataset (98.21% accuracy; ConvNetQuake, fine-tuned for this dataset, achieves a 97.32% accuracy), corresponding to a totally different geographical region.

Introduction

The number of seismic networks and monitoring sensors have steadily increased in recent years, and the continuous growth of seismic records call for new processing algorithms that

assist in solving problems in seismology. A fundamental endeavor is earthquake recognition, which pertains to the identification of seismic events in continuous data, with real-time application to early warning systems, or offline data postprocessing in the search for undetected past earthquakes. Based on time-dependent or spectral analyses of seismic traces, earthquakes of large or moderate magnitudes are easy to detect by conventional algorithms, some of which are cited below. Such earthquakes may impact the economy and human lives, but they are more rare and therefore scarce in seismic catalogs. Alternatively, small-magnitude earthquakes are more frequent, but they sometimes pass undetected by trained analysts or automatic recognition if, for instance, traces present poor signal-to-noise (SNR) ratios or recordings of overlapping events. Thus, seismic catalogs may be incomplete in the low magnitude range, as mentioned in Rydelek and Sacks (1989), Mignan and Woessner (2012), Wiemer and Wyss (2002) and Ross et al. (2019). Such catalogs are used in seismic hazard analysis for ground motion estimation under a given earthquake magnitude, which serves as a basis for building codes. In addition, accurate detection of the foreshocks and aftershocks associated with a main shock are used to constrain the actual fault configuration, total rupture area, and evolution of the tectonic stresses. All of these topics are fundamental in seismology, and they allow for better understanding of the hosting fault system and assessment of its destructive potential under surrounding geological conditions.

Conventional algorithms have been developed for automatic detection of seismic phases and picking the arrival times of primary (P) and secondary (S) waves. Most of these techniques rely on general properties of seismic waves, waveform attributes, and statistical correlation or wave polarization analyses. A popular P-wave detector developed in Allen (1978) and Allen (1982) calculates the ratio of the average of the absolute amplitude of a signal in a short time window (STA) to the average of the signal in a long time window (LTA). In cases in which the STA/LTA ratios exceed a given threshold at a minimum number of network stations, the seismic event is considered as an earthquake. Some generalizations of the STA/LTA algorithm that also allow phase picking were later proposed in Baer and Kradolfer (1987), Earle and Shearer (1994) and Nippress et al. (2010). The phase identification of S waves is usually difficult due to superposition with P coda and converted phases, and some elaborate picking methods were developed in Roberts et al. (1989), Cichowicz (1993), Diehl et al. (2009) and Baillard et al. (2014). The P and S detection algorithm presented by Ross and Ben-Zion (2014) combines some of the aforementioned waveform analyses with STA/LTA measures. By using recordings from local stations, this method is also able to pick fault-zone head waves that may be triggered during earthquakes.

A different class of detection algorithms are based on pattern recognition or intensive autocorrelation. Most methods build appropriate pattern sets of earthquake and noise signals, and the association of new trace windows with one set based on sufficient similarity serves as the discrimination basis for detection. Among the early pattern recognition methods, one can find Anderson (1982), Chen (1984), Joswig (1990) and Klumpen and Joswig (1993). Earthquake signals that share the same source mechanism, along with similar path and site conditions, should present strong waveform similarities. Earthquake detection by intensive autocorrelation is undertaken by the methods in Gibbons and Ringdal (2006) and Brown et al. (2008). This category of detection algorithms naturally adapts to general regional and network conditions and behaves reliably, but obtaining results is computationally intensive when processing large datasets. To reduce the computational cost, this class has evolved

into the template matching algorithms that maintain a reduced database of representative waveforms to be fully compared against new traces or that extract and store key discriminative features from waveforms to be used in the similarity search. Some works focused on earthquake swarms and sequences of foreshocks and aftershocks are those of Shelly et al. (2006), Harris and Dodge (2011), Skoumal et al. (2015), Benz et al. (2015), and the Fingerprint and Similarity Thresholding (FAST) algorithm in Yoon et al. (2015). In FAST, discriminative features are binary fingerprints, stored in a special dictionary according to their similarity, that allow for efficient searches owing to a locality-sensible hashing. Special FAST adaptations to sparse seismic networks have been recently proposed in Bergen and Beroza (2018).

Artificial neural networks (ANNs) represent a suitable framework for seismic phase detection due to their proficiency in complex pattern recognition. Since the mid-1990s, there has been a tremendous amount of contributions that vary in terms of the underlying learning technique, network architecture and functionality, and spatial extent of the documented application, among other distinct aspects. Early works based on supervised learning involve simple feed forward neural networks (FFNNs), which were followed by some scattered cases of recurrent neural networks (RNNs), and we then find some recent applications of very deep convolutional neural networks (CNNs). FFNNs for seismic phase detection have been proposed in Dai and MacBeth (1995), Wang and Teng (1995), Enescu (1996), Dai and MacBeth (1997), Wang and Teng (1997) and Zhao and Takano (1999) and applied to local and regional earthquake data. In particular, the testing dataset used in Dai and MacBeth (1995) consists of slightly more than 850 local events, while Zhao and Takano (1999) employed over 1200 seismograms of the IRIS network for training and validation. Alternatively, the FFNNs introduced in Tiira (1999) make use of a dataset of P-wave signals from 193 teleseismic events recorded at 3 short period stations in central Finland. In Gentili and Michelini (2006), the singular network IUANT2 presents a problem adaptive structure that is inferred during the training process. This technique is used for phase picking by employing records of 342 local earthquakes recorded by 23 different stations (approximately 5K traces). Wiszniowski et al. (2014) implement an RNN for real-time detection of small magnitude (below 2.5 Mw) earthquakes in populated and noisy areas owing to an elaborate design of filter banks processing the STA/LTA ratios of seismic waveforms. This process allows for the extraction of relevant frequencies from input data, making this network less prone to false detection occurrences. The initial training relies on 170 events, including regional and teleseismic earthquakes, and the full operating network was tested during different time periods in 2009. Recent applications of RNNs to early warning systems and to earthquake prediction are given in Ibrahim et al. (2018) and Bhandarkar et al. (2019), respectively.

Among the state-of-the-art CNNs for earthquake detection, we find ConvNetQuake developed by Perol et al. (2018), the generalized phase detection (GPD) introduced in Ross et al. (2018), and PhaseNet proposed by Zhu and Beroza (2019). ConvNetQuake operates on three-channel seismograms from local stations for P-wave detection and has been used for the recognition of natural or human-induced (related to waste water injection) low-magnitude events in central Oklahoma, USA (hereafter referred to as the OKLAHOMA dataset). For training, the authors employ a seismic catalog of over two thousand events that occurred during 2014-2016 and then validate the network by using 209 additional events. This network is extensively used in this work as a reference, so additional ConvNetQuake characteristics

and new performance assessments will be discussed in the following sections. The training and validation of GDP and PhaseNet make use of the extremely large seismic datasets, properly labeled by human analysts, that are available from the Californian seismic networks. In particular, GDP detection has also been tested on the 2016 Bombay Beach, California swarm of small and moderate (≤ 4.8 Mw) events and with the data of the 7.0 Mw Kumamoto earthquake that occurred in 2016.

The aforementioned detection ANNs, as well as multiple alternative applications, could also be compared in terms of data preprocessing (trace normalization, filtering, windowing, etc.) and achieved accuracy, among other possible aspects. An interested reader, in addition to directly inspecting our listed references, may review any of the following survey papers: Baan and Jutten (2000), Florido et al. (2016), Kong et al. (2019) and Rojas et al. (2019). For locating the earthquake hypocenter, as part of the core information in seismic catalogs, only a few scattered estimation approaches incorporating ANNs can be found. For instance, ConvNetQuake maps hypocenters to a cluster of different geographical areas based on a Voronoi partition. Alternatively, the detection CNN proposed by Kriegerowski et al. (2019) is trained on an earthquake swarm of 2000 events recorded by several local three-component stations. During testing, this network locates approximately 900 earthquakes with standard deviations of nearly 56 m, 124 m and 136 m along the east-west, north-south and vertical directions, respectively. Lastly, the detection CNN introduced in Zhang et al. (2018) constructs a 3D probability volume of location likelihood inside the Earth. This work employs the OKLAHOMA dataset for training and evaluating the model, and reports an average error of 4.9 km in the epicenter location estimation (1.0 km in the depth estimation).

Here we address the earthquake detection and epicentral estimation problems for seismic data collected by broadband stations in north-central Venezuela during the time period from April 2018 to April 2019. Two main contributions of this study are as follows:

- A new analyst-labeled dataset, called CARABOBO according to the recording region, was built and made public for result reproducibility and benchmarking purposes. This dataset contains seismic data related to 949 earthquakes with magnitudes ranging between 1.1 Mw and 5.2 Mw.
- A deep CNN is proposed, called UPC-UCV and inspired by ConvNetQuake, to approach the P-wave detection problem by processing three-channel seismic signals in the mentioned geographic area. The method also performs source region estimation, mapping positive events to geographical partitions (automatically obtained with the k-means clustering algorithm) of the study region. Although our epicentral estimation lacks depth approximation for complete hypocentral location, it represents information that is available in real time and is complementary to standard earthquake location procedures.

Methodology

The CARABOBO Dataset

The CARABOBO dataset of analyst-labeled P-wave arrivals is based on the data provided by The Venezuelan Foundation for Seismological Research (FUNVISIS), the governmental agency for monitoring and reporting seismic activity in Venezuela. The FUNVISIS network includes 35 broadband stations recording three-channel continuous data at 100 Hz. The stations are mainly deployed in the regions with higher seismic activity which are close to the active fault systems in northern Venezuela. The dataset is the first for this region and contains seismic data (in miniSEED format) collected by broadband stations in north-central Venezuela, in the region of 9.5 to 11.5°N and 67.0 to 69.0°W, during the time period from April 2018 to April 2019. During this period, 949 earthquakes were recorded in that area by 5 seismological stations (BAUV, BENV, MAPV, TACV and TURV). The dataset also includes a catalog with the metadata related to the events (hypocenter, P-wave arrival times, magnitude, etc.) in Nordic format.

The study region (figure 1), characterized by shallow seismicity, includes the central part of the Caribbean mountain system with important cities such as Caracas, Valencia and Maracay. The magnitudes of the earthquakes range between 1.1 and 5.2 Mw (figure 2 shows the magnitude distribution of the events); P- and S wave arrival times are reported in the FUNVISIS catalogs (FUNVISIS (2018), FUNVISIS (2019)). Most seismic events belong to a seismic swarm with epicenters located within northwestern Valencia (Carabobo state). The seismicity of the region is associated with the San Sebastián and La Victoria fault systems, along with some minor faults such as the Las Trincheras and Morón faults, all of which are right lateral strike-slip faults (Audemard et al. (1995), Singer et al. (2014)). The San Sebastián and La Victoria faults belong to the continental scale Boconó - San Sebastián - El Pilar fault system along the interface between the Caribbean and South American plates (Audemard (2007)).

It is worthy of mentioning that the study region in figure 1 concentrates a large population and a variety of big industries, and the human and industrial activities induce a high level of seismic noise. Unfortunately, some of the seismic stations of the local FUNVISIS network (also shown in figure 1) failed during the recording period of this study, and the seismic processing of the recorded data was carried out by trained analysts assisted by STA/LTA detection software. The combination of these facts lead to a detection deficiency of small events, and reflect on the apparent scarcity of the seismic activity below 2 Mw, as depicted in figure 2. This cumulative earthquake distribution does not follow a standard frequency magnitude relation, such as the GutenbergRichter law. However, more elaborate earthquake detection procedures could reveal several small events, allowing a future enrichment of the CARABOBO dataset.

Earthquake Detection and Source Region Estimation

Earthquake detection and source region estimation are both approached as a single multiclass classification problem, in which fixed-size windows of waveform data must be classified into $K + 1$ classes. The first K classes are for windows containing a P-wave from an event whose

epicenter is located in one of the given K source regions. The last class is for windows that are free of any P-wave arrival. The proposed method (UPC-UCV) is inspired by ConvNetQuake (Perol et al. (2018)), which addresses these problems with a supervised learning approach. ConvNetQuake consists of a series of data preprocessing steps to convert the input waveforms into a set of fixed-size windows and the application of a deep convolutional network to infer the best class for a given window. ConvNetQuake was the first choice to perform detection over the CARABOBO dataset. However, the obtained results (81.95% detection accuracy and 82.08% source region estimation accuracy with $K = 5$) were not adequate, and it was necessary to perform a series of enhancements, both in the preprocessing strategy and in the network design, which are described below.

Data Preprocessing

Several data preprocessing steps are applied in order to obtain the training and evaluation datasets. First, input streams with no data in any of the three channels are purged. Then, each input stream is normalized (independently for each of the three channels), and if the stream contains seismograms from multiple stations, it is divided into multiple 3-channel streams. Because different events can be detected by a different set of seismic stations (in different locations, a station is not always operative, etc.), processing single-station data enables obtaining a more homogeneous and larger training dataset. Each 3-channel stream is split into multiple 3-channel temporal windows of fixed size (window sizes in the range of 10-60 sec have been tested). With the event information obtained from the metadata files, the windows are divided into $K + 1$ classes (K classes for positive detection originating in one of the given K source regions and one class for negative detection). These preprocessing steps are identical to those performed by ConvNetQuake, with the exception that ConvNetQuake only reports results with 10 sec windows. However, to improve the detection performance, the proposed method incorporates some preprocessing improvements. Unlike ConvNetQuake, which splits the input streams into non-overlapping windows, the proposed method applies a sliding window approach so that the resulting windows partially overlap. Another difference with respect to ConvNetQuake involves dataset balancing. ConvNetQuake works with an extremely imbalanced dataset, but the proposed method balances the training data by undersampling the negative data. Finally, unlike ConvNetQuake, bandpass filtering ($[0.5, 10]$ Hz) is applied to attenuate the noise. This preprocessing setup applied over the CARABOBO dataset results in sixty thousand windows, 50% positive and 50% negative.

Geographical Clustering

The proposed method is essentially a signal window classifier. With only two classes/labels (P-wave or not P-wave), the method behaves similar to a P-wave detector. Increasing the number of classes and properly labeling the windows that contain a P-wave enable inference of properties about the event that triggered the detected P-wave. One possibility, the one addressed in this work, is attempting to determine the epicentral source region of the event. First, the study area is partitioned into K geographic subdivisions. Second, the positive windows are labeled with the identifier of the subdivision to which the event source location

belongs. Third, a classifier with $K + 1$ classes is trained. The first step, geographic partitioning, can be performed in many different ways, including manually. In this work, it is performed by clustering all of the event locations in the dataset with k-means (MacQueen (1967)) for different values of K (figure 1 shows the partitioning for $K = 5$). The tested values of K are the optimal ones resulting from the application of both the Elbow method and the Silhouette method. Figure 3 shows the results of both methods for finding the appropriate number of source regions.

UPC-UCV network architecture

The proposed model is a deep convolutional network inspired by the one described in Perol et al. (2018) but enhanced to be able to work properly with the CARABOBO dataset. First, the convolutional part of the network is flattened by enlarging the size of the convolutional kernels (from 3 to 20) and reducing the number of convolutional layers to only 4 (from 8 in ConvNetQuake). Greater robustness to noise and distortions is obtained by applying max pooling after each convolutional layer. Second, two additional fully connected layers are incorporated. All of these changes result in a significant increase in the number of trainable parameters (to 70K, from 23K in ConvNetQuake) but not in the training time, as the depth of the network becomes slightly reduced. Figure 4 shows the proposed network architecture.

The network takes a multiple-channel seismogram window as input and outputs $K + 1$ detection probabilities. The first K values estimate whether the window contains a P-wave triggered within one of the given source regions, and the last output is the probability that the window does not contain any P-wave. The input and output of the network are the same as those in ConvNetQuake, with the exceptions of the input window size (10 sec in ConvNetQuake; 50 sec in the best configuration of UPC-UCV) and the value of K (6 for the OKLAHOMA dataset targeted by ConvNetQuake; 5 and 3 for the CARABOBO dataset targeted by UPC-UCV).

The input of the model is a window of, in the best configuration of UPC-UCV, 50 sec. As the signal is sampled at 100 Hz, the window contains 5000 samples. Each sample presents three values, one for each component (N-S, E-W, and up-down). This would result in a tensor of shape $[5000, 3]$ (*[window size \times sampling rate, number of channels]*). However, in the same way as in ConvNetQuake, UPC-UCV’s network works with an input tensor of $[5000, 1, 3]$ (*[window size \times sampling rate, 1, number of channels]*), a more convenient shape to feed a ConvNet network, which is typically used to work with color images received as tensors with shape $[width, height, color\ channels]$. Thus, from a conventional ConvNet point-of-view, UPC-UCV’s input is like a just-one-row image with three color channels. Figure 4 attempts to clarify this by showing the input tensor as a flat cuboid (while a color image would be a normal 3D cuboid).

The network has four convolutional layers, each with an associated max pooling layer. All of the layers have 32 convolutional kernels. Conventionally, the deeper layers of a ConvNet, which work with higher-level features, contain fewer kernels than the first layers. However, after testing different configurations, it was decided to retain the same configuration as in ConvNetQuake for this aspect.

The convolutions are applied in 1D fashion (only through the temporal axis), but the kernels are 2D to process the multiple input channels. Given a 2D kernel k of size $s \times c$

(width s over c channels) of a given layer l , the weight matrix $W^{k,l} \in \mathbb{R}^{2s+1 \times c}$ is:

$$W^{(k,l)} = \begin{pmatrix} W_{-s,1}^{(k,l)} & \cdots & W_{-s,c}^{(k,l)} \\ \vdots & \vdots & \vdots \\ W_{0,1}^{(k,l)} & \cdots & W_{0,0}^{(k,l)} & \cdots & W_{0,c}^{(k,l)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ W_{s,1}^{(k,l)} & \cdots & W_{s,c}^{(k,l)} \end{pmatrix} \quad (1)$$

Empirically, the value of the s hyperparameter (size of the kernel) has been set to 20 (3 in ConvNetQuake). Thus, each convolutional kernel is a $20 \times c$ matrix, with c denoting the number of output channels of the previous layer (3 for the first convolutional layer and 32 for the following ones). The 20×3 cuboids in figure 4 correspond to the convolutional kernels of the first layer, i.e., to Equation 1 when $l = 1$, $s = 20$ and $c = 3$. Each kernel is applied to all of the 20-size windows of the input tensor with a stride of 2 (the same as in ConvNetQuake). For each of these windows, the discrete convolution of the input 2D tensor Y with kernel $W^{k,l}$ at position t of the input tensor (the kernel only moves in 1D) is given by:

$$(Y * W^{k,l})_t = \sum_{u=-s}^s \sum_{v=1}^c W_{u,v}^{k,l} * Y_{t+u,v} \quad (2)$$

The result of equation 2 is known as a *convolved feature map*. As each kernel is applied to different windows of the input tensor (different values of t), many convolved feature maps are obtained for each kernel (e.g., 2500 in the first convolutional layer). A bias and an activation function are applied to each convolved feature map to provide nonlinearity and obtain the final output of the layer (also known as a *rectified feature map*) at each position. The output at position t of a convolutional layer l and kernel k is computed as:

$$Y_{k,t}^{(l)} = \sigma(b_k^{(l)} + (Y^{(l-1)} * W^{k,l})_{t'}) \quad (3)$$

in which t' is the index of the input tensor. t' (t in Equation 2) depends not only on t but also on the stride (2 in this case). $\sigma(x) = \max(0, x)$ is the nonlinear ReLU (rectified linear unit) activation function, and $b_k^{(l)}$ is the bias term for kernel k in layer l . After each convolutional layer, we apply a max pooling layer with a pooling window of size 5 and stride 3.

After the last convolutional layer, the resulting 2D tensor is flattened into a 1D tensor. This feature vector is then processed by three fully connected layers with ReLU activation functions (10 neurons for the first ones and $K + 1$ neurons for the last one). The size of the last fully connected layer depends on the number of classes (two if only P-waves are being detected and $K + 1$ if the source region where the P-wave originated is also being inferred from K predefined regions). The outputs of the last fully connected layer, the so-called *logits*, are raw prediction values with a range of $[-infinity, +infinity]$. To transform these values into probabilities (range of $[0, 1]$, sum to one) a *softmax* function is applied. Each of the K outputs of the network is the probability (a float value within the range of $[0, 1]$) that the input window contains a P-wave that originated in one of the K regions. One output

of the network contains the probability that the input window does not contain a P-wave. This is the same output that is defined in ConvNetQuake.

The network is trained with 80% of the balanced dataset (48K windows), and the remaining 20% (12K windows) is used for the evaluation.

Experiments and Results

The main goal of the experiments is to evaluate the detection performance of the different aspects (including the data preprocessing strategy and the network architecture) of the UPC-UCV method. All of the reported results are obtained when evaluating each configuration over a test dataset of previously unseen windows (20% of the dataset). The experiments were conducted on one of the computing clusters of the Computer Architecture Department of the UPC. The cluster is accessed through a Sun Grid Engine (SGE) batch-queuing system and comprises 200 heterogeneous nodes, the newest ones being equipped with 2x Intel Xeon E5-2630L v4 processors running at 2.20 GHz with 128 GB of RAM.

Before deciding the final configuration, different data preprocessing configurations (window size, frequency filtering, number of channels, etc.) and different network geometries and parametrizations were tested. Table 1 summarizes the impacts of the different preprocessing choices on the performance. The obtained results confirm that all of the proposed modifications with respect to Perol et al. (2018) exert a positive impact on the performance. The preprocessing modification with the highest impact is the slicing of the input streams into highly overlapping windows, which can be seen as a phase shifting-based data augmentation technique. The usage of the three channels, which is mainly aimed at improving source region estimation, exerts a small impact on the P-wave detection accuracy, as expected.

One important hyperparameter is the window size. Window sizes in the range of 10-60 sec have been tested. Figure 5 shows the impact of the window size on the detection accuracy for both UPC-UCV and ConvNetQuake. It is interesting to observe how the optimal window size differs between methods, with larger windows (approximately 50 sec) being better for the proposed method.

Table 2 summarizes the results of UPC-UCV (with all of the described preprocessing and network architecture improvements) in comparison with the reference method, ConvNetQuake. UPC-UCV achieves a 95.27% detection accuracy, which is significantly better than the 81.95% (balanced) accuracy obtained by ConvNetQuake in its original setup (10 sec windows and without the proposed preprocessing improvements). UPC-UCV is also better if the proposed preprocessing improvements are also applied to ConvNetQuake while retaining its optimal 10 sec window size. Table 2 summarizes the results of UPC-UCV (with all the described preprocessing and network architecture improvements) in comparison with the reference method, ConvNetQuake. While the goal of this work is to provide reliable detection and source region estimation for the target region, its performance when trained and evaluated on the dataset employed in Perol et al. (2018) (the OKLAHOMA dataset) has also been assessed to study the applicability of the proposed network setup to different seismic data. Because the experiment is based on an already preprocessed dataset, the test is limited to a configuration of 10 sec windows without overlap. Although the UPC-UCV hyperparameters were not fine-tuned over this dataset, it achieves a 98.21% detection accuracy,

while ConvNetQuake reaches a similar 97.32% (balanced) accuracy.

Figures 6 and 7 compare test cases with satisfactory and wrong P-wave detections. Most of the false positives are windows shortly after the P-wave (out-of-phase detection). When applied in a real setup, these false positives could be avoided by introducing an exclusion window after the detection of the P-wave (if there is a true positive detection). In the worst cases, out-of-phase detection implies a false negative followed by a false positive. Irregular waveforms related to the heterogeneous characteristics and states of the sensors are the other main source of detection errors.

Table 3 shows the results obtained in the performed source region estimation experiments (for both the reference model and the proposed model). These results, a 95.68% accuracy with $K = 3$ and a 93.36% accuracy with $K = 5$, are satisfactory considering that the source region is being estimated with the information of just one seismic station, and they confirm that the conclusions obtained by Perol et al. (2018) are also extrapolatable to other seismicities. The UPC-UCV results are also significantly better than those achieved with ConvNetQuake.

Regarding the computational efficiency of UPC-UCV during inference time, processing a window (which happens every 10 sec of the input stream) takes an average of 4.4 milliseconds. Such performance would enable our network to process a one hour stream in less than 2 seconds, allowing an efficient real-time processing. This performance is very similar to that observed for ConvNetQuake, which takes an average of 3.8 milliseconds to process each window. Regarding the memory footprint, UPC-UCV (which has 70K parameters) occupies about 3.5 MB of memory during inference time. ConvNetQuake, which has only 23K parameters, occupies about 3 MB, that represents a similar cost. Some of the overheads introduced by TensorFlow are not actually proportional to the number of network parameters.

Conclusions

Results of the proposed UPC-UCV method, consisting of applying a convolutional neural network to single-station 3-channel waveforms for P-wave earthquake detection and source region estimation in north-central Venezuela, have been reported. In addition, the CARABOBO dataset of analyst-labeled P-wave arrivals, named after the studied seismic region, has been built and made public for reproducibility and benchmarking purposes. Both the UPC-UCV network and the CARABOBO dataset are the first developed for this geographic region. This convolutional network is inspired by ConvNetQuake developed by Perol et al. (2018) but incorporates many improvements, both in the data preprocessing strategy and the network architecture, to yield higher detection accuracy (13.3 percentage point increase) for the new target seismicity. Regarding the preprocessing strategy, the main modifications are the usage of larger and overlapping windows, the balancing of the dataset and the application of a bandpass filter to input seismic signals. Regarding the network architecture, the main enhancements are the flattening of the convolutional part (fewer layers with larger kernels), max pooling and more fully connected layers. UPC-UCV achieves a 95.27% detection accuracy over a subset of windows of the CARABOBO dataset that were never seen during training, while ConvNetQuake obtains a 81.95% detection accuracy over the same subset. Regarding source region estimation, UPC-UCV achieves a 95.68% accuracy

with $K = 3$ geographic partitions, while ConvNetQuake obtains an 84.58% accuracy. In the case of $K = 5$ geographic partitions, UPC-UCV reaches a 93.36% accuracy, compared to ConvNetQuake that yields an 82.08% accuracy. The dataset partitioning into K geographical clusters is automatically performed by the k-means algorithm, and the optimality of these K values has been assessed using the Elbow and the Silhouette methods. In addition, without any modification, this network yields excellent detection results when trained and evaluated on the OKLAHOMA dataset, which corresponds to a totally different geographical region (98.21% detection accuracy; ConvNetQuake, fine-tuned for this dataset, achieves a 97.32% detection accuracy).

Data and Resources

Both the data and the source code used in this work are publicly available in order to enable the reproducibility of our results and to facilitate the evaluation and benchmarking of other methods. The source code of the UPC-UCV method is available on <https://github.com/rtous/deepquake> (last accessed May 15, 2020). The CARABOBO dataset is available on <https://github.com/rtous/CARABOBO2019> (last accessed May 15, 2020).

Acknowledgments

We are thankful to BSSA editors and Ursula Iturraran Viveros for insightful comments and revisions that help us to improve this manuscript. We also acknowledge FUNVISIS for providing the seismic data subject of reported studies. This work is partially supported by the Spanish Ministry of Economy and Competitiveness under contract TIN2015-65316-P, by the Spanish Ministry of Science and Innovation under contract PID2019-107255GB-C22, and by the SGR programmes (2017-SGR-1414 and 2017-SGR-962) of the Catalan Government. We also thank the Generalitat de Catalunya for providing additional funds under the RIS3CAT DRAC project (001-P-001723). The research leading to these results has received funding from the European Union’s Horizon 2020 research and innovation programme under the ChEESE project, grant agreement No. 823844. Additionally, this project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 777778 MATHROCKS.

References

- Allen, R. (1982). Automatic phase pickers: Their present use and future prospects, *Bull. Seism. Soc. Am.*, **72**(6B) S225–S242.
- Allen, R. V. (1978). Automatic earthquake recognition and timing from single traces, *Bull. Seism. Soc. Am.*, **68**(5) 1521–1532.
- Anderson, K. R. (1982). Syntactic analysis of seismic waveforms using augmented transition network grammars, *Geoexploration*, **20**(1) 161–182, doi:10.1016/0016-7142(82)90013-8.

- Audemard, F. A. (2007). Revised seismic history of the El Pilar fault, Northeastern Venezuela, from the Cariaco 1997 earthquake and recent preliminary paleoseismic results, *J. Seismol.*, **11** 311–326, doi:10.1007/s10950-007-9054-2.
- Audemard, F. A., F. De-Santis, A. Singer, and C. Ramos (1995). Sistema de falla de La Victoria, Venezuela norcentral: trazas activas, complejidades estructurales, cinemática y sismicidad asociada, in M. de Energía y Minas, editor, *Trabajos de autores venezolanos presentados en el IX Congreso Latinoamericano de Geología*, pages 1–15, Caracas, Venezuela, ISSN 0505-1169.
- Baan, M. V. D. and C. Jutten (2000). Neural networks in geophysical applications, *Geophysics*, **65**(4) 1032–1047, doi:10.1190/1.1444797.
- Baer, M. and U. Kradolfer (1987). An automatic phase picker for local and teleseismic events, *Bull. Seism. Soc. Am.*, **77**(4) 1437–1445.
- Baillard, C., W. C. Crawford, V. Ballu, C. Hibert, and A. Mangeney (2014). An automatic kurtosis-based P- and S-phase picker designed for local seismic networks, *Bull. Seism. Soc. Am.*, **104**(1) 394–409, doi:10.1785/0120120347.
- Beltrán, C. (1993). Mapa Neotectónico de Venezuela, 1:2,000,000 scale, FUNVISIS.
- Benz, H. M., N. D. McMahon, R. C. Aster, D. McNamara, and D. B. Harris (2015). Hundreds of earthquakes per day: The 2014 Guthrie, Oklahoma, earthquake sequence, *Seismol. Res. Lett.*, **86**(5) 1318–1325, doi:10.1785/0220150019.
- Bergen, K. J. and G. C. Beroza (2018). Detecting earthquakes over a seismic network using single-station similarity measures, *Geophys. J. Int.*, **213**(3) 1984–1998, doi:10.1093/gji/ggy100.
- Bhandarkar, T., K. Vardaan, N. Satish, S. Sridhar, R. Sivakumar, and S. Ghosh (2019). Earthquake trend prediction using long short-term memory RNN, *Int. J. Electr. Comput. Eng.*, **9**(2) 1304–1312, doi:10.11591/ijece.v9i2.pp1304-1312.
- Brown, J. R., G. C. Beroza, and D. R. Shelly (2008). An autocorrelation method to detect low frequency earthquakes within tremor, *Geophys. Res. Lett.*, **35**(L16305) 1–5, doi:10.1029/2008GL034560.
- Chen, C. H. (1984). On a segmentation algorithm for seismic signal analysis, *Geoexploration*, **23**(1) 35–40, doi:10.1016/0016-7142(84)90022-X.
- Cichowicz, A. (1993). An automatic S-phase picker, *Bull. Seism. Soc. Am.*, **83**(1) 180–189.
- Dai, H. and C. MacBeth (1995). Automatic picking of seismic arrivals in local earthquake data using an artificial neural network, *Geophys. J. Int.*, **120**(3) 758–774, doi:10.1111/j.1365-246X.1995.tb01851.x.
- Dai, H. and C. MacBeth (1997). The application of back-propagation neural network to automatic picking seismic arrivals from single-component recordings, *J. Geophys. Res. B Solid Earth Planets*, **102**(B7) 15105–15113, doi:10.1029/97JB00625.

- Diehl, T., N. Deichmann, E. Kissling, and S. Husen (2009). Automatic S-wave picker for local earthquake tomography, *Bull. Seism. Soc. Am.*, **99**(3) 1906–1920, doi:10.1785/0120080019.
- Earle, P. S. and P. M. Shearer (1994). Characterization of global seismograms using an automatic-picking algorithm, *Bull. Seism. Soc. Am.*, **84**(2) 366–376.
- Enescu, N. (1996). Seismic data processing using nonlinear prediction and neural networks, in H. U. of Technology, editor, *IEEE Nordic Signal Processing Symposium*, pages 1–4, Espoo, Finland.
- Florido, E., J. L. Aznarte, A. Morales-Esteban, and F. Martínez-Álvarez (2016). Earthquake magnitude prediction based on artificial neural networks: A survey, *Croatian Operational Research Review (CRORS)*, **7**(2) 159–169, doi:10.17535/corr.2016.0011.
- FUNVISIS (2018). Boletín Sismológico de Venezuela, **1–4**(35).
- FUNVISIS (2019). Boletín Sismológico de Venezuela, **1**(36).
- Gentili, S. and A. Michelini (2006). Automatic picking of P and S phases using a neural tree, *J. Seismol.*, **10** 39–63, doi:10.1007/s10950-006-2296-6.
- Gibbons, S. J. and F. Ringdal (2006). The detection of low magnitude seismic events using array-based waveform correlation, *Geophys. J. Int.*, **165**(1) 149–166, doi:10.1111/j.1365-246X.2006.02865.x.
- Harris, D. B. and D. A. Dodge (2011). An autonomous system for grouping events in a developing aftershock sequence, *Bull. Seism. Soc. Am.*, **101**(2) 763–774, doi:10.1785/0120100103.
- Ibrahim, M. A., J. Park, and N. Athens (2018). Earthquake warning system: Detecting earthquake precursor signals using deep neural networks, Technical Report CS 230, School of Earth, Energy and Environmental Sciences, Stanford University.
- Joswig, M. (1990). Pattern recognition for earthquake detection, *Bull. Seism. Soc. Am.*, **80**(1) 170–186.
- Klumpen, E. and M. Joswig (1993). Automated reevaluation of local earthquake data by application of generic polarization patterns for P- and S-onsets, *Comput. Geosci.*, **19**(2) 223–231, doi:10.1016/0098-3004(93)90120-T.
- Kong, Q., D. T. Trugman, Z. E. Ross, M. J. Bianco, B. J. Meade, and P. Gerstoft (2019). Machine learning in seismology: Turning data into insights, *Seismol. Res. Lett.*, **90**(1) 3–14, doi:https://doi.org/10.1785/0220180259.
- Kriegerowski, M., G. M. Petersen, H. Vasyura-Bathke, and M. Ohrnberger (2019). A deep convolutional neural network for localization of clustered earthquakes based on multistation full waveforms, *Seismol. Res. Lett.*, **90**(2A) 510–516, doi:10.1785/0220180320.

- MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations, in L. M. L. Cam and J. Neyman, editors, *Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297, University of California Press, Berkeley, California.
- Mignan, A. and J. Woessner (2012). Estimating the magnitude of completeness for earthquake catalogs, *Community Online Resource for Statistical Seismicity Analysis*, pages 1–45, doi:10.5078/corssa-00180805.
- Nippres, S. E. J., A. Rietbrock, and A. E. Heath (2010). Optimized automatic pickers: Application to the ANCORP data set, *Geophys. J. Int.*, **181**(2) 911–925, doi:10.1111/j.1365-246X.2010.04531.x.
- Perol, T., M. Gharbi, and M. Denolle (2018). Convolutional neural network for earthquake detection and location, *Sci. Adv.*, **4**(2), doi:10.1126/sciadv.1700578.
- Roberts, R. G., A. Christoffersson, and F. Cassidy (1989). Real-time event detection, phase identification and source location estimation using single station three-component seismic data, *Geophys. J. Int.*, **97**(3) 471–480, doi:10.1111/j.1365-246X.1989.tb00517.x.
- Rojas, O., B. Otero, L. Alvarado, S. Mus, and R. Tous (2019). Artificial neural networks as emerging tools for earthquake detection, *Computación y Sistemas*, **23**(2) 335–350, doi:10.13053/CyS-23-2-3197.
- Ross, Z. E. and Y. Ben-Zion (2014). Automatic picking of direct P, S seismic phases and fault zone head waves, *Geophys. J. Int.*, **199**(1) 368–381, doi:10.1093/gji/ggu267.
- Ross, Z. E., M.-A. Meier, E. Hauksson, and T. H. Heaton (2018). Generalized seismic phase detection with deep learning, *Bull. Seism. Soc. Am.*, **108**(5A) 2894–2901, doi:10.1785/0120180080.
- Ross, Z. E., D. T. Trugman, E. Hauksson, and P. M. Shearer (2019). Searching for hidden earthquakes in Southern California, *Science*, **364**(6442) 767–771, doi:10.1126/science.aaw6888.
- Rydelek, P. A. and I. S. Sacks (1989). Testing the completeness of earthquake catalogues and the hypothesis of self-similarity, *Nature*, **337** 251–253, doi:10.1038/337251a0.
- Shelly, D. R., G. C. Beroza, and S. Ide (2006). Low frequency earthquake swarms and non-volcanic tremor under Shikoku, Japan, in *AGU Fall Meeting Abstracts*, volume 1, pages T41A–1546, American Geophysical Union.
- Singer, A., F. Nevado, A. Gómez, and L. M. Rodríguez (2014). Evidencias de actividad tectónica en la falla de Las Trincheras, VI Coloquio de Microzonificación Sísmica, Valencia, Venezuela.
- Skoumal, R. J., M. R. Brudzinski, B. S. Currie, and J. Levy (2015). Optimizing multi-station earthquake template matching through re-examination of the Youngstown, Ohio, sequence, *Earth Planet. Sci. Lett.*, **405** 274–280, doi:10.1016/j.epsl.2014.08.033.

- Tiira, T. (1999). Detecting teleseismic events using artificial neural networks, *Comput. Geosci.*, **25**(8) 929–938, doi:10.1016/S0098-3004(99)00056-4.
- Wang, J. and T.-L. Teng (1995). Artificial neural network-based seismic detector, *Bull. Seism. Soc. Am.*, **85**(1) 308–319.
- Wang, J. and T.-L. Teng (1997). Identification and picking of S phase using an artificial neural network, *Bull. Seism. Soc. Am.*, **87**(5) 1140–1149.
- Wiemer, S. and M. Wyss (2002). Mapping spatial variability of the frequency-magnitude distribution of earthquakes, *Adv. Geophys.*, **45** 259–302, doi:10.1016/S0065-2687(02)80007-3.
- Wiszniowski, J., B. M. Plesiewicz, and J. Trojanowski (2014). Application of real time recurrent neural network for detection of small natural earthquakes in Poland, *Acta Geophysica*, **62**(3) 469–485, doi:10.2478/s11600-013-0140-2.
- Yoon, C. E., O. O’Reilly, K. Bergen, and G. C. Beroza (2015). Earthquake detection through computationally efficient similarity search, *Sci. Adv.*, **1**(11) 1–13, doi:10.1126/sciadv.1501057.
- Zhang, X., J. Zhang, C. Yuan, S. Liu, Z. Chen, and W. Li (2018). Locating earthquakes with a network of seismic stations via a deep learning method, *CoRR*, **abs/1808.09603** 1–14, doi:10.1038/s41598-020-58908-5.
- Zhao, Y. and K. Takano (1999). An artificial neural network approach for broadband seismic phase picking, *Bull. Seism. Soc. Am.*, **89**(3) 670–680.
- Zhu, W. and G. C. Beroza (2019). PhaseNet: A deep-neural-network-based seismic arrival time picking method, *Geophys. J. Int.*, **216**(1) 261–273, doi:10.1093/gji/ggy423.

Table 1: Impact of different hyperparameters.

| Overlapping | Filtering [0.5, 10] Hz | Components | Accuracy |
|-------------|------------------------|------------|---------------|
| yes | yes | 3 | 95.27% |
| no | yes | 3 | 89.9% |
| yes | no | 3 | 92.48% |
| yes | yes | 1 | 94.11% |

Impact of different hyperparameters on the detection accuracy of the proposed method on the CARABOBO dataset.

Table 2: Final P-wave detection results.

| Preprocessing | Model | Win. size | Accuracy |
|---------------|--------------|-----------|---------------|
| ConvNetQuake | ConvNetQuake | 10 | 81.95%* |
| ConvNetQuake | ConvNetQuake | 50 | 83.25%* |
| UPC-UCV | ConvNetQuake | 10 | 89.63% |
| UPC-UCV | ConvNetQuake | 50 | 90.10% |
| UPC-UCV | UPC-UCV | 10 | 89.96% |
| UPC-UCV | UPC-UCV | 50 | 95.27% |

Detection accuracy results for both the reference model (ConvNetQuake) and the proposed model (UPC-UCV) on the CARABOBO dataset with two different window sizes (10 sec and 50 sec). *Accuracy values for the ConvNetQuake preprocessing are *balanced accuracies*, as the dataset is imbalanced.

Table 3: Source region estimation accuracy results.

| Model | Accuracy (K=3) | Accuracy (K=5) |
|--------------|----------------|----------------|
| ConvNetQuake | 84.58% | 82.08% |
| UPC-UCV | 95.68% | 93.36% |

Source region estimation accuracy results for both the reference model (ConvNetQuake) and the proposed model (UPC-UCV) on the CARABOBO dataset.

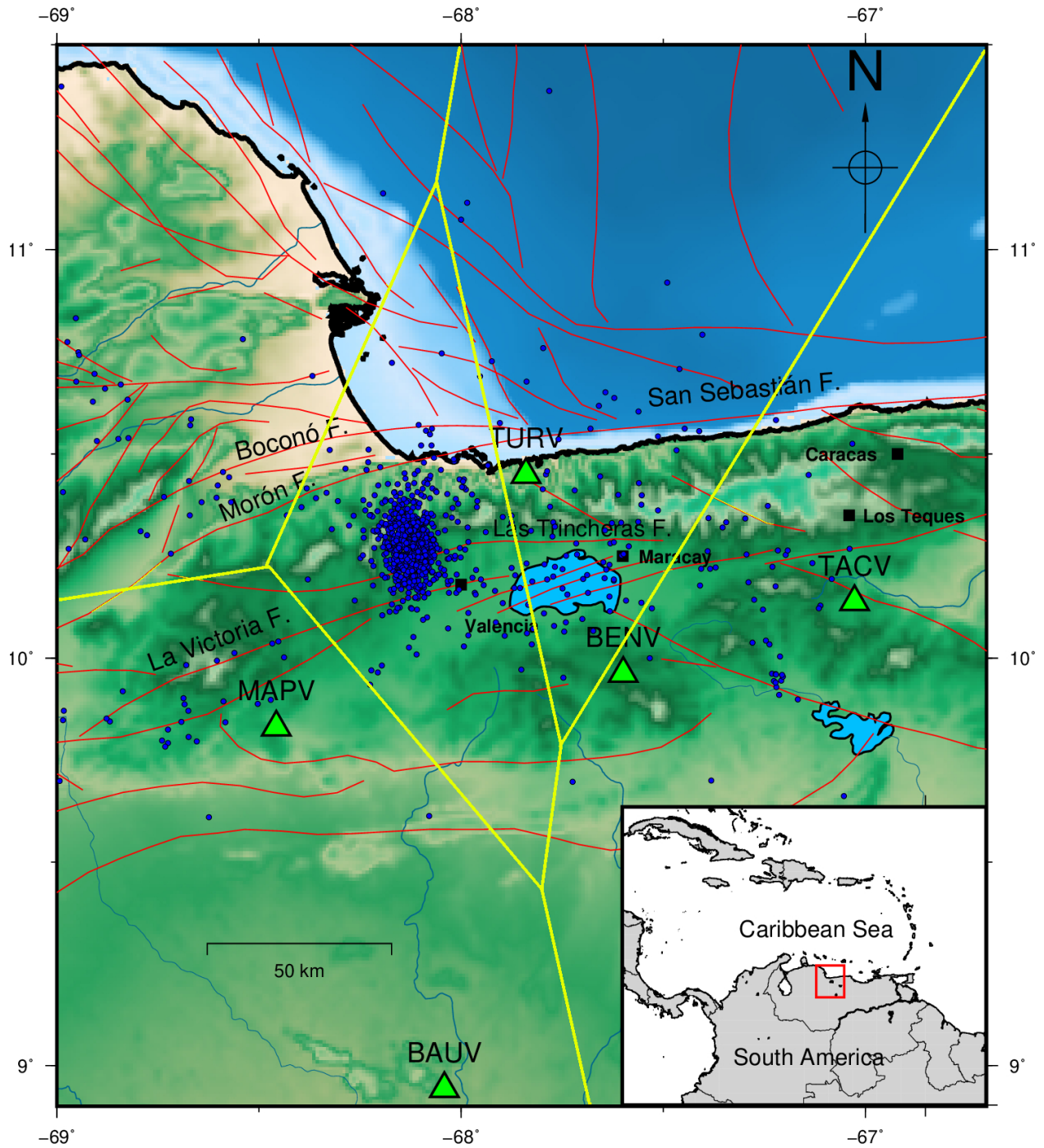


Figure 1: Epicenters of seismic activity in the center-north region ($67^{\circ}\text{W} - 69^{\circ}\text{W}$, $9.5^{\circ}\text{N} - 11.5^{\circ}\text{N}$) of Venezuela (circles), seismological stations (triangles), and active faults (red lines) compiled by Beltrán (1993); inset: relative location of the study area at the Caribbean-South American plate interaction. The figure also shows the geographical partitioning (obtained with k-means and $K=5$) of the events (yellow lines).

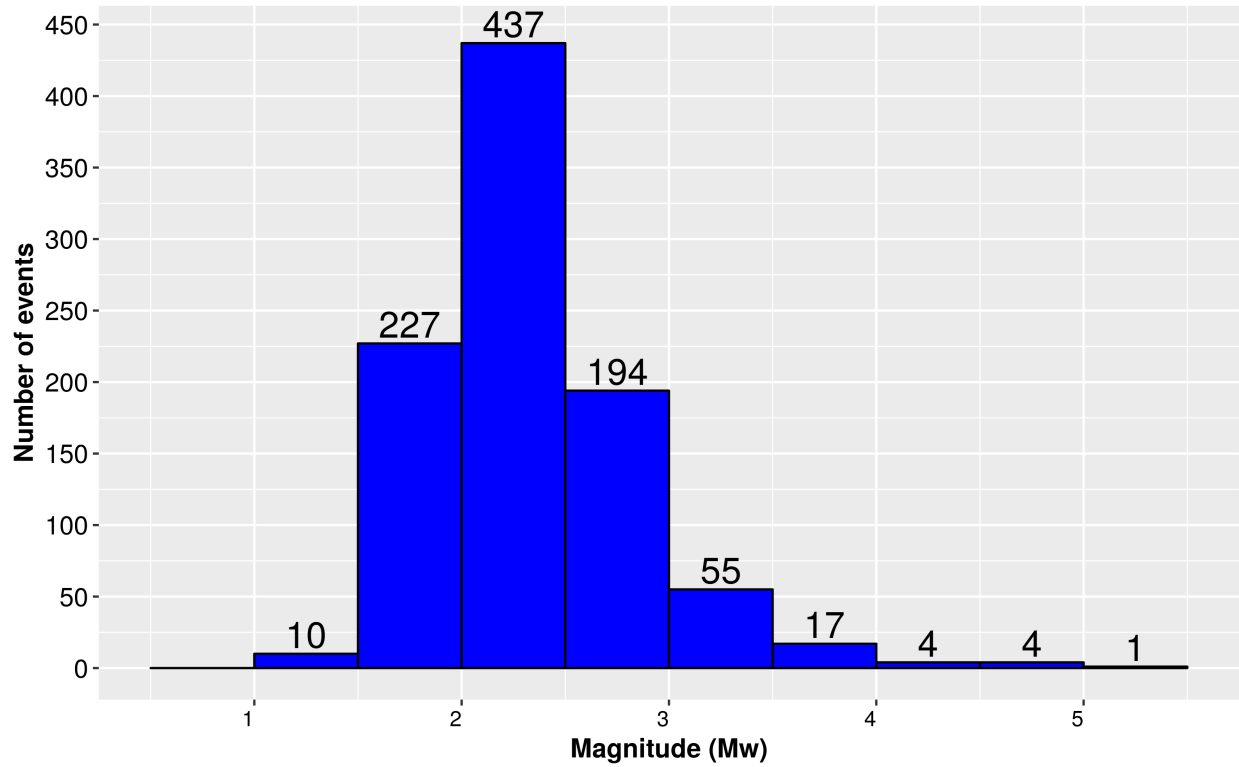


Figure 2: Magnitude distribution of the events.

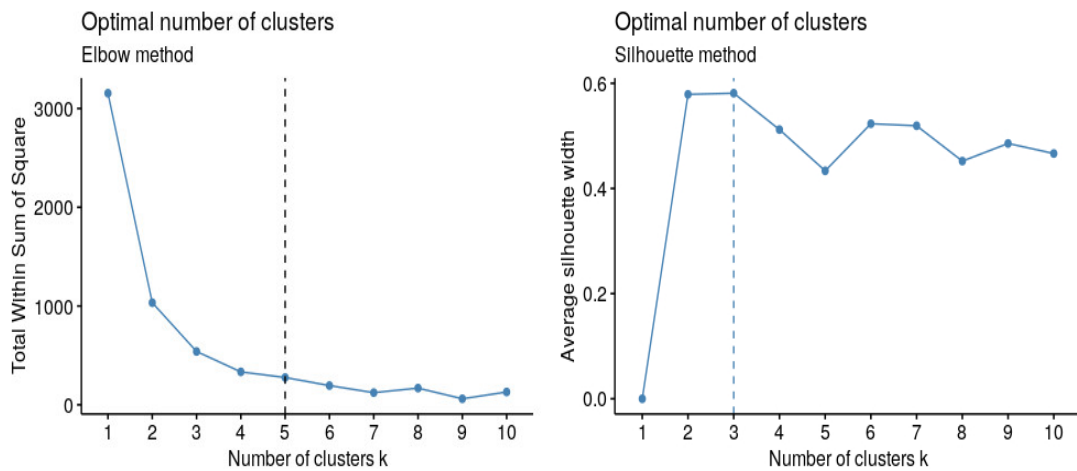


Figure 3: Results of the methods for finding the appropriate number of source regions (clusters). Elbow method (left), and Silhouette method (right).

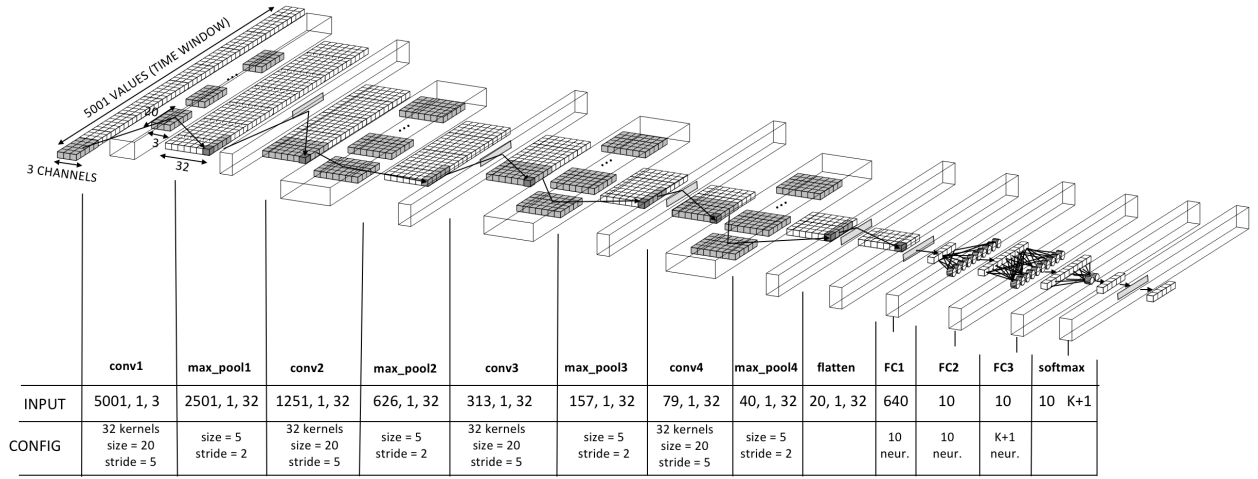


Figure 4: Network architecture with 3-channel input, 4 convolutional+max pooling layers, 3 fully connected layers and a softmax layer.

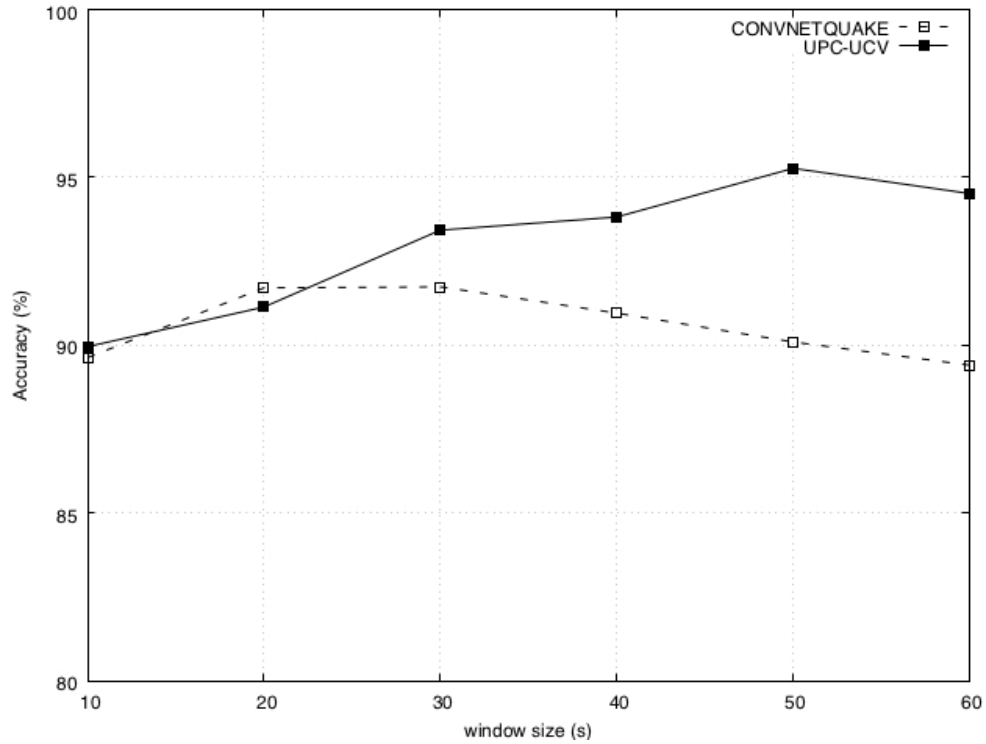


Figure 5: Impact of window size on accuracy for both the reference model (ConvNetQuake) and the proposed model (UPC-UCV). Models are trained with the CARABOBO dataset.

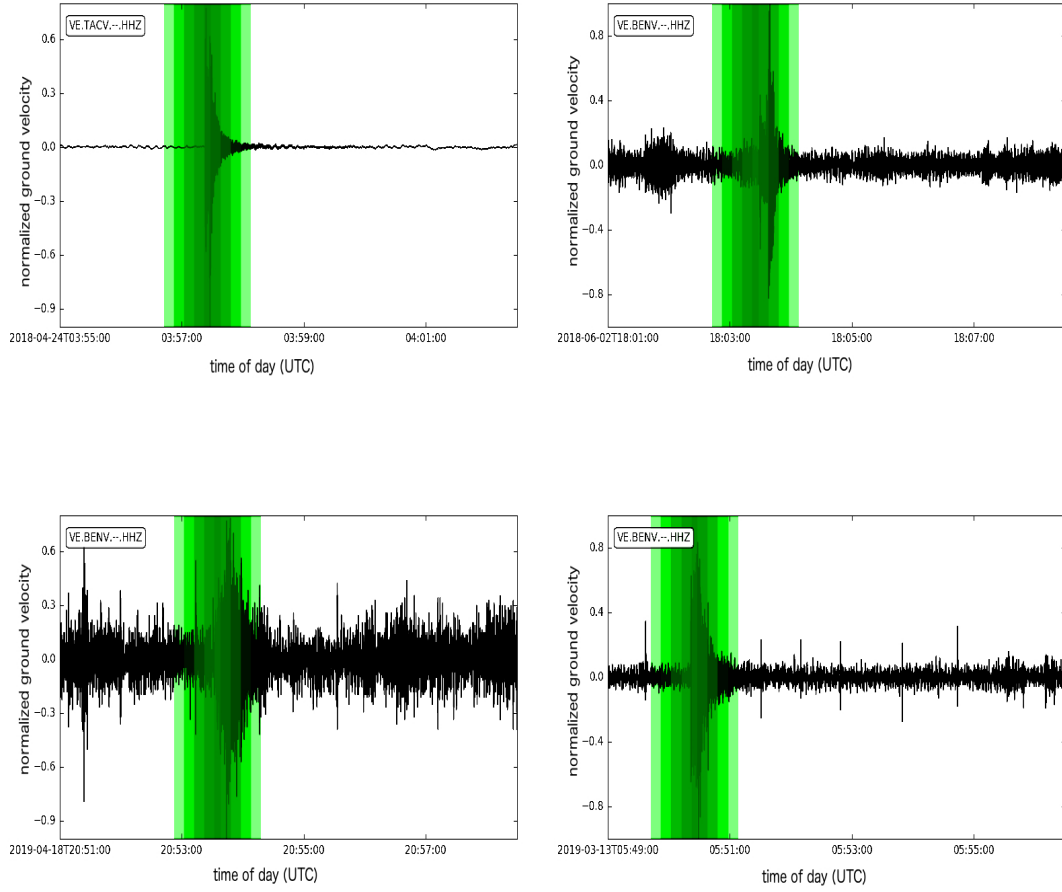


Figure 6: Example seismograms (only Z component for readability) with P-waves successfully detected (true-positive windows are overlaid in green).

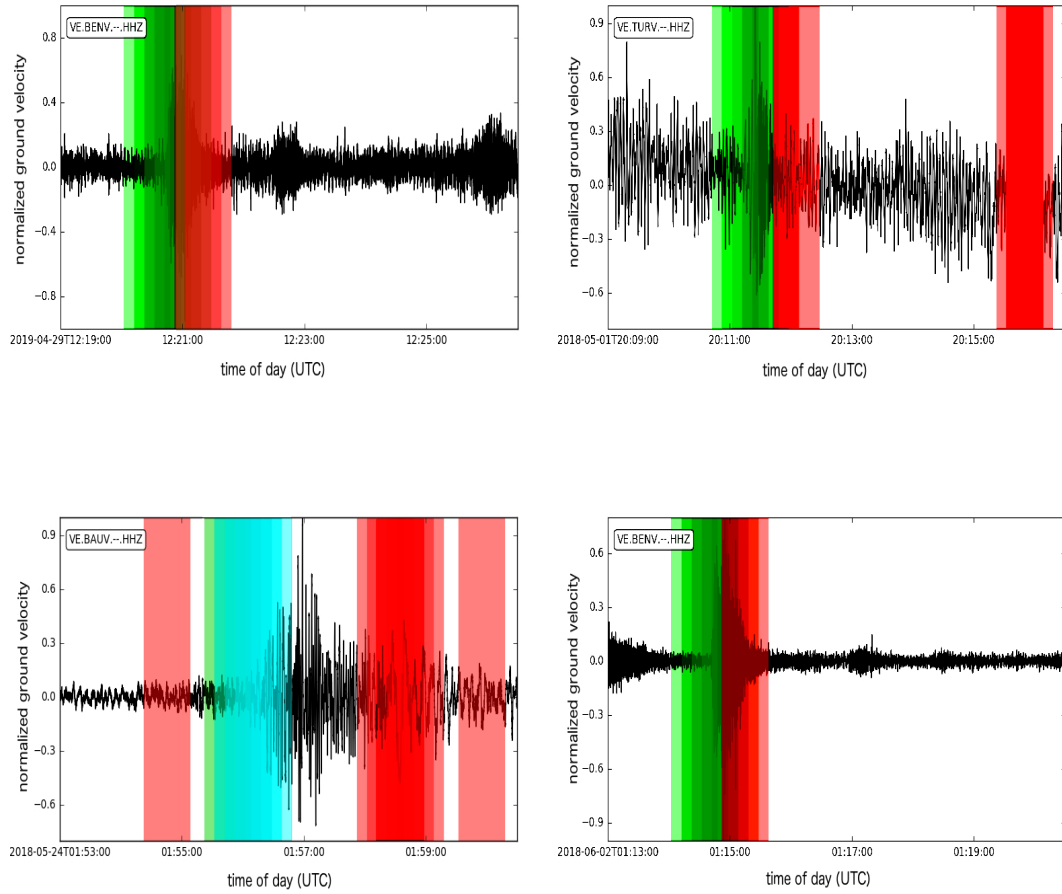


Figure 7: Example seismograms (only Z component for readability) with detection errors (false positive in red and false negative in cyan).