Fully Convolutional Networks for Local Earthquake Detection

Youness Choubik[1]
LIMIARF Laboratory
Faculty of Sciences
Mohammed V University
in Rabat, Morocco

Abdelhak Mahmoudi[2]
Ecole Normale Supérieure
Mohammed V University
in Rabat, Morocco

Mohammed Majid Himmi[3]
LIMIARF Laboratory
Faculty of Sciences
Mohammed V University
in Rabat, Morocco

Abstract—**Automatic earthquake detection is widely studied to replace manual detection, however, most of the existing methods are sensitive to seismic noise. Hence, the need for Machine and Deep Learning has become more and more significant. Regardless of successful applications of the Fully Convolutional Networks (FCN) in many different fields, to the best of our knowledge, they are not yet applied in earthquake detection. In this paper, we propose an automatic earthquake detection model based on FCN classifier. We used a balanced subset of STanford EArthquake Dataset (STEAD) to train and validate our classifier. Each sample from the subset is re-sampled from 100Hz to 50Hz then normalized. We investigated different, widely used, feature normalization methods, which consist of normalizing all features in the same range, and we showed that feature normalization is not suitable for our data. On the contrary, sample normalization, which consists of normalizing each sample of our dataset individually, improved the accuracy of our classifier by ~16% compared to using raw data. Our classifier exceeded 99% on training data, compared to ~83% when using raw data. To test the efficiency of our classifier, we applied it to real continuous seismic data from XB Network from Morocco and compared the results to our catalog containing 77 earthquakes. Our results show that we could detect 75 out of 77 earthquakes contained in the catalog.**

Keywords—*Earthquake detection; fully convolutional networks; data normalization; classification*

I. INTRODUCTION

Earthquake detection requires discriminating real earthquakes from noise signals, which makes it a classification problem. Earthquake detection is a very crucial and challenging phase in seismic processing, especially for single station-based detection, because every station records a very wide range of non-earthquake waveforms. Manual detection is a time consuming work due to the huge amount of seismic data, therefore, automatic earthquake detection is essential and widely studied.

A large number of automatic earthquake detection methods exist [35], some of them are time domain methods, such as the short term average to long term average (STA/LTA), which is the most used in seismic stations. Other time domain methods are used, such as the maximum likelihood detector [9], envelope-based detector [3], and modified data envelope detector [29]. On the other hand, some frequency domain methods are based on the Power Spectral Density (PSD) [25] and the Walsh transform [12]. However, most of the existing

methods are sensitive to noise and suffer from false and missed detections [32].

In recent years, methods based on Machine and Deep Learning have shown great potentials, especially Artificial Neural Networks, which are widely used in seismic detection [8], [10], [1]. The Convolutional Neural Networks (CNNs) known as very successful in the computer vision area become more and more popular in seismic area [23], [39], [37], [34]. Recurrent Neural Networks (RNNs), another architecture of Neural Networks known to be suitable for many time-series applications such as text to speech and voice recognition [36], are also used in seismology [40], [19], [6].

Unsupervised clustering methods are also used in seismology. They can cluster seismic samples into different clusters without prior knowledge of labels. Different clustering methods are used in many seismic studies, such as k-means [5], [28], Deep Convolutional Autoencoders [21], and Self-Organizing Maps [16], [17], [27].

Fully Convolutional Networks (FCN) [24] are a Neural Network architectures that have been successfully applied in many different fields, such as image segmentation[13], [4], medical image analysis [7], [18], character recognition [31], time-series classification [33], [14], [22] and also in seismology; for earthquake localization, by taking a window of three-component waveform data from multiple stations and predicting the earthquake location with a 3D image [38], and for fault detection, where the FCN model extracts fault features from synthetic seismic data and recognize the locations of faults with an accuracy of ~97% [26]. Despite their higher achievements, to the best of our knowledge, FCN had not yet been applied to seismic detection.

In this study, we describe the application of FCN for earthquake detection using seismic waveforms from a single seismic station. The basic of the earthquake detection problem is turned into a classification problem by using a subset of STanford EArthquake Dataset (STEAD) to train our classifier. Our approach does not require a feature extraction technique, which makes it independent of the choice of sensitive features. We tested the effectiveness of our classifier by applying it to real continuous data From the XB seismic network implemented in morocco between 2009 and 2013 [2].

In the following, we first describe the dataset used to train our classifier and the real continuous dataset used for testing. Then, we present the method and the steps applied in the
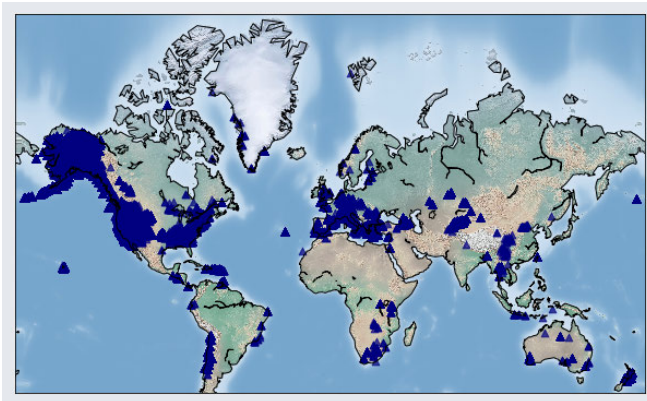
(a) Locations of seismic stations recording earthquakes
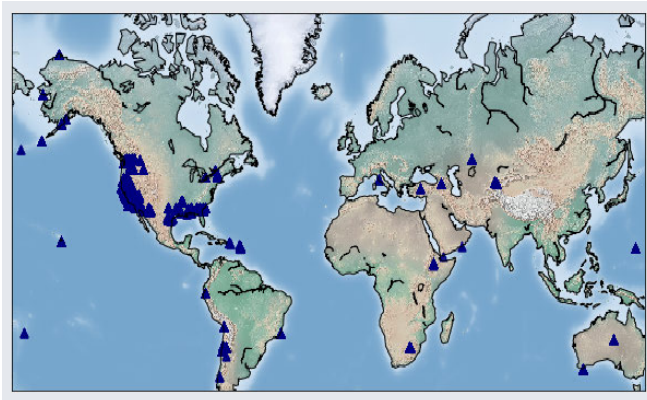
(b) Locations of seismic stations recording seismic noise

Fig. 1. Distribution of Seismic Stations used to Record Earthquakes (a) and Noise (b) [20].
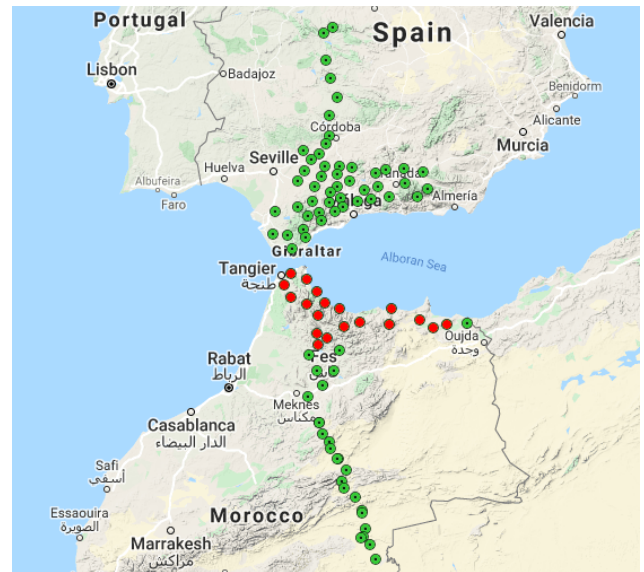

Fig. 2. XB Seismic Network Installed in both Morocco and Spain, it Contains 93 Seismic Stations, from which 19 Stations are used in this Study and are Colored in Red.

training process. Finally, we describe the results and discuss the performance of our classifier Using real continuous seismic data.

II. DATA DESCRIPTION

The STanford EArthquake Dataset (STEAD) [20] is a large-scale and global dataset that contains two waveform classes; seismic noise and local earthquake waveforms, which are recorded at local distances (within 350 km of earthquakes). STEAD comprises about 1.2 million waveforms, recorded by worldwide located seismometers, resampled at 100Hz, and have 60 seconds duration (6000 features). Local-earthquakes class contains about 1 050 000 three-component seismograms associated with ∼450 000 earthquakes that occurred between January 1984 and August 2018 (Fig. 1(a)). The seismic noise class contains about 100 000 waveforms that have been recorded since 2000 in the United States and Europe (Fig. 1(b)).

The earthquake waveforms are requested from continuous time-series archived at the Incorporated Research Institutions for Seismology Data Management Center (IRIS DMC). Three types of arrival statuses exist, "Manual" picks; picked manually by human analysts, "automatic" picks; measured by automatic algorithms and "autopicker" that are determined using an AI-

based model. STEAD is provided as individual arrays containing three waveforms that correspond to three-component seismograms, each waveform has 6000 features.

The XB network used to test our model was deployed in both Morocco and Spain, in the frame of the Project to Investigate Convective Alboran Sea System Overturn (PICASSO), from 2009 to 2013. The XB network contained 93 seismic stations labeled as PICASSO Morocco (PM) and PICASSO Spain (PS). Fig. 2 shows the stations of the XB network, where 44 stations were installed in Morocco. We used data of January 2011 from 19 stations, measured by High gain Broadband (BH) seismic instruments and sampled at 50Hz, to test our model.

III. TRAINING WITH THE FULLY CONVOLUTIONAL NETWORKS

To train our model, we choose a subset of STEAD measured by BH seismic instruments, since we have only BH waveforms from XB network. We found 7874 unique noise waveforms of BH type in STEAD. In order to create a balanced dataset, we extracted the same quantity of waveforms from the earthquake class. because classification is affected by imbalanced datasets and resulting a reduction in accuracy as shown by [30]. The selected waveforms are associated with a wide range of earthquake sizes from magnitude 0 to magnitude 6.3. Earthquakes were recorded within 330 km of the earthquakes, are mainly shallower than 210 km and have Signal to Noise Ratio (SNR) between -5 and 100 decibels.

Our dataset is comprised of 15 748 samples and divided into train/validation/test subsets as shown in Table I. The portion of the test-set is small because we will test our model on real continuous data from the XB seismic network. The

TABLE I. TRAIN/VALIDATION/TEST SUBSETS DISTRIBUTION USED IN OUR STUDY

Training-set	Validation-set	Test-set
12000	3000	748
(6000 from earthquake class, 6000 from noise class)	(1500 from earthquake class, 1500 from noise class)	(374 from earthquake class, 374 from noise class)

samples used in STEAD are 60 seconds waveforms sampled at 100Hz. Since we are applying our model to data from XB network that is sampled at 50Hz, we resampled our dataset to 50Hz, so that every sample have 3000 features instead of 6000.

The Fully Convolutional Network classifier used in this study is comprised of four convolutional layers with different filter numbers and sizes (Fig. 3), followed by batch normalization that normalizes the output of the convolution layer and a ReLU activation function, which enables better training of deeper networks, compared to other activation functions [11], then a Global Pooling layer that reduces the amount of parameters in the network to an output prediction for the model. Finally, since the output is One Hot Encoded, a softmax function is placed in the output layer that normalizes the output into two probabilities corresponding to belonging to the two classes earthquake and noise. The adaptive moment estimation algorithm (Adam) is used as optimizer for our classifier.

The classifier is trained to distinguish between earthquake and noise signals using the STEAD subset described above. The training/validation subsets were randomly split using a 5-fold cross-validation. The training was performed on 100 epochs, where each epoch is a complete pass through the entire training dataset, with early stopping enabled, which stop the training when the loss (error) does not decrease during training. We used a learning rate decay, where the learning rate is reduced by a factor of 10 once learning stagnates for a number of epochs. The predictions were compared to the real classes then the loss and accuracy are calculated.

Normalization is one of the most used data preparation techniques in deep learning, because features often have different ranges of values, which make the training process takes a long time to converge. Feature normalization and standardization are the most used methods. To select the best normalization method, we compared different methods against raw data (without applying any normalization method) and selected the one that gave the best accuracy on the training/validation data. The methods applied to our input data are the following:

- MinMax: Transform features by scaling each feature individually between zero and one.

- MaxAbs: Scale each feature by its maximum absolute value such that the maximal absolute value of each feature in the training set will be 1.0.

- Standard: Standardize features by removing the mean and scaling to unit variance.

- RobustScaler: Removes the median and scales the data according to the quantile range independently on each feature.

IV. RESULTS AND DISCUSSIONS

In this section, we will present and discuss the effect of using different normalization methods and batch sizes on the classifier accuracy. We investigated the effect of normalizing features on the model accuracy and compared it against using raw data. We conducted many experiments using the same training process for the normalization methods described above. Table II shows the mean accuracy of 5-fold cross-validation. We can see that MinMax, MaxAbs and Robust normalizations decrease the model accuracy compared to raw data, while standardization improves slightly the accuracy. Overall, we see that normalizing the input features did not bring a big improvement to our model, so we suspected the feature normalization to be not suitable for our data. Therefore, we tried to normalize our data per sample instead of normalizing per feature. By normalizing per sample, we mean that each sample of our dataset is normalized individually. We reported the results in Table III.

By using sample-normalized data, we can clearly see an improvement of the accuracy compared to feature-normalized and raw data. All the methods improved the accuracy without exception, compared to feature-normalized, especially the MinMax method, which is improved by ∼35%. The sample-standardization method made the best accuracy over the other methods, it reaches 99% on the training data, with an improvement of ∼16% and ∼14% compared to raw data feature-standardization respectively.

As seen in Fig. 4(b), when standardizing per feature, the range of earthquake classes is very large compared to that of noise classes, which makes no difference with raw data (Fig. 4(a)), except for the scale of the signals. It can be observed from Fig. 4(c) that both earthquake and noise samples have close ranges when standardized per sample. Hence, the classifier is forced to classify samples based on their shape instead of their amplitude. In the rest of our tests, only the sample-standardization method will be presented, since it outperformed the other methods.

Different batch sizes are investigated, where each batch is a subset of signals given to the network at once. Fig. 5 shows an example of the evolution of the loss function during the training process and it is clear that our classifiers converge as the training progresses. We can see that for larger batch sizes, the training loss is bigger and the validation loss is smoother, because large batch sizes are less sensitive to outliers, and converge slower than small batch sizes as stated by other studies [15].

Fig. 6(a) shows the accuracy during the training process, we can clearly see that larger batch sizes have lower accuracies compared to smaller batch sizes. While for the validation dataset (Fig. 6(b)), larger batch sizes tend to be slower and
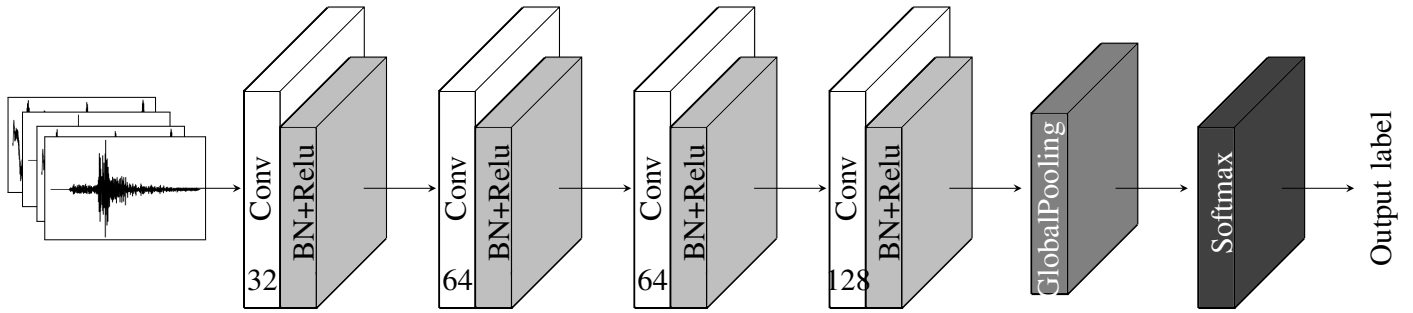
Fig. 3. The FCN Architecture Consists of 4 Convolutional Layers Followed by Batch-Normalization (BN) then a Rectified Linear Units (ReLU) Activation Function, Finally Global-Pooling and Softmax Layers.

TABLE II. THE ACCURACY FOR DIFFERENT FEATURE NORMALIZATION METHODS

	Raw data	Standard	MinMax	MaxAbs	Robust
Training	0.838	0.858	0.639	0.822	0.778
Validation	0.785	0.885	0.619	0.683	0.718
Test	0.611	0.894	0.502	0.513	0.657

TABLE III. THE ACCURACIES FOR DIFFERENT PER-SAMPLE NORMALIZATION METHODS

	Raw data	Standard	MinMax	MaxAbs	Robust
Training	0.838	0.990	0.986	0.985	0.982
Validation	0.785	0.987	0.985	0.986	0.916
Test	0.611	0.998	0.995	0.993	0.973

(a) Raw signal

(b) Normalize per feature
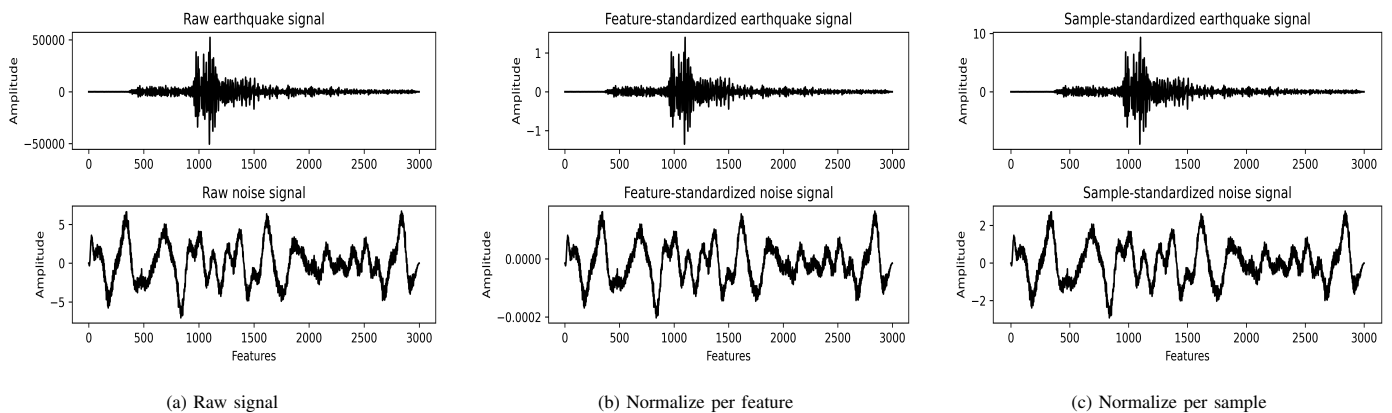
(c) Normalize per sample

Fig. 4. Standardization Effect for the Same Two Signals, in Top an Earthquake Signal while in Bottom a Noise One. Figure (a) Shows Raw Signals, in (b) Both Signals are Standardized Per Feature, (c) Shows the Sample-Standardization Version for Both Signals.
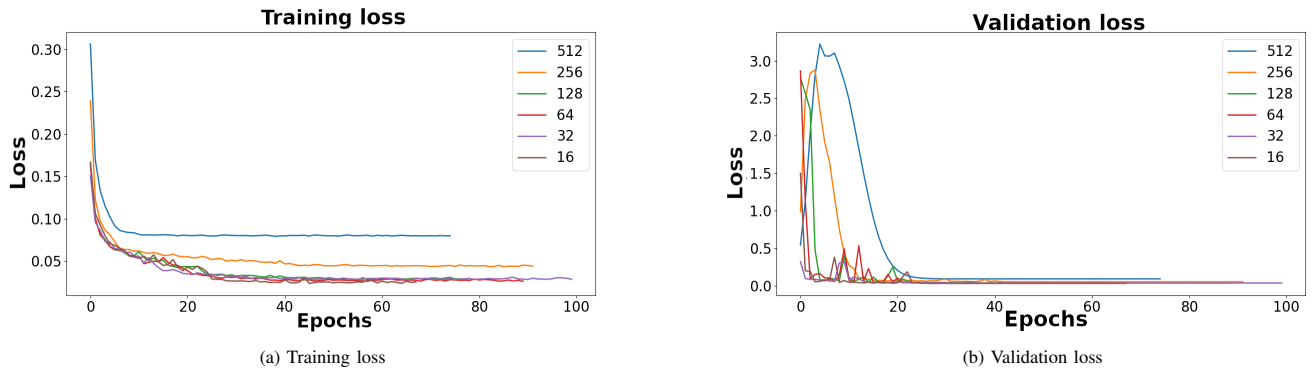
(a) Training loss

(b) Validation loss

Fig. 5. Model Loss Per Batch Size.

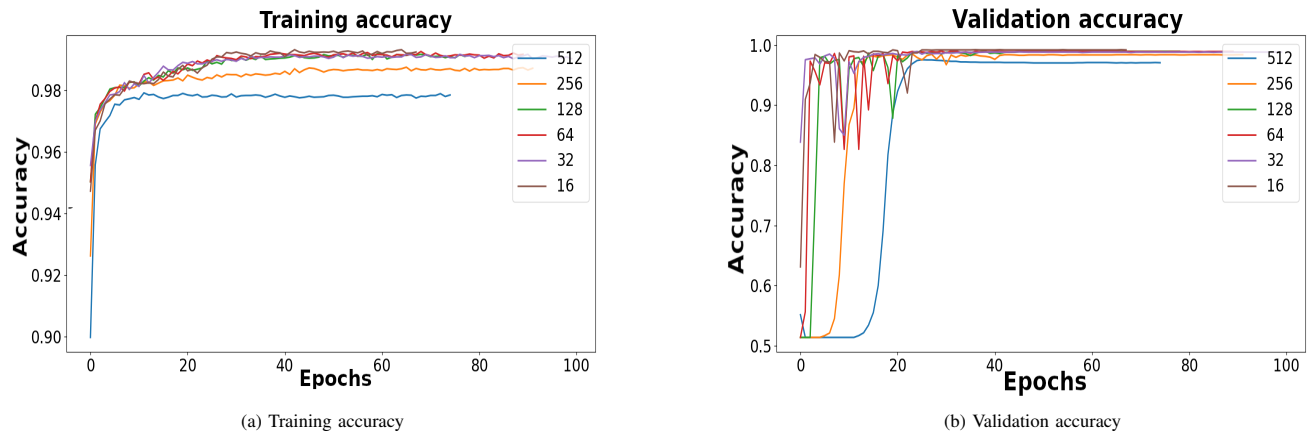(a) Training accuracy

(b) Validation accuracy

Fig. 6. Training and Validation Subsets Accuracies Per Batch Size. Large Batches Tend to have Lower Accuracies for Both Subsets.
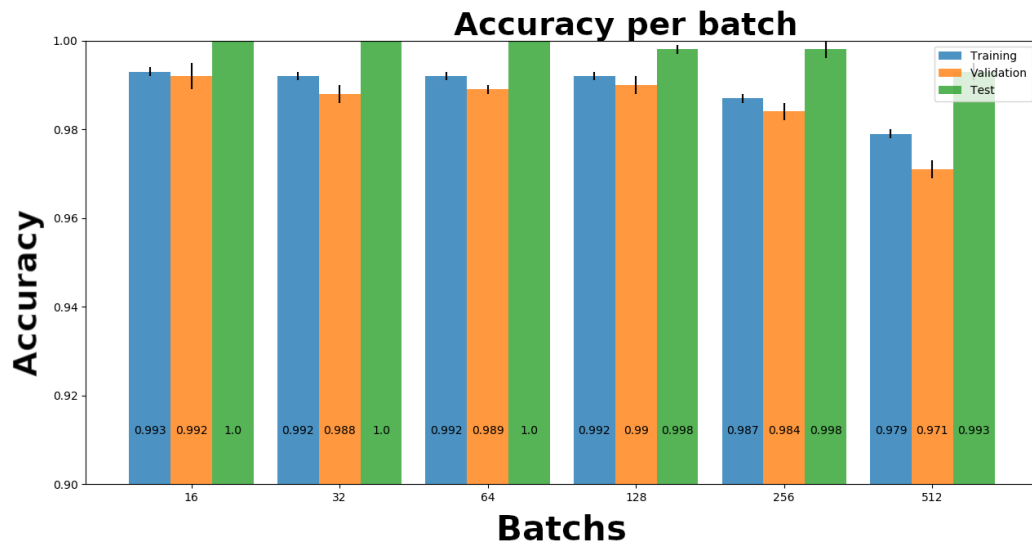

Fig. 7. The Mean Accuracy Per Batch Size. The Best Accuracies in Train/Validation/Test Subsets are Obtained by using a 16 Batch Size.

more stable. Fig. 7 shows the mean accuracy of 5-fold cross-validation. The best accuracies in training, validation and test are 99.3%, 99.2% and 100% respectively, obtained by using a 16 batch size. For smaller batches, the accuracy in training reached 100%, but in validation it has fallen to 70%, which means that the model over-fit and can not generalize for new data. The high accuracy in test-set is due to the small amount of data, because we are interested in testing our classifier on continuous data from XB network.

To check the effectiveness of our best classifier, we tested it on real three-component seismic data from the XB network. The test was applied to data from the first month of 2011, from 19 seismic stations, presented in red in fig. 2. The frequency of the seismic data is about 50Hz, and the input feature, which will be fed to the classifier, is a sliding window of 60 seconds length (3000 features), and the window is moved by 15 sec after each test.

To verify our results, we compared the earthquakes detected by our classifier to a seismic catalog that we have. Our catalog contains 77 earthquakes of magnitude > 2, located in the region of XB network. By comparing our results with the catalog, we found that our classifier detected 75 out of the 77 earthquakes contained in the catalog. Our analysis shows that our classifier is able to reliably detect local earthquake signals in continuous real data.

V. Conclusion

In this paper, we have presented a seismic detection model, based on a Fully Convolutional Networks classifier which is trained on STanford EArthquake Dataset (STEAD) and tested on real continuous seismic data. By making a separate standardization for each sample of our dataset, instead of normalizing per feature, the performance of our classifier is increased significantly by ∼16% compared to raw data. Our experiments show that the use of small batch sizes is more adequate for our dataset, however, very small batch sizes (8 and lower) make the model over-fit and can not generalize for new data. By applying our classifier to real continuous data from XB network in Morocco, we were able to detect local earthquakes already existing in our catalog. Our method does not require hand-engineered features and is able to discriminate between earthquakes and seismic noise with high accuracy. Our results demonstrated that FCN classifier holds vast promise for making seismic detection more accurate.

References

[1] Jubran Akram, Oleg Ovcharenko, and Daniel Peter. A robust neural network-based approach for microseismic event detection. In *SEG Technical Program Expanded Abstracts 2017*, pages 2929–2933, 08 2017.

[2] Levander Alan and Humphreys Gene. Program to investigate convective alboran sea system overturn, 2009.

[3] Rex Allen. Automatic earthquake recognition and timing from single trace. *Bulletin of the Seismological Society of America*, 68:1521–1532, 10 1978.

[4] Lei Bi, Jinman Kim, Euijoon Ahn, Ashnil Kumar, David Dagan Feng Feng, and Michael Fulham. Step-wise integration of deep class-specific learning for dermoscopic image segmentation. *Pattern Recognition*, 85:78–89, 01 2019.

[5] Yangkang Chen, Guoyin Zhang, Min Bai, Shaohuan Zu, Zhe Guan, and Mi Zhang. Automatic waveform classification and arrival picking based on convolutional neural network. *Earth and Space Science*, 6, 04 2019.

[6] Jana Doubravová, Jan Wiszniowski, and Josef Horalek. Single layer recurrent neural network for detection of swarm-like earthquakes in w-bohemia/vogtland - the method. *Computers & Geosciences*, 93, 05 2016.

[7] Jingfan Fan, Xiaohuan Cao, and Pew-Thian Yap. Birnet: Brain image registration using dual-supervised fully convolutional networks. *Medical Image Analysis*, 54, 02 2018.

[8] Luigi Fortuna, Salvatore Graziani, M. Presti, and Giuseppe Nunnari. A neural network for seismic events classification. In *[Proceedings] IGARSS'91 Remote Sensing: Global Monitoring for Earth Management*, pages 1663 – 1666, 07 1991.

[9] Walter Freiberger. An approximate method in signal detection. *Quarterly of Applied Mathematics*, 20, 01 1963.

[10] Flora Giudicepietro, Anna Esposito, and Patrizia Ricciolino. Fast discrimination of local earthquakes using a neural approach. *Seismological Research Letters*, 88:1089–1096, 07 2017.

[11] Xavier Glorot, Antoine Bordes, and Y. Bengio. Deep sparse rectifier neural networks. *Proceedings of the 14th International Conference on Artificial Intelligence and Statisitics (AISTATS) 2011*, 15:315–323, 01 2011.

[12] Tom Goforth. An automatic signal detection algorithm based on the walsh transform. *Bulletin of the Seismological Society of America*, 71:1351, 01 1981.

[13] Yuan Huang, Fugen Zhou, and Jerome Gilles. Empirical curvelet based fully convolutional network for supervised texture image segmentation. *Neurocomputing*, 349, 04 2019.

[14] Fazle Karim, Somshubra Majumdar, Houshang Darabi, and Shun Chen. Lstm fully convolutional networks for time series classification. *IEEE Access*, PP, 09 2017.

[15] Nitish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tang. On large-batch training for deep learning: Generalization gap and sharp minima. 09 2016.

[16] Andreas Köhler, Matthias Ohrnberger, and Frank Scherbaum. Unsupervised pattern recognition in continuous seismic wavefield records using self-organizing maps. *GEOPHYSICAL JOURNAL INTERNATIONAL*, 182:1619–1630, 09 2010.

[17] Andreas Köhler, Matthias Ohrnberger, Carsten Riggelsen, and Frank Scherbaum. Unsupervised feature selection for pattern search in seismic time series. In *Proceedings of the 2008 International Conference on New Challenges for Feature Selection in Data Mining and Knowledge Discovery - Volume 4*, FSDM'08, page 106–120. JMLR.org, 2008.

[18] Chao Li, Xinggang Wang, Wenyu Liu, Longin Jan Latecki, Bo Wang, and Junzhou Huang. Weakly supervised mitosis detection in breast histopathology images using concentric loss. *Medical Image Analysis*, 53, 02 2019.

[19] Men-Andrin Meier, Zachary Ross, Anshul Ramachandran, Ashwin Balakrishna, Suraj Nair, Peter Kundzicz, Zefeng Li, Jennifer Andrews, Egill Hauksson, and Yisong Yue. Reliable real-time seismic signal/noise discrimination with machine learning. *Journal of Geophysical Research: Solid Earth*, 12 2018.

[20] S.Mostafa Mousavi, Yixiao Sheng, Zhu Weiqiang, and Gregory Beroza. Stanford earthquake dataset (stead): A global data set of seismic signals for ai. *IEEE Access*, PP:1–1, 10 2019.

[21] S.Mostafa Mousavi, Zhu Weiqiang, William Ellsworth, and Gregory Beroza. Unsupervised clustering of seismic signals using deep convolutional autoencoders. *IEEE Geoscience and Remote Sensing Letters*, PP:1–5, 05 2019.

[22] Kaushal Paneri, Vishnu Tv, Pankaj Malhotra, Lovekesh Vig, and Gautam Shroff. Regularizing fully convolutional networks for time series classification by decorrelating filters. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:10003–10004, 07 2019.

[23] Thibaut Perol, Michaël Gharbi, and Marine Denolle. Convolutional neural network for earthquake detection and location. *Science Advances*, 4, 02 2017.

[24] Evan Shelhamer, Jonathon Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1–1, 05 2016.

[25] Shensa. The deflection detector: its theory and evaluation on short-period seismic data. technical report. 01 1977.

[26] Y. Shi, L. Huang, X. Dong, T. Liu, and J. Ning. Application of Fully Convolutional Neural Network on Fault Detection. In *AGU Fall Meeting Abstracts*, volume 2018, pages S13B–04, December 2018.

[27] Chengyun Song, Zhining Liu, Yaojun Wang, Xingming Li, and Guangmin Hu. Multi-waveform classification for seismic facies analysis. *Computers & Geosciences*, 101:1–9, 04 2017.

[28] Chengyun Song, Zhining Liu, Yaojun Wang, Feng Xu, Xingming Li, and Guangmin Hu. Adaptive phase k-means algorithm for waveform classification. *Exploration Geophysics*, 49, 01 2017.

[29] S.W. Stewart. Real time detection and location of local seismic events in central california. *Bulletin of the Seismological Society of America*, 67(2):433–452, 04 1977.

[30] Mustafa Üstüner, Fusun Balik Sanli, and Saygin Abdikan. Balanced vs imbalanced training data: Classifying rapideye data with support vector machines. volume XLI-B7, 07 2016.

[31] Felipe Such, Suhas Pillai, Frank Brockler, Vatsala Singh, Paul Hutkowski, and Raymond Ptucha. Intelligent character recognition using fully convolutional neural networks. *Pattern Recognition*, 88, 12 2018.

[32] Yoones Vaezi and Mirko Baan. Comparison of the sta/lta and power spectral density methods for microseismic event detection. *Geophysical Journal International*, 203:1896–1908, 12 2015.

[33] Zhiguang Wang, Weizhong Yan, and Tim Oates. Time series classification from scratch with deep neural networks: A strong baseline. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 1578–1585, 05 2017.

[34] Andy Wilkins, Andrew Strange, Yi Duan, and Xun Luo. Identifying microseismic events in a mining scenario using a convolutional neural network. *Computers & Geosciences*, 137:104418, 04 2020.

[35] Mitchell Withers, Richard Aster, Christopher Young, Judy Beiriger, Mark Harris, Susan Moore, and Julian Trujillo. A comparison of select trigger algorithms for automated global seismic phase and event detection. *Bulletin of the Seismological Society of America*, 88:95–106, 02 1998.

[36] Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. A review of recurrent neural networks: Lstm cells and network architectures. *Neural Computation*, 31:1–36, 05 2019.

[37] Sanyi Yuan, Jiwei Liu, Shangxu Wang, Tieyi Wang, and Peidong Shi. Seismic waveform classification and first-break picking using convolution neural networks. *IEEE Geoscience and Remote Sensing Letters*, PP:1–5, 01 2018.

[38] Xiong Zhang, Jie Zhang, Congcong Yuan, Sen Liu, Zhibo Chen, and Weiping Li. Locating induced earthquakes with a network of seismic stations in oklahoma via a deep learning method. *Scientific Reports*, 10, 12 2020.

[39] Bendong Zhao, Shanzhu Xiao, Huanzhang Lu, and Junliang Liu. Waveforms classification based on convolutional neural networks. In *2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, pages 162–165, 03 2017.

[40] Jing Zheng, Jiren Lu, Suping Peng, and Tianqi Jiang. An automatic microseismic or acoustic emission arrival identification scheme with deep recurrent neural networks. *Geophysical Journal International*, 212:1389–1397, 02 2018.