Machine Learning Based Seismic Region Classification

Samuel da S. Oliveira, Anne M. P. Canuto, Bruno M. Carvalho, Marcio E. Kreutz
Department of Computer Science and Applied Mathematics Universidade Federal do Rio Grande do Norte Natal, Brazil
Email: samuel@imd.ufrn.br, {anne, bruno, kreutz}@dimap.ufrn.br

***Abstract*—It has become increasingly common in academic and industrial environments the necessity to process huge amounts of seismic signals. Several researchers have been seeking for ways to improve and optimize the processing of these enormous amounts of data that are related to routine demands of geophysicists. One of these demands is the classification of distinct seismic regions captured by the same seismograph, a task that could take up to months of manual data processing. In this paper, we propose the usage of machine learning techniques to the task of the classification of seismic regions, in order to achieve accurate results with better performance and speed. The algorithms K-NN, MLP, Naive Bayes and Decision Tree were used for tests as base classifiers and also combined on ensemble methods. We also employed Deep Learning techniques, namely, a pure RNN network, and a variation of RNN called LSTM. The best results were achieved when using heterogeneous classifiers, showing accuracy rates of up to 98.52%. The results show that one can build an efficient seismic region classification system even when few classified data are already available for a specific seismograph setting.**

***Index Terms*—Seismc Signals, Seismic Processing, Machine Learning**

I. Introduction

Seismic events can be found in the world since its origin and have been studied on a large scale by several researchers for several centuries. These events carry information about specific characteristics such as differences in their size and nature. Passive seismics techniques consider these diverse types of events, trying to understand more about them. A prevalent task in passive seismics [1] is the classification and cataloguing of these events, with the aim of maintaining an extensive catalog for reference and research. One of the main steps of this cataloguing is finding out the exact region where the seismic events occurred.

A single seismograph can record several distinct seismic regions, making it difficult to distinguish the event's origin. This analysis is usually carried out by several researchers observing only the format of the amplitude of the signal captured by the seismographs, since each region has different features in their waveforms. The position of the seismograph in relation to the seismic regions is important, as it interferes with the acquired signals.

Such a work can take hours or even days since all the analysis are usually performed manually. In this paper, we propose the usage of several machine learning techniques to the classification of different seismic regions captured by a single seismograph.

The data used in this research were obtained from a regional seismograph in the state of Rio Grande do Norte, Brazil. Geophysical analysts took about 3 months to classify 3 different regions read by the seismograph. Our goal is to maintain a very low misclassification rate while reducing the classification time as much as possible.

This paper is organized as follows: Section II talks about related works while Section III presents the problem and gives a description of the database used. Section IV details the experiments performed and presents the results and a discussion of them. Finally, Section V presents our conclusions and future work related to this task.

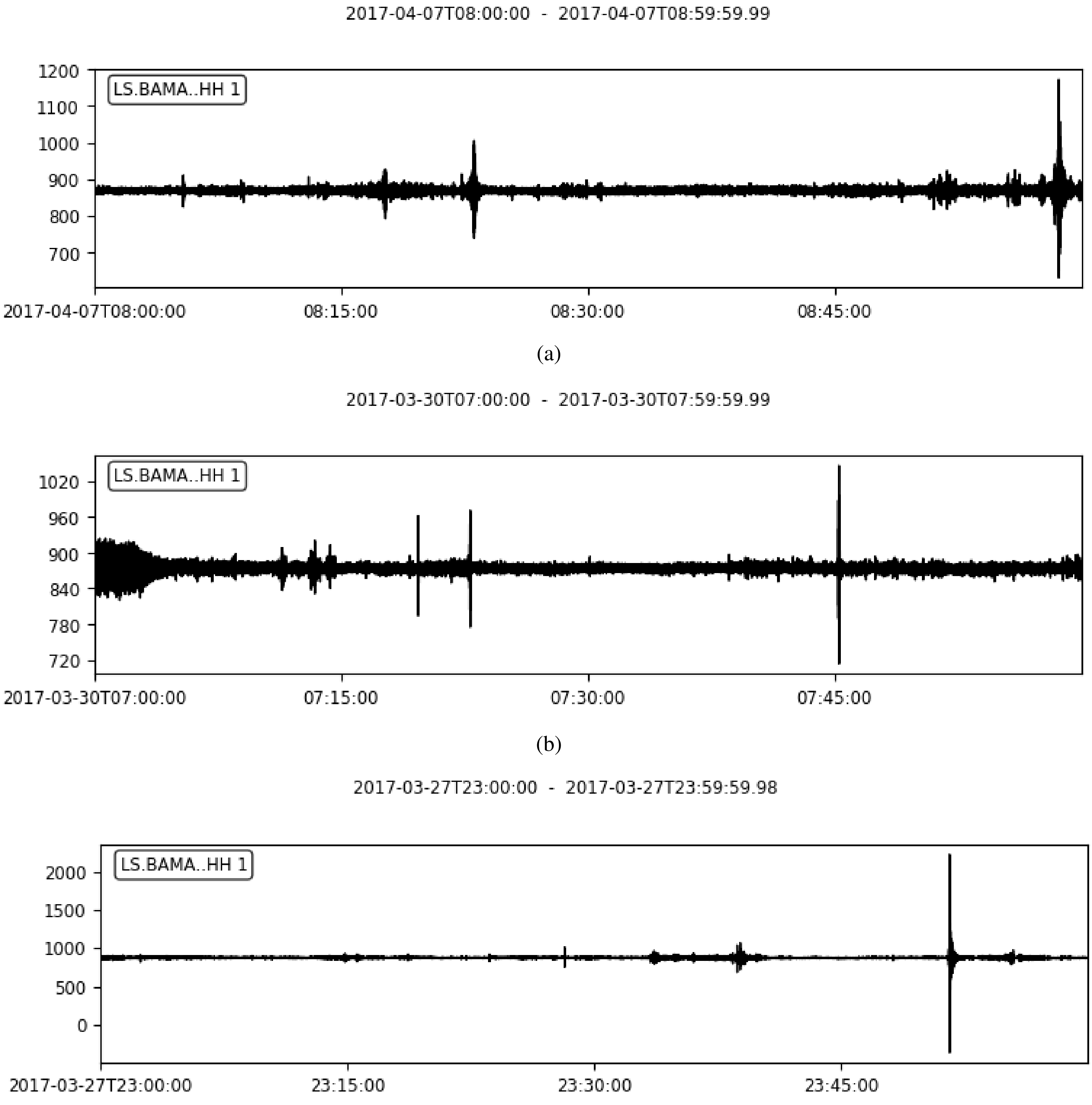
II. Related Works

There are several studies in literature that employ machine learning tools to process seismic signals. We now briefly describe some of them.

The work of Zhang et al. [2] introduces a tool for seismic velocity model building (VMB), whose purpose is to assist the interpreters during the initial stages of the VMB, when no seismic data has been migrated. Their method employs machine learning techniques and can automatically identify and localize faults from seismic data that have not been migrated. They targeted the fault localization problem, but most of the results are obtained using processed seismic data or images as input. Results show that a fully automated VMB was not achieved because the human knowledge was difficult to formalize in a way that could be systematically applied. However, if the framework is extended to other seismic events or attributes, it might become a powerful tool to alleviate the interpreters' work.

Ramirez Jr and Meyer [3] consider the supervised learning problem of seismic phase classification. It was proposed for the classification of seismic phases in three-channel seismic data collected within a network of regional recording stations. Their method improves on current techniques, inserting concepts from machine learning, seeking to learn the characteristics associated with different data patterns, by first using a multi-scale feature extraction technique for clustering seismic data on low-dimensional manifolds. Also, they designed an information theoretic measure used to merge regression scores across the multi-scale feature manifolds. This technique is applied to a set of seismic data from several US states collected during 2005 and 2006. Through cross-validation the method

978-1-7281-6926-2/20/$31.00 ©2020 IEEE



(a)

(b)

(c)

Fig. 1. Examples of instances of Region 1 (a), Region 2 (b) and Region 3 (c).

achieved a 74.6% average correct classification rate when compared to analyst classifications.

Li et. al [4] propose to use a generative adversarial network (GAN) to learn the characteristics of first-arrival earthquake P waves, using 300,000 waveforms recorded in southern California and Japan. The GAN was used as an automatic feature extractor, and a Random Forest classifier was trained with about 700,000 earthquake and noise waveforms. It shows that the discriminator can recognize 99.2% of the earthquake P waves and 98.4% of the noise signals. This study demonstrates that GANs can discover a compact and effective representation of seismic waves and has the potential for broad applications in seismology.

The work proposed by Titos et. al [5] used recurrent neural networks (RNN), long short-term memory (LSTM) and gated recurrent units (GRU) to detect and classify continuous sequences of volcano-seismic events at the Deception Island Volcano, in Antarctica. The data are formed by a representative set of volcano-tectonic earthquakes, long-period events, volcanic tremors, and hybrid events and was used to train these models. Experimental results show that RNN, LSTM, and GRU can exploit temporal and frequency information from continuous seismic data, attaining close to 90%, 94%, and 92% events correctly detected and classified. Despite the variations in the geophysical properties of the seismic events within the volcano across eruptive periods, results provide good generalization accuracy.

Different from other works, our paper approaches seismic regions classification on seismograms that are captured by the same seismograph. We test several machine learning algorithms to cope with the challenge of finding the best results for this problem.

III. Problem Description and Database

The problem addressed here is the classification of seismic regions that are monitored by the same seismograph. The signals originated from these regions may concentrate in the same reading area of a particular seismograph, hindering their exact identification, except by the manual work of an analyst. This paper proposes to identify these different seismic regions through the waveform and its amplitude signal. Figures 1a, 1b and 1c show seismograms with events happening in the three different seismic regions addressed in our problem.

The data used was provided by the Department of Geophysics of UFRN (Federal University of Rio Grande do Norte) with a pre-processing performed through the SAC software that is used by geophysicists for analysis. The data is stored in the SAC [6] file format and unable to be worked on except in the software itself, so they were converted to the ASCII format using the Spyder tool [7]. Each file has 360,000 different amplitude signals, i.e., samples. By varying these values every 0.01 seconds, we averaged every 100 signals, transforming them into attributes, thus, giving us a signal value for every second, totaling 3,600 attributes. In all, we get 54 different files, counting 54 instances and three different regions. In all these instances, a bandpass filter was applied, removing any amplitude signal that was outside the range [5, 25] Hz, thus creating the database.

It is important to note that such a system for seismic region classification has to be trained using data acquired on the same location, since the positioning of the seismographs interfere with the acquired signal. A reasonable goal then would be to build an efficient (fast and accurate) classification system for seismic regions using as little labelled (classified) data as possible.

IV. Experimental Results

This section presents the different experiments done with different machine learning algorithms on supervised, ensembles of classifiers, and deep learning approaches. This section is then divided into four subsections, where the first three show tables with comparative results and the last performs an analysis of the results.

A. *Supervised Classifiers*

For these experiments, four types of classifiers are used in order to compare them and to find the classifier and configuration most suitable for this database. The classifiers were chosen for their differences, performances, and for being widely used in the academic community. The supervised methods used in this work are the K-NN [8], Decision Tree [9], Naive Bayes [10], and MLP [11] with backpropagation. The tests were executed on the Weka tool [12] and Scikit-Learn [13].

With the K-NN, twelve experiments were performed with the standard and staggered values (the values were selected on a scale between 0 and 1) changing the K value, and for each value was applied or not weights. The best K value found was 2, for both staggered and weighted approaches. The Decision Tree algorithm used was the J48 present in WEKA, and the attributes converted from numerical to nominal. Two parameters were considered for the tree, pruned and unpruned. Since the values obtained were the same for both approaches, we selected the pruned option for the remaining experiments in this paper.

The Naive Bayes method is also used with two different parameters, the normal and the kernel distribution. The results of the tests showed that the normal distribution had more significant values, which allowed us to choose it for the continuation of this study. Finally, the neural network algorithm used is the MLP Backpropagation, and with it, 27 different experiments were performed, varying the learning rate, the number of neurons in the hidden layer and the number of iterations of the algorithm. The values chosen for the number of interactions were: 100, 500, and 1,000, while the values used for the number of neurons in the hidden layer werre 200, 250, and 300. For the learning rate, the values used were 0.1, 0.01, and 0.001. These values are chosen for analysing the behavior of the network, and the results obtained so that the training was not too excessive. The experiments for the neural network were performed with 2-fold cross-validation. The best result was the MLP with 1000 interactions with 300 neurons in the hidden layer and a learning rate of 0.01, and a new experiment was performed with 10-fold cross-validation. While we used the ADAM activation function in the experiments with deep learning, we did not use in the MLP experiments because the Weka tool does not have the ADAM function.

The results shown in Table I refer to the best values obtained in the base classifiers.

TABLE I
Base Classifiers

Classifier	Accuracy
K-NN	57.4074 ± 18.2614
Decision Tree	37.0370 ± 18.0484
Naive Bayes	$\mathbf{92.5996 \pm 9.7789}$
MLP	83.3333 ± 15.9862

We then performed a statistical analysis between the 4 models of classifiers used. We set the rate of significance for all tests as 0.05 [14]. We ran the Friedman test, and, since its result reported a p-value < 0.001, another test was applied to know which samples were statistically different from each other. Table II presents the Wilcoxon test results [15] comparing the results from the classifiers.

TABLE II
WILCOXON TEST - CLASSIFIERS

Wilcoxon Test	
Classifiers	p-Values
K-NN and AD	0.001
K-NN and NB	0.005
K-NN and MLP	< 0.001
AD and NB	< 0.001
AD and MLP	< 0.001
NB and MLP	0.009

The Wilcoxon test reveals that all samples reject the null hypothesis and have a significant difference. So, we need to check which classifier is the best, using the accuracy. Naive Bayes $>$ MLP $>$ K-NN $>$ Decision Tree. In other words, the NB is the best classifier for this experiment.

B. Ensembles of Classifiers

In a second batch of tests, we used ensembles of classifiers. The ensembles bundle several classifiers together in order to achieve greater precision than when using a single base classifier. The experiments were executed with 10, 15, and 20 classifiers with 10-fold and ten repetitions per experiment. The settings used in these tests are the same as those used in the supervised approach.

The experiments use three different homogeneous ensemble algorithms, Boosting [16], Bagging [17] and Stacking [18]. The ensemble using three Stackings was the one that obtained the best results. Stacking is also used in its heterogeneous form. The Tables III and IV show the results of Stacking's homogeneous and heterogeneous approaches. We can observe that the increase in the number of classifiers did not always translate into better results.

The Friedman test was also performed comparing the two different Stacking approaches, where the p-value was set to 0.92. Since the p-value is greater than 0.05, no further testing is required, as the null hypothesis is not rejected.

C. Deep Learning

In the last batch of experiments, we applied two deep learning techniques, the pure RNN (Recurrent Neural Network) [19] and LSTM (Long Short-Term Memory) [20], which is a variation of the RNN. We have chosen these networks because several studies in the literature show that RNNs are indicated for processing database temporal series [21]–[24].

The LSTM was trained with several different parameters, and we retained the one with the best result. In this experiment, LSTM and Dense layers were created, and the activation function used was a sigmoid and the optimization function used was ADAM [25]. The LSTM chosen used 10 epochs and 200 neurons. For the pure RNN network, the same settings were used, but we changed the LSTM layer to a SimpleRNN layer. Table V shows the results of this experiment.

The result of the Friedman test showed a p-value of 0.65, indicating that the results produced by the two techniques do not have a significant difference, but observing the standard deviation we can see that the LSTM obtained a better performance.

D. General Analysis

Table VI shows the execution times in seconds, for each technique used. Since the manual classification took around 3 months, we can see that by using machine learning techniques we have a huge time saving for the classification and recognition of the seismic regions. The experiments were performed on a computer with an AMD A10-9600P Quad-Core processor with 3.30 GHz and 4 GB of RAM.

Figure 2 shows a comparative graph between all classifiers showing their associated accuracy and standard deviation. For the experiments performed with homogeneous and heterogeneous stacking due to the use of different amounts of classifiers, the values with the best accuracy / standard deviation were chosen to be presented. Below we can observe this behavior on the chart.

After all the experiments and statistical tests we can see that the results of the ensembles were usually better than the results of the individually based classifiers. The heterogeneous stacking technique composed by using 15 Naive Bayes and MLP classifiers achieved the best results for the all classifiers, obtaining an accuracy of 98.52%.

Even when compared to the results produced by deep learning techniques for time series, the Heterogeneous Stacking achieved better results. We believe that this advantage in comparison to the use of RNN and LSTM is due to the low number of instances in our base, and that, given a substantial amount of data, these later techniques would also achieve excellent accuracy scores.

When compared to the results of [5], which also makes use of an RNN, LSTM and GRU (Gated Recurrent Unit) techniques for detecting and classifying volcano-seismic events, we were able to obtain better results. Their results achieved successful classification rates close to 90%, 94% and 92%. In our experiments using RNN and LSTM, we achieved accuracy rates of 92.98% and 93.38% respectively, which are results very close to those was reported in [5], but when we compare it with our approach using Heterogeneous Stacking, mixing Naive Bayes and MLP with the result of 98.52%, we can see that our approach stands out for this particular classification of seismic data.

We believe that the big difference between the classifiers is due to the fact that the data are from time series and have values that are very close, which made it difficult to use the K-NN and the decision trees. As there were few instances in the database, MLP also did not perform well, making Naive have the best values of the supervised classification.

TABLE III
HOMOGENEOUS STACKING

	Accuracy		
Classifier	10	15	20
K-NN	53.1666 ± 18.1734	53.1666 ± 18.1734	53.1666 ± 18.1734
DT	38.2333 ± 17.5199	38.2333 ± 17.5199	38.2333 ± 17.5199
NB	**93.2626 ± 10.4695**	**93.2626 ± 10.4695**	**93.2626 ± 10.4695**
MLP	85.0000 ± 14.5578	84.7500 ± 16.2862	80.0000 ± 16.3299

TABLE IV
HETEROGENEOUS STACKING

	Accuracy		
Classifier	10	15	20
NB/MLP	88.7000 ± 12.0128	**98.5200 ± 5.6445**	72.3400 ± 10.3203
NB/K-NN	91.5000 ± 12.0266	92.2000 ± 11.2637	90.6060 ± 14.0393
MLP/K-NN	64.5667 ± 18.9876	64.5600 ± 8.6123	72.4540 ± 15.2387
NB/MLP/K-NN	78.4500 ± 10.3267	70.3400 ± 26.5312	84.4540 ± 13.5502

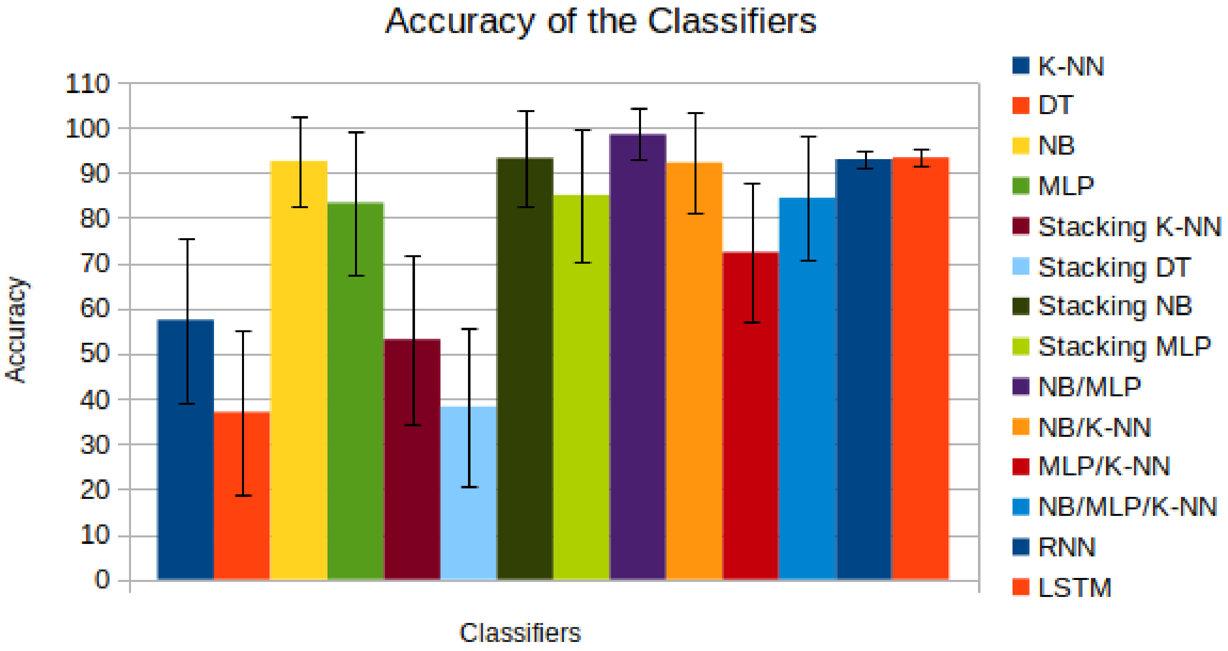


Fig. 2. Accuracy of the analyzed classifiers.

TABLE V
DEEP LEARNING

Classifier	Accuracy
RNN	92.9836 ± 1.9288
LSTM	93.3836 ± 1.8735

Given the small number of samples we have available, it is understandable that the deep learning techniques tested did not produce results with accuracies as high as the Heterogeneous Stacking. We are currently in the process of acquiring and classifying more data. However, the results presented here point out to the usage of heterogenous ensembles for classifying seismic regions from seismographs when few data are available.

V. CONCLUSION AND FUTURE WORK

This work proposes the use of several machine learning techniques for seismic region classification, seeking to find approaches that can facilitate the work of analysts. As we mentioned before, it is important to point out that this is a task that is very time consuming and that the patterns that characterize the signals belonging to the different regions are dependent on the position of the seismograph in relation to them.

TABLE VI
RUNTIME

Classifier	Runtime (s)
K-NN	73
Decision Tree	65
Naive Bayes	50
MLP	122
Stacking K-NN	90
Stacking DT	87
Stacking NB	76
Stacking MLP	187
NB/MLP	278
NB/K-NN	180
MLP/K-NN	230
NB/MLP/K-NN	343
RNN	480
LSTM	440

The main goal of this work was to show that it is possible to build a classification system for this task by using a reasonably low number of labelled samples. Among the techniques used, the Stacking Heterogeneous approach was the one that obtained the best accuracy values, reaching an accuracy of 98.52%. The individual base classifiers that achieved the in our experiments were the Naive Bayes and MLP, according to the supervised experiments. Thus, as expected, when we use Stacking with MLP and Naive Bayes as the base classifiers, the best accuracy values were obtained.

As future directions, we intend to increase the number of instances to be automatically classified by the system and confirmed by the analysts. By doing so, we can produce a larger amount of labelled samples and build a high accuracy RNN.

ACKNOWLEDGMENT

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior Brasil (CAPES) Finance Code 001.

REFERENCES

[1] Peter M Duncan. Is there a future for passive seismic? *First Break*, 23(6), 2005.

[2] Chiyuan Zhang, Charlie Frogner, M Araya-Polo, and D Hohl. Machine-learning based automated fault detection in seismic traces. In *76th EAGE Conference and Exhibition 2014*, 2014.

[3] Juan Ramirez Jr and François G Meyer. Machine learning for seismic signal processing: Phase classification on a manifold. In *2011 10th International Conference on Machine Learning and Applications and Workshops*, volume 1, pages 382–388. IEEE, 2011.

[4] Zefeng Li, Men-Andrin Meier, Egill Hauksson, Zhongwen Zhan, and Jennifer Andrews. Machine learning seismic wave discrimination: Application to earthquake early warning. *Geophysical Research Letters*, 45(10):4773–4779, 2018.

[5] Manuel Titos, Angel Bueno, Luz García, M Carmen Benítez, and Jesús Ibañez. Detection and classification of continuous volcano-seismic signals with recurrent neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 57(4):1936–1948, 2018.

[6] S Mahin, J Malley, and R Hamburger. Overview of the FEMA/SAC program for reduction of earthquake hazards in steel moment frame structures. *Journal of Constructional Steel Research*, 58(5-8):511–528, 2002.

[7] Robert Johansson. *Numerical Python: A Practical Techniques Approach for Industry*. Apress, 2015.

[8] Zhenyun Deng, Xiaoshu Zhu, Debo Cheng, Ming Zong, and Shichao Zhang. Efficient k-NN classification algorithm for big data. *Neurocomputing*, 195:143–148, 2016.

[9] S Rasoul Safavian and David Landgrebe. A survey of decision tree classifier methodology. *IEEE transactions on systems, man, and cybernetics*, 21(3):660–674, 1991.

[10] Irina Rish. An empirical study of the naive Bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46, 2001.

[11] Nevio Benvenuto and Francesco Piazza. On the complex backpropagation algorithm. *IEEE Transactions on Signal Processing*, 40(4):967–969, 1992.

[12] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The WEKA data mining software: An update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

[13] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

[14] Edgar Brunner, Sebastian Domhof, Frank Langer, and Edgar Brunner. *Nonparametric Analysis of Longitudinal Data in Factorial Experiments*. J. Wiley New York, 2002.

[15] William H Kruskal. Historical notes on the Wilcoxon unpaired two-sample test. *Journal of the American Statistical Association*, 52(279):356–360, 1957.

[16] Yoav Freund, Robert Schapire, and Naoki Abe. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612, 1999.

[17] J Ross Quinlan et al. Bagging, boosting, and C4.5. In *AAAI/IAAI, Vol. 1*, pages 725–730, 1996.

[18] Li Deng, Dong Yu, and John Platt. Scalable stacking and learning for building deep architectures. In *2012 IEEE International conference on Acoustics, speech and signal processing (ICASSP)*, pages 2133–2136. IEEE, 2012.

[19] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.

[20] Felix A Gers and E Schmidhuber. Lstm recurrent networks learn simple context-free and context-sensitive languages. *IEEE Transactions on Neural Networks*, 12(6):1333–1340, 2001.

[21] Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, and Puneet Agarwal. Long short term memory networks for anomaly detection in time series. In *Proceedings*, volume 89. Presses universitaires de Louvain, 2015.

[22] Haibin Cheng, Pang-Ning Tan, Jing Gao, and Jerry Scripps. Multistep-ahead time series prediction. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 765–774. Springer, 2006.

[23] Toru Nakashika, Tetsuya Takiguchi, and Yasuo Ariki. Voice conversion using rnn pre-trained by recurrent temporal restricted boltzmann machines. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):580–587, 2014.

[24] Chao Che, Cao Xiao, Jian Liang, Bo Jin, Jiayu Zho, and Fei Wang. An RNN architecture with dynamic temporal matching for personalized predictions of Parkinson's disease. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 198–206. SIAM, 2017.

[25] D. P. Kingma and J. Ba. ADAM: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.