Seismic Event Detection from Volcanic Regions using Feature Extraction and Deep Learning

Tavishi Priyam[1], Elizabeth Bradley[1,2], Anne Sheehan[3], Rey Koki[1]

1. Department of Computer Science, University of Colorado Boulder 2. Department of Computer Science, Santa Fe Institute
3. CIRES and Department of Geological Sciences, University of Colorado Boulder

ABSTRACT

Volcanic eruptions are usually accompanied by seismic events with a variety of temporal signatures: volcano-tectonic earthquakes, low-frequency earthquakes, tremors, and explosions. The detection of these events in seismograph data has been done manually for several decades now. This process produces accurate results but is time and labor-intensive. A variety of machine-learning methods have been developed over the past decade to automate elements of this task. Few of these methods incorporate expert knowledge about the process; rather, they train a model with a large labeled data set and then simply work with the raw seismograph data. Our goal is to build and train a model that combines supervised machine learning and expert knowledge to aid in this process. The main technical challenge here is feature engineering: identification of task-specific, salient, higher-level attributes that can be extracted from the raw data and used as inputs to the model. A well-chosen feature set can greatly improve both training and deployment. Through this research, we describe a neural network that identifies the arrival times of the P and S waves of small earthquakes preceding the 2012 New Zealand Mount Tongariro/Te Maari craters eruption. We utilize seismic waveform and analyst-reviewed phase arrival data, which are available through GeoNet New Zealand. The initial implementation involves training a neural network that contains convolutional and fully connected layers on the raw waveform data. The subsequent implementation involves training a similar neural network model, without the convolutional layers, using a set of engineered features. We compare the performance of both models to determine the efficiency and effectiveness of using a feature set as opposed to using raw waveform data.

KEYWORDS

Seismic Event Detection, Seismic Phase Picking, Machine Learning, Deep Learning, Feature Extraction

1 Introduction

Artificial neural networks have been successful in automating a multitude of tasks over the past decade. This automation reduces manual labor and standardizes processes. One such task is picking arrival times of P and S waves from seismograph data. This task requires analysts to study the seismographs in order to be able to detect the event as well as the picks. Due to the tedious nature of the task, attempts have been made at automating this process through the application of machine learning and artificial neural networks. These attempts have shown great success in achieving the desired goal. However, as any other automation process, such models have their limitations.

Artificial neural networks require a great amount of computation resources and time in order to train and learn patterns from data. In addition to

this, most neural network models work like a black-box with little to no understanding of how they are performing a certain task. This makes identification and correction of errors harder and sometimes even impossible. Through our research, we aim to integrate scientifically meaningful feature engineering with the artificial neural networks. We hypothesize improved understanding and performance through our methods.

2 Related Work

A great amount of research has been done in this field and a greater amount is still ongoing. To understand and approach the process of combining phase picking with a computational deep learning model, we conducted a literature review of numerous research articles that describe integrative models between the world of computer science and geology.

Ross et al. [1] propose a convolutional neural network approach to detect earthquakes without sacrificing detection sensitivity. They compare the task of recognising seismic phases in a waveform time series to that of recognising objects in two-dimensional images since both objects can be represented using pixels in a two-dimensional space. They apply the recent advances in computer vision to seismological applications and report noteworthy results.

Mousavi et al. [2] propose a global deep learning model for simultaneous earthquake identification and phase picking to ease the process of handling and preprocessing noisy data. Proposed model involves a combination of layers of multiple convolutional neural networks (CNN) and bidirectional long short-term memory neural networks (bi-LSTM). Following these layers is an amalgam of transformers that work with attention weights which help define the importance of each input to be considered for predicting the output. The model performs well due to factors such as specifics of the training set, architecture of the training model, attention weight mechanism, depth of the network and the augmentations used during the training process.

Zhu et al. propose an arrival-time picking model called Phasenet [3]. This is a deep-learning model that can identify the arrival times of both P and S waves from seismic data. They utilize the architecture of U-net (deep neual network used to process biomedical images) and modify the architecture such that it can process one dimensional time-series waveforms. They use downsampling and upsampling throughout the process of training such that the neurons can focus on the most important features of the input and expand learning on those features. This model exhibits great performance and is able to pick 94 percent of P-waves and 85 percent of S-waves within 0.1 seconds of the analyst pick times.

Yoon et al. [4] propose a Fingerprint and Similarity Thresholding (FAST) model which focuses on performing an efficient similarity search and using it to detect the occurrence of

seismic events. The model works by converting the time series seismic waveform data to fingerprints (similar to audio fingerprinting. Eg-Shazam). The FAST algorithm builds on the waveprint audio fingerprinting algorithm by combining different computer vision and large scale data processing techniques to match similar waveforms. In their research, the authors observed that FAST can detect several uncataloged earthquakes in one week of continuous time series data. Along with the advantages, the proposed algorithm has some limitations. It trades off higher memory and storage requirements for quicker runtime and reduced complexity. It needs a significant amount of memory in order to be able to successfully detect previously undetected earthquakes.

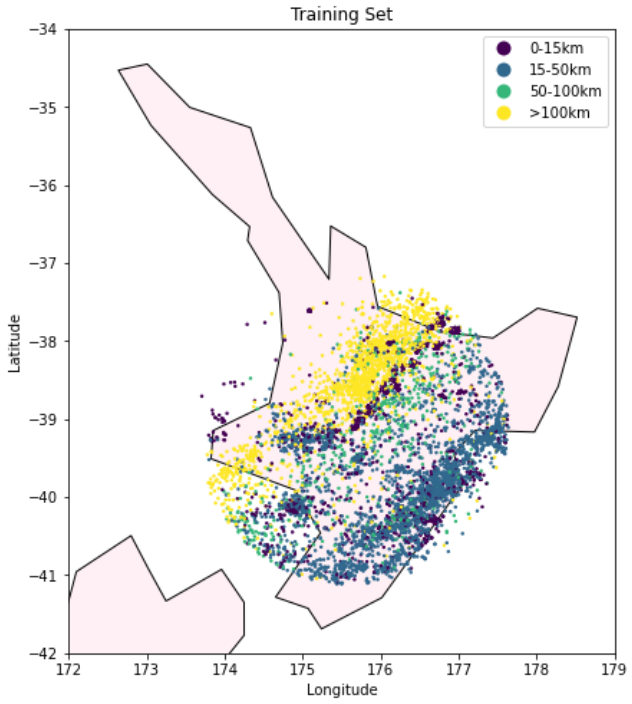

Figure 1: **Training Earthquakes**

3 Data

In order to train a neural network, a great amount of labeled data is required. The area of focus for this research is the Southern Taupo Volcanic Zone, New Zealand. Due to the volcanic nature of the target region, there has been a lot of seismic activity observed over the past decade which suits our training needs. Figure 1 shows the target area and earthquakes used for training.

3.1 Data Extraction

Seismic data from the specified region is provided opensource by GeoNet. We extract one year of seismic data, collected using a short-period seismometer with three component sensor aligned to North from our region of interest. The data extraction process is carried out using ObsPy [5]. ObsPy is an open-source framework that allows extraction and processing of seismic data using simple functions that can be used seamlessly in Python using the ObsPy library. The extracted data is divided into training and testing sets in a 70:30 ratio.

This data is segmented into short time windows of 30 seconds. For the purpose of training, we filter out the time windows without any seismic activity and windows containing only P-wave arrivals. The resultant is a dataset containing windows that have both P and S wave arrivals. An example is described in Figure 2. We create tensors using this data by stacking the waveforms from three different sensor channels: EHE, EHN, EHZ. The labels are created by

creating tensors of the same length that have zeros representing all the timesteps and ones representing the arrival time of the P and S waves.

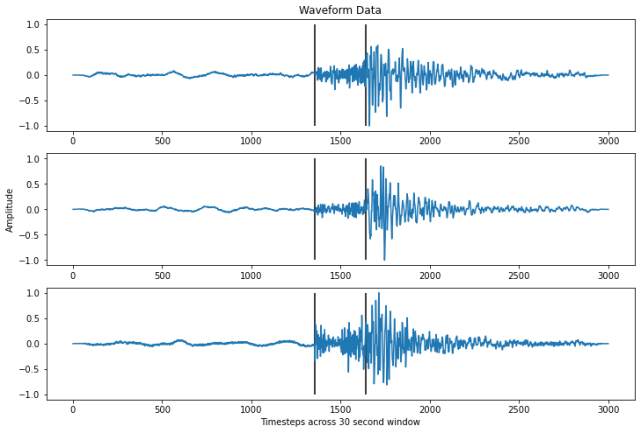

Figure 2: **30 second waveform window**

3.2 Data Pre-processing

The extracted data is pre-processed to ensure uniform learning of the artificial neural network. We apply the detrend function, provided by Obspy, on the waveforms in order to remove the mean from every signal. We also apply a taper of five percent to the ends of each time window using the taper function provided by Obspy.

In order to scale down the amplitude to a uniform scale, we use the normalize function provided by ObsPy. The resultant waveforms are saved in a numpy array format and used for the training purpose. Figure 3 describes a waveform before and after the pre-processing steps have been applied.

3.3 Feature Engineering

In order to incorporate scientific knowledge into the training process, we engineered features using the original signals. Seismic events such as earthquakes possess characteristic temporal signatures. Our aim is to capture these temporal signatures and use them to help the model learn. Templates have been used in the past to perform this task [8]. For our study, we make use of Wavelets to operationalize the proposed methods since they have proven to work well for seismic data analysis [9].

A wavelet is an oscillating signal that mimics the action of a wave. Wavelets are localized in time and can be manipulated using their scale and location. A greater scale refers to stretched out shape of the wavelet whereas a smaller scale would result in a compact wavelet. The location of the wavelet helps in shifting left or right on any signal.

Wavelet transform helps in capturing temporal as well as frequency information from a signal. There exist two main types of wavelet transform: discrete and continuous. The main difference between the two exists in their scale. Discrete wavelet transform, used in this study, convolves the original signal over a mother wavelet on a defined set of scales and locations whereas the continuous wavelet transform convolves a signal over an infinite number of scales and locations. This results in much higher computation time for the continuous wavelet transform.

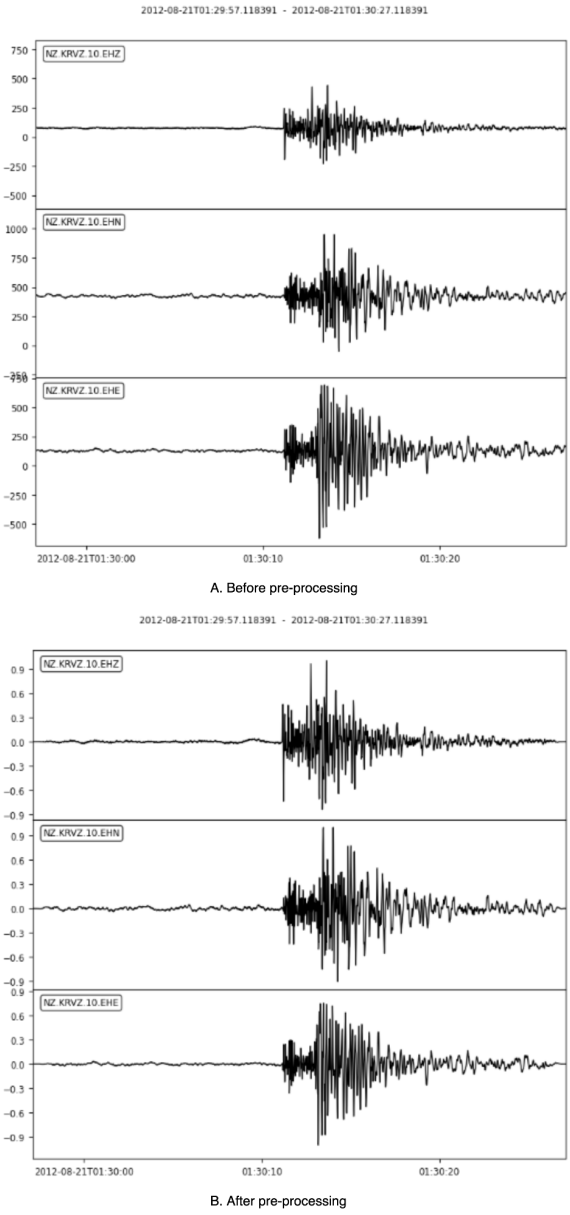

Figure 3: **Results of pre-processing**

The python package *pywt* [10] offers a great selection of mother wavelets than can be used to transform a signal. We use the discrete Debauchies 9 wavelet for our engineered features. An example of a transformed signal is described in Figure 4.

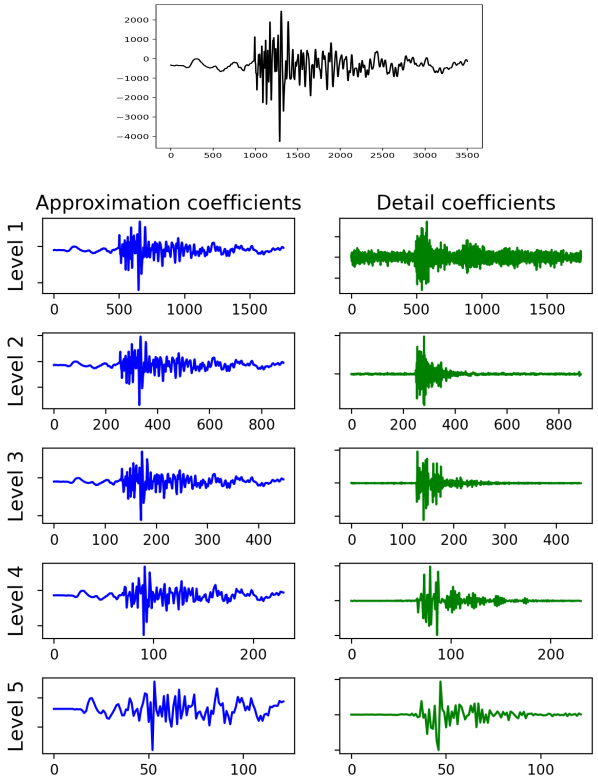

Figure 4: **Wavelet Transform**

4 Proposed Model

We propose two different model architectures that can capture seismic activity from waveform data. The first architecture focuses on using pre-processed signals as input and leverages the feature maps created by convolutional layers in a neural network to get the desired result. The model is a deep neural network comprising several convolutional, pooling and

fully-connected layers. The architecture is described in Figure 5.

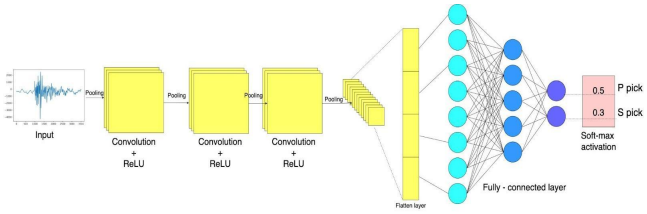

Figure 5: **Signal input model architecture**

The second model aims at using characteristic temporal signatures of seismic waves through engineering features using these signatures in the form of wavelets. The wavelets are passed through a deep neural network as input and the output probabilities describe the P and S-wave picks. This network contains a lesser number of convolutional layers and hence reduces runtime and complexity while producing the desired results. The architecture is described in Figure 6.

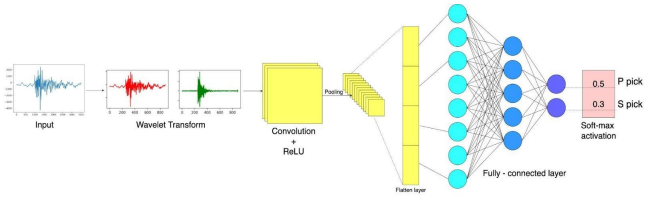

Figure 6: **Wavelet input model architecture**

4.1 Convolutional Neural Network

Convolutional Neural Networks are a type of artificial neural networks that have been majorly been applied to visual/image data. These networks mimic the patterns of biological neural networks present in the human brain in order to train a machine to be able to predict and analyse patterns in data like humans.

The key performance development exhibited by such networks comes from the convolutional layers. These layers are the building blocks of these networks. They contain a set of filters or *kernels* that are modified throughout the training process. These filters are applied on the training data through multiple iterations and convolve the training input into feature maps. The feature maps are used by the sequential fully-connected layers to produce the desired output.

Fully connected layers refer to a subset of the deep neural network where every neuron from one layer is connected to every neuron in the next layer. They work by performing a linear transformation on the input to each layer. This transformation involves multiplication with weights and the addition of a bias term. The weights and bias are learnt throughout multiple iterations of training the network. The last fully-connected layer usually contains the size of the desired output and helps compile all the data learnt throughout the training process.

For the purpose of training, we have created a deep neural network comprising three convolutional layers followed by a pooling layer and 3 fully-connected layers. Using a pooling layer helps reduce the dimensionality of feature maps and reduces the overall complexity of the network. The model used to train the wavelets

contains lesser number of convolutional layers with the same network architecture.

The neural network is compiled using a focal loss function. Focal loss proves to be extremely helpful in dealing with class imbalance. It applies a modulating term to a basic cross-entropy loss which focuses the model learning on the hard and misclassified examples in the training set [13]. We also make use of the adam optimizer which applies an optimized gradient descent to the training process. We design the training architecture using PyTorch [14].

5 Performance Evaluation

The proposed models' performance is tested on previously unseen earthquakes from the test set. We observe that both models are able to pick a majority of seismic phase arrivals within 0.5 seconds of the analyst pick times. It is seemingly clear from the model performance that P waves are easier to capture in comparison to S waves.

	P waves within 0.5s	S waves within 0.5s
Signal Input	0.87	0.79
Wavelet Input	0.92	0.83

Table 1: **Model Performance**

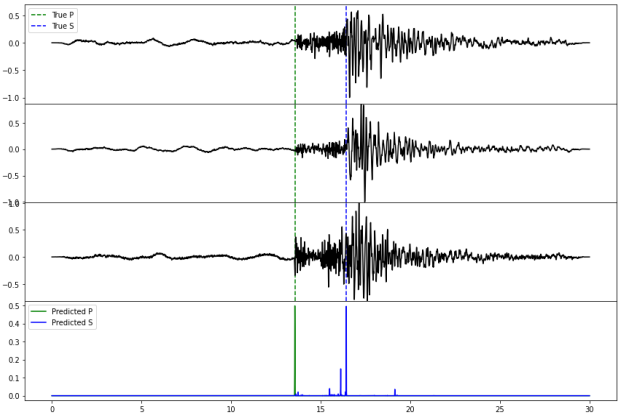

Figure 7: **Pre-processed signal model performance**

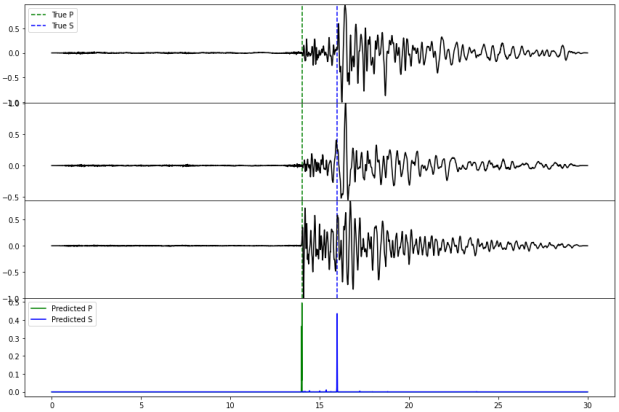

Figure 8: **Wavelet input model performance**

Despite exhibiting comparable performance, we observe a minor improvement in the performance exhibited by the model with feature engineered inputs. This combined with the lesser complexity, runtime and integration of scientific knowledge make it the better choice among both proposed models. A comparison of the performance of both models is demonstrated in Table 1.

The performance of these models is not up to the state of the art pickers such as PhaseNet and

EQTransformer. Additional training data, pre-processing and computational layers can help improve the performance.

5 Future Work

There is great amount of work that can potentially improve our current hypothesis. One area to be explored is the feature engineering. Addition of a varying set of features involving basic statistical features such as mean, standard deviation and interquartile range as well as advanced features that have proven to work well on seismic data such as covariance matrix [15], Shannon entropy and coherency function may help improve the models performance.

We plan to expand the current model and incorporate the classification of various kinds of seismic events such as tremor and low-frequency earthquakes.

7 Conclusion

Through this research, we have established that seismic phase picking can be automated through the application of artificial intelligence and the use of scientifically meaningful engineered features helps improve the performance. The current application has a lot of limitations and would need to be improved in order to be deployed at a larger scale.

REFERENCES

1. Zachary E. Ross, Men-Andrin Meier, Egill Hauksson, Thomas H. Heaton; Generalized Seismic Phase Detection with Deep Learning. *Bulletin of the Seismological Society of America* 2018;; 108 (5A): 2894–2901. doi: https://doi.org/10.1785/0120180080

2. Mousavi, S.M., Ellsworth, W.L., Zhu, W. *et al.* Earthquake transformer—an attentive deep-learning model for simultaneous earthquake detection and phase picking. *Nat Commun* 11, 3952 (2020). https://doi.org/10.1038/s41467-020-17591-w

3. Weiqiang Zhu, Gregory C Beroza, PhaseNet: a deep-neural-network-based seismic arrival-time picking method, Geophysical Journal International, Volume 216, Issue 1, January 2019, Pages 261–273, https://doi.org/10.1093/gji/ggy423

4. Yoon, Clara & O'Reilly, Ossian & Bergen, Karianne & Beroza, Gregory. (2015). Earthquake detection through computationally efficient similarity search. Science Advances. 1. e1501057-e1501057. 10.1126/sciadv.1501057.

5. M. Beyreuther, R. Barsch, L. Krischer, T. Megies, Y. Behr and J. Wassermann (2010) ObsPy: A Python Toolbox for Seismology SRL, 81(3), 530-533, DOI: 10.1785/gssrl.81.3.530

6. T. Megies, M. Beyreuther, R. Barsch, L. Krischer, J. Wassermann (2011) ObsPy – What can it do for data centers and observatories?, Annals Of Geophysics, 54(1), 47-58, DOI: 10.4401/ag-4838

7. L. Krischer, T. Megies, R. Barsch, M. Beyreuther, T. Lecocq, C. Caudron, J. Wassermann (2015) ObsPy: a bridge for seismology into the scientific Python ecosystem, Computational Science & Discovery, 8(1), 014003, DOI: 10.1088/1749-4699/8/1/014003

8. Mu, Dawei & Lee, En-Jui & Chen, Po. (2017). Rapid earthquake detection through GPU-Based template matching. Computers & Geosciences. 109. 10.1016/j.cageo.2017.09.009.

9. Adhikari, B., Dahal, S., Karki, M. et al. Application of wavelet for seismic wave analysis in Kathmandu Valley after the 2015 Gorkha earthquake, Nepal. Geoenviron Disasters 7, 2 (2020). https://doi.org/10.1186/s40677-019-0134-8

10. Gregory R. Lee, Ralf Gommers, Filip Wasilewski, Kai Wohlfahrt, Aaron O'Leary (2019). PyWavelets: A Python package for wavelet analysis. Journal of Open Source Software, 4(36), 1237, https://doi.org/10.21105/joss.01237.

11. Illsley-Kemp, F., Barker, S. J., Wilson, C. J. N., Chamberlain, C. J., Hreinsdóttir, S., Ellis, S., et al. (2021). Volcanic unrest at Taupō volcano in 2019: Causes, mechanisms and implications. Geochemistry, Geophysics, Geosystems, 22, e2021GC009803. https://doi.org/10.1029/2021GC009803

12. Harish Sangireddy, Colin P. Stark, Anna Kladzyk, Paola Passalacqua, GeoNet: An open source software for the automatic and objective extraction of channel heads, channel network, and channel morphology from high resolution topography data, Environmental Modelling & Software, Volume 83, 2016, Pages 58-73, ISSN1364-8152, https://doi.org/10.1016/j.envsoft.2016.04.026.

13. T. Lin, P. Goyal, R. Girshick, K. He and P. Dollar, "Focal Loss for Dense Object Detection" in IEEE Transactions on Pattern Analysis & Machine Intelligence, vol. 42, no. 02, pp. 318-327, 2020. doi: 10.1109/TPAMI.2018.2858826

14. Paszke, A. et al., 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Advances in Neural Information Processing Systems 32. Curran Associates, Inc., pp. 8024–8035. Available at: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

15. Shi, P., Seydoux, L., & Poli, P. (2021). Unsupervised learning of seismic wavefield features: clustering continuous array seismic data during the 2009 L'Aquila earthquake. Journal of Geophysical Research: Solid Earth, 126, e2020JB020506. https://doi.org/10.1029/2020JB020506.

16. https://ataspinar.com/2018/12/21/a-guide-for-using-the-wavelet-transform-in-machine-learning/

https://towardsdatascience.com/multiple-time-series-classification-by-using-continuous-wavelet-transformation-d29df97c0442

https://medium.com/visionwizard/understanding-focal-loss-a-quick-read-b914422913e7

https://pytorch.org/docs/stable/optim.html

https://www.deeplearningwizard.com/deep_learning/practical_pytorch/pytorch_convolutional_neuralnetwork/