Summer 7-15-2022

Machine Learning Techniques for Seismic Data Analysis and Explosion Monitoring

Mohammad Ashraf Siddiquee
University of New Mexico

Mohammad Ashraf Siddiquee

Candidate

Computer Science

Department

This dissertation is approved, and it is acceptable in quality and form for publication:

Approved by the Dissertation Committee:

Abdullah Mueen	Chair
Shuang Luan	Member
Huiping Cao	Member
Trilce Estrada	Member
Jonathan MacCarthy	Member

Machine Learning Techniques for Seismic Data Analysis and Explosion Monitoring

BY

Mohammad Ashraf Siddiquee
B.S., Computer Science & Engineering, University of Dhaka, 2013

DISSERTATION

Submitted in Partial Fulfillment
of the Requirements for the Degree of
Doctor of Philosophy

Computer Science
The University of New Mexico
Albuquerque, New Mexico

August, 2022

DEDICATION

This dissertation is dedicated to my wife, Humayra

my daughter Zunaira, and to my parents

ACKNOWLEDGMENTS

I have been very fortunate to work with a set of extraordinary people without whom the dissertation would not be even possible. First and foremost, I would like to thank my advisor Prof. Abdullah Mueen for his extraordinary guidance and support over the last six years. He not only taught me how an idea can be transformed into a research project, but also how to communicate the work effectively. Mueen's advice and guidance pushed me beyond my limits at the most difficult situations and helped me grow as an independent researcher. The journey with him has been very enjoyable yet effective and this will remain invaluable throughout the rest of my career.

I am also extremely grateful to my other committee members Shuang Luan, Huiping Cao (NMSU), Trilce Estrada and Jonathan MacCarthy (LANL) for their suggestion, comment and feedback on my work. Shuang's advice helped me navigate through not only my academic journey at UNM, but also my relocation here in the US. I thank Trilce and Huiping for their guidance in machine learning part of my research. Jonathan was my go-to person for any seismology related questions. I thank him for clarifying tons of nebulous seismology concepts.

I would like to extend my thanks to Glenn Eli Baker (AFRL) and Kenny Ryan (AFRL) for their contributions in my research. I have enjoyed and learned a lot from the in-depth bi-weekly meetings with Eli. I would like to thank Keith Koper (U of Utah) for contributing in my research. I express my gratitude to Biplob Debnath (NEC Laboratories) and Ofer immanuel (Meta Inc.) for being great mentors in my internships.

I would like to thank all my fellow students at the lab for being wonderful peers and amazing friends. In particular, I would like to mention Vinicius Souza, Farhan Asif Chowdhury and Sheng Zhong for engaging in my work and for the brainstorming sessions we had together. I would also like to

thank Rashidul Hasan, Dheeman Saha, Lawrence Allen, Zeinab Akhavan, José Abel Castellanos Joo, and Parvez Mollah for their wonderful friendship. Special thanks to Abirul Islam for the extended problem solving sessions we have had together.

I can not express my gratitude enough to my parents Tafazzal Ali and Sultana Mahbuba for believing in me and shaping me into who I am today. I thank my mother-in-law Khurshid Jahan who travelled thousands of miles to take care of us when we were in need. I thank my sister Mahbuba Ashrafi and brother-in-law Imrul Hasan for taking care of all the difficult situations back in Bangladesh. I thank my little brother Ahbab Siddique for being an inspiration in my life.

I am forever grateful to Humayra for her continuous support, encouragement, and patience throughout this journey. Without her, I wouldn't have come this far. Lastly, I thank Zunaira, my nephews Jawad and Zahiyan for being a part of our life. Watching them grow has been the most satisfying experience in my life.

Machine Learning Techniques for Seismic Data Analysis and Explosion Monitoring

by

Mohammad Ashraf Siddiquee

B.S., Computer Science & Engineering, University of Dhaka, 2013

PhD, Computer Science, University of New Mexico, 2022

ABSTRACT

The current seismic data processing pipeline is surprisingly human-dependent. With the rapid increase of seismic-sensor data availability, all manual data processing approaches fail to detect, classify, and analyze seismic activity within a reasonable amount of time. An automated, fast, and reliable seismic data processing pipeline is desired for the meaningful analysis of massive seismic datasets. In this thesis, we show how advanced time-series data-mining and machine learning techniques can be leveraged to resolve this issue. We precisely focus on seismic activity detection, classification, and inspection using our techniques that would help us better understand the surrounding earth structure, earthquake evaluation, and seismic monitoring

In this dissertation, (a) we demonstrate a semi-supervised motif discovery algorithm that forms a nearest neighbor graph to discover novel seismic events from static continuous waveforms. (b) We exhibit a seismic data repository system that can extract thousands of seismic waveforms including annotations using complex queries within seconds. (c) We design and implement a hierarchical neural network that can predict seismic depth from seismograms and classify deep and shallow earthquakes with 86.5% F1 score.

TABLE OF CONTENTS

List of Figures

List of Tables

Chapter 1

Introduction

The seismic signal processing pipeline is heavily human-dependent. From signal detection to aggregation of multiple channels and association of multiple networks, trained domain experts must annotate the seismic signals at each step. Professional annotators manually examine the traces recorded by seismic sensors and use physics-based techniques and their expertise to catalog seismic activity. Manual annotation is tedious and time-consuming. Consider the International Data Center (IDC) that employs human analysts to annotate seismic events. The human analysts at IDC processed 126 events per day in 2007, on average [15]. In contrast, thousands of earthquakes of magnitude 2.0 and above happen everyday across the globe [28]. Thus, human involvement seriously limits the processing capability of a global seismic network, such as the International Monitoring System (IMS) for the Comprehensive Nuclear Test-Ban Treaty (CTBT).

The main goal of this dissertation is to automate the seismic data processing pipeline by analyzing spatio-temporal features in seismic waveforms and introducing state-of-the-art data mining and machine learning techniques. To do this, we introduce a semi-supervised time series motif discovery system for seismic signal detection (SeiSMo). We discuss this in Chapter 2. SeiSMo is able to detect seismic activity in the trace from one channel. Us-

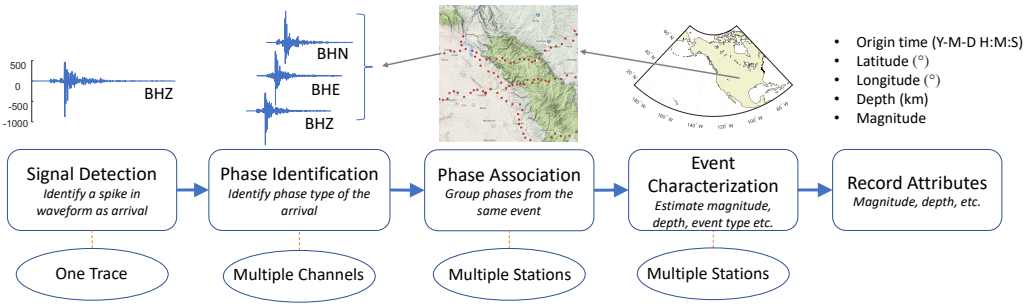

Figure 1.1: Seismic data processing pipeline. This pipeline refers to the process of transforming a set of seismic signals into a bulletin of seismic events (i.e earthquakes, explosions, etc). We discuss the seismic data processing pipeline in-detail in Chapter 4.

ing six different datasets, we show how SeiSMo can be effective to detect hidden novel seismic events that were missed by legacy systems. In Chapter 3, we demonstrate a seismic data repository system that enables fast retrieval of seismic waveform data and corresponding annotations. The system facilitates with data sources for research projects on intelligent seismic phase identification, aftershock detection, and seismic depth estimation. In Chapter 4, we introduce a seismic depth prediction model (Septor), Septor takes advantage of multiple stations in a seismic network in order to learn a predictive model. To the best of our knowledge, Septor is the first attempt at learning depth estimation of a seismic event.

1.1 Semi-supervised Seismic Event Detection

The seismic data processing pipeline (Figure 1.1) starts with seismic event detection, and the accuracy of subsequent steps relies heavily on accurate event detection. Therefore, a fast and accurate event detection system is essential for the later part of the pipeline as the rest of the pipeline works on the detected events only. While it is straightforward to detect the higher

magnitude seismic events, lower magnitude events are harder to detect due to the low signal-to-noise ratio. Lower magnitude seismic events are important for a better understanding of the earth's structure near faults, volcanoes, and so on. Moreover, it is essential to detect lower magnitude events for seismic monitoring purposes as human-induced seismic activities are mostly lower magnitude. We develop a semi-supervised earthquake detection system (SeiSMo) which is able to detect undiscovered seismic events from static waveform. Our algorithm takes advantage of already discovered or known seismic events and creates a nearest neighbor graph using the known events.

We use four real-world and two geoscientific synthetic datasets to measure the performance of our system and find impressive results. SeiSMo is able to discover seismic events that are missed by human annotators and also missed by other state-of-the-art event detection systems. We observe that the detected events from SeiSMo have a very low signal-to-noise ratio compared to the catalog events. Also, SeiSMo can run in seconds for a million samples which makes it a proper choice for real-time detection.

1.2 Seismic Data Repository

We are living in the era of big data. The availability of various sensors and widespread use of IoT (internet of things) devices allows us to generate more data and seismology is no different. As seismic sensors are getting cheaper and more available, we have a larger number of denser seismic networks globally. To store and retrieve seismic waveforms including proper annotation is a challenge due to the scale of such operations. In this work, we design and implement a storage system to support the development of intelligent processing algorithms over seismic data from hundreds of seismic stations over ten years. The data is collected from the Air Force Research Laboratory. We discuss the organization of the database and access method for waveform data with examples. The system is currently in use to support a group of researchers in machine intelligence for *Nuclear Explosion Monitor-*

ing (MINEM) consortium[1].

1.3 A Machine Learned Depth Prediction System for Seismic Events

The depth of a seismic event is an essential feature to discriminate natural earthquakes from events induced or created by humans. However, estimating the depth of a seismic event with a sparse set of seismic stations is a daunting task, and there is no globally usable method. The existing methods to estimate seismic depth uses travel time equations and velocity models for specific geographic region. Such methods are physics-driven and require lots of human effort. Moreover, physics-driven methods are most accurate when there is a seismic station right above the origin on the earth's surface, and the error rate is proportional to the distance of the station from the epicenter.

In this work, we focus on developing a machine learning model (Septor) to accurately estimate the depth of arbitrary seismic events straight from seismic waveforms. Our proposed deep learning architecture is not-so-deep compared to commonly found models in the literature for related tasks, consisting of two loosely connected levels of neural networks, associated with the seismic stations at the higher level and the individual channels of a station at the lower level. Thus, the model has significant advantages, including a reduced number of parameters for tuning and better interpretability for geophysicists. We evaluate our solution on seismic data collected from SCEDC (Southern California Earthquake Data Center) catalog for regional events in California. The model achieved 86.5% F1-score in separating deep events from shallow ones. For prediction purposes, our model achieved 70.1% correlation between predicted and actual depth while having a root-mean-squared error of 2.89 km.

[1] ara.com

Chapter 2

Seismic Signal Detection using Semi-supervised Motif

2.1 Introduction

We model seismic signal detection as the data mining task called motif discovery from time series. Motif discovery from time series data is a well studied problem in data mining. The typical objective in motif discovery is to identify approximately repeating segments in a time series. Each pattern that repeats significantly, either with high number of occurrences or with high similarity among the occurrences, is called a motif. For example, in Figure 2.1, there are four occurrences of a sinusoidal motif.

Existing motif discovery algorithms perform unsupervised search for the most similar occurrences of a motif and, expand the motif set by identifying relatively fewer similar occurrences [36][67]. However, it may be possible that some occurrences of a motif are known. Such information can lead us to *semi-supervised* motif discovery algorithms, which would be *faster* than the completely unsupervised algorithms and, more *robust* than simple similarity search with a domain dependent parameter (i.e. distance radius for a radial search).

Semi-supervised motif discovery has the potential to enable data mining

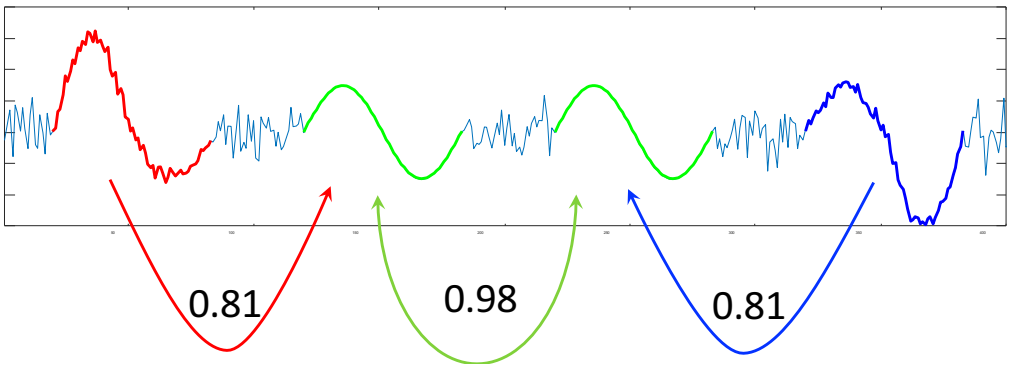

Figure 2.1: A time series with a sinusoidal motif appearing four times. The arrows below show the nearest neighbor and correlation coefficients. If the red(left) and blue(right) sinusoids are given as example occurrences of the motif, it is intuitive that pure/middle sinusoids should also be occurrences. Traditional unsupervised motif discovery algorithms can find the pure/middle sinusoids at a high computational cost, of quadratic time complexity. Moreover, knowing the middle/green sinusoids as motif does not help selecting threshold for further similarity search.

in domains where significant manual annotation exists already. In seismic data analysis, well curated catalogs of seismic events are maintained by IRIS [2], NCEDC [3], SCEDC [4] etc. Yet, many events of interest, especially low magnitude ones, are hidden in historical data that can produce valuable insights for geological scientists.

In this work, we propose a semi-supervised motif discovery algorithm. We name our technique **SeiSMo** (Semi-Supervised Motif). The algorithm performs an iterative nearest neighbor search to find other occurrences of the given motifs. SeiSMo has the following desirable properties:

- SeiSMo is an order of magnitude faster than unsupervised algorithms and can scale to millions of samples.
- SeiSMo works with any distance measure.

- SeiSMo is deterministic and less sensitive to parameters than supervised methods (i.e. radial search).
- SeiSMo can discover motifs with domain significance.

We develop two optimization strategies to speedup the computation of SeiSMo over a naive approach. We have applied SeiSMo on four seismological datasets from three states in the U.S. to discover numerous uncataloged seismic events that were missed by existing unsupervised techniques. SeiSMo is very promising in detecting *low magnitude* events which are not commonly cataloged by any manual effort.

2.1.1 Challenges in Semi-supervised Motif Discovery

One may consider simple similarity search within a threshold of the given motifs to identify the remaining occurrences of the motifs. However, depending on the quality of the given events, similarity search achieves mixed results.

Motif discovery is a special case of clustering where the goal is to identify a few clusters (typically less than five) of highly coherent subsequences in a time series, while leaving most of the data unclustered. In semi-supervised motif discovery, we define that one or more subsequences from one or more motifs are labeled; and the goal is to find all of the motifs. Most of the current work on semi-supervised algorithms for time series data focus on classification and clustering [8][13], in general. We provide an efficient semi-supervised motif discovery algorithm with applications to seismic data analysis.

To elaborate this idea, consider a set of points in two dimensional space in Figure 2.2(a). Note that time series segments are points in very high dimensional space. We consider two dimensional points for better illustration. The dense lump is a motif with *five* occurrences or repetitions. State-of-the-art motif discovery algorithms will find the closest pair of points (marked in red color) in Figure 2.2(a), and then search the region within a radius/threshold, typically proportional to the closest pair distance. In the Figure 2.2(a), all

of the five occurrences can be identified by existing techniques for the illustrated radius/threshold.

Now consider the same set of points, along with the star-marked points showing the given occurrences representing some physical events (e.g. earthquakes). In Figure 2.2(b), all of the points in the dense lump are closer to the stars, hence, they are very well detectable by similarity search [67] based on the same radius as in 2.2(a). In Figure 2.2(c), no point is closer to the stars, hence, no motif would be found. In Figure 2.2(d), two of the five points will be found by the same search radius. Thus, the number of similar points detected by similarity search depends on the position of the stars and the radius/threshold value.

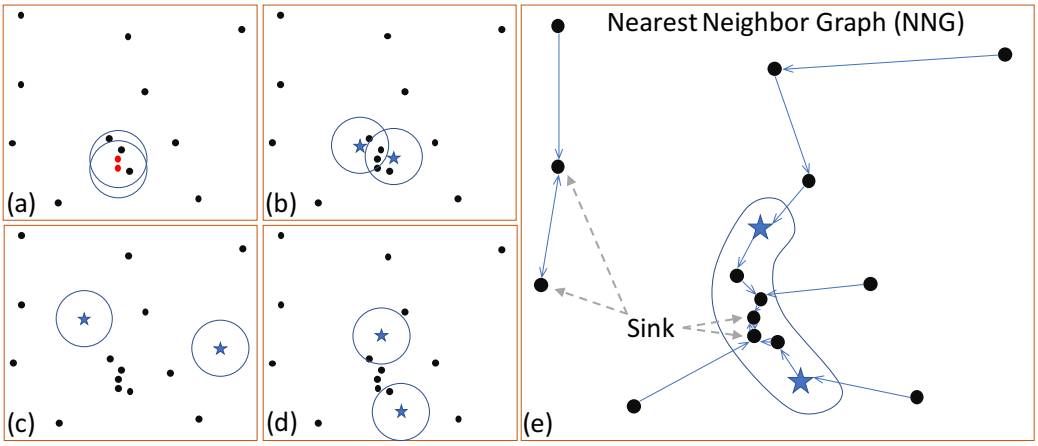

Figure 2.2: A set of two-dimensional points. Unsupervised radial search at the closest-pair (a). Successful (b), failed (c) and partially successful (d) similarity search at given (i.e., star) points. 2(e) is an enlarged version of 2(d). Nearest neighbor chains starting from the stars contains all the five points (e).

To draw a quick contrast and build an intuition, we enlarge Figure 2.2(d) in 2.2(e), where SeiSMo can find all five occurrences in all of the scenarios (b-d). An arrow in Figure 2.2(e) connects a point to its nearest neighbor. If we follow the chain of nearest neighbors starting the star points, we reach the

same pair of points, which we name *sink*, that are the nearest neighbors of each other. If we label all nodes on these chains as occurrences of the given motif, we correctly identify the five points without any threshold parameter.

2.2 Background and Notation

We define *time series* as a sequence of real numbers measuring a quantity at a fixed sampling rate. A *time series subsequence* is a continuous segment of a time series. We can extract $n - m + 1$ time series subsequences of length m from a long time series of length $n >> m$. Time series subsequences are not independent of each other. Overlapping subsequences are trivially close in the high dimensional space, however, they are not interesting for mining purposes.

To elaborate on this fact, in Figure 2.3, we show a toy time series and all the subsequences of length three as points in three dimensional space. The points are not independently scattered, instead, the points form a trail where the nearest neighbor of a point (i.e. a window) is the next point on the trail (i.e. next slide of the window). To avoid such trivial nearest neighbors, we define a threshold O_t (typically 50%) to set *the minimum overlap between two subsequences* required to identify them as trivial matches. In other words, if two subsequences share more than half of their observations, we consider them as trivial. Because almost all computational properties (moments, frequency distribution, etc.) of two half-overlapping subsequences are governed by the overlapping segments, and the lost information is at the Nyquist frequency. The choice of $O_t = 50\%$ has been empirically validated in related work in the literature [67][69]. In Figure 2.3(right), we connect a point with its non-overlapping ($< O_t$) nearest neighbor. Clearly, the nearest neighbors now contain more information, hence, non-trivial.

Nearest Neighbor (NN): Given a subsequence $S_{i,m}$, that starts at i^{th} index in the time series and of length m, we define the nearest neighbor of $S_{i,m}$ as below,

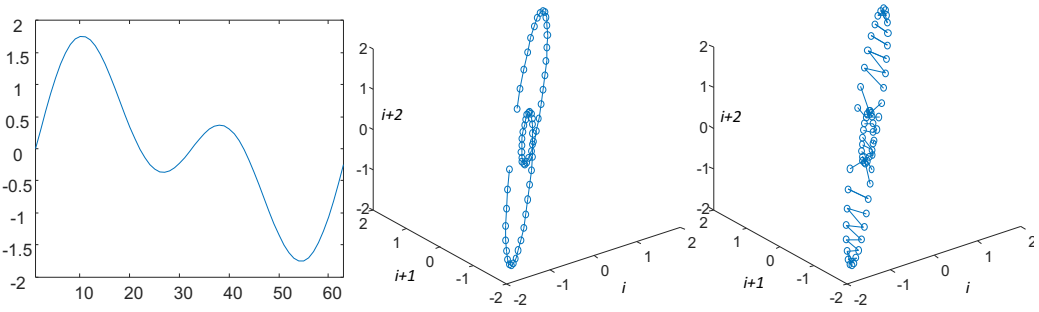

Figure 2.3: (left) A toy time series. (middle) The subsequences of length three in a 3D space form a trail. The nearest neighbor of a point/subsequence is trivially the next point on the trail. (right) The nearest non-overlapping neighbors are not trivial and, can possibly be in anywhere on the trail.

$$\min_{|j-i|\geq m} d(zNorm(S_{i,m}), zNorm(S_{j,m})) \tag{2.1}$$

Here, d can be any distance function defined to calculate distance between two equi-length time series. Typical distance functions are Euclidean distance [37], Dynamic Time Warping distance [44], Longest Common Subsequences distance [56], and Move-Split-Merge distance [52]. $zNorm$ is the standard z-normalization defined as $zNorm(x) = \frac{x-\mu}{\sigma}$, where μ and σ are estimates of mean and standard deviation of the observations in the vector x. Searching for the nearest neighbor of a query subsequence is extensively studied under all of these distance measures. In this paper we search for nearest neighbors under Euclidean distance using MASS [37] and DTW using UCR_Suite [44].

Nearest Neighbor Graph (NNG): An NNG is a graph (V, E) where V is the set of all subsequences of length m from a time series of length n. $(u \rightarrow v) \in E$ if v is the nearest neighbor of u as defined above. The *length* of an edge $(u \rightarrow v) \in E$ is the z-normalized distance between u and v.

Properties of NNG:

- The NNG is a collection of paths.

- Only cycles of size two are possible, which are named as *sink* in this paper.
- The edges along the paths are non-increasing in lengths. Every path ends in a *sink*.

The above properties of an NNG are invariant to the distance measure used to find the nearest neighbors. Multiple unique paths can lead to a sink. We define such paths as **Confocal Paths** highlighting the fact that they all end in the same sink. We define the **support** of a sink, P, by the number of given occurrences of a motif (i.e. star points in Figure 2.2 and Figure 2.4) on any confocal path leading to the sink. The support of the sinks in Figure 2.2(e) are zero and two. See Figure 2.4 for more examples.

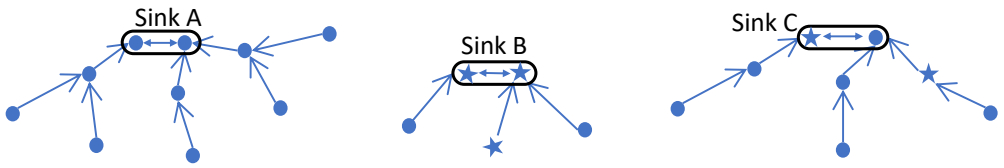

Figure 2.4: Three sinks (A,B and C) and their sets of Confocal Paths. The supports of A,B and C are zero, three and two, respectively

The intuition behind SeiSMo is that, if there are several paths from several labeled nodes/subsequences that lead to the same sink, the likelihood of the paths containing more unlabeled nodes (i.e., motif) is high. The reason is that the distances are progressively shorter as we follow paths from the labeled nodes/subsequences towards the sink. Since every node leads to a sink, we do not consider sinks with support $P = 1$. We only label sinks with $P \geq 2$ and, all nodes on paths from the labeled nodes to that sink. The enclosed region in Figure 2.2(e) shows the labeled nodes by SeiSMo. The advantage of this method is that there is no domain dependent parameter to tune. One may simply perform more strict motif discovery by choosing a higher threshold of support, e.g. $P \geq 3$. Note that, the value of P is a

design choice for the practitioners, and not a parameter to the algorithm. In Section 6, we analyze the sensitivity to this parameter.

2.3 Related Work

Searching for the nearest neighbor of a query subsequence is done under Euclidean [37] or Dynamic Time Warping [44] distance. Both of these techniques are efficient and suitable for our algorithmic optimizations. SeiSMo is not limited to any specific distance measure, moreover an ensembling effect is observed when we combine outputs from SeiSMo using Euclidean and DTW distances.

Existing time series motif discovery algorithms are all unsupervised [36] [34] [33], focusing on various operational aspects of the tasks such as exact motifs [36], online motifs [34], variable length motifs [33], multi-dimensional motifs [66] and rare motifs [9]. Domain knowledge has been incorporated in motif discovery in the form of annotation vectors [16]; however, such annotation denotes preferred regions of the time series to guide the search for motifs, as opposed to providing instances/occurrences of a motif.

Semi-supervised approaches have been developed for time series classification extensively [59][13]. Classification tasks propagate labels to all unlabeled instances of the training data, while semi-supervised motif discovery algorithms are not required to label all subsequences of a time series. Time series classification, in general, works a on set of independent time series. SeiSMo works on subsequences of a long time series. In Section 6, we apply semi-supervised classification to detect hidden motifs to demonstrate the difference.

The proposed method is very relevant to classic density based techniques for unsupervised clustering (e.g., DBSCAN [20]) and outlier detection (e.g., Local Outlier Factor [11]). However, the concept of density connectedness has never been applied to motif discovery. To be more specific, SeiSMo has prefixed the two parameters in those algorithms, $minPts(=1)$ and $\epsilon(=\infty)$,

that most density based techniques require tuning.

NNG on independent data points have been studied in computational geometry for decades [43]. However, our problem definition requires careful handling of overlapping subsequences in a time series to avoid trivial matches, and our optimization techniques greatly improve the construction of NNG on high dimensional data.

One important point of distinction is worth noting. Motif discovery from time series data is fundamentally different from motif discovery from discrete sequences such as DNA or protein sequences. To elaborate on the distinction, consider the query matching problem. Exactly matching a query string takes $O(n)$ time using the KMP (Knuth-Morris-Pratt) algorithm. Exact matching is not defined for real sequences, while approximate matching takes at least $O(n \log n)$ time [37]. Similarly, motif discovery from time series data faces several challenges because of normalization, trivial matches and diverse distance functions, while motif discovery from discrete sequence data enjoys simple match/no-match relationships between observations.

A recent practice in Seismology is to apply machine learning algorithms to learn from vast amount of labeled data collected over many decades. As shown later in this paper, SeiSMo can detect events that well-trained deep learning models cannot detect[42]. For the sake of argument, if we assume the nearest neighbor search for all subsequences can be modeled by a convolutional and a pooling layer, the recursive search for the nearest neighbors in SeiSMo would need variable number of layers. Although, with sufficient large number of layers and units in each layer, it should be possible to model SeiSMo , however, that is beyond the purpose of this paper.

2.4 SeiSMo : Semi-supervised Motif

In Algorithm 1, we describe SeiSMo . The algorithm requires the positions of the known/seed subsequences in the time series as input. The output of the algorithm is a set of positions of newly detected events. We assume there is

a maximum length for all the given and output occurrences of motifs. Such a length, that can contain most events, can be identified in most domains. For example, in seismology, 10-second window is generally enough for the mine blasts, and 20-second window is good for earthquakes.

The algorithm computes the nearest neighbors (NN), recursively, from each seed subsequence (Line 3-9). The iterative search ends when the current subsequence and the nearest of the nearest neighbor subsequence are more than O_t overlapped. Thus, we identify sinks using overlapping subsequences, as opposed to matching the exact locations of a subsequence. This is a necessary deviation from an exact location-based sink identification. To explain why, consider $S_{1,100}$. If the nearest neighbor of $S_{1,100}$ is $S_{501,100}$ and the nearest neighbor of $S_{501,100}$ is $S_{2,100}$, the algorithm will terminate iterating at Line 5, which would not be the case if the algorithm checked for $NN(NN(currentNode) = currentNode$ in Line 5.

Since the algorithm stops on a condition that uses overlap between subsequences, our algorithm is dependent on the order of the data as many other data mining algorithms are [45][35]. In addition, in SeiSMo, several sinks with large overlaps are considered as one sink. All of the sinks reachable from all of the seed events are collected in a list, $SinkList$.

The second half of the algorithm counts the *support* of each sink (Line 10-13). Again, two sinks, (u, v) and (p, q), are considered the same or equivalent (i.e. $(u, v) \equiv (p, q)$) if overlap between p and u, and q and v, <u>both</u> are more than O_t, without loosing generality. The sink equivalence relation has the following properties.

- If (u, v) is a sink, then u and v are guaranteed to be non-overlapping by the NN operation. The order of u and v can be swapped without loosing generality.
- sink equivalence is not transitive; $(u, v) \equiv (p, q)$ and $(u, v) \equiv (s, t)$ do not entail $(u, v) \equiv (s, t)$.
- Sink equivalence is symmetric. $(u, v) \equiv (p, q)$ entails $(p, q) \equiv (u, v)$.

Since we count support for one sink at a time, the lack of transitivity does not impact the counting process because of the symmetric property. Once high enough support is gathered, all nodes on the path from any seed event to the highly supported sink are added to the set of $newEvents$.

Algorithm 1 $SeiSMo(TS, SP)$

Require: $TS \leftarrow$ a continuous time-series, $SP \leftarrow$ positions of seed events

Ensure: Output newly detected seismic events from the time-series TS using the given seeds SP

1: **for** every seed $s \in SP$ **do**
2: $currentNode \leftarrow s$
3: $neighborNode \leftarrow \text{NN}(currentNode)$
4: $neighborOfNeighbor \leftarrow \text{NN}(neighborNode)$
5: **while** Overlap between $neighborOfNeighbor$ and $currentNode < O_t$ **do**
6: $currentNode \leftarrow neighborNode$
7: $neighborNode \leftarrow neighborOfNeighbor$
8: $neighborOfNeighbor \leftarrow \text{NN}(neighborNode)$
9: Insert $(currentNode, neighborNode)$ in the $SinkList$
10: **for** each $sink$ in $SinkList$ **do**
11: Count $support$ of $sink$
12: **if** $support$ of $sink \geq 2$ **then**
13: $newEvents \leftarrow newEvents \cup$ Nodes from all seeds to $sink$
14: deduplicate $newEvents$ using O_t
15: **return** $newEvents$

In the final stage, SeiSMo performs a **deduplication** operation to ensure non-overlapping events are output (Line 14). The deduplication operation is a self-join of the set of $newEvents$ on overlap more than O_t. One may consider a more restrictive overlap threshold to limit the number of output patterns, however, we use the same overlap threshold O_t for consistency.

The Algorithm 1 does not require the seed events to be identical in length.

If seeds are of different lengths, their chains of nearest neighbors will be of different lengths. When counting support, the algorithm considers O_t along with the length of the longer time series. More precisely, $(u, v) \equiv (p, q)$ if overlap between p and u, and q and v, both are more than $\lceil O_t m \rceil$, where m is the maximum of the lengths of u, v, p and q.

2.5 Optimizations to SeiSMo

The most time critical part of SeiSMo is searching for the nearest neighbors in iterative manner. The properties of NNG allow us to develop two optimization techniques to speedup the search process.

Recursive search initialization: The Nearest neighbor search algorithm [44] uses an initial *best-so-far* value to begin the search. If no prior knowledge exists, *best-so-far* is set to ∞. Since the distances along a path on NNG are always non-increasing, we can initialize the search in Line 8, by the last nearest neighbor distance on the path. More precisely, $neighborOfNeighbor \leftarrow NN(neighborNode, d(currentNode, neighborNode))$ This helps the early-abandoning process greatly to stop calculating unpromising distances.

Path pruning: The in-degree of a node on NNG can be zero or more; however, the out-degree of a node is exactly one. If a nearest neighbor discovered in Line 8 is a repeated discovery; we can prune the path by breaking the loop in Line 5. At the implementation level, we keep a hash of all visited nodes on NNG and check for repeated visit in constant time. This approach ensures no node is visited more than once. Note that SeiSMo must increase the support of the sink the path was heading to before pruning any path. This requires some additional bookkeeping to maintain the support count for each sink dynamically.

2.5.1 Complexity of SeiSMo

Time complexity of SeiSMo algorithm is dominated by the iterative nearest neighbor search. If we use DTW distance, the cost of an NN search is $O(nm^2)$ where n is the length of the time series and m is the length of the motifs. The number of NN searchers is proportionate to the number of seeds S and, lengths of the paths from seed to sink, P. Hence, the total complexity is $O(SPnm^2)$. In most applications, S and P are on the order of tens, while n is in millions. Hence, the complexity of the algorithm is largely dominated by the length of the time series.

Our pruning approaches do not reduce the asymptotic complexity of the algorithm. However, recursive search initialization reduces effective value n and path pruning reduces effective value of P. The memory requirement for these optimization techniques is proportional to the number of sinks and average path size. These quantities are much smaller than the data size (n), hence, the algorithm is effectively in-situ.

2.6 Experimental Evaluation

All our experiments are reproducible. The code, data, spreadsheet and figures are available in [5]. We use six datasets for experiments including four real, one synthetic, and one simulated datasets:

- ***Synthetic*** white noise with implanted sinusoids containing 100,000 observations.
- ***California*** seismograms from station NC.CCOB between January 8, 2011 (00:00:00) and January 15, 2011 (00:00:00). The dataset contains 12,096,000 observations recorded at the rate of 20 samples per second.
- ***Oklahoma*** seismograms from station GS.OK029 between April 1, 2014 (00:00:00) and April 8, 2014 (00:00:00). The dataset contains 12,096,000 observations recorded at the rate of 20 samples per second.

- ***Wyoming*** seismograms from station ZH.RPCE between July 19, 2010 (00:00:00) and July 21, 2010 (00:00:00). The dataset contains 3,456,000 observations recorded at the rate of 20 samples per second.

- ***LabQuake*** or *LabQ* seismic data [62] generated by a frictional experiment in the Pennsylvania State University Rock and Sediment Mechanics Laboratory. This dataset is very well labeled and specifically suited for precision and recall analysis because of the known ground truth. Also, this data is suited for testing precision and recall on different SNR values. The dataset is publicly available in [1]. The dataset contains 125,000 observations.

- ***Southern California*** seismograms from station CI.DRE between April 3, 2010 (00:00:00) and April 10, 2010 (00:00:00). The dataset contains 12,096,000 observations recorded at the rate of 20 samples per second.

We define **Precision** and **Recall** for motif discovery in the following way. Let us assume the number of occurrences of a motif is N. Let us assume a motif discovery algorithm finds Q occurrences of the motif and, P of them are from the set of known occurrences. The *precision* is $\frac{P}{Q}$ and *recall* is $\frac{P}{N}$. SNR is a measure of signal strength relative to background noise. Our objective is to empirically evaluate precision and recall of our method using the same seismogram modified in such a way that each version of seismogram has a different SNR value.

Precision is our primary metric of comparison because false detection costs valuable human-hours for further filtering. In contrast, there is no ground truth about the total number of undetected events in a seismogram. Therefore, recall calculation is not practical on real world datasets.

2.6.1 Sanity Check

We generate a synthetic dataset by implanting sinusoids of length 200 in a series of 100K white noise samples. The variances of the sinusoids are 10%

of the variances of the noise. We use random 40% of the sinusoids as seed events. We vary the number of implants from 64 to 1024 by iterative doubling, and measure the precision and recall of motif detection. Our precision is always **100%** on synthetic data. This is reassuring, and not surprising, as motif discovery algorithms are precision-focused by construction. SeiSMo achieved a recall rate of 35.02% on average, with 3.51% standard deviation. To improve the recall rate, one may relax the algorithm further by using a minimum support of one. However, on real datasets, the impact on precision can be significant.

2.6.2 Comparison

We compare SeiSMo with a naive semi-supervised motif discovery algorithm based on similarity search within a radius (RS). We set the search radius equivalent to 80% Pearson's correlation coefficient [67]. We compare the methods on two synthetic datasets: *Synthetic* and *LabQuake* datasets. Both of these datasets have manually inserted events enabling us to calculate precision and recall. The seeds are all of length 400 for this experiment.

Quantitative comparison to similarity search within a radius (RS) is unfair in the sense that there is a difference in degree of freedom. RS can perform anywhere between very good to the worst. In contrast, SeiSMo has no degree of freedom, it is exact and deterministic. The output produced by SeiSMo are robust and significant. Similarity search within a threshold is very useful in interactive data mining, however, for the purpose of motif (i.e. hidden event) discovery SeiSMo off load the human involvement in the mining process by eliminating the threshold parameter.

On simple sinusoidal synthetic data, both RS and SeiSMo achieve perfect precision, while RS achieves higher recall for a search threshold of 0.87. Note that sinusoidal data is the best case for motif discovery. In contrast, on *LabQuake* data, SeiSMo performs much better than radial search in precision and recall. Although simulated, *LabQuake* events contains similar noise level as real earthquake signals contain.

Table 2.1: There is no ground truth for seismic datasets, because the complete set of underground events is generally not know. Hence, we show precision and recall of different methods on simulated and synthetic datasets below.

Dataset	**LabQuake**	**Synthetic**
Length	125K	100K
Number of Seed Events	82	59
Total Number of Events	130	150
SeiSMo - New Detection	33	36
RS - New Detection	0	88
SeiSMo - Precision	96.97%	100%
RS - Precision	0%	100%
SeiSMo - Recall	66.67%	39.56%
RS - Recall	0%	96.70%

2.6.3 Comparison with Existing Technique

Comparison to methods in seismology

Although SeiSMo is more robust that naive radial search, we must compare SeiSMo with existing earthquake detection algorithms from the seismology community. Two of our datasets, *California* and *Oklahoma* are taken from existing work on hidden event discovery. We compare to both of them to check if SeiSMo adds any novel detection. The results are shown in Figure 2.5. All of our novel detections are manually verified.

In *California* data, we observe that SeiSMo adds few more (18) unique events to the events detected by existing technique (ConvnetQuake), while radial search is adding no value. We show the events in the subsequent sections. However, the fact that SeiSMo is precisely detecting new events in two independent real datasets from two states is showing the generalizability of our technique.

In *Oklahoma* data, we observe an interesting fact that all methods are

detecting large number of unique events. This suggests that the dataset is rich in number of events and, the methods are capturing unique aspects of the events to be able to work collaboratively.

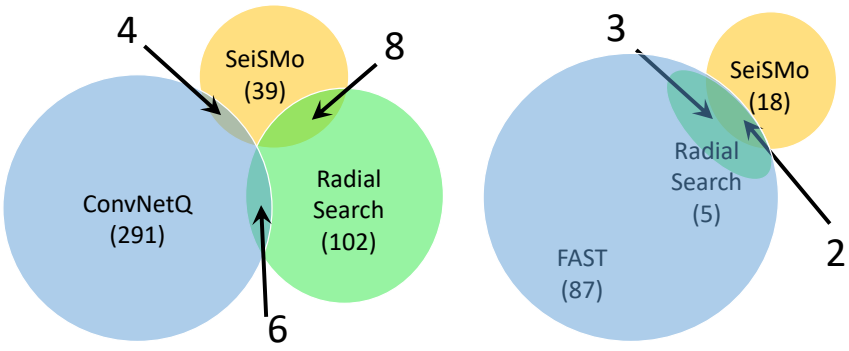

Figure 2.5: (left) Comparison to ConvNetQuake [42] and similarity search within a radius (RS) on Oklahoma dataset. (right) Comparison to FAST [68] and similarity search within a radius (RS) on California dataset.

Comparison to semi-supervised data mining method

Historically, semi-supervised methods have been developed to perform supervised learning in the absence of labeling. SeiSMo focuses on exploiting labels in performing unsupervised motif discovery. Although this key difference makes SeiSMo unique, we consider existing semi-supervised learning algorithm for time series data, DTW-D [13], to detect hidden events.

Our approach is to train a semi-supervised classifier on the seed events (i.e. control) and on a mix of seed and SeiSMo events (i.e. treatment). In both of the scenarios, we consider the seismic events as positive instances and add equal number of non-event segments as negative instances to balance the datasets. If the SeiSMo events are all detectable by DTW-D classifier, we expect to see an increase in classifier performance. If the SeiSMo events contain same amount of confusion as the seed events, the classifier accuracy should stay the same.

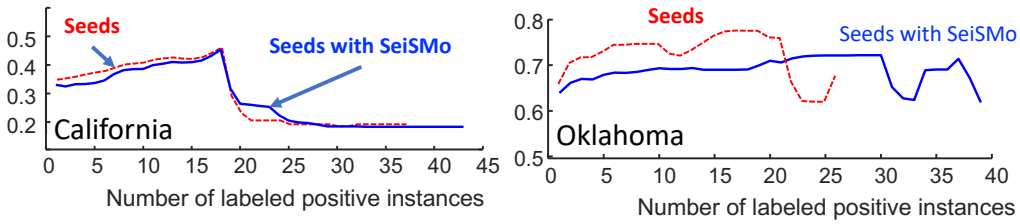

Figure 2.6: Applying semi-supervised classification (DTW-D) algorithm to detect new events.

In Figure 2.6, we show the results on two of our datasets: California and Oklahoma. We see no significant difference between control and treatment on California dataset, which ideally implies that DTW-D classifier finds SeiSMo events very similar to the seed events. However, the accuracy of the DTW-D classifier is at most equal to the default accuracy (50%) on California dataset. Therefore, no conclusion can be made for this dataset. We see a significant decrease in accuracy once we add SeiSMo events to the seed events in the training dataset (Figure 2.6(right)). We conclude that DTW-D failed to learn about the unique SeiSMo events.

Note that DTW-D is not an event discovery algorithm. Hence, our demonstration of DTW-D classifier failing in event detection should not undermine its goodness as a classifier.

2.6.4 Efficiency

SeiSMo recursively searches for the nearest neighbors. The number of searches is generally more than the number of searches in radial search based methods. In Figure 2.7 (left), we show the execution time in seconds as we increase the data size upto a million observations. We use the synthetic data with 14 seeds. The optimization techniques give us roughly **2**$\times$ speedup. The growth in execution time is roughly linear, suggesting scalability to even hundreds of millions of observations.

Unsupervised motif discovery algorithms using recent Matrix Profile tech-

nique [67] requires an order of magnitude more time to identify the seed motif and, expand to more occurrences. To find motifs in a 10^6 long time series, Matrix Profile runs for a day without any GPU, and for a shorter window size of 256 (see Table III of [67]); while SeiSMo finishes in 25 seconds.

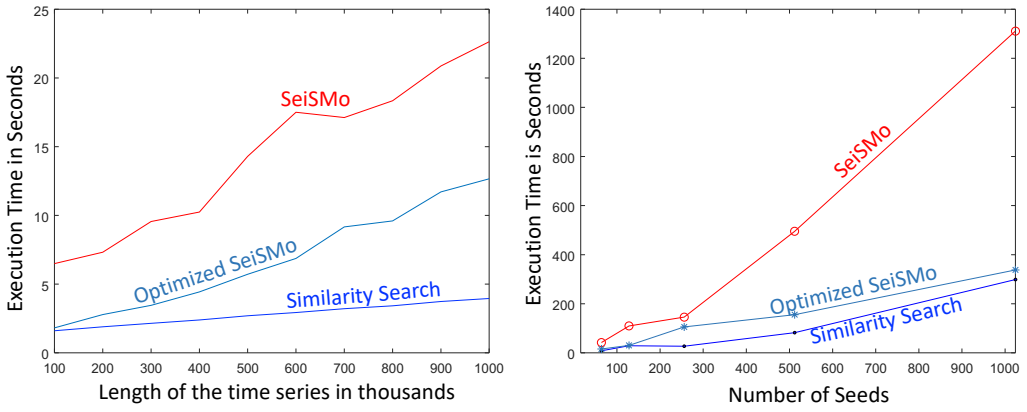

Figure 2.7: (left) Linear scalability and speedup by Optimization techniques. (right) Scalability with respect to number of seeds.

Execution speed grows with number of seed events. Figure 2.7(right) shows the change in execution time as we implant more events and use more seeds. Similarity Search within a radius is generally faster, however, the optimized SeiSMo grows almost linearly with respect to n, without any sensitive parameter such as the radius.

2.6.5 Parameter Sensitivity

We have one parameter in SeiSMo. The support of the sink, P, is set to 2 for all of the experiments in this paper. However, to develop a reasoning about this parameter, we perform an experiment by varying the parameter P, and recording the precision and recall. In the Figure 2.8(right), we show the result. For both synthetic and *LabQuake* (LabQ) data, precision is generally insensitive to increasing support, while recall decreases with increasing support. This is intuitive because, increasing support restricts SeiSMo to

detect only patterns with high confidence. Since precision is the key metric for automated signal detection, we claim insensitivity to the parameter P.

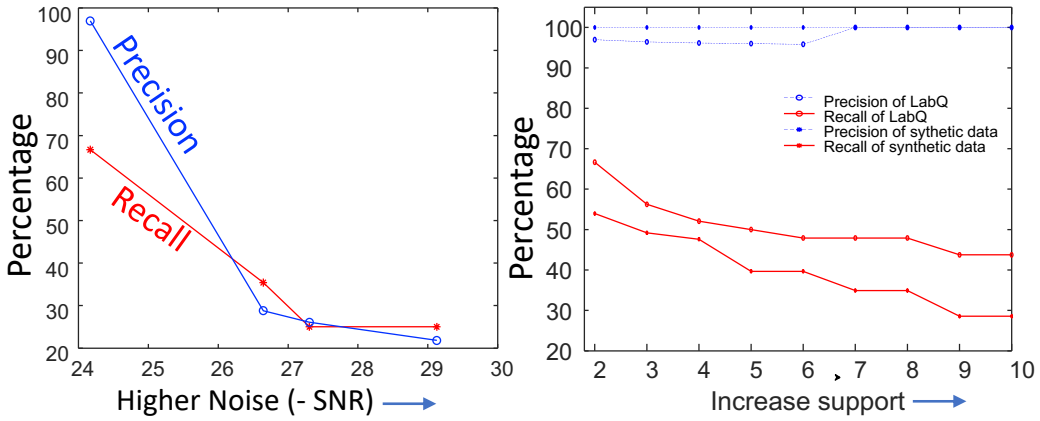

Figure 2.8: Parameter sensitivity of SeiSMo . (left) Precision and recall graph of SeiSMo with different noise level. Both precision and recall decreases when noise level is increased. (right) Precision and recall for varying support count, P.

To evaluate the effect of noise, we incrementally add noise to *LabQuake* data and measure precision and recall. The results are shown in Figure 2.8(left). We see a drop in precision and recall when signal-to-noise-ratio (SNR), measured in decibel, is decreased. Decibel is a logarithmic scale, hence, we attribute the drop in precision and recall as slow decay.

2.6.6 Effect of Distance Measure

We find that Euclidean and Dynamic Time Warping distances can produce unique hidden events from the same data. Hence, we suggest parallel execution of SeiSMo under all available distance measures and taking a union of the outputs as final output. We empirically observe, on average in all our experiments, DTW contributes 60% to the newly detected events, while Euclidean distance contributes 40%. We hypothesize that more distance

measures will add more detections at a diminishing rate. We leave it for future work to validate the hypothesis.

2.6.7 Evaluation

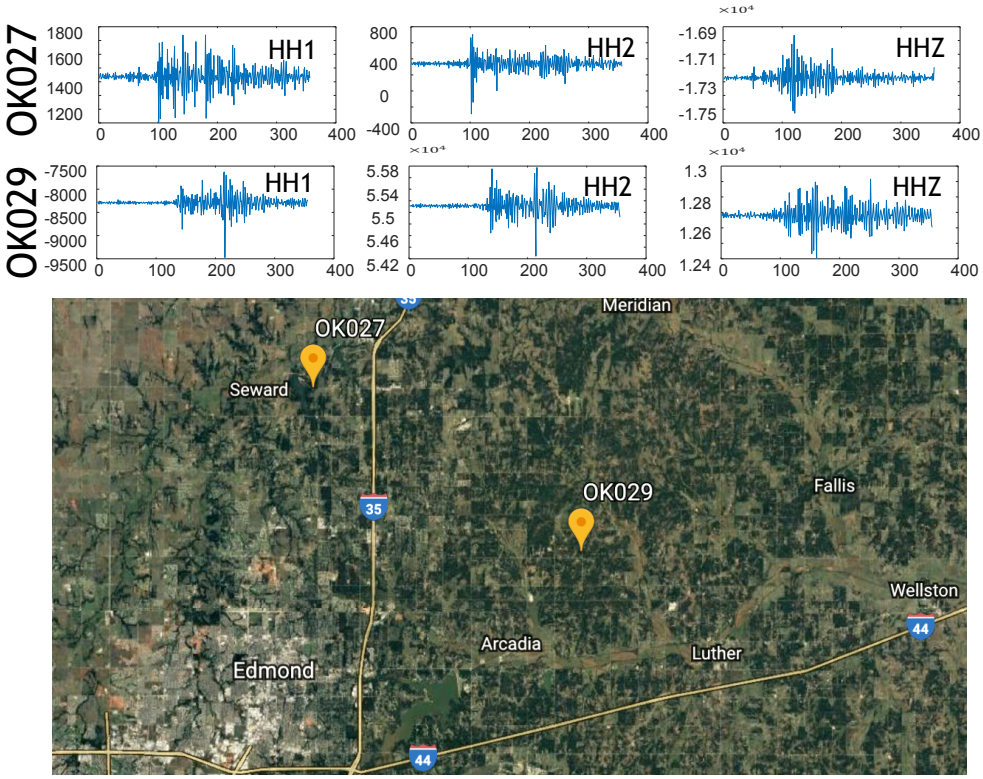

Figure 2.9: Example of validation using additional channels from the same and additional station. SeiSMo detects the signal in OK029-HH1. In this figure, We have other components (HH2, HH3) of OK029 station on first row. Moreover, we picked another station, OK027 which is 13 miles apart from OK029 (exact position is shown in the map). We have shown waveforms from all three components from OK027 in the second raw. We observe the presence of the event in all six waveforms, thus, validating the detection.

It is very important to ensure that the events we detect are not caused

by any electrical/mechanical malfunction; or by any minor incident (i.e. a vehicle passing by). We can ensure this by examining the signals from all three (horizontal North-South, horizontal East-West vertical) components, and from all of the components in nearby stations. Therefore, once detected, we evaluate if the detected signal is a real seismic event by visualizing the other channels/components in the same stations and channels in nearby stations. If more than one stations record the same event within reasonable delay, we confirm the signal as a real event. To illustrate, in Figure 2.9, we show the detected signal in the HH1 channel of OK029 station. We see that other two channels, HH2 and HHZ, depict strong signals. To further validate, we probe a nearby station, in this case OK027, to visualize the channels at the same time window. We see presence of strong signals in all channels of OK027, which confirm that the detected signal represents a real event. Note that the signals in OK027 are consistently ahead of the signals in OK029 for all channels. This further validates that the seismic source is closer to OK029.

The challenges associated with such visual validation technique are absence of all the channels and absence of a nearby station. If none available, we pessimistically consider the signal as false detection.

2.7 Natural Seismic Events in California

2.7.1 Northern California Seismic Network

We present a motivating case study to demonstrate the utility of semi-supervised motif discovery from time series data. Seismic events are observable in seismographs, especially when the events are of high magnitude. However, low magnitude events are more frequent than higher ones, and often escape expert attention. Such events are not found in IRIS-hosted [1] catalogs while they are important for various geo-scientific analysis.

[1](Incorporated Research Institutions for Seismology)

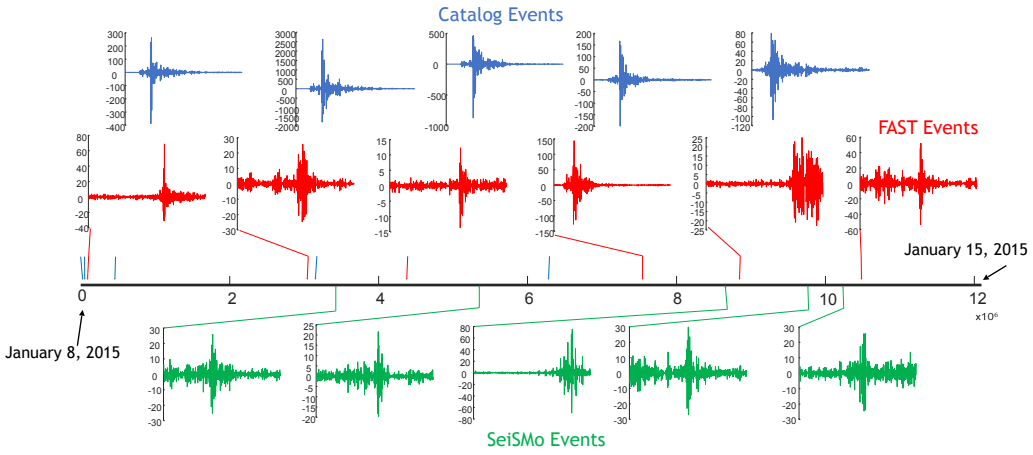

Figure 2.10: Randomly picked results of California dataset. Top row of events is cataloged. Middle row of events is detected by FAST [68]. And bottom row of events is detected by SeiSMo .

We apply our SeiSMo algorithm to discover hidden seismic events using existing cataloged events as labeled data. The Calaveras Fault in central California is known to have repeating earthquakes [47]. We have collected seven days (from January 8, 2011 to January 15, 2011) of single channel (horizontal north-south component) seismic data from a station (NC.CCOB) in Northern California Seismic Network from Northern California Earthquake Data Center (NCEDC) [3]. We have collected 24 cataloged events originated within 100 miles of the station. FAST [68] is a hash based method, which, in principle, is a radial search technique under a specific distance measure. We have augmented the catalog events with 87 events detected by FAST to create the seed set. Seismograms for each event were 20 seconds long which is good enough to fit a single event and had a sampling rate of 20 per samples second. We applied a 4- to 10-Hz bandpass filter to remove any unwanted noise. We have discovered 18 new events that have neither been cataloged in IRIS nor been detected by FAST. We use only one channel (EHN) of data for detection. The events we have discovered are all confirmed by manually checking the other two channels as well as other nearby stations.

In Figure 2.10, we show some random events for further discussion. All the events show presence of significant signals. The catalog events are generally high magnitude events. FAST detects higher magnitude events than those SeiSMo detects. Northern California is a very active region. Automated seismic signal discovery is the first step towards data driven understanding of seismic activity over time and space. Thus, SeiSMo contributes significantly towards that objective.

2.7.2 Southern California Seismic Network

Southern California Seismic Network (SCSN) recorded a good number of seismic activities after a 7.2 magnitude earthquake at Baja California, Mexico in April, 2010. We apply SeiSMo algorithm to discover hidden seismic events in a dataset collected from Southern California Seismic Network using existing cataloged events as labeled data. We have collected seven days (from April 3, 2010 (00:00:00) to April 10, 2010 (00:00:00)) of single channel (vertical component, BHZ) seismic data from a station (CI.DRE) in Southern California Seismic Network from Southern California Earthquake Data Center (SCEDC) [4]. Unlike the *Northern California* experiment in the previous section, we perform a study on the effect of event magnitudes in the seed set using this dataset.

Table 2.2: Increasing performance with seed quality.

Maximum Magnitude	4.10	4.20	4.40	7.20
Minimum Magnitude	0.30	1.66	2.50	4.10
Number of Detection	2	17	30	35

We have collected 60 cataloged events recorded at this station with a range of magnitudes from 0.3 to 7.2 in Richter scale. We order the events in increasing order of magnitudes and take sets of 30 consecutive seeds from the ordered list of events. For each of these sets of seeds, we run SeiSMo to detect new events and chart the performance in Table 2.2. We see an increase

in detection performance as the magnitude of the seed events increases. High magnitude events show better contrast with respect to the baselines around them compared to low magnitude events. This corroborates to the observations in Table 2.2.

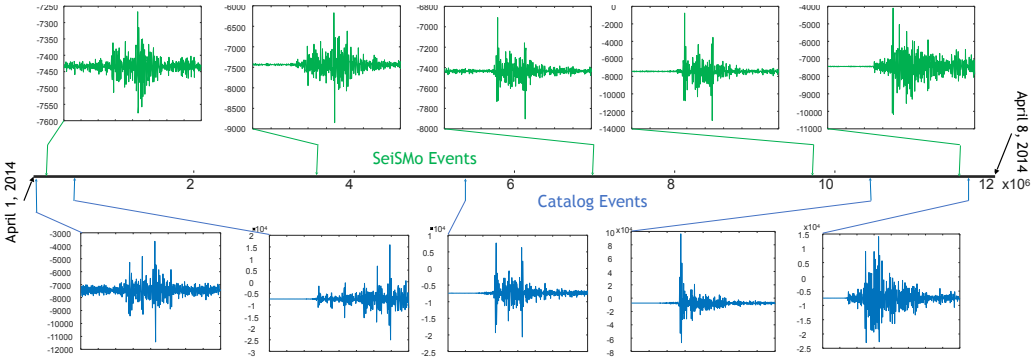

Figure 2.11: Results of Oklahoma dataset shows that, SeiSMo detects lower magnitude events than those in the IRIS catalog.

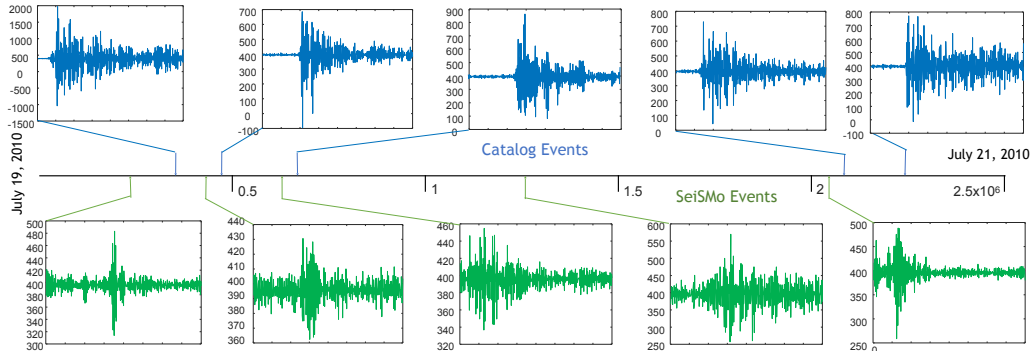

Figure 2.12: Some randomly picked events from the Wyoming dataset results. Top row of events is known. Bottom row of events is detected by SeiSMo .

2.8 Induced Seismic Events in Oklahoma

Human-induced seismicity includes seismic activities that are not directly initiated by humans (e.g. nuclear tests), rather are indirect outcome of human activity. Such as seismic activity because of waste water injection [58] [60]. In ConvNetQuake [42], authors have identified a recent increase in induced seismicity in Guthrie county of central Oklahoma.

We have applied SeiSMo to a single channel (HH1) dataset from the same station and for the same time duration of 7 days (April 1, 2014 (00:00:00) to April 8, 2014 (00:00:00)). We have collected 77 seed seismograms from IRIS [2]; each of the those was 20 seconds long with a sampling rate of 20 samples per second, just like the previous experiment. Also, we applied a 4- to 10-Hz bandpass filter to remove any unwanted noise. We run SeiSMo on this dataset and using these seeds we detect 39 more events. All of these 39 events are confirmed by manually checking all of the channels and few neighboring stations. Figure 2.11 shows five random cataloged and newly detected events on a time line. The cataloged events are higher in magnitude than those detected by SeiSMo .

2.9 Seismicity Due to Controlled Explosion in Wyoming

The mining industry in Wyoming uses borehole mining technology [61], which creates very low energy yet significant seismic footprint in nearby stations across the Bighorn Mountain [65] [41]. We collected a dataset from the experiment of [40], which contains 24 active source borehole events that were labeled by the authors. We used 16 borehole shot events in two consecutive days among those 24 as seed to explore if there are hidden/unlabeled borehole events. SeiSMo identified 12 new events that are confirmed to be borehole shots. Randomly picked SeiSMo detections and seeds are shown in Figure 2.12.

Consistent with the previous cases, the detected events are significantly lower in magnitude compared to catalog events. This suggest that SeiSMo is capable of discovering hidden low-magnitude seismic events.

2.10 Conclusion and Future Work

We propose a semi-supervised motif discovery algorithm to discover motifs or hidden events in a time series when a small number of arbitrarily similar events are known apriori. Using six different datasets, We show that our method is more accurate and robust than a naive similarity search on large datasets. We apply the algorithm on four seismic datasets and discover novel seismic events that were unknown to experts. All of our code, data and result used in this article are available [5] for reproducibility.

SeiSMo is good at spotting unique low magnitude events, which were not cataloged because analysts were not certain about their origin and characteristics. We plan on analyzing SeiSMo events to produce high quality information towards a fully automated cataloging system.

Chapter 3

UNM Seismic Data Repository

3.1 Introduction

A robust and faultless seismic data repository is a must to study historical seismic activity. While there exist numerous number of seismic networks, only a handful of them have precise accuracy and follow specific standards. As a result, analyzing historical seismic data becomes unreliable due to lack of proper ground truth. Moreover, almost all of the seismic networks have a local coverage which makes is more difficult to compare and datasets across regions. In this work, we create a global repository of seismic activity that are well-curated by domain experts and follow same standards for all regions globally. The seismic waveforms and labels are collected by International Data Center (IDC) using 127 seismic arrays over 10 years. We create *three* MySQL databases to store the bulletin which allows fast searches for complex queries. In short, the seismic bulletin contains:

- 650 thousand seismic events
- 13.2 million Seismic arrivals
- Collected from 127 global seismic arrays

3.2 Repository Description

Our UNM seismic data repository consists of *three* main storage unites: *(i) MySQL databases, (ii) wfdisk text files and (iii) waveform binary files.* At the center of the repository, we have *Data Generator* module which combines all storage units and communicates with the users.

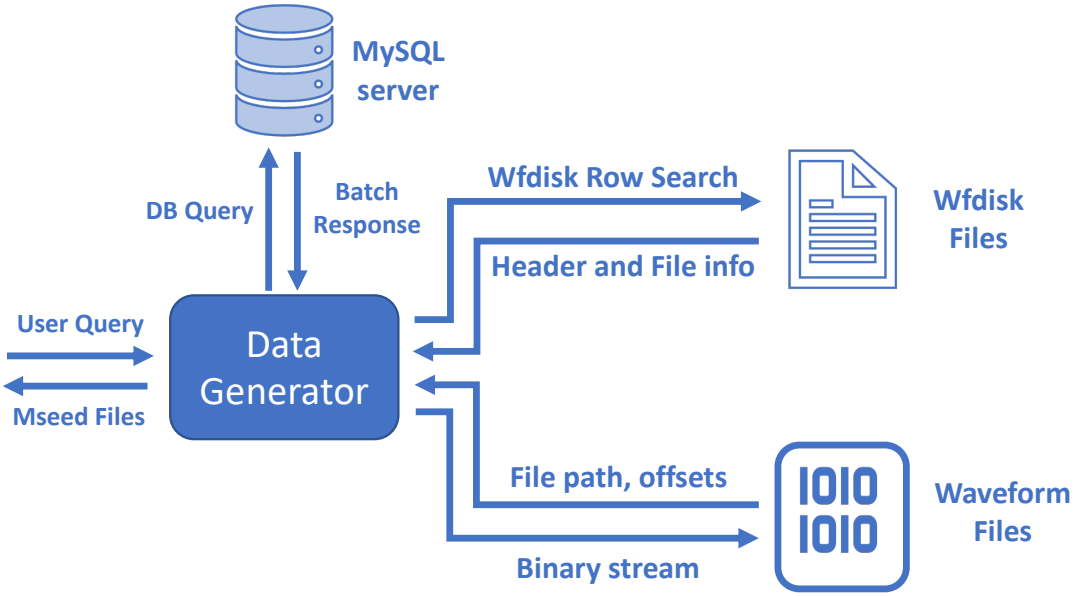

Figure 3.1: A high level workflow of UNM seismic data repository.

3.2.1 Workflow of Data Generation

In Figure 3.1, we show a workflow diagram of the UNM data repository. The workflow starts when a user search is fed into the *Data Generator* module. *Data Generator* converts the user search into a database query and fetches results from the corresponding MySQL database. The response from the database is a list of arrivals. Since the number of arrivals can be too large to process in one transaction, the arrivals are fetched in multiple (configurable,

default 5000) batches. Each arrival information then matched with the corresponding row in the wfdisk file. *Two* types of information are fetched from the wfdisk files: *(i)* Waveform descriptions like station, channel, sampling rate, start and end time, number of data points, clipping flag; and *(ii)* File pointer to waveform like the directory path, file index and byte offsets for the trace in the waveform file. Using the file pointer, the waveform trace is fetched from the binary waveform files. Finally, waveform descriptions and traces are combined together to create the miniseed file which is the output returned to the user of written in the disk.

3.3 Databases

We have created *three* databases to store global bulletins:

1. International Data Center (IDCx)
2. Late Event Bulletin (LEB)
3. Selected Event List (SEL3)

Each database consists of *six* tables. The table schema of identical for different databases. In this section, we describe the schema of all the tables. Also, in the figures 3.2, to 3.7 we show a couple of sample rows randomly picked from each table.

3.3.1 Table: Arrival

1. *sta*: VARCHAR(6). Contains station code. This column value is the code name of a seismic, hydroacoustic, or infrasonic observatory and identifies a geographic location recorded in the site table. Generally, only three to five upper case characters are used.
2. *time*: FLOAT. Epoch time, given as seconds since midnight, January 1, 1970, and stored in a double-precision floating number. time refers

Column	Value 1	Value 2	Column	Value 1	Value2	Column	Value 1	Value 2
sta	AAK	AAK	**deltim**	1.398	1.403	**logat**	-999	-999
time	1232535603.444	1232535608.544	**azimuth**	208.46	193.57	**clip**	-	-
arid	48191627	48191628	**delaz**	91.53	26.47	**fm**	-	-
jdate	2009021	2009021	**slow**	1.87	6.25	**snr**	6.38	6.34
Stassid	3746387	3746387	**delslo**	2.68	2.86	**qual**	-	-
chanid	-1	-1	**ema**	6.42	21.6	**auth**	NULL	NULL
chan	Z1020	Z1530	**rect**	0.884	0.867	**commid**	-1	-1
iphase	P	tx	**amp**	1.63	1.23	**lddate**	2009-01-29 12:09:55	2009-01-29 12:09:55
stype	-	-	**per**	0.67	0.44			

Figure 3.2: Snapshot of arrival table with two rows of randomly picked data entry.

to the table in which it is found; for example, in arrival it is the arrival time, in origin it is the origin time, in wfdisc it is the start time of data, and in siteaux it is the start time for which measurements are valid. Where the date of historical events is known, time is set to the start time of that date. Where the date of contemporary arrival measurements is known but no time is given, then time is set to the NA Value. The double-precision floating point number allows 15 decimal digits. At one milli- second accuracy, this is a range of 3 *104 years. Where the date is unknown, or prior to February 10, 1653, time is set to the NA Value.

3. *arid*: INT. Arrival identifier. Each arrival is assigned a unique positive integer identifying it with a unique station, channel and time.

4. *jdate*: INT. Julian date; date of an arrival, origin, seismic recording, and so on. The same information is available in epoch time, but the Julian date format is more con- venient for many types of searches. Dates B.C. are negative. The year will never equal 0000, and the day will never equal 000. Where only the year is known, the day of the year is 001; where only year and month are known, the day of year is the first day of the month. Only the year is negated for B.C., so 1 January of 10 B.C. is –0010001.

5. *stassid*: INT. Station association identification. The wavetrain from a single event may be made up of a number of arrivals. A unique stassid joins those arrivals believed to have come from a common event as measured at a single station. Stassid is also the key to the stassoc table, which contains additional signal measure- ments not contained within the arrival table, such as station magnitude estimates and computed signal characteristics.

6. *chanid*: INT. Channel recording identifier. The value is a surrogate key used to uniquely identify a specific recording. Chanid duplicates the information of the com- pound key station, channel and time.

7. *chan*: VARCHAR(8). Channel identifier. The value is an eight-character code that specifies a particular channel within a network (station), which, taken together with station and time, uniquely identifies seismic timeseries data, including the geographic location, spatial orientation, sensor, and subsequent data processing.

8. *iphase*: VARCHAR(8). Reported phase. This eight-character column holds the name initially given to a seismic phase. Standard seismological labels for the types of signals (or phases) are used (for example, P, PKP, PcP, pP). Both upper and lower case letters are available and should be used when appropriate, for example, pP or PcP.

9. *stype*: VARCHAR(1). Signal type. This single-character flag indicates the event or signal type. The following definitions hold:

    - l = local event
    - r = regional event
    - t = teleseismic event
    - g = glitch
    - e = calibration activity

10. *delim*: FLOAT. Arrival time uncertainty. This column is an estimate of the standard deviation of an arrival time.

11. *azimuth*: FLOAT. Observed azimuth. This value is the estimated station-to-event azimuth measured clockwise from North. The estimate is made from f-k or polarization analysis. In stassoc, the value may be an analyst estimate.

12. *delaz*: FLOAT. Azimuth uncertainty. This column is an estimate of the standard deviation of the azimuth of a signal.

13. *slow*: FLOAT. Observed slowness of a detected arrival. Units are seconds/kilometer in detection, seconds/degree in arrival and p-arrival.

14. *delslo*: FLOAT. Slowness uncertainty. This column is an estimate of the standard deviation of the slowness of a signal.

15. *ema*: FLOAT. Emergence angle. This column is the emergence angle of an arrival, as observed at a three-component station or array. The value increases from the vertical direction towards the horizontal.

16. *rect*: FLOAT. Signal rectilinearity defined as $1–(l3 + l2)/2 * l1$ where l1, l2, and l3 are the three eigenvalues from the decomposition of the covariance matrix. This column value is the maximum rectilinearity for all overlapping time windows.

17. *amp*: FLOAT. Measured amplitude defined by amplitude type.

18. *per*: FLOAT. Measured period at the time of the amplitude measurement.

19. *logat*: FLOAT. Log of amplitude divided by period. This measurement of signal size is often reported instead of the amplitude and period separately. This column is only filled if the separate measurements are not available.

20. *clip*: VARCHAR(1). Clipped data flag. The value is a single-character flag to indicate whether or not the data were clipped.

21. *fm*: VARCHAR(2). First motion. This is a two-character indication of first motion. The first char- acter describes first motion seen on short-period channels and the second holds for long-period instruments. Compression on a short-period sensor is denoted by c, dilatation by a d; and compression on a long-period sensor is denoted by u, dilatation by an r. Empty character positions will be indicated by dots (for example, ".r" for dilatation on a long-period sensor).

22. *snr*: FLOAT. Signal-to-noise ratio. This is an estimate of the ratio of the amplitude of the signal to amplitude of the noise immediately preceding it. For apma, this value is based on the maximum 3-component amplitudes. This column is the average signal-to-noise ratio for the frequency bands that contributed to the final polarization estimates.

23. *qual*: VARCHAR(1). Onset quality. This single-character flag is used to denote the sharpness of the onset of a seismic phase.

24. *auth*: VARCHAR(15). Author, the originator of the data. Auth may also identify an application generating the record, such as an automated interpretation or signal-processing program.

25. *commid*: INT. Comment identifier. The value is a key that points to free-form comments entered in the remark table. These comments store additional information about a record in another table. The remark table can have many records with the same commid and different lineno, but the same commid will appear in only one other record among the rest of the tables in the database.

26. *lddate*: DATETIME. Load date. The date and time the record was inserted into the database.

3.3.2 Table: Assoc

Column	Value 1	Value 2	Column	Value 1	Value2	Column	Value 1	Value 2
arid	47561181	47561216	**esaz**	284.77	1.76	**emares**	-999	-999
orid	5131589	5131589	**timeres**	0.165	-0.169	**wgt**	0.484	0.628
sta	ESDC	ARCES	**timedef**	d	26.47	**vmodel**	IASPEI	IASPEI
phase	P	P	**azres**	3.7	5.4	**commid**	-1	-1
belief	-1	-1	**azdef**	d	d	**lddate**	2009-01-16 09:19:56	2009-01-16 09:19:56
delta	21.085	32.35	**slores**	-0.74	0.64			
seaz	87.98	183.99	**slodef**	d	d			

Figure 3.3: Snapshot of assoc table with two rows of randomly picked data entry.

1. *arid:* INT. Arrival identifier. Each arrival is assigned a unique positive integer identifying it with a unique station, channel and time.

2. *orid:* INT. Origin identifier that relates a record in these tables to a record in the origin table.

3. *sta:* VARCHAR(6). Contains station code. This column value is the code name of a seismic, hydroacoustic, or infrasonic observatory and identifies a geographic location recorded in the site table. Generally, only three to five upper case characters are used.

4. *phase:* VARCHAR(8). Phase type. The identity of a phase that has been associated to an arrival. Standard labels for phases are used (for example, P, PKP, PcP, pP, and so on). Both upper and lower case letters are available and should be used when appropriate; for example, pP or PcP.

5. *belief:* FLOAT. Phase identification confidence level. The value is a qualitative estimate of the confidence that a seismic phase is correctly identified.

6. *delta:* FLOAT. Source-receiver distance. This column is the arc length, over the earth's surface, of the path the seismic phase follows from

source to receiver. The location of the origin is specified in the Origin record referenced by the column orid. The column arid points to the record in the Arrival table that identifies the receiver. The value of the column can exceed 360 degrees. The geographic distance between source and receiver is delta modulo(180).

7. *seaz:* FLOAT. Station-to-event azimuth calculated from the station and event locations and measured clockwise from north.

8. *esaz:* FLOAT. Event-to-station azimuth measured in degrees clockwise from North.

9. *timeres:* FLOAT. Time residual. This column is a travel-time residual measured in seconds. The residual is found by taking the observed arrival time (saved in the Arrival table) of a seismic phase and subtracting the expected arrival time. The expected arrival time is calculated by a formula based on an earth velocity model (column vmodel), an event location and origin time (saved in table Origin), the distance to the station (column dist in table assoc), and the particular seismic phase (column phase in table assoc).

10. *timedef:* VARCHAR(1). Time-defining code. This one-character flag indicates whether or not the time of a phase was used to constrain the event location. This column is defining ($timedef = d$) or nondefining ($timedef = n$).

11. *azres:* FLOAT. Azimuth residual. The value is the difference between the measured station-to-event azimuth for an arrival and the true azimuth. The true azimuth is the bearing to the inferred event origin.

12. *azdef:* VARCHAR(1). Azimuth-defining code. The one-character flag indicates whether or not the azimuth of a phase was used to constrain the event location solution. This column is defining ($azdef = d$) if it was used in the location, nondefining ($azdef = n$) if it was not.

13. *slores:* FLOAT. Slowness residual. This column gives the difference between an observed slowness and a theoretical prediction. The prediction is calculated for the related phase and event origin described in the record.

14. *slodef:* VARCHAR(1). Slowness defining code. This one-character flag indicates whether or not the slowness of a phase was used to constrain the event location. This column is defining ($slodef = d$) or nondefining ($slodef = n$) for this arrival.

15. *emares:* FLOAT. Emergence angle residual. This column is the difference between an observed emergence angle and the theoretical prediction for the same phase, assuming an event location as specified by the accompanying *orid*.

16. *wgt:* FLOAT. Location weight. This column gives the final weight assigned to the allied arrival by the location program. This column is used primarily for location programs that adaptively weight data by their residuals.

17. *vmodel:* VARCHAR(15). Velocity model. This character string identifies the velocity model of the earth used to compute the travel times of seismic phases. A velocity model is required for event location (if phase is defining) or for computing travel-time residuals.

18. *commid*: INT. Comment identifier. The value is a key that points to free-form comments entered in the remark table. These comments store additional information about a record in another table. The remark table can have many records with the same *commid* and different *lineno*, but the same *commid* will appear in only one other record among the rest of the tables in the database.

19. *lddate*: DATETIME. Load date. The date and time the record was inserted into the database.

3.3.3 Table: Event

Column	Value 1	Value 2	Column	Value 1	Value 2	Column	Value 1	Value 2
evid	2762179	2762180	**prefor**	2762179	2762180	**commid**	NULL	NULL
evname	-	-	**auth**	-1	-1	**lddate**	2009-11-20 15:02:37	2009-11-20 15:02:37

Figure 3.4: Snapshot of event table with two rows of randomly picked data entry.

1. *evid*: INT. Event identifier. Each event is assigned a unique positive integer that identifies it in a database. Several records in the origin table can have the same evid. Analyst have several opinions about the location of the event.

2. *evname*: VARCHAR(15). Event name. This is the common name of the event identified by *evid*.

3. *prefor*: INT. Preferred origin. This column holds the origin identifier (*orid*) that points to the preferred origin for a seismic event.

4. *auth*: VARCHAR(15). Author, the originator of the data. Auth may also identify an application generating the record, such as an automated interpretation or signal-processing program.

5. *commid*: INT. Comment identifier. The value is a key that points to free-form comments entered in the remark table. These comments store additional information about a record in another table. The remark table can have many records with the same *commid* and different lineno, but the same *commid* will appear in only one other record among the rest of the tables in the database.

6. *lddate*: DATETIME. Load date. The date and time the record was inserted into the database.

3.3.4 Table: Origin

Column	Value 1	Value 2	Column	Value 1	Value 2	Column	Value 1	Value 2
lat	0.369964	52.407644	**ndef**	3	5	**ms**	-999	-999
lon	132.37178	34.071173	**ndp**	0	0	**msid**	-1	-1
depth	0	0	**grn**	611	724	**ml**	-999	-999
time	12587060 26.70587	12587076 77.26363	**srn**	39	49	**mlid**	-1	-1
			etype	-	-	**algorithm**	LocSAT	LocSAT
orid	2762179	2762180	**depdp**	-999	-999	**auth**	NULL	NULL
evid	2762179	2762180	**dtype**	g	g	**commid**	-1	-1
jdate	2009324	2009324	**mb**	3.84	-999	**lddate**	2009-11-20 15:02:37	2009-11-20 15:42:44
nass	4	7	**mbid**	13939842	-1			

Figure 3.5: Snapshot of origin table with two rows of randomly picked data entry.

1. *lat*: FLOAT. Geographic latitude. Locations north of the equator have positive latitudes.

2. *lon*: FLOAT. Geographic longitude. Longitudes are measured positive east of Greenwich meridian.

3. *depth*: FLOAT. Source depth. This column gives the depth (positive down) of the event origin. Negative depth implies an atmospheric event. In *stassoc*, this may be an analyst estimate.

4. *time*: FLOAT. Epoch time, given as seconds since midnight, January 1, 1970, and stored in a double-precision floating number. time refers to the table in which it is found; for example, in arrival it is the arrival time, in origin it is the origin time, in wfdisc it is the start time of data, and in siteaux it is the start time for which measurements are valid. Where the date of historical events is known, time is set to the start time of that date. Where the date of contemporary arrival measurements is known but no time is given, then time is set to the NA Value. The double-precision floating point number allows 15 decimal digits. At one millisecond accuracy, this is a range of $3 * 104$ years.

Where the date is unknown, or prior to February 10, 1653, time is set to the NA Value.

5. *orid:* INT. Origin identifier that relates a record in these tables to a record in the origin table.

6. *evid*: INT. Event identifier. Each event is assigned a unique positive integer that identifies it in a database. Several records in the origin table can have the same evid. Analyst have several opinions about the location of the event.

7. *jdate*: INT. Julian date; date of an arrival, origin, seismic recording, and so on. The same information is available in epoch time, but the Julian date format is more con- venient for many types of searches. Dates B.C. are negative. The year will never equal 0000, and the day will never equal 000. Where only the year is known, the day of the year is 001; where only year and month are known, the day of year is the first day of the month. Only the year is negated for B.C., so 1 January of 10 B.C. is –0010001.

8. *nass*: INT. Number of associated arrivals. This column gives the number of arrivals associated with the origin.

9. *ndef*: INT. Number of time-defining phases.

10. *ndp*: INT. Number of depth phases.

11. *grn*: INT. Geographic region number.

12. *srn*: INT. Seismic region number.

13. *etype*: VARCHAR(8). Event type. Describes the type of event.

14. *depdp*: FLOAT. Depth as estimated from depth phases. The value is a measure of event depth estimated from a depth phase or an average of several depth phases. Depth is measured positive in a downwards direction starting from the earth's surface.

15. *dtype*: VARCHAR(1). Depth determination flag. This single-character flag indicates the method by which the depth was determined or constrained during the location process. The recommended values are f (free), d (from depth phases), r (restrained by location program) or g (restrained by geophysicist). In cases r or g, either the *auth* column should indicate the agency or person responsible for this action.

16. *mb*: FLOAT. Body wave magnitude, mb. This is the body wave magnitude of an event. The identifier *mbid* that points to *magid* in the *Netmag* table is associated with this column.

17. *mbid*: INT. Magnitude identifier for mb. This column stores the *magid* for a record in *Netmag*.

18. *ms*: FLOAT. This is the surface wave magnitude for an event. The identifier *msid*, which points to *magid* in the *Netmag* table, is associated with this column. The information in that record summarizes the method of analysis and the data used (see *imb*, *iml*, *ims*, *magnitude*, *magtype*, *mb*, and *ml*).

19. *msid*: INT. Magnitude identifier for ms. This column stores the *magid* for a record in *Netmag*.

20. *ml*: FLOAT. Local magnitude (ML) of an event. The identifier *mlid*, which points to *magid* in the *Netmag* table, is associated with this column.

21. *mlid*: INT. Magnitude identifier for ml. This column stores the magid for a record in Netmag. *Mlid* is a foreign key joining origin to *netmag*, where *table.mlid* = *table.magid*.

22. *algorithm* VARCHAR(15). Location algorithm used. This column is a brief textual description of the algorithm used for computing a seismic origin.

23. *auth*: VARCHAR(15). Author, the originator of the data. Auth may also identify an application generating the record, such as an automated interpretation or signal-processing program.

24. *commid*: INT. Comment identifier. The value is a key that points to free-form comments entered in the remark table. These comments store additional information about a record in another table. The remark table can have many records with the same *commid* and different lineno, but the same *commid* will appear in only one other record among the rest of the tables in the database.

25. *lddate*: DATETIME. Load date. The date and time the record was inserted into the database.

3.3.5 Table: Netmag

Column	Value 1	Value 2	Column	Value 1	Value2	Column	Value 1	Value 2
magid	12457161	12457171	**magtype**	mb	mb	**auth**	-1	-1
net	SEISMIC	SEISMIC	**nsta**	3	4	**commid**	NULL	NULL
orid	5099452	5099457	**magnitude**	3.88	3.51	**lddate**	2009-01-01 12:16:53	2009-01-01 12:16:53
evid	5099452	5099457	**uncertainty**	0.66	0.65			

Figure 3.6: Snapshot of netmag table with two rows of randomly picked data entry.

1. *magid*: INT. Network magnitude identifier. This value is assigned to identify a network magnitude in the *netmag* table. This column is required for every network magnitude. Magnitudes given in Origin must reference a network magnitude with *magid* = *mbid*, *mlid* or *msid*, whichever is appropriate .

2. *net*: VARCHAR(8). Unique network identifier. This character string is the name of a seismic network. One example is WWSSN.

3. *orid:* INT. Origin identifier that relates a record in these tables to a record in the origin table.

4. *evid*: INT. Event identifier. Each event is assigned a unique positive integer that identifies it in a database. Several records in the origin table can have the same evid. Analyst have several opinions about the location of the event.

5. *magtype*: VARCHAR(6). Magnitude type, for example, *mb*.

6. *nsta*: INT. Number of stations. This column is the number of stations contributing to the network magnitude estimate.

7. *magnitude*: FLOAT. Magnitude. This column gives the magnitude value of the type indicated in *magtype*. The value is derived in a variety of ways, which are not necessarily linked directly to an arrival.

8. *uncertainty*: FLOAT. Magnitude uncertainty. This column value is the standard deviation of the accompanying magnitude measurement.

9. *auth*: VARCHAR(15). Author, the originator of the data. Auth may also identify an application generating the record, such as an automated interpretation or signal-processing program.

10. *commid*: INT. Comment identifier. The value is a key that points to free-form comments entered in the remark table. These comments store additional information about a record in another table. The remark table can have many records with the same *commid* and different lineno, but the same *commid* will appear in only one other record among the rest of the tables in the database.

11. *lddate*: DATETIME. Load date. The date and time the record was inserted into the database.

Column	Value 1	Value 2	Column	Value 1	Value2	Column	Value 1	Value 2
orid	2762179	2762180	**syz**	-1	-1	**strike**	63.44	108.06
sxx	1592.08	483.423	**stx**	-0.1361	-55.0804	**sdepth**	-1	-1
syy	920.343	103.245	**sty**	2.5761	16.8975	**stime**	1.954	4.413
szz	-1	-1	**stz**	-1	-1	**conf**	0.9	0.9
stt	1.4083	7.1861	**sdobs**	0.0943	2.0422	**commid**	-1	-1
sxy	447.673	-138.749	**smajax**	91.3985	49.314	**lddate**	2009-11-20 15:02:37	2009-11-20 15:42:44
sxz	-1	-1	**sminax**	56.6074	16.333			

Figure 3.7: Snapshot of origerr table with two rows of randomly picked data entry.

3.3.6 Table: OrigErr

1. *orid:* INT. Origin identifier that relates a record in these tables to a record in the origin table.

2. *sxx, syy, szz, stt, sxy, sxz, syz, stx, sty, stz*: FLOAT. Elements of the covariance matrix for the location identified by *orid*. The covariance matrix is symmetric (and positive definite) so that $sxy = syx$, and so on, (x, y, z, t) refer to latitude, longitude, depth, and origin time, respectively. These columns (together with *sdobs*, *ndef*, and *dtype*) provide the information necessary to construct the K-dimensional (K = 2, 3, 4) confidence ellipse or ellipsoids at any confidence limit desired.

3. *sdobs*: FLOAT. Standard error of one observation. This column is derived from the discrepancies in the arrival times of the phases used to locate an event. This column is defined as the square root of the sum of the squares of the time residuals divided by the number of degrees of freedom. The latter is the number of defining observations minus the dimension of the system solved (4 if depth is allowed to be a free variable, 3 if depth is constrained).

4. *smajax*: FLOAT. Semi-major axis of error ellipse for a given confidence. The column value is the length of the semi-major axis of the location error ellipse. The value is found by projecting the covariance matrix onto the horizontal plane.

5. *sminax*: FLOAT. Semi-minor axis of error ellipse. The column value is the length of the semi-minor axis of the location error ellipse. The value is found by projecting the covariance matrix onto the horizontal plane.

6. *strike*: FLOAT. Strike of major axis of error ellipse. This column is the strike of the semi-major axis of the location error ellipse, measured in degrees clockwise from the North.

7. *sdepth*: FLOAT. Depth error. This is the maximum error of a depth estimate for a level of confidence.

8. *stime*: FLOAT. Origin time error.

9. *conf*: FLOAT. Confidence measure for a particular event identification method.

10. *commid*: INT. Comment identifier. The value is a key that points to free-form comments entered in the remark table. These comments store additional information about a record in another table. The remark table can have many records with the same *commid* and different lineno, but the same *commid* will appear in only one other record among the rest of the tables in the database.

11. *lddate*: DATETIME. Load date. The date and time the record was inserted into the database.

3.4 Wfdisk Files

The widisk files are text files that contain waveform headers and description in tabular format. The table has 20 columns, each representing meta information of a specific trace of waveform in the waveform file. This table also

provides a pointer (or index) to raw waveforms stored on disk. In figure 3.8 standard wfdisk column descriptions[1] are shown.

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	sta	varchar2(6)	a6	1-6	station code
2	chan	varchar2(8)	a8	8-15	channel code
3	time	f1oat(53)	f17.5	17-33	epoch time of first sample in file
4	wfid	number(9)	i9	35-43	waveform identifier
5	chanid	number(8)	i8	45-52	channel identifier
6	jdate	number(8)	i8	54-61	julian date
7	endtime	f1oat(53)	f17.5	63-79	*time +(nsamp* -1)/samles
8	nsamp	number(8)	is	81-88	number of samples
9	samprate	f1oat(24)	f11.7	90-100	sampling rate in samples/sec
10	calib	f1oat(24)	f16.6	102-117	nominal calibration
11	calper	f1oat(24)	f16.6	119-134	nominal calibration period
12	instype	varchar2(6)	a6	136-141	instrument code
13	segtype	varchar2(1)	a1	143-143	indexing method
14	datatype	varchar2(2)	a2	145-146	numeric storage
15	clip	varchar2(1)	a1	148-148	clipped flag
16	dir	varchar2(64)	a64	150-213	directory
17	dfile	varchar2(32)	a32	215-246	data file
18	foff	number(10)	i10	248-257	byte offset of data seg ment within file
19	commid	number(9)	i9	259-267	comment identifier
20	Iddate	date	a19	269-287	load date

Figure 3.8: A wfdisk table [27] that contains meta information of waveforms stored in a disk.

[1] http://nappe.wustl.edu/antelope/css-formats/wfdisc.htm

3.5 Waveform Files

Waveform files are binary files with an extension **.w*. These files contain multiple traces of seismic waveforms which can be read from the file using the pointer for the trace in corresponding *wfdisk* file. For example, let's say we are looking for a specific waveform trace. The corresponding wfdisk file will be the starting point, Rows 16, 17 and 18 contain the information regarding the directory the waveform file is located, name of the waveform file and byte offset respectively. Therefore, for a specific trace of waveform, we need to use the columns 16 and 17 of the corresponding row in the wfdisk file to get the appropriate waveform file. We can fetch the waveform by reading from the start offset that is listed in column 17. The end offset can be calculated using columns 7 and 8.

Chapter 4

Seismic Depth Prediction

4.1 Introduction

Accurate depth estimation of seismic events is a critical procedure to discriminate between man-made and natural events. While anthropogenic seismic sources are overwhelmingly less than 1 km depth, nearly all earthquakes nucleate below a 2 or 3 km depth. Most earthquakes considered shallow occur between a few and 70 kilometers depth, while deep earthquakes can occur as deep as 700 kilometers [51]. Distinguishing man-made events from natural events has several applications in nuclear non-proliferation [14, 49], and security of underground assets such as optical fiber, among others.

Theoretically, the depth of a seismic event is estimated by inverting the travel time equations to individual observing stations. However, the correctness of the estimation largely depends on the locations and the number of observing stations. When stations are far from an event (e.g., more than 100 km), and the number of stations observing the event is small (e.g., three or less), the uncertainty in the estimated depth grows beyond tolerance. In contrast, the depth is most accurately estimated when a seismic station is located exactly above the event's origin. Unfortunately, no single seismic network can guarantee global coverage, leading to inaccurate depth estimation for novel seismic hot spots.

This research considers estimating seismic depth directly from the waveforms generated by the events (i.e., time series, or seismograms) employing modern machine learning (ML) techniques. Such an approach to evaluating ML methods and potentially replacing physics-based estimation methods is gaining significant interest among geophysicists [14]. In this paper, we develop the first hierarchical neural network model named **Septor** (Seismic depth estimator) aimed to estimate the depth of seismic events from waveforms of multiple channels at multiple stations. Besides its novelty in automated depth estimation, our model can potentially be a support tool to distinguish seismic events (i.e., man-made vs. natural), and has several applications in automated seismic monitoring. We train Septor using a set of 8,359 highly calibrated (by human analysts) events from the Southern California Earthquake Data Center (SCEDC) spanning over forty years of monitoring data. We have achieved an impressive root-mean-squared-error (RMSE) of 2.89 km in predicting the depth (with 70.1% correlation to actual depth) of these events using only a few close-by stations from the same network. We have also considered a binary classification problem to distinguish shallow from deep events. Our model achieves a 86.5% F1-score in shallow-deep discrimination, promising a step closer to fully automatic seismic monitoring. Even though the model can learn waveform features from a specific network of station, we neither expect nor claim the model learning the underlying travel-time inversion process. In an attempt to evaluate the generalizability, we perform three experiments on disjoint train-test sets with no common source and station. The model shows a gradual decrease in performance as the distances between source-station pairs increase.

Septor architecture is not very deep compared to modern ML models used in computer vision or speech processing. The reason is the lack of labeled data to train a deeper network. The calibrated events used to train Septor were labeled by well-trained analysts working on daily shifts for years. Hence, we tailored the model to fit our data instead of fitting a model to tailored data. Our network architecture has two hierarchies, each consisting

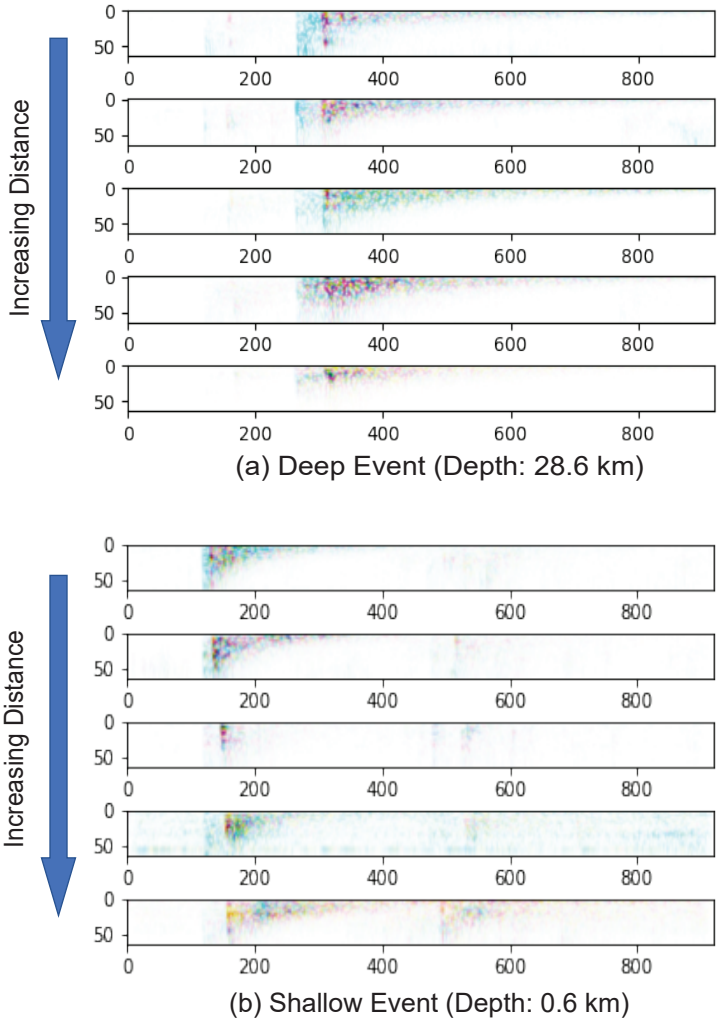

Figure 4.1: CWT images from the 5 closest stations for a deep (top) and a shallow (bottom) earthquake. For the same earthquake, CWT images are ordered ascending according to the *epicenter to station distance* from top to bottom.

of Convolutional Neural Network (CNN) layers followed by Long-Short Term Memory (LSTM) layers. Such simple architecture provides a great deal of efficiency and interpretability for geophysicists. We demonstrate that our

model conforms to the general scientific understanding of depth estimation and can be employed in at least two different seismic regions.

Why is depth estimation from waveforms challenging? *(i)* Physics-based depth estimation suffers from uncertainty due to noise in the signal and the lack of nearby stations. For example, in the SCEDC original catalog [4, 39], the mean quantified uncertainty is 6.89 km. Such uncertain labels in training data hardly lead to accurate models, making seismologists skeptical about ML-based systems for depth estimation. However, we consider the highly calibrated SCEDC catalog [23] for training our model with a very low uncertainty of 0.357 km (due to high station density in the network). Thus, the uncertainty in training labels cannot accumulate into the validation error. *(ii)* Physics-based estimated depth for the same earthquake event can be different in different earthquake catalogs. For example, the root-mean-squared difference of depths between the SCEDC original catalog and the SCEDC highly calibrated catalog is **3.81** km for the same set of earthquakes. Moreover, different algorithms may calculate the depth from different reference points. For example, in Northern California, NCSN uses depth relative to the geoid (essentially sea level) [10], whereas the double-difference catalog uses depth from the surface [57]. Therefore, ambiguity among data sources makes it hard to evaluate learned models on new catalogs. We contribute experiments to test our model in multiple geographic regions to demonstrate equivalent performance; *(iii)* Only experts in geophysics and seismology can produce confident labeled information. Unfortunately, it is very difficult (or even impossible) for a non-expert to spot any pattern in raw waveforms or other visual data representation. To better illustrate this difficulty, consider the Continuous Wavelet Transform (CWT) representation of a deep earthquake captured by different stations at varying distances illustrated in Figure 4.1 (a) and a shallow earthquake shown in Figure 4.1 (b). Therefore, crowd-sourced annotations are not feasible for this application, ruling out the option to train deep models on a large-scale labeled dataset. For this reason, we propose a model based on a small two-hierarchy network, just enough to

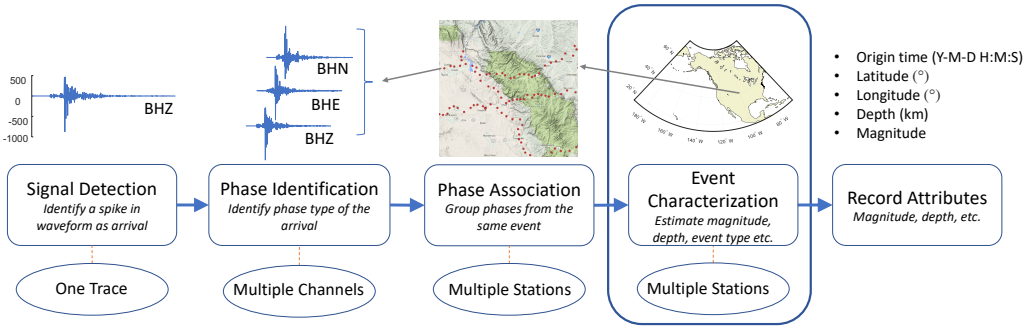

Figure 4.2: Seismic data processing pipeline. We perform this task of depth prediction in the "Event Characterization" step without the human supervision.

learn from the set of available well-calibrated events.

The remaining of this chapter is organized as follows. Section 4.2 introduces essential concepts of seismology related to the task of depth prediction. Section 4.3 discuss related work. We describe Septor in Section 4.4. Data description and preprocessing are discussed in Section 4.5. The experimental setup and our evaluations considering regression and classification settings are presented in Section 4.6 and Section 4.7. The use of Septor at a different region is discussed in Section 4.8. Finally, our conclusions are presented in Section 4.9.

4.2 Background

In this section, we discuss the main steps of a typical seismic data processing pipeline illustrated in Figure 4.2. This pipeline refers to the process of transforming a set if seismic signals into a bulletin of seismic events. (i.e earthquakes, explosions, etc). The data processing starts when a network of stations detects a seismic signal that could be from a man-made or a natural event.

Signal detection: This step consists of monitoring a continuous wave-

form obtained by seismometers to detect events. Typically, a seismometer captures ground motion in multiple directions (up-down, north-south, and east-west) to obtain a three-dimensional ensemble. Any seismic event must excite at least one dimension or channel (e.g., BHZ, BHN, and BHE) of a seismometer. When all channels simultaneously show signals, one can be confident about the arrival of a seismic wave at the station. Once an event is detected, the remaining steps are executed in sequence.

Phase identification: Any movement of earth materials generates multiple types of waves and each wave is affected differently by the unique velocity structure along it's own propagation path. Common phase types are compressional (P) and shear (S) waves, which travel through the earth; and surface waves, which travel along the earth's surface. A phase picker detects different types of signals and labels them as P, S, or other types of waves, as illustrated in Figure 4.3. For example, when an earthquake or underground explosion occurs, seismic waves propagate away from the source. The scattered energy immediately following the P and S waves is called coda. Characteristics of each part of the waveform carry information about the source, including about its depth. Accurate phase picking is essential to obtain parameters that can constrain location, depth, and event type.

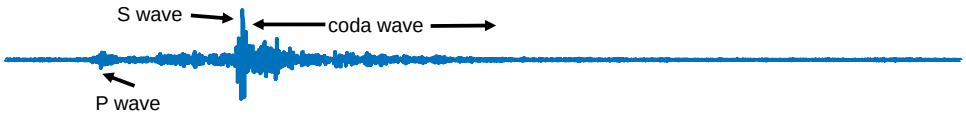

Figure 4.3: A 230 second long seismic waveform shows the P wave arrives first for an earthquake, followed by the S wave. From 30 seconds before the P arrival time to 200 seconds after is sufficient to capture the entire earthquake.

Phase association: Association of phases across multiple stations is done by a phase associator. Phase associators aggregate multiple waveforms generated from the same earthquake and triangulate to compute the epi-

center[1] and depth of the earthquake. Phase associators work on multiple stations' data and may generate other meta information such as an initial location of the earthquake.

Event characterization: Typically, the depth estimate of an event may be refined in this stage of the pipeline. Depth can be estimated by minimizing the propagation time residuals relative to some prediction across all the observing stations. However, considering the complexity of the earth's structure, depth estimation involves many geophysical priors, including local, regional, and global velocity models. Misassociated phases are common in automated associators and at the very least can bias depth estimates (and even lead to false events being formed), so human supervision of output is necessary. In this work, we propose to perform the depth estimation after the initial event formation to refine the initial depth estimate. The goal is to estimate and exploit the depth of an event automatically without human supervision. After the event characterization step, an event, with its location, depth, magnitude, etc. are listed in the earthquake catalog.

4.3 Related Works

State-of-the-art seismic event depth prediction techniques require considerable manual oversight in data annotation and expertise in data analysis. Depths are determined using 3D earth models and seismic wave travel-times. They can be refined by consideration of travel times relative to other seismic events, providing extremely accurate relative locations, which can in turn improve absolute locations. The range of uncertainty associated with the estimated depth though often makes it hard to consider them as "ground truth". For these reasons, the number of examples to train and test predicting models is limited to less than one thousand instances in the literature proposals. Such a reduced number of examples make challenging the use

[1]The epicenter is the location of an earthquake on the earth surface. The epicenter does not provide information about depth.

of modern machine learning models, such as deep learning based. Besides, these works consider the information from a single station for prediction, as discussed follows.

Seismologists have used physics based approaches to identify waveform features that indicate source depth, and have applied metrics based on those features to estimate depth. Kafka [29] used the existence of higher frequency fundamental mode Rayleigh waves, or Rg, as an indication of shallow source depth, typically less than 3 or 4 km. Rg detectors are useful tools for identifying shallow seismic sources in some areas, but especially in tectonically active areas such as southern California, Rg may only propagate 10 km before it is attenuated to below background levels of the S wave coda [2], which arrives at the same time. The ratio of the S wave peak amplitude to the duration of the S wave coda also is affected by and can be used to roughly estimate the source depth, with longer coda durations occurring for shallow sources [30]. This may be due to trapping of high frequency shear waves in shallow low velocity waveguides, where they propagate more slowly, so extend the wavetrain. Alternately, scattering of Rg at or near the surface can also extend the wavetrain. Both mechanisms may be important, but the physical basis of this observation is still not fully settled. The ratio of P to S wave energy is often also indicative of depth, as S waves are more likely to be trapped in highly attenuative near surface waveguides than are P waves, thus increasing the ratio of P to S wave energy for shallow events relative to deeper ones [22]. While the accuracy and broad applicability of these methods are limited, we are encouraged that there are such features in seismic waveforms that ML methods can learn, and optimize the use of to improve estimations. Further, we expect that other features may exist that we haven't discovered using physics-based approaches, but that algorithms can learn with sufficient training data.

Ochoa et al. [38] proposed to use Support Vector Regression (SVR) for depth estimation of events collected by a single station at Bogota, Colombia.

[2] coda wave is the part of the waveform that follows the peak amplitude

The original catalog has 2,164 events observed between 1998 and 2008. However, events with a lower magnitude and possibly related to man-made events were discarded in the experimental evaluation, leaving only 863 events. A set of 25 features related to the magnitude, epicenter distance, and source location was extracted from data for training the SVR model.

Recently, Yang et al. [63] compared the performance of conventional feature-based classification models (e.g., Support Vector Machine, Random Forest, Naive Bayes, and k-Nearest Neighbors) with a 10 layers convolutional neural network using CWT representation for distinguishing between deep and shallow seismic events. The comparisons were performed using 444 micro-earthquake events correlated with an underground collapse of a cavern in South Louisiana. The signals were sampled at 200 Hz by 8 broad-band three-component monitoring devices, and the hypocentre depths range from 1 km to 2 km for deeper events and depths between 40 m and 400 m for shallow events.

From machine learning perspective, classification of deep and shallow earthquakes from waveforms can be described as a *Time Series Classification* problem (TSC) [6, 17, 7], in which the goal is to predict a discrete label for a series from a finite set of categories. In the past decade, TSC problems have been of great interest among data mining researchers, and have been applied in many different domains. In seismology, TSC has been used in the context of phase detection and identification [14, 32, 42]. However, there is a lack of solutions for depth prediction as a regression of seismic events using waveforms from multiple stations, as proposed in this paper. Although hierarchical networks have been used in other domains like text classification [64], information extraction [21].

Depth estimation from seismic waveform falls under the category of *Time Series Extrinsic Regression* (TSER) [53] where a single scalar continuous value is predicted based on the whole time series. Unlike *Time Series Forecasting* [26], where the prediction *mostly* depends on recent values, TSER considers the whole time series *equally* for prediction. Although TSER has

been used in a wide range of domains [18, 53], it has not been employed in seismology. In this paper, we compare our solution with Rocket [18], a state-of-the-art TSER method.

4.4 Septor: Hierarchical Network

Convolutional Neural Networks (CNN) and Long-Short Term Memory (LSTM) networks are the building blocks of our model Septor . A CNN is a class of deep neural networks widely employed on image mining problems. CNNs perform convolutions on images with multiple fixed-sized kernels. A convolution operation can be seen as sliding the kernel over the image and computing the dot product at each step. Each convolution extracts different higher-order representations from the feature map. A convolutional layer is usually followed by a nonlinear activation function (such as ReLU) and a max-pooling function.

Long Short-Term Memory (LSTM) networks are an improved variant of traditional Recurrent Neural Networks (RNNs) [24]. RNNs can model temporal dependencies in the data by feedback connections considering both the input at the current time step as well as the output of the last time step's hidden state. However, vanilla RNNs suffer from the vanishing gradient problem, which prevents the model from learning long range dependencies. LSTM tackles this problem by introducing three gating mechanisms to update the memory cell c_t and hidden state h_t at each step t based on the current time step input x_t and the previous time step's hidden state output h_{t-1}. The input gate i_t, forget gate f_t, output gate o_t, memory cell c_t and hidden state h_t at step t are computed as follows:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i \tag{4.1}$$

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f \tag{4.2}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o \tag{4.3}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{4.4}$$

$$h_t = o_t \odot tanh(c_t) \tag{4.5}$$

Here, σ is the logistic sigmoid function, tanh is the hyperbolic tangent function, and $\odot$ denotes the element wise multiplication. Each LSTM unit is composed of a memory cell and three main gates: input, output and forget. By this architecture, the LSTM manages to create a controlled information flow by deciding which information it must forget and which information to remember. To understand the mechanism behind the architecture, we can view f_t as the function that controls to what extent the information from the old memory cell is going to be thrown away, i_t controls how much new information is going to be stored in the current memory cell, and o_t controls what to output based on the memory cell c_t.

4.4.1 Architecture of Septor

Septor consists of two loosely connected hierarchies, as illustrated in Figure 4.4: (1) waveform aggregator, and (2) station aggregator. Waveform aggregator (Figure 4.5) is a CNN-LSTM based network and receives 3 dimensional linearly spaced Continuous Wavelet Transform (CWT) image for a single station and outputs a 2D feature array. Multiple output from the waveform aggregator is fed into the station aggregator (Figure 4.6). Station aggregator receives the features from waveform aggregator in a distance (distance from the epicenter of the earthquake to the station) preserving ordering. Station aggregator is another CNN-LSTM based network which finally predicts the depth of the earthquake.

Our *waveform aggregator* consists of four CNN layers followed by two LSTM layers followed by three fully connected (FC) layers. Each of the

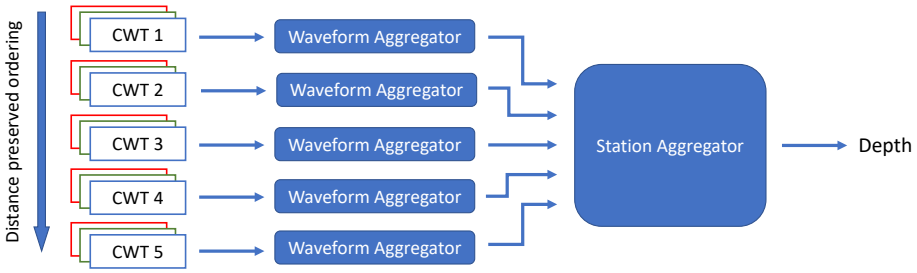

Figure 4.4: General view of Septor architecture.

CNN layer is followed by a batch normalization layer and a ReLU activation layer. 30% dropout is applied after ReLU activation layers to prevent from overfitting the model. We used 8, 16, 32 and 64 size kernels of size 3×3 for the convolution layers. The output from the fourth CNN layer is fed to the two stacked LSTM layers after a time preserving transformation. Extracted features from the LSTM layers are then passed through three FCN layers. After each FCN layer, batch normalization and ReLU activation is applied. Figure 4.5 contains the major parts of the waveform aggregator.

The *station aggregator* consists of one CNN layer, one LSTM layer and a stack of three FCN layers. CNN layer in *station aggregator* has 16 kernels of 3×3 size and is followed by a ReLU activation and a max pooling layer. Output from the CNN layer is fed into the LSTM layer which has 32 hidden units. *Station aggregator* hierarchy ends with FCN layers with 64, 32 and 1 output units. Figure 4.6 contains the details of *station aggregator*. Both for waveform aggregator and station aggregator, we define the number of layers according to preliminary results on training sets.

4.5 Data Description

We carry out our experimental evaluation with a highly calibrated earthquake catalog from Southern California [23]. This catalog used either single station locations with a 3D velocity model [31] or a multiple event location method,

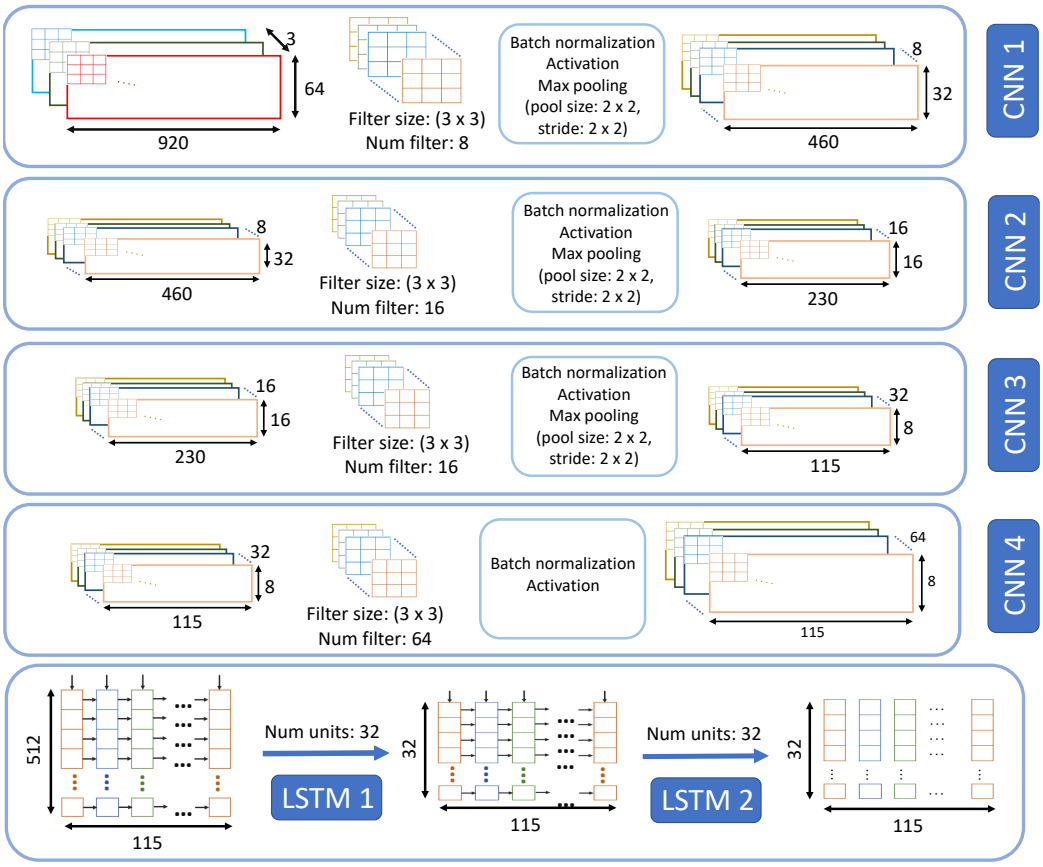

Figure 4.5: *Waveform aggregator* consists of four CNN and two LSTM layers.

GrowClust [54] for depth calculation, both of which are more accurate than the baseline Southern California Earthquake Data Center (SCEDC) depth calculation. The catalog includes uncertainty bounds for the depths. The reported median vertical uncertainty is 0.4 km, which is considered very good for many seismic monitoring tasks. Therefore, this dataset is a close-to-ideal candidate for our experimental evaluation. We collected more than 650,000 multi-channel waveforms that are associated with earthquakes having magnitude 2.0 and 4.0, recorded by 423 densely located seismic stations in Southern California region.

We filter out the earthquake events for which we did not find any station

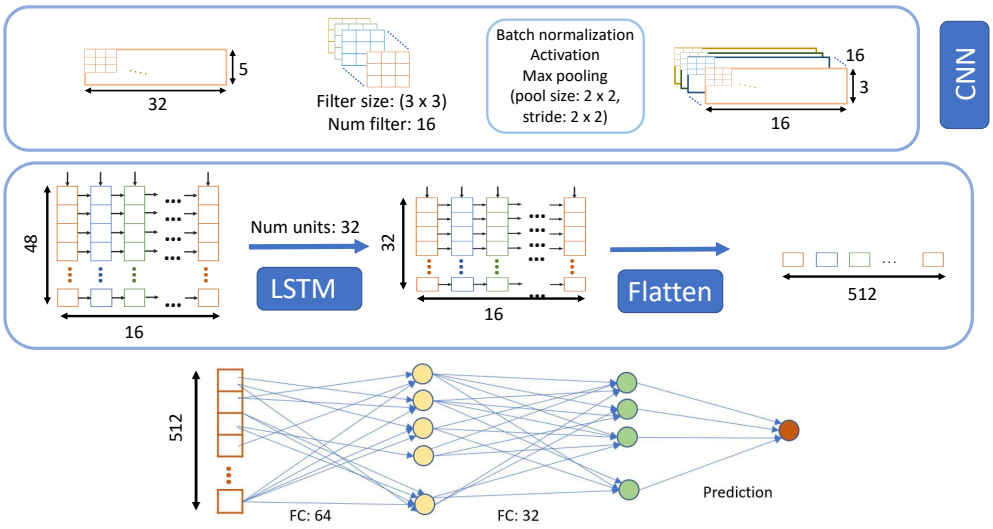

Figure 4.6: *Station aggregator* consists of a CNN and an LSTM layers. Three fully connected layers are used to get the final prediction.

with a distance less than 1.2 times the reported depth. Such close proximity of an observing station ensures greater accuracy of the depth estimate. After selecting events measured by at least five stations and containing waveforms from all three broadband channels (BHZ, BHN, BHE), we build a dataset with 8,359 earthquake events. In Figure 4.7, we show the SCEDC station map and depth distribution of our collected earthquakes.

For each of the 3-channels, we collected 230 seconds of waveforms, starting 30 seconds prior to the first P arrival time and ending at 200 seconds after the P arrival time. This time window is large enough to capture seismic waves generated by any regional earthquake. Since the waverforms are sampled at 40Hz, the length of each waveform is 9,200 data points. In summary, our dataset consists of:

- A total of 8,359 earthquake events;
- Each event is recorded at 40Hz by five observing stations;
- Each station records three broadband waveforms;

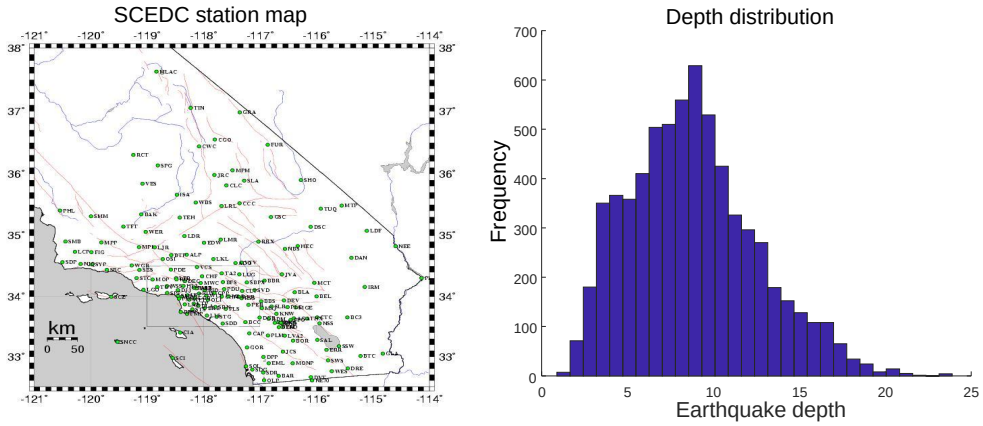

Figure 4.7: (left) Distribution of Broadband stations in Southern California shows a dense seismic network [46]. (right) Distribution of earthquake depth (in km.) of our dataset.

- Each waveform contains 9,200 numeric observations.

4.5.1 Number of Observing Stations

To build our dataset, we select from the highly calibrated catalog events that were measured by at least five observing stations close to the epicenter. Although a higher number of stations can provide more information for each event, only a reduced number of events could be measured by many close stations due to the geographic limit in station coverage. In practice, the number of available events decreases if we consider increasingly more observing stations, as shown in Figure 4.8. We consider five stations as a reasonable number to balance this trade-off between the number of available events and stations.

4.5.2 Waveform Preprocessing

Following conventional seismic signal preprocessing techniques, we remove from the waveforms any instrument response associated with the station. Be-

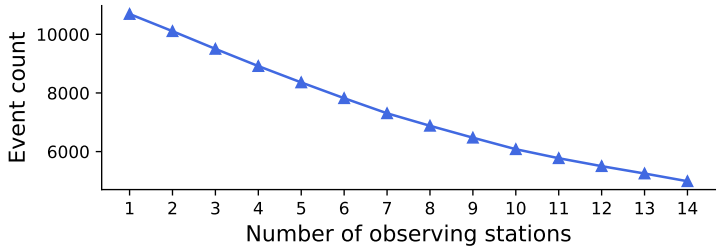

Figure 4.8: Number of available earthquake events declines for increasing number of observing stations.

sides, we convert the horizontal, vertical north-south, and vertical east-west components (Z, N, E) to horizontal, radial, and tangential components (Z, R, T). We pass the waveforms through a 0.4Hz to 10Hz bandpass filter, detrend each sample, and remove the mean. We max-normalize the data across each channel, thus retaining the relative amplitudes among components of a station. Afterward, we use the 64-scale CWT to obtain a spectral-temporal representation, as previously illustrated in Figure 4.1. Each RGB linearly spaced CWT image has $64 \times 920 \times 3$ dimensions, where the vertical, radial and tangential components are represented by red, green, and blue colors, respectively.

4.6 Experimental Setup

In this work, depth prediction is a regression task from three dimensional spectral-temporal images. Several existing methods can be used to produce depth estimates. We compare the results of SeiSMo against two classic ML-based models: CNN, LSTM; and two state-of-the-art algorithms for time series regression: Rocket [18] and XGBoost [12]. The algorithms are briefly described below.

CNN based models have been widely used in image classification, time series classification and seismic data classification [19]. We use linearly-spaced CWT images of waveforms as input to simple four layer CNN network to

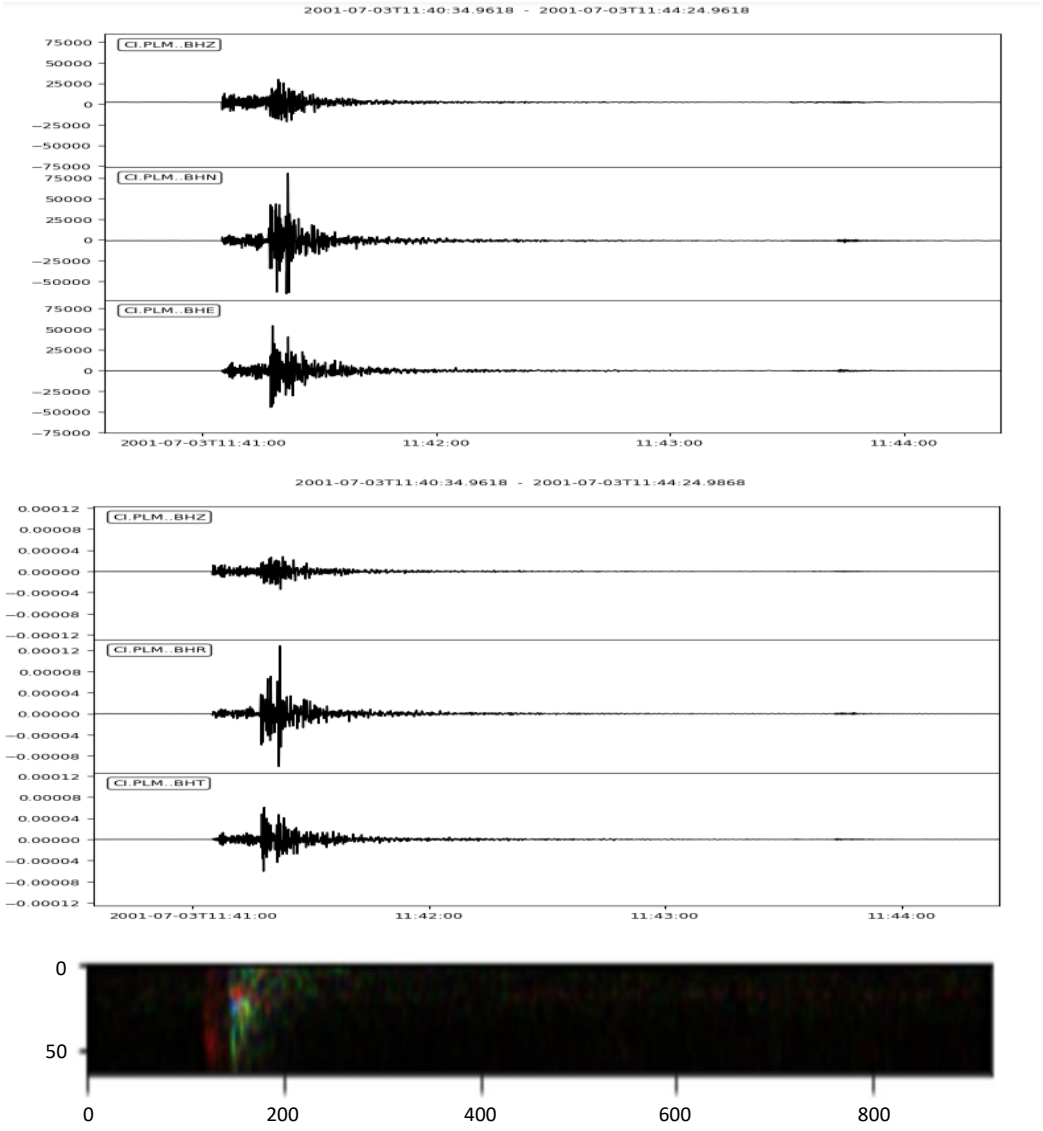

Figure 4.9: (a) 3-channel raw waveform collected from station CI.PLM (b) vertical, tangential and radial components generated from the raw waveforms (c) linearly spaced CWT image from the ZRT channels

train towards actual depths.

LSTM based models can capture long-term temporal dependencies, hence, it is very effective for time series classification and regression [24]. We flatten the CWT images to feed into a two layer LSTM network and train towards actual depth.

XGBoost (Extreme Gradient Boosting) is a decision-tree based ensemble algorithm that uses boosting technique to sequentially add new trees to the classification or regression model [12].

Rocket (Random Convolutional KErnel Transform) is originally a classifier for time series that transforms the data using a large number of random convolutional kernels that can capture basic patterns or shapes from the series. Recently, Rocket was adapted for extrinsic regression and achieves the highest overall accuracy in a comprehensive experimental evaluation [53].

Rocket and XGBoost were trained using the vertical component (Z-axis) of the waveforms in case of single-channel resolution; by concatenating the three components (Z, R, T) waveforms into a single vector in the case of multi-channel resolution; and by concatenating the three-channel waveforms from five stations for multi-station resolution. For Rocket, the number of kernels was set to 10,000.

We consider Root-Mean-Square Error (RMSE) as a loss function and Stochastic Gradient Descent (SGD) as the optimizer to train our model. The learning rate is set to 0.01 with a 10% decay per epoch. Both dropout ratio and recurrent dropout ratio for LSTM were set to 0.3 for all models. These values were set based on preliminary evaluations on training sets. In all experiments, we consider a split of 80/10/10 for training, validation, and testing after random shuffling. The results are the average of five separate training sets using five-fold cross-validation of 300 epochs each.

In addition to RMSE to measure the predictions' error, we calculate Pearson's correlation coefficient to qualitatively show the relationship between the predicted and actual depth. Note that, Pearson's correlation has no unit while RMSE is measured in km. The two metrics complement each other to demonstrate the robustness of performance evaluation.

Pearson's correlation coefficient (ρ) is defined according to Equation 4.6, in which x_i is the predicted depth and y_i is the actual depth ($1 \leq i \leq$ *number of test events*). Whereas, $\overline{y}$ and $\overline{x}$ are the average of predicted depth and average of actual depth respectively.

$$\rho = \frac{\Sigma(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\Sigma(x_i - \overline{x})^2 \Sigma(y_i - \overline{y})^2}} \tag{4.6}$$

To demonstrate the performance of SeiSMo as a binary classifier, we modify the output layer of the station aggregator and use a softmax activation function. We use Adam optimizer instead of SGD and binary cross-entropy as loss function. In the evaluation, we consider accuracy, precision, recall, and F1-score as measures. Our primary evaluation metric is accuracy and we also calculate precision, recall and F1-score.

Precision, recall, accuracy and F1 score is computed using true positive(TP), true negative(TN), false positive(FP) and false negative(FN) classification results of our binary classifier:

$$precision = \frac{TP}{TP + FP} \tag{4.7}$$

$$recall = \frac{TP}{TP + FN} \tag{4.8}$$

$$accuracy = \frac{TP + FP}{TP + FP + TN + FN} \tag{4.9}$$

$$F1 - score = 2.\frac{precision.recall}{precision + recall} \tag{4.10}$$

We train Septor using a GPU server with four Nvidia RTX2080 GPUs (total of 44GB GPU memory), 256 GB of RAM and 32-core CPU. Septor has 8.2 million trainable parameters and for 300 epochs, the server takes 1 hour and 48 minutes to train. All of our experiments are reproducible. The source code of data preprocessing and model training, waveforms and meta-data, figures, and additional results can be found in our supporting website [50].

4.7 Empirical Evaluation

4.7.1 Regression Performance

Prior works in seismology consider that the difference between local (M_L) and coda (M_C) magnitude correlates with the depths of seismic events [25]. However, we discover that such physics driven method struggle to find any correlation between depth and $M_L - M_C$ for our dataset (see Figure 4.10). This conflicting discovery motivated us to develop a data-driven method for depth estimation.

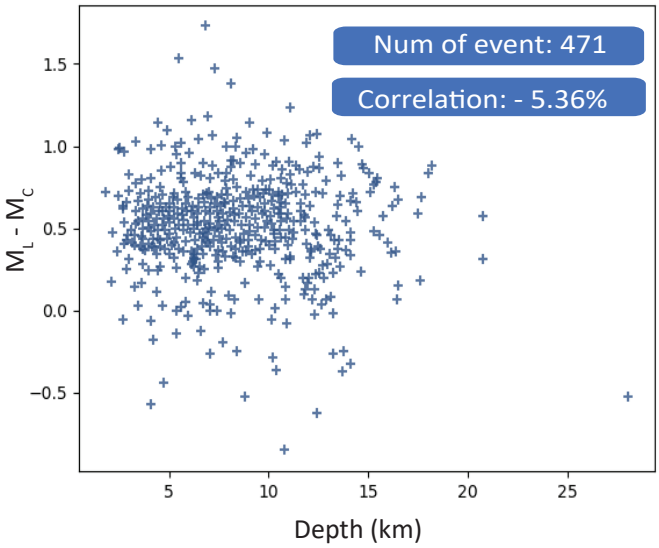

Figure 4.10: Actual depth and $M_L - M_C$ values does not show any linear correlation as advocated by literature.

In Table 4.1, we compare the performance of Septor with different rival methods and data resolution. The RMSE of Septor for depth prediction is 2.89 km, which is lower than one standard deviation (3.71 km) of depth values in the test data. Furthermore, we achieve an impressive 70.1% Pearson's correlation coefficient.

Table 4.1: Performance comparison of Septor with baseline multiple methods.

Model	Data resolution	RMSE (km)	Corr. (%)
CNN	Multi-channel	3.26	56.0
LSTM	Multi-channel	3.38	52.0
XGBoost	Single-channel	3.53	37.0
XGBoost	Multi-channel	3.58	36.0
XGBoost	Multi-station	3.39	44.3
Rocket	Single-channel	3.11	46.2
Rocket	Multi-channel	3.12	46.0
Rocket	Multi-station	3.51	36.5
Septor	**Multi-station**	**2.89**	**70.1**

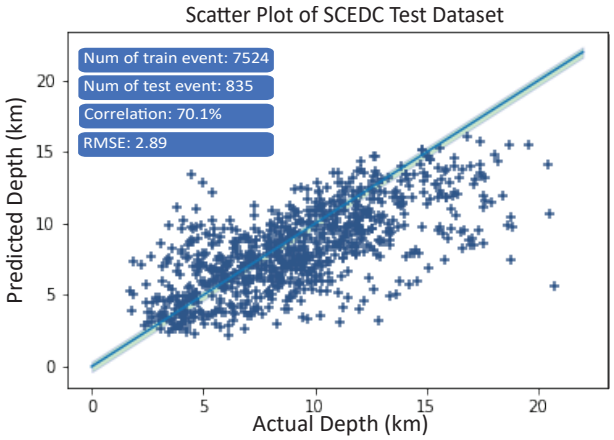

Figure 4.11: Scatter plot shows 70.1% correlation between actual vs. predicted depth on SCEDC dataset. The blue straight line represents $y = x$.

To confirm the statistical significance of our predictions, we run a t-test where we represent predicted depth with variable X, true depth with variable with Y, and our null hypothesis H_0 is: *there is no significant linear correlation between X and Y*. From the test, we found a p-value lower than

the threshold value ($\alpha = 0.05$). Therefore, we reject the null hypothesis H_0 and conclude that the correlation between X and Y is statistically significant and did not occur by chance. Our model, Septor , significantly outperforms all other data-driven methods in terms of RMSE and Pearson's Correlation. In Figure 4.11, the scatter plot shows the dots representing the predictions close to the expected $y = x$ line.

Sensitivity to the magnitudes of earthquakes

The observed noise in seismic signals depends on the earthquake's magnitude. Given that all other factors are the same, high magnitude earthquakes produce better signals than low magnitude ones. For this reason, we evaluate whether our model is sensitive to different magnitudes range, which could potentially decrease its utility.

We evaluated our model considering depth values in two distinct magnitude ranges: from 2 to 3 and from 3 to 4. The results are shown in Table 4.2. For both ranges, Pearson's correlation coefficients are more than 86% with negligible ($<< 0.05$) p-value. We achieve an RMSE of 2.216 km for magnitude range 2-3 and 2.217 km for magnitude 3-4, where the standard deviation is 3.831 km and 3.712 km, respectively. The relationship between actual and predicted depths is shown in Figure 4.12. We conclude that the model is invariant to magnitude ranges.

Table 4.2: Performance evaluation on earthquakes of two magnitude ranges. RMSE and standard deviation are in km.

Magnitude	**Test event Std Dev.**	**RMSE**	**Corr. (%)**
2 - 3	3.831	2.216	86.5
3 - 4	3.712	2.217	88.2

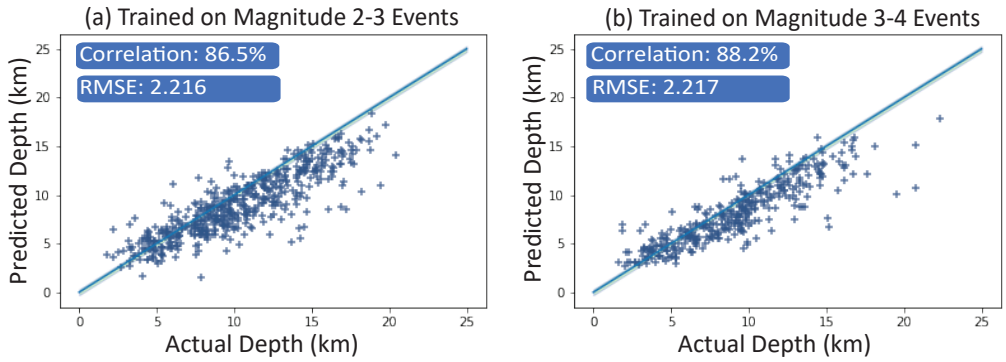

Figure 4.12: Scatter plots of predicted and actual depths for two magnitude ranges.

Sensitivity to origin location

Earth's non-uniformity is the greatest challenge to generalizable model development for seismic monitoring applications. Southern California has numerous dramatic lateral variations in seismic velocity structure, for example, with rapid transitions from the San Jacinto mountains with their deep crustal roots, to the Salton Trough, with its very thin crust, thick sediments, and high heat flow, and from areas where the crust is being sheared, to areas where it is under compression. Because of such geographic structural variability we evaluate our model on smaller sets of sub-regional events. As a simple test we divide our dataset into north and south sub-regions of Southern California and train and test using these smaller subsets. This is intended to test the performance of our model with earthquakes generated and captured within regions with at least somewhat less variation in earth structure. In Table 4.3, we show the performances on *North* and *South* datasets. RMSE values are lower than those obtained using the full dataset, and the correlation coefficients are higher for both sub-regions (Figure 4.13). This indicates that geographic restrictions help the model to perform better.

Table 4.3: Performance evaluation after splitting the dataset into two geographic locations. RMSE and standard deviation are in km.

Location	Test event Std Dev.	RMSE	Corr. (%)
North	3.065	1.907	85.4
South	3.804	2.264	85.9

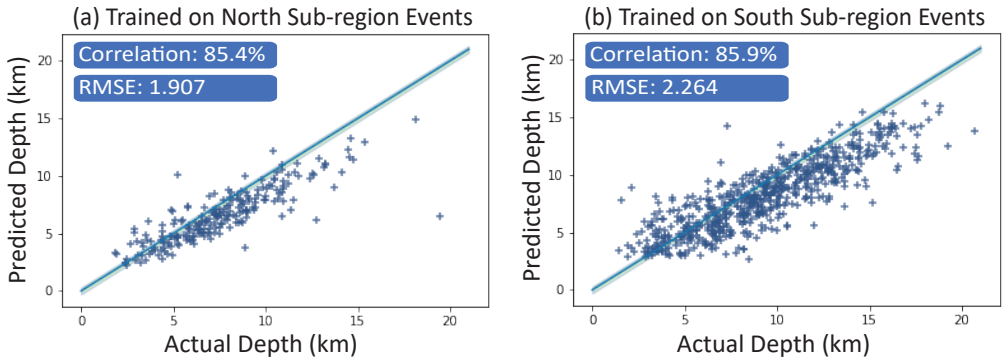

Figure 4.13: The model performs better for both North (left) and South (right) splitted dataset than whole dataset.

Effects of different data pre-processing choices

In this experiment, we evaluate the effects of our choices in the preprocessing pipeline. First, we use spectrograms instead of linearly-spaced CWTs, since this data representation is widely employed in seismology. For the same SCEDC dataset, we observe a 5% decline in Pearson's Correlation coefficient when using spectrograms. Second, we use horizontal east-west, horizontal north-south components (Z, N, E channels) instead of rotating into radial and transverse components (Z, R, T channels). In this case, we observe Pearson's Correlation coefficient declines around 4% for the same set of earthquakes. Both results are shown in Figure 4.14. This empirical evaluation confirms that our choices work better than the alternatives.

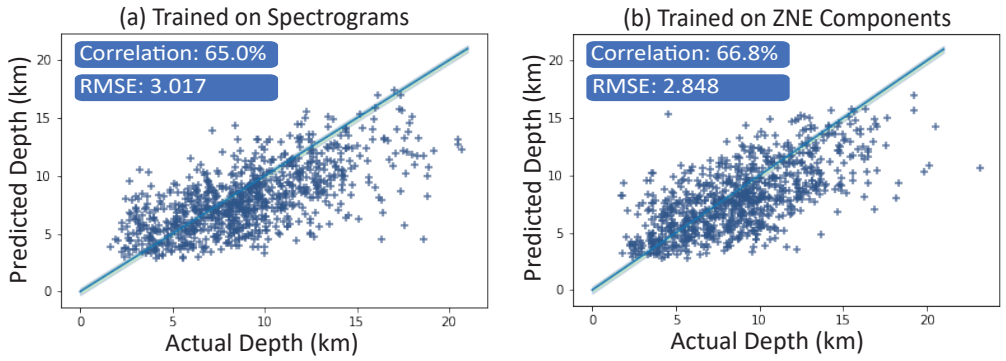

Figure 4.14: (a) Using spectrograms instead of CWTs results in a 5% performance decrease. (b) Using ZNE components of seismograms instead of rotating into ZRT components results in a 4% performance decrease.

Sensitivity to distance between stations and epicenters.

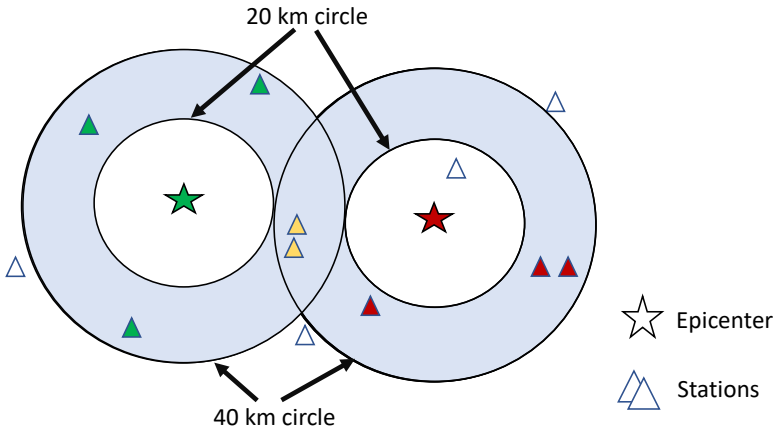

Figure 4.15: Example of the selection of earthquakes based on distance radius.

The performance of depth prediction depends on the distance between the source event and observing stations. Information regarding depth in a waveform may vary with distance from the epicenter to the station. In this section,

we demonstrate the performance of our model based on stations that fall into a fixed distance range. For example, if the distance range is 20-40 km, all the five stations for each earthquake are between 20 km and 40 km from the epicenter. This example is illustrated in Figure 4.15, in which we draw a doughnut shape around each earthquake epicenter with an inner circle of 20 km and an outer circle of 40 km. Then, we use the stations that fall within the doughnut-shaped region. Since Septor works after the Phase Association step, we have the epicenter distance information beforehand. Table 4.4 shows the performance in different distance ranges. Performance improves with distance up to the 40 to 60 km range, likely because the different seismic phases have a great deal of overlap at short distances. Performance then generally decreases with distance, possibly due to decreased signal-to-noise ratios and the accumulation of propagation effects on the waveforms.

Table 4.4: Performance evaluation after separating stations for train and test dataset. RMSE and standard deviation are in km.

Distance band (km)	**Test event Std Dev.**	**RMSE**	**Corr. (%)**
0 - 20	4.258	4.766	30.3
20 - 40	3.977	4.506	14.1
40 - 60	3.889	3.665	38.8
60 - 80	4.014	4.600	36.1
80 - 100	3.882	4.088	8.5
100 - 120	3.803	5.969	18.1

4.7.2 Classification Performance

In this section, we evaluate the performance of our network to classify deep and shallow earthquakes. Natural earthquakes can originate anywhere from the surface up to 700 km deep into the earth. United States Geological Survey (USGS) [55] defines three categories of earthquakes depending on depth:

Table 4.5: Performance of Septor as a binary classifier for a subset of training data based on source to origin distance. The results are shown in percentage (%).

Distance (km)	Accuracy	Precision	Recall	F1-score
0-20	88.4	91.8	89.3	90.5
20-40	75.1	85.2	89.6	87.3
40-60	74.1	92.0	74.4	82.3
60-80	73.1	84.4	92.6	88.3
80-100	70.6	91.0	88.6	89.8
100-120	73.7	89.3	93.3	91.3
>120	67.8	67.1	68.9	68.0
Full dataset	73.0	78.2	75.5	86.5

(i) Shallow earthquakes originate between 0 to 70 km; *(ii)* Earthquakes that occur between 70 and 300 km are intermediate; *(iii)* Deep earthquakes originate deeper than 300 km into the ground. However, the cutoff depth to separate shallow and deep earthquakes can vary depending on the application. For most seismic monitoring purposes, it is sufficient to find out whether the earthquake occurred on the earth's surface or not. Therefore, an earthquake with a depth of 10 km can be labeled as a deep earthquake as man-made earthquakes (i.e., mining blasts, borehole shots, or nuclear explosions) can never originate 10 km below the surface.

In our experiments, we divide our full dataset into two balanced subsets of shallow and deep earthquake events considering the median depth (8.73 km) as the cutoff depth for shallow and deep earthquakes. We consider accuracy as a performance measure. The results are shown in the last row of Table 4.5. The confusion matrix is shown in Figure 4.16 (left).

Sensitivity to distance between station and epicenter

In this section, we demonstrate the performance of our binary classifier based on stations that fall into a fixed distance range.

Based on our full dataset, we take seven different subsets depending on distance ranges. In Table 4.5, we show detailed results of our binary classifier for various distance ranges. Our results in Figure 4.16 (right) show that the accuracy for both deep and shallow earthquakes drop as the distance increases. Although the accuracy graph is not monotonic, we expect a monotonic decrease in the accuracy with increasing distance for larger datasets. We leave it for our future work.

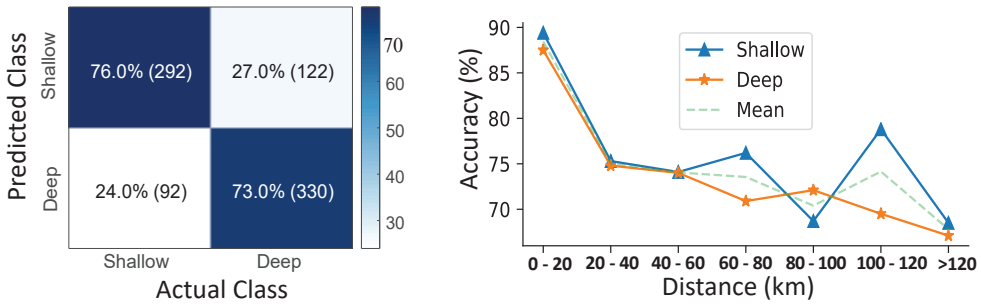

Figure 4.16: (left) Confusion matrix for binary classification for deep and shallow earthquakes. (right) Accuracy of the binary classifier drops with the source to origin distance of training data.

4.8 Case Study: Novel Geographical region

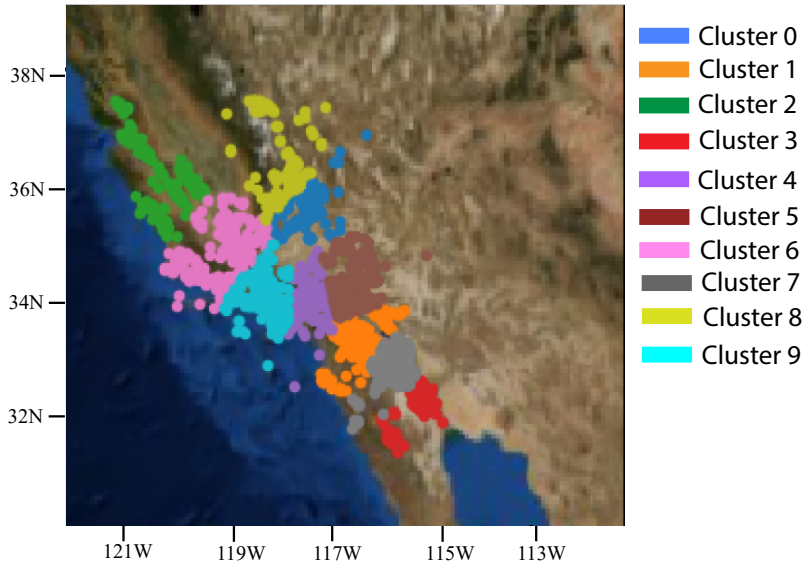

Figure 4.17: Locations of ten clusters found by DBSCAN based on epicenters of the events.

The ultimate goal of an ML-based depth estimator is to perform at a new location with a new set of stations. We ask if Septor can generalize to a novel geographical region without seeing any training instance from that region. We perform three different tests to evaluate generalizability of Septor.

No common source or station between train and test set

To simulate novel source-station scenarios, we sample the events and and stations in our dataset uniformly to create a test set, and use the remaining events and stations for training.

In Figure 4.18(a) we show the scatter plot results for separated stations. The results show a clear decrease in correlation to 44.8% from 70.1%, while the RMSE increases to 3.53 km. Although we create disjoint training and test

sets with no common station and event source, the overall dataset is limited within the Southern California region, resulting a positive correlation.

Spatial separation between train and test sets

We cluster the events from SCEDC using the DBSCAN algorithm [48]. First, the earthquake events are clustered into ten regions based on epicenter (i.e., latitude and longitude), and then we perform a leave-one-out test. The clusters are shown in Figure 4.17 on the Southern California map.

In our test, we take each of these ten clusters in turn for testing, and use the remaining clusters for training; excluding the clusters adjacent to the test cluster. For example, if we take cluster 2 (green dots in Figure 4.17) as our test cluster, we train on clusters 0, 1, 3, 4, 5, 7, and 9. We exclude clusters 6 and 8 from this experiment as they are adjacent to our test cluster (2). The combined result for all test clusters are shown in Figure 4.18(b). RMSE of actual and predicted depth increases to 4.836 after spatial separation.

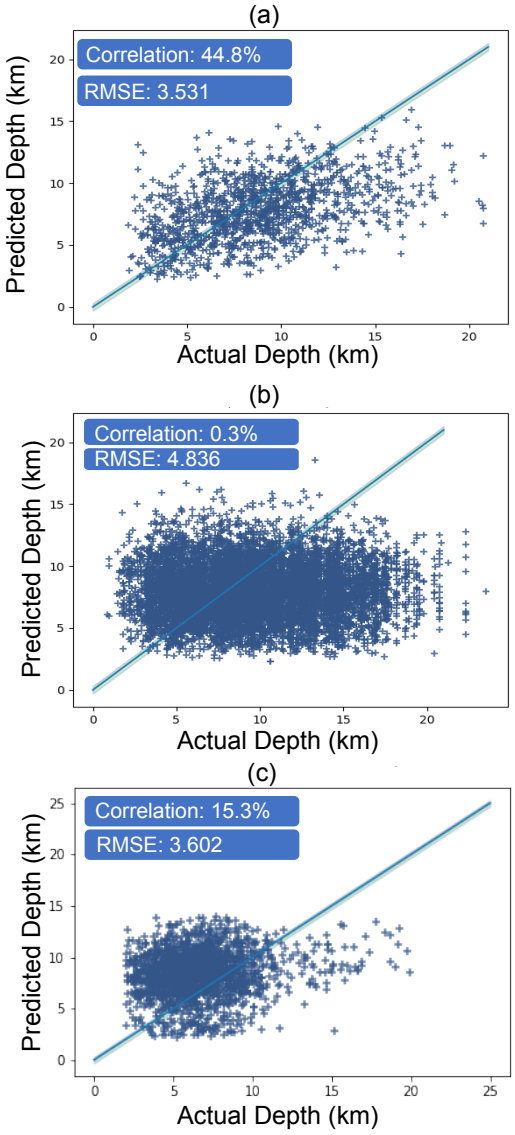

Figure 4.18: (a) Scatter plot for the model trained and tested on Southern California data after uniformly separating the stations in training and testing datasets. (b) Scatter plot of ten test cluster combined after spatial separation. (c) Scatter plot for the model trained on Southern California dataset and tested on Northern California dataset. In each sub-figures,the blue straight line represents $y = x$.

Test set from completely new region

We collected 1,777 natural earthquake events, each with seismograms from the five closest stations from the Northern California (NCEDC) seismic data center. We perform similar preprocessing to the seismograms and test our original model trained using the events from the SCEDC. The results are shown in Figure 4.18 (c). The model underperforms on this test dataset achieving a RMSE of 3.602 km with Pearson's Correlation coefficient of 15.3%. This is not surprising since Southern California and Northern California have different earth structures. Due to geographical dissimilarity, SCEDC and NCEDC seismograms generate different feature spaces, leading to poor model transferability across regions.

4.9 Conclusion and Future Work

We present a two-level neural network model (Septor) to predict the depth of a seismic event using the waveforms recorded at multiple nearby stations. Septor achieves an impressive RMSE of 2.89 km, suitable for discriminating man-made events from most earthquakes. We demonstrate Septor 's effectiveness in predicting depths of low magnitude earthquakes, which will enable fine-grained monitoring of underground events. When working in a classification scenario, our model can identify deep and shallow seismic events with 86.5% F1-score. For our future work, we wish to evaluate Septor with larger datasets.

Chapter 5

Conclusion and Future Work

The main goal of this dissertation is to automate the seismic data processing pipeline by analyzing spatio-temporal features in seismic waveforms and introducing state-of-the-art data mining and machine learning techniques. From seismic event detection to event association, each step of the seismic data processing pipeline is heavily dependent on expert analysts. In this thesis, we discuss the limitations of such legacy pipeline We also discuss the challenges to solve these problems. The primary challenge to devise a data-driven solution is the inadequate and inaccurate ground truth. To solve the shortcomings of existing solutions and automate the process, we leveraged state-of-the-art data mining, machine learning, and digital signal processing techniques.

In Chapter 2, we proposed and implemented a semi-supervised time series motif discovery system (SeiSMo) for seismic signal detection. SeiSMo creates a nearest neighbor graph using cataloged seismic events to discover novel seismic activity from static seismic sensor data. SeiSMo is good at spotting unique low magnitude events, which were not cataloged because analysts were not certain about their origin and characteristics. SeiSMo was able to discover hundreds of novel seismic events in six real-world and curated datasets.

We demonstrate a comprehensive seismic data repository in Chapter 3. The repository contains 10-year-long catalog and waveform data that was captured and curated by Air Force Technical Applications Center (AFTAC). The repository can yield complex query results and corresponding waveforms within seconds. We have used this repository for multiple research projects (i.e. seismic phase identifications, seismic aftershock detection). The repository is also facilitating with data sources for multiple Department of Defense (DoD) research projects.

A seismic depth prediction system (Septor) from raw waveforms is illustrated in Chapter 4. Septor is a two-hierarchy not-so-deep neural network that can be trained using continuous wavelet transform images from seismic waveforms. We demonstrate Septor 's effectiveness in predicting depths of low magnitude earthquakes, which will enable fine-grained monitoring of underground events. We also show that Septor can be converted into a classifier to separate deep and shallow earthquakes. We describe and discuss experimental findings under various conditional settings differentiating shallow-deep, large-small, and mountain-valley earthquakes.

From the research work presented here, there are many possible directions one can follow. We briefly describe a few of them in this section:

1. **Analyze detected events to get better insight:** Analyzing novel events detected by SeiSMo can lead to producing a high-quality seismic catalog. Which in turn would help us better understand earth's structure near faults, volcanoes, etc.

2. **Integrate SeiSMo for streams of seismic sensor data:** For real-time detection, SeiSMo can be integrated with seismic data streams. This would lead us to fully automated earthquake detection.

3. **Explore transferability across regions for depth prediction:** In our experiments on Septor , we describe the results by training and testing on earthquakes generated from the same geographical region.

However, it would be interesting to see the results across regions for train and test datasets.

4. **Explore possibilities of other research projects using the IMS data repository:** As the UNM data repository is being used by multiple academic research groups and by the MINEM consortium, we believe the data repository would open up possibilities for other research projects that requires precisely annotated seismic sensor data.

Bibliography

[1] Deepdetect dataset public repository., 2018.

[2] The facilities of iris data services, and specifically the iris data management center, were used for access to waveforms, related metadata, and/or derived products used in this study. iris data services are funded through the seismological facilities for the advancement of geoscience and earthscope (sage) proposal of the national science foundation under cooperative agreement ear-1261681, 2018.

[3] Ncedc (2014), northern california earthquake data center. uc berkeley seismological laboratory. dataset. doi:10.7932/ncedc., 2018.

[4] Scedc (2013): Southern california earthquake center. caltech.dataset. doi:10.7909/c3wd3xh1, 2018.

[5] Public github repository to download code, spread- sheet, slides and datasets, 2019., 2019.

[6] Anthony Bagnall, Jason Lines, Aaron Bostrom, James Large, and Eamonn Keogh. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Min. Knowl. Discov.*, 31(3):606–660, 2017.

[7] Anthony Bagnall, Jason Lines, Jon Hills, and Aaron Bostrom. Time-series classification with cote: The collective of transformation-based

ensembles. *IEEE Trans. on Knowl. and Data Eng.*, 27(9):2522–2535, September 2015.

[8] Sugato Basu, Arindam Banerjee, and Raymond J Mooney. Semi-supervised Clustering by Seeding. In *Proceedings of the Nineteenth International Conference on Machine Learning*, ICML '02, pages 27–34, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

[9] Nurjahan Begum and Eamonn Keogh. Rare time series motif discovery from unbounded streams. *Proceedings of the VLDB Endowment*, 8(2):149–160, 10 2014.

[10] U Berkeley. The NCSS now reports earthquake depth relative to the geoid (sea level), 10 2015.

[11] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. Lof: Identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, SIGMOD '00, pages 93–104, New York, NY, USA, 2000. ACM.

[12] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *SIGKDD*, pages 785–794. ACM, 2016.

[13] Yanping Chen, Bing Hu, Eamonn Keogh, and Gustavo E.A.P.A Batista. DTW-D: Time Series Semi-supervised Learning from a Single Example. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '13*, page 383, New York, New York, USA, 8 2013. ACM Press.

[14] Farhan Asif Chowdhury, M Ashraf Siddiquee, Glenn Eli Baker, and Abdullah Mueen. Faser: Seismic phase identifier for automated monitoring. In *SIGKDD*, pages 2714–2721, 2021.

[15] CTBTO. Waveform data processing and analysis, 10 2015.

[16] Hoang Anh Dau and Eamonn Keogh. Matrix Profile V: A Generic Technique to Incorporate Domain Knowledge into Motif Discovery. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '17*, pages 125–134, New York, New York, USA, 2017. ACM Press.

[17] Hoang Anh Dau, Eamonn Keogh, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, Yanping, Bing Hu, Nurjahan Begum, Anthony Bagnall, Abdullah Mueen, Gustavo Batista, and Hexagon-ML. The ucr time series classification archive, 2018. https://www.cs.ucr.edu/ eamonn/time$_s eries_d ata_2$018/.

[18] Angus Dempster, François Petitjean, and Geoffrey I Webb. Rocket: exceptionally fast and accurate time series classification using random convolutional kernels. *Data Min. Knowl. Discov.*, 34(5):1454–1495, 2020.

[19] Ramin MH Dokht, Honn Kao, Ryan Visser, and Brindley Smith. Seismic event and phase detection using time–frequency representation and convolutional neural networks. *Seismological Research Letters*, 90(2A):481–490, 2019.

[20] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise, 1996.

[21] S Gao, M. T Young, J. X Qiu, J. B. Christian, P. A. Fearn, G. D. Tourassi, and A Ramanthan. Hierarchical attention networks for information extraction from cancer pathology reports. pages 321–330. Journal of the American Medical Informatics Association : JAMIA, 2018.

[22] Peter Goldstein. Slopes of p- to s-wave spectral ratios—a broadband regional seismic discriminant and a physical model. *Geophysical Research Letters*, 22(23):3147–3150, 1995.

[23] Egill Hauksson, Wenzheng Yang, and Peter M Shearer. Waveform relocated earthquake catalog for southern california (1981 to june 2011). *BSSA*, 102(5):2239–2244, 2012.

[24] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–1780, 1997.

[25] Monique M. Holt, Keith D. Koper, William Yeck, Sebastiano D'Amico, Zongshan Li, J. Mark Hale, and Relu Burlacu. On the Portability of ML–Mc as a Depth Discriminant for Small Seismic Events Recorded at Local Distances. *Bulletin of the Seismological Society of America*, 109(5):1661–1673, 09 2019.

[26] Rob J Hyndman. A brief history of forecasting competitions. *International Journal of Forecasting*, 36(1):7–14, 2020.

[27] Washington University in St. Louis. The wfdidc table. Available at http://nappe.wustl.edu/antelope/css-formats/wfdisc.htm (June 2, 2022), 2022.

[28] Iris. How Often Do Earthquakes Occur?, 10 2015.

[29] Alan L. Kafka. Rg as a depth discriminant for earthquakes and explosions: A case study in New England. *Bulletin of the Seismological Society of America*, 80(2):373–394, 04 1990.

[30] K. D. Koper, J. C. Pechmann, R. Burlacu, K. L. Pankow, J. R. Stein, J. M. Hale, P. Roberson, and M. K. McCarter. Magnitude Based Discrimination of Manmade Seismic Events From Naturally Occurring Earthquakes in Utah, USA. In *AGU Fall Meeting Abstracts*, volume 2016, pages S31A–2721, December 2016.

[31] Guoqing Lin, Peter M Shearer, and Egill Hauksson. Applying a three-dimensional velocity model, waveform cross correlation, and cluster analysis to locate southern california seismicity from 1981 to 2005. *Journal of Geophysical Research: Solid Earth*, 112(B12):1–14, 2007.

[32] S Mostafa Mousavi, Weiqiang Zhu, Yixiao Sheng, and Gregory C Beroza. Cred: A deep residual network of convolutional and recurrent units for earthquake signal detection. *Scientific reports*, 9(1):1–14, 2019.

[33] Abdullah Mueen. Enumeration of time series motifs of all lengths. In *Proceedings - IEEE International Conference on Data Mining, ICDM*, ICDM, pages 547–556, 2013.

[34] Abdullah Mueen and Eamonn Keogh. Online discovery and maintenance of time series motifs. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '10*, page 1089, New York, New York, USA, 7 2010. ACM Press.

[35] Abdullah Mueen, Eamonn Keogh, and Nima Bigdely-Shamlo. Finding time series motifs in disk-resident data. In *Proceedings - IEEE International Conference on Data Mining, ICDM*, pages 367–376, 2009.

[36] Abdullah Mueen, Eamonn Keogh, Qiang Zhu, Sydney Cash, and Brandon Westover. Exact Discovery of Time Series Motifs. In *Proceedings of the 2009 SIAM International Conference on Data Mining*, pages 473–484, 2009.

[37] Abdullah Mueen, Krishnamurthy Viswanathan, Chetan Gupta, and Eamonn Keogh. The Fastest Similarity Search Algorithm for Time Series Subsequences under Euclidean Distance, 8 2015.

[38] Luis H Ochoa, Luis F Niño, and Carlos A Vargas. Fast determination of earthquake depth using seismic records of a single station, implementing machine learning techniques. *Ingeniería e Investigación*, 38(2):97–103, 2018.

[39] Division of Geological and California Institute of Technology Planetary Sciences. Scsn catalog change history, 2020. https://scedc.caltech.edu/eq-catalogs/change-history.html.

[40] Colin T. O'Rourke, G. Eli Baker, and Anne F. Sheehan. Using p/s amplitude ratios for seismic discrimination at local distancesusing p/s amplitude ratios for seismic discrimination at local distances. *Bulletin of the Seismological Society of America*, 106(5):2320, 2016.

[41] CT O'Rourke, AF Sheehan, EA Erslev, and KC Miller. Estimating basin thickness using a high-density passive-source geophone array. *Earth and Planetary Science Letters*, 402:120–126, 2014.

[42] Thibaut Perol, Michaël Gharbi, and Marine Denolle. Convolutional neural network for earthquake detection and location. *Science Advances*, 4(2), 2018.

[43] Franco P. Preparata and Michael Ian Shamos. *Computational Geometry*. Springer New York, New York, NY, 1985.

[44] Thanawin Rakthanmanon, Bilson Campana, Abdullah Mueen, Gustavo Batista, Brandon Westover, Qiang Zhu, Jesin Zakaria, and Eamonn Keogh. Searching and mining trillions of time series subsequences under dynamic time warping. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 262–270, 2012.

[45] Thanawin Rakthanmanon and E Keogh. Fast shapelets: A scalable algorithm for discovering time series shapelets. In *Proceedings of the 2013 SIAM International Conference on Data Mining. 2013.*, pages 668–676, 2013.

[46] SCEDC. Station metadata and maps. southern california earthquake data center at caltech. Available at https://scedc.caltech.edu/data/station/index.html (September 2, 2021), 2021.

[47] David P Schaff, Götz HR Bokelmann, Gregory C Beroza, Felix Waldhauser, and William L Ellsworth. High-resolution image of calav-

eras fault seismicity. *Journal of Geophysical Research: Solid Earth*, 107(B9):ESE–5, 2002.

[48] Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. Dbscan revisited, revisited: why and how you should (still) use dbscan. *ACM Transactions on Database Systems (TODS)*, 42(3):1–21, 2017.

[49] Y. Shimshoni and N. Intrator. Classification of seismic signals by integrating ensembles of neural networks. *IEEE Transactions on Signal Processing*, 46(5):1194–1201, 1998.

[50] M. Ashraf Siddiquee, V. M. A. Souza, and Abdullah Mueen. Public github repository to download code, spread- sheet, slides and datasets, 2022. https://github.com/mashrafsiddiq/Septor.

[51] William Spence, Stuart A Sipkin, and George L Choy. Measuring the size of an earthquake. *Earthquake Information Bulletin (USGS)*, 21(1):58–63, 1989.

[52] Alexandra Stefan, Vassilis Athitsos, and Gautam Das. The Move-Split-Merge Metric for Time Series. *IEEE Transactions on Knowledge and Data Engineering*, 25(6):1425–1438, 6 2013.

[53] Chang Wei Tan, Christoph Bergmeir, François Petitjean, and Geoffrey I Webb. Time series extrinsic regression. *Data Mining and Knowledge Discovery*, 35(3):1032–1060, 2021.

[54] Daniel T. Trugman and Peter M. Shearer. GrowClust: A Hierarchical Clustering Algorithm for Relative Earthquake Relocation, with Application to the Spanish Springs and Sheldon, Nevada, Earthquake Sequences. *Seismological Research Letters*, 88(2A):379–391, 02 2017.

[55] USGS. United states geological survey. Available at https://www.usgs.gov/ (September 2, 2021), 2021.

[56] Michail Vlachos, Marios Hadjieleftheriou, Dimitrios Gunopulos, and Eamonn Keogh. Indexing Multi-dimensional Time-series with Support for Multiple Distance Measures. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, pages 216–225, New York, NY, USA, 2003. ACM.

[57] Felix Waldhauser and David P. Schaff. Large-scale relocation of two decades of northern california seismicity using cross-correlation and double-difference methods. *Journal of Geophysical Research: Solid Earth*, 113(B8), 2008.

[58] F. Rall Walsh and Mark D. Zoback. Oklahoma's recent earthquakes and saltwater disposal. *Science Advances*, 1(5), 2015.

[59] Li Wei and Eamonn Keogh. Semi-supervised time series classification. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '06*, page 748, New York, New York, USA, 2006. ACM Press.

[60] Matthew Weingarten, Shemin Ge, Jonathan W Godt, Barbara A Bekins, and Justin L Rubinstein. High-rate injection is associated with the increase in us mid-continent seismicity. *Science*, 348(6241):1336–1340, 2015.

[61] Lindsay L Worthington, Kate C Miller, Eric A Erslev, Megan L Anderson, Kevin R Chamberlain, Anne F Sheehan, William L Yeck, Steven H Harder, and Christine S Siddoway. Crustal structure of the bighorn mountains region: Precambrian influence on laramide shortening and uplift in north-central wyoming. *Tectonics*, 35(1):208–236, 2016.

[62] Y. Wu, Y. Lin, Z. Zhou, D. C. Bolton, J. Liu, and P. Johnson. Deepdetect: A cascaded region-based densely connected network for seismic event detection. *IEEE Transactions on Geoscience and Remote Sensing*, 57(1):62–75, Jan 2019.

[63] De-He Yang, Xin Zhou, Xiu-Ying Wang, and Jian-Ping Huang. Mirco-earthquake source depth detection using machine learning techniques. *Information Sciences*, 544:325–342, 2021.

[64] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, California, June 2016. Association for Computational Linguistics.

[65] William L Yeck, Anne F Sheehan, Megan L Anderson, Eric A Erslev, Kate C Miller, and Christine S Siddoway. Structure of the bighorn mountain region, wyoming, from teleseismic receiver function analysis: Implications for the kinematics of laramide shortening. *Journal of Geophysical Research: Solid Earth*, 119(9):7028–7042, 2014.

[66] Chin-Chia Michael Yeh, Nickolas Kavantzas, and Eamonn Keogh. Matrix Profile VI: Meaningful Multidimensional Motif Discovery. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 565–574. IEEE, 11 2017.

[67] Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Diego Furtado Silva, Abdullah Mueen, and Eamonn Keogh. Matrix Profile I: All Pairs Similarity Joins for Time Series: A Unifying View That Includes Motifs, Discords and Shapelets. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 1317–1322. IEEE, 12 2016.

[68] Clara E. Yoon, Ossian O'Reilly, Karianne J. Bergen, and Gregory C. Beroza. Earthquake detection through computationally efficient similarity search. *Science Advances*, 1(11), 2015.

[69] Y. Zhu, Z. Zimmerman, N.S. Senobari, C.-C.M. Yeh, G. Funning, A. Mueen, P. Brisk, and E. Keogh. Matrix profile II: Exploiting a novel

algorithm and GPUs to break the one hundred million barrier for time series motifs and joins. In *Proceedings - IEEE International Conference on Data Mining, ICDM*, 2017.