SeiSMo: Semi-supervised Time Series Motif Discovery for Seismic Signal Detection

M Ashraf Siddiquee
The University of New Mexico
Albuquerque, New Mexico
siddiquee@unm.edu

Zeinab Akhavan
The University of New Mexico
Albuquerque, New Mexico
zakhavan@unm.edu

Abdullah Mueen
The University of New Mexico
Albuquerque, New Mexico
mueen@unm.edu

ABSTRACT

Unlike semi-supervised clustering, classification and rule discovery; semi-supervised motif discovery is a surprisingly unexplored area in data mining. *Semi-supervised Motif Discovery* finds hidden patterns in long time series when a few arbitrarily known patterns are given. A naive approach is to exploit the known patterns and perform similarity search within a radius of the patterns. However, this method would find only similar shapes and would be limited in discovering new shapes. In contrast, traditional unsupervised motif discovery algorithms detect new shapes, while missing some patterns because the given information is not utilized.

We propose a semi-supervised motif discovery algorithm that forms a nearest neighbor graph to identify chains of nearest neighbors from the given events. We demonstrate that the chains are likely to identify hidden patterns in the data. We have applied the method to find novel events in several geoscientific datasets more accurately than existing methods.

CCS CONCEPTS

• **Information systems** → *Nearest-neighbor search*; • **Applied computing** → *Earth and atmospheric sciences*;

KEYWORDS

motif discovery, earthquakes, seismology, time series analysis, nearest neighbor graph

ACM Reference format:

M Ashraf Siddiquee, Zeinab Akhavan, and Abdullah Mueen. 2019. SeiSMo: Semi-supervised Time Series Motif Discovery for Seismic Signal Detection. In *Proceedings of The 28th ACM International Conference on Information and Knowledge Management, Beijing, China, November 3–7, 2019 (CIKM '19)*, 10 pages.

<https://doi.org/10.1145/3357384.3357931>

1 INTRODUCTION

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

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CIKM '19, November 3–7, 2019, Beijing, China
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6976-3/19/11...\$15.00
<https://doi.org/10.1145/3357384.3357931>

that repeats significantly, either with high number of occurrences or with high similarity among the occurrences, is called a motif. For example, in Figure 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 less similar occurrences [15][33]. 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.

Figure 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 the cost of expensive because computation of quadratic time complexity. Moreover, knowing the middle/green sinusoids as motif does not help selecting threshold for further similarity search.

Semi-supervised motif discovery has the potential to enable data mining 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 paper, we propose a semi-supervised motif discovery algorithm. We name our technique **SeiSMo** (**S**emi-**S**upervised **M**otif). 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 observations.
- SeiSMo works with any distance measure.
- SeiSMo is deterministic and less sensitive to parameters.
- 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.

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 [6][9], 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(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(a), and then search the region within a radius/threshold, typically proportional to the closest pair distance. In the Figure 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(b), all of the points in the dense lump are closer to the stars, hence, they are very well detectable by similarity search [33] based on the same radius as in 2(a). In Figure 2(c), no point is closer to the stars, hence, no motif would be found. In Figure 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.

To draw a quick contrast and build an intuition, we enlarge Figure 2(d) in 2(e), where SeiSMo can find all five occurrences in all of the scenarios (b-d). An arrow in Figure 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

Figure 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).

all nodes on these chains as occurrences of the given motif, we correctly identify the five points without any threshold parameter.

In the rest of the paper, we define the problem, and discuss the algorithm and optimization techniques (Section 4-5). We empirically evaluate effectiveness, efficiency, and parameter sensitivity (Section 6). We demonstrate utility of SeiSMo in four seismic datasets from three different states (Section 7-9).

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 $n \gg m$. 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 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 [33][35]. In Figure 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: 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,

$$\min_{|j-i| \geq m} d(\text{zNorm}(S_{i,m}), \text{zNorm}(S_{j,m})) \quad (1)$$

Figure 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.

Here, d can be any distance function defined to calculate distance between two equi-length time series. Typical distance functions are Euclidean distance [16], Dynamic Time Warping distance [21], Longest Common Subsequences distance [25], and Move-Split-Merge distance [24]. $zNorm$ 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 [16] and DTW using UCR_Suite [21].

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 and Figure 4) on any confocal path leading to the sink. The support of the sinks in Figure 2(e) are zero and two. See Figure 4 for more examples.

Figure 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(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.

3 RELATED WORK

Searching for the nearest neighbor of a query subsequence is done under Euclidean [16] or Dynamic Time Warping [21] 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 [15][13][12], focusing on various operational aspects of the tasks such as exact motifs [15], online motifs [13], variable length motifs [12], multi-dimensional motifs [32] and rare motifs [7]. Domain knowledge has been incorporated in motif discovery in the form of annotation vectors [10]; 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 [27][9]. 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 on a 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 [11]) and outlier detection (e.g., Local Outlier Factor [8]). 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 [20]. 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 [16]. 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 models cannot detect[19]. 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.

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 [22][14]. 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 , both are more than $[O_t m]$, where m is the maximum of the lengths of u, v, p and q .

- 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 losing 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:    $neighborNode \leftarrow NN(currentNode)$ 
4:    $neighborOfNeighbor \leftarrow NN(neighborNode)$ 
5:   while Overlap between  $neighborOfNeighbor$  and  $currentNode < O_t$  do
6:      $currentNode \leftarrow neighborNode$ 
7:      $neighborNode \leftarrow neighborOfNeighbor$ 
8:      $neighborOfNeighbor \leftarrow NN(neighborNode)$ 
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 $[O_t m]$, where m is the maximum of the lengths of u, v, p and q .

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 [21] 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.

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.

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 [30] 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 CLDRE 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.

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.

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 [33]. 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 parameter.

Table 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%

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.

6.3 Comparison with Existing Technique

6.3.1 Comparison to Methods in Seismology. Although SeiSMo is more robust than 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 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 more 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.

6.3.2 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 [9], 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

Figure 5: (left) Comparison to ConvNetQuake [19] and similarity search within a radius (RS) on Oklahoma dataset. (right) Comparison to FAST [34] and similarity search within a radius (RS) on California dataset.

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

In Figure 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 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.

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 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× 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 technique [33] 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 [33]); while SeiSMo finishes in 25 seconds.

Figure 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 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.

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 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 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 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.

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.

6.7 Evaluation

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 10, 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.

7 NATURAL SEISMIC EVENTS IN CALIFORNIA

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 (Incorporated Research Institutions for Seismology) catalogs while they are important for various geo-scientific analysis.

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 [23]. 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

Figure 9: Randomly picked results of California dataset. Top row of events is cataloged. Middle row of events is detected by FAST [34]. And bottom row of events is detected by SeiSMo .

within 100 miles of the station. FAST [34] 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 9, 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.

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 (CLDRE) 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.

Figure 10: 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 row. We observe the presence of the event in all six waveforms, thus, validating the detection.

Table 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. 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.

Figure 11: Results of Oklahoma dataset shows that, SeiSMo detects lower magnitude events than those in the IRIS catalog.

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 [26] [28]. In ConvNetQuake [19], 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 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.

9 SEISMICITY DUE TO CONTROLLED EXPLOSION IN WYOMING

The mining industry in Wyoming uses borehole mining technology [29], which creates very low energy yet significant seismic footprint in nearby stations across the Bighorn Mountain [31] [17]. We

Figure 12: Some randomly picked events from the Wyoming dataset results. Top row of events is known. Bottom row of events is detected by SeiSMo.

collected a dataset from the experiment of [18], 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 37 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 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.

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.

11 ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 1527127 and Air Force Research Laboratory under contract no. 17C0024

REFERENCES

- [1] 2018. DeepDetect dataset public repository. (2018). <https://sites.psu.edu/chasbolton/>
- [2] 2018. 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). <https://www.iris.edu/hq/>
- [3] 2018. NCEDC (2014), Northern California Earthquake Data Center. UC Berkeley Seismological Laboratory. Dataset. doi:10.7932/NCEDC. (2018). <http://ncedc.org/>
- [4] 2018. SCEDC (2013): Southern California Earthquake Center. Caltech.Dataset. doi:10.7909/C3WD3xH1. (2018). <http://scedc.caltech.edu>
- [5] 2019. Public github repository to download code, spread- sheet, slides and datasets, 2019. (2019). <https://github.com/mashrafsiddiq/SeiSMo>

- [6] Sugato Basu, Arindam Banerjee, and Raymond J Mooney. 2002. Semi-supervised Clustering by Seeding. In *Proceedings of the Nineteenth International Conference on Machine Learning (ICML '02)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 27–34. <http://dl.acm.org/citation.cfm?id=645531.656012>
- [7] Nurjahan Begum and Eamonn Keogh. 2014. Rare time series motif discovery from unbounded streams. *Proceedings of the VLDB Endowment* 8, 2 (10 2014), 149–160. <https://doi.org/10.14778/2735471.2735476>
- [8] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. 2000. LOF: Identifying Density-based Local Outliers. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data (SIGMOD '00)*. ACM, New York, NY, USA, 93–104. <https://doi.org/10.1145/342009.335388>
- [9] Yanping Chen, Bing Hu, Eamonn Keogh, and Gustavo E.A.P.A Batista. 2013. 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*. ACM Press, New York, New York, USA, 383. <https://doi.org/10.1145/2487575.2487633>
- [10] Hoang Anh Dau and Eamonn Keogh. 2017. 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*. ACM Press, New York, New York, USA, 125–134. <https://doi.org/10.1145/3097983.3097993>
- [11] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. (1996), 226–231 pages. <https://dl.acm.org/citation.cfm?id=3001507>
- [12] Abdullah Mueen. 2013. Enumeration of time series motifs of all lengths. In *Proceedings - IEEE International Conference on Data Mining, ICDM (ICDM)*. 547–556. <https://doi.org/10.1109/ICDM.2013.27>
- [13] Abdullah Mueen and Eamonn Keogh. 2010. 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*. ACM Press, New York, New York, USA, 1089. <https://doi.org/10.1145/1835804.1835941>
- [14] Abdullah Mueen, Eamonn Keogh, and Nima Bigdely-Shamlo. 2009. Finding time series motifs in disk-resident data. In *Proceedings - IEEE International Conference on Data Mining, ICDM*. 367–376. <https://doi.org/10.1109/ICDM.2009.15>
- [15] Abdullah Mueen, Eamonn Keogh, Qiang Zhu, Sydney Cash, and Brandon Westover. 2009. Exact Discovery of Time Series Motifs. In *Proceedings of the 2009 SIAM International Conference on Data Mining*. 473–484. <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Exact+Discovery+of+Time+Series+Motifs#0>
- [16] Abdullah Mueen, Krishnamurthy Viswanathan, Chetan Gupta, and Eamonn Keogh. 2015. The Fastest Similarity Search Algorithm for Time Series Subsequences under Euclidean Distance. (8 2015).
- [17] CT O'Rourke, AF Sheehan, EA Erslev, and KC Miller. 2014. Estimating basin thickness using a high-density passive-source geophone array. *Earth and Planetary Science Letters* 402 (2014), 120–126.
- [18] Colin T. O'Rourke, G. Eli Baker, and Anne F. Sheehan. 2016. Using P/S Amplitude Ratios for Seismic Discrimination at Local Distances Using P/S Amplitude Ratios for Seismic Discrimination at Local Distances. *Bulletin of the Seismological Society of America* 106, 5 (2016), 2320. <https://doi.org/10.1785/0120160035>
- [19] Thibaut Perol, Michaël Gharbi, and Marine Denolle. 2018. Convolutional neural network for earthquake detection and location. *Science Advances* 4, 2 (2018). <https://doi.org/10.1126/sciadv.1700578> arXiv:<https://advances.sciencemag.org/content/4/2/e1700578.full.pdf>
- [20] Franco P. Preparata and Michael Ian Shamos. 1985. *Computational Geometry*. Springer New York, New York, NY. <https://doi.org/10.1007/978-1-4612-1098-6>
- [21] Thanawin Rakthanmanon, Bilson Campana, Abdullah Mueen, Gustavo Batista, Brandon Westover, Qiang Zhu, Jesin Zakaria, and Eamonn Keogh. 2012. 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*. 262–270. <https://doi.org/10.1145/2339530.2339576>
- [22] Thanawin Rakthanmanon and E Keogh. 2013. Fast shapelets: A scalable algorithm for discovering time series shapelets. In *Proceedings of the 2013 SIAM International Conference on Data Mining*. 2013. 668–676. <http://epubs.siam.org/doi/abs/10.1137/1.9781611972832.74>
- [23] David P Schaff, Götz HR Bokelmann, Gregory C Beroza, Felix Waldhauser, and William L Ellsworth. 2002. High-resolution image of Calaveras fault seismicity. *Journal of Geophysical Research: Solid Earth* 107, B9 (2002), ESE–5.
- [24] Alexandra Stefan, Vassilis Athitsos, and Gautam Das. 2013. The Move-Split-Merge Metric for Time Series. *IEEE Transactions on Knowledge and Data Engineering* 25, 6 (6 2013), 1425–1438. <https://doi.org/10.1109/TKDE.2012.88>
- [25] Michail Vlachos, Marios Hadjieleftheriou, Dimitrios Gunopulos, and Eamonn Keogh. 2003. 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)*. ACM, New York, NY, USA, 216–225. <https://doi.org/10.1145/956750.956777>
- [26] F. Rall Walsh and Mark D. Zoback. 2015. Oklahoma's recent earthquakes and saltwater disposal. *Science Advances* 1, 5 (2015). <https://doi.org/10.1126/sciadv.1500195> arXiv:<https://advances.sciencemag.org/content/1/5/e1500195.full.pdf>
- [27] Li Wei and Eamonn Keogh. 2006. Semi-supervised time series classification. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '06*. ACM Press, New York, New York, USA, 748. <https://doi.org/10.1145/1150402.1150498>
- [28] Matthew Weingarten, Shemin Ge, Jonathan W Godt, Barbara A Bekins, and Justin L Rubinstein. 2015. High-rate injection is associated with the increase in US mid-continent seismicity. *Science* 348, 6241 (2015), 1336–1340.
- [29] 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. 2016. Crustal structure of the Bighorn Mountains region: Precambrian influence on Laramide shortening and uplift in north-central Wyoming. *Tectonics* 35, 1 (2016), 208–236.
- [30] Y. Wu, Y. Lin, Z. Zhou, D. C. Bolton, J. Liu, and P. Johnson. 2019. DeepDetect: A Cascaded Region-Based Densely Connected Network for Seismic Event Detection. *IEEE Transactions on Geoscience and Remote Sensing* 57, 1 (Jan 2019), 62–75. <https://doi.org/10.1109/TGRS.2018.2852302>
- [31] William L Yeck, Anne F Sheehan, Megan L Anderson, Eric A Erslev, Kate C Miller, and Christine S Siddoway. 2014. 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 (2014), 7028–7042.
- [32] Chin-Chia Michael Yeh, Nickolas Kavantzaz, and Eamonn Keogh. 2017. Matrix Profile VI: Meaningful Multidimensional Motif Discovery. In *2017 IEEE International Conference on Data Mining (ICDM)*. IEEE, 565–574. <https://doi.org/10.1109/ICDM.2017.66>
- [33] Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Diego Furtado Silva, Abdullah Mueen, and Eamonn Keogh. 2016. 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)*. IEEE, 1317–1322. <https://doi.org/10.1109/ICDM.2016.0179>
- [34] Clara E. Yoon, Ossian O'Reilly, Karianne J. Bergen, and Gregory C. Beroza. 2015. Earthquake detection through computationally efficient similarity search. *Science Advances* 1, 11 (2015). <https://doi.org/10.1126/sciadv.1501057> arXiv:<https://advances.sciencemag.org/content/1/11/e1501057.full.pdf>
- [35] Y. Zhu, Z. Zimmerman, N.S. Senobari, C.-C.M. Yeh, G. Funning, A. Mueen, P. Brisk, and E. Keogh. 2017. 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*. <https://doi.org/10.1109/ICDM.2016.126>