An Enhanced Time Series Motif Discovery Using Approximated Matrix Profile

Chanapon Onwongsa
Department of Computer Engineering Chulalongkorn
University, Bangkok 10330, Thailand

Chotirat Ann Ratanamahatana
Department of Computer Engineering Chulalongkorn
University, Bangkok 10330, Thailand

ABSTRACT

Motif discovery of time series data is one of the most prevalent data mining tasks in finding repeated patterns that contain important information in a time series sequence. In particular, its purpose is to find the most similar non-overlapping subsequences pair(s). Recent methods have adopted a matrix profile as a novel data structure for motif discovery problem. However, their $O(n^2)$ time complexity is considered exceedingly high especially for massive time series data.

In this paper, we propose a simple dimensionality reduction method as well as an enhanced approximated matrix profile structure to speed up motif discovery task in massive time series data. As large parts of the matrix profile and raw time series sequences are omitted, our experiments on both synthetic and real datasets demonstrate that we could substantially outperform other rival methods in terms of computation time while maintaining high accuracy of the motif discovery results.

CCS CONCEPTS

• **Computing methodologies** → Motif discovery.

KEYWORDS

Motif discovery, Time series mining, Matrix profile

ACM Reference Format:
Chanapon Onwongsa and Chotirat Ann Ratanamahatana. 2020. An Enhanced Time Series Motif Discovery Using Approximated Matrix Profile. In *2020 2nd International Conference on Image Processing and Machine Vision (IPMV 2020), August 05–07, 2020, Bangkok, Thailand.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3421558.3421586

1 INTRODUCTION

A time series motif discovery task is to locate repeated patterns in a time series sequence. It has been widely applied in many domains such as entomology [1], seismology [2], music [3], weather prediction [4], etc. Figure 1 illustrates a motif detected in the motion capture data [5], which contains various martial arts movements animation, e.g., punches, kicks, retracts, and block movements. The data comprise the time series of z-coordinate values from the sensor detecting a 2-3-second movement of the actor's left arm. The figure illustrates the most similar motif pair detected, presenting a scenario where the actor repeats the same blocking movement with slightly uneven lengths. The two motif occurrences were extracted along with four corresponding frames.

Figure 2 illustrates a time series data in seismology domain [2, 6]. The extracted seismic data include the discovery of aftershocks, foreshocks, volcanic activity, induced seismicity and triggered earthquakes. The figure shows an excerpt of a 9,000-point sequence extracted from 604,781 points (9.5 days) of seismic data. Using a matrix profile structure with a query length of 200 data points, a motif pair is detected at time 4,050 and 7,800. In particular, the seismic data is transformed by Locality-Sensitive Hashing (LSH) based techniques for the time series similarity search [2], and the motif are identified with the use of a matrix-profile-based STAMP algorithm [6]. We will discuss the matrix profile and The STAMP algorithm in the next section.

Recently, there have been various attempts to improve a motif discovery task. In 2016, a Profile-Based Motif Discovery (PBMD) algorithm [6] using Scalable Time series Anytime Matrix Profile (STAMP) was proposed to solve an exact motif discovery problem, along with MASS algorithm. However, as its time complexity was as high as $O(n^2 \log n)$, another method called Scalable Time series Ordered-search Matrix Profile (STOMP) [7] was then proposed. It has an overall of $O(n^2)$ time complexity due to its optimized nested loop within the distance profile calculation. Nevertheless, STAMP algorithm [6] was actually the preferred solution for most applications due to its fast converging anytime algorithm property, making its solution converged faster than those of STOMP algorithm. However, the time and space complexities of both STAMP and STOMP are independent of the length of the motif. Therefore, they may not seem to be very scalable for large data.

In 2018, Approximated Matrix Profile (AMP) [8] has been proposed and become a state-of-the-art method to find an approximated matrix profile, which is much faster than the original matrix profile approach, while providing very high accuracy of the motif results. However, we see that AMP could be further improved to speed up the motif discovery task while maintaining highly accurate results.

In this work, we propose to simply reduce the dimensionality of the data before applying to our novel approximated matrix profile calculation to speed up the overall time series motif discovery task, comparing with the calculation of the full matrix profiles. We then will compare the results of our proposed work with the state-of-the-art algorithms, STAMP and AMP, for an exact and approximated motif discovery problems, respectively.

IPMV 2020, August 05–07, 2020, Bangkok, Thailand
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-8841-2/20/08…$15.00
https://doi.org/10.1145/3421558.3421586

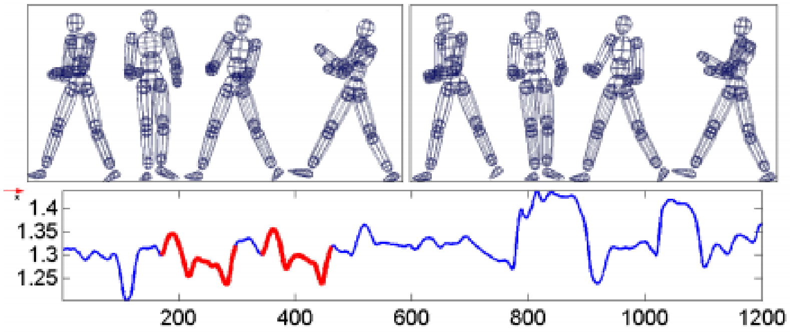

Figure 1: Motion capture data [5]: a motif pair is detected where the actor repeats the same blocking movement.

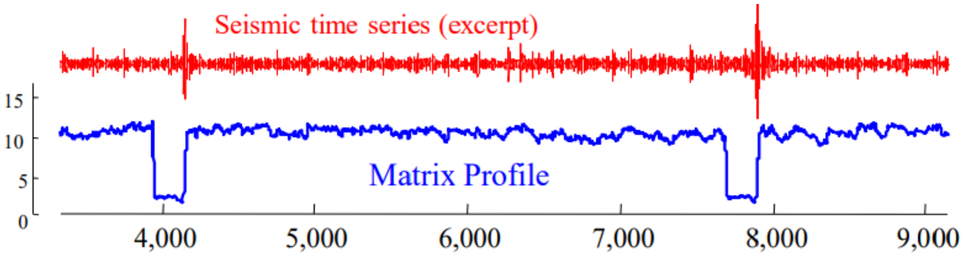

Figure 2: An excerpt of seismic time series data [6] with two occurrences of motif events at time 4,050 and 7,800. respectively.

2 BACKGROUND AND RELATED WORKS

2.1 Definitions

We provide the following definitions to help with the consistency of the understanding in our proposed work.

Definition 1 A *time series* T is a sequence of ordered real values t_i

$$T = t_1,\ t_2,\ \ldots,\ t_n \tag{1}$$

where n is the length of T.

Definition 2 A *subsequence* $T_{i,\ m}$ of T is a contiguous set of values in T starting from i with length m.

$$T_{i,m} = t_i,\ t_{i+1},\ \ldots,\ t_{i+m-1} \tag{2}$$

where $1 \le i \le n - m + 1$.

Definition 3 z-normalized Euclidean distance.

Given a pair of time series sequences $T = t_1,\ t_2,\ \ldots,\ t_n$ and $Q = q_1,\ q_2,\ \ldots,\ q_n$ with length n. The z-normalized Euclidean distance between the pair of T and Q is calculate by

$$\mathrm{Dist}\,(T,\ Q) = \sqrt{\sum_{i=1}^{n}\left(\frac{t_i - \mu_T}{\sigma_T} - \frac{q_i - \mu_Q}{\sigma_Q}\right)^2} \tag{3}$$

for $i = 1,\ 2,\ 3,\ \ldots,\ n$. μ_T and μ_Q are arithmetic means of T and Q, respectively. σ_T and σ_Q are standard deviations of T and Q, respectively.

Definition 4 A *Motif in Time Series* is the most similar non-overlapping subsequence pair(s) that has a minimum distance among all subsequences in T of length n, e.g., $T_{i,m}$ and $T_{j,m}$ are motif pair of length m such that Dist($T_{i,m}$,$T_{j,m}$) $\le$ Dist($T_{x,m}$,$T_{y,m}$) if and only if $|i - j| \ \ge\ m$ and $|x - y| \ \ge\ m$ for all $x,\ y\ \in \{1,\ 2, 3, \ldots,\ n - m + 1\}$

2.2 Related Works

For very large time series data, a massive number of subsequence pairs makes motif discovery task quite difficult. Various attempts have been made to make it more feasible, including dimensionality reduction techniques [9], exact motif discovery algorithms [1, 6, 7], approximated motif discovery algorithms [8-10], anytime algorithms [6, 11], and motif discovery algorithm for online data [12].

In this work, we will present an approximated algorithm that could further speed up the motif discovery task using an enhanced approximated matrix profile. Therefore, we start this section with an introduction to a matrix profile, which is a core data structure for both PBMD [6] and AMP [8], state-of-the-art algorithms for exact and approximated motif discovery problems, respectively.

2.2.1 Matrix Profile.

Distance Profile [6]. A distance profile $Dist_i$ of a time series T is a vector containing z-normalized Euclidean distances between a subsequence $T_{i,m}$ and each corresponding subsequence of length m in time series T.

$$Dist_i = \left[d_{i,1},\ d_{i,2},\ d_{i,3},\ \ldots,\ d_{i,n+m-1}\right] \tag{4}$$

where $d_{i,j}$ is a z-normalized Euclidean distance between $T_{i,m}$ and $T_{j,m}$ with $1 \le (i\ and\ j) \le n - m + 1$.

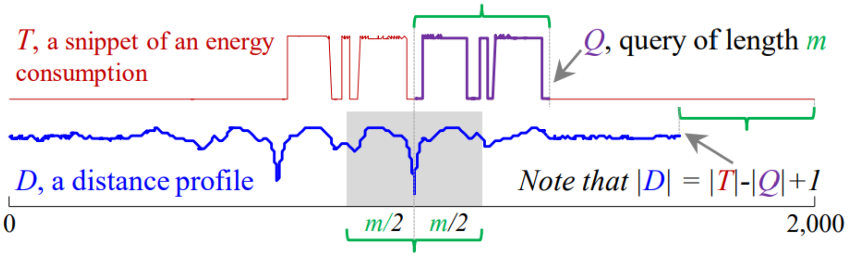

Figure 3: A time series T is extracted into any subsequence Q, which is used as a query to compute a z-normalized Euclidean distance between a query and every subsequence in T movement [6].

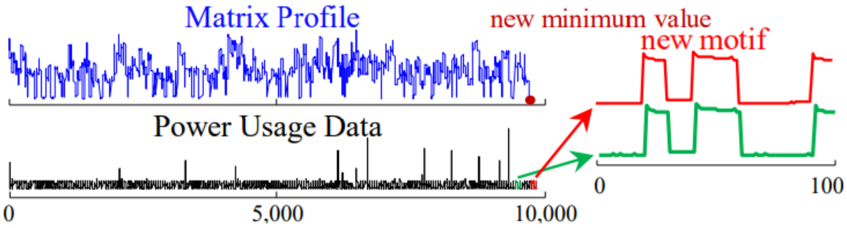

Figure 4: (left) The Power Usage Data and its corresponding matrix profile [6], along with the motif pair detected according to the minimum value of the matrix profile. (right) A zoom-in of the motif pair detected.

To illustrate the distance profile, Figure 3 shows a snippet of energy consumption data of 2,000 data points, along with its corresponding distance profile of length 2,000− m+1 data points [6].

Note that to avoid trivial matches (a few positions just to the left and right of i^{th} location) that could result in zero or near-zero distance profile $Dist_i$ [1], we could set values of $d_{i,j}$ to infinity where $i - \frac{m}{2} \leq j \leq i + \frac{m}{2}$.

Matrix Profile [6]. A matrix profile P of time series T is a vector containing the minimum distance along with the corresponding index between every query of T and its nearest neighbor in T.

$$P = [\min(Dist_1), \ \min(Dist_2), \ldots, \min(Dist_{n-m+1})] \qquad (5)$$

where $Dist_i$ for $i = 1, \ 2, \ \ldots \ , \ \ n-m+1$ is the distance profile calculated by the z-normalized Euclidean distance associated with a subsequence $T_{i,m}$ and a time series T.

Figure 4 illustrates a time series Power Usage Data and its corresponding Matrix Profile [6]. The minimum value of the matrix profile corresponds to a 100-minute motif pair occurring at 9,864 minutes (green) and 10,473 minutes (red).

- Matrix Profile Computation

The two main algorithms are Mueen's ultra-fast Algorithm for Similarity Search (MASS) [13], which was then used as a subroutine in the state-of-the-art algorithm to solve a motif discovery task for massive data so-called Scalable Time series Anytime Matrix Profile (STAMP) algorithm [6].

- MASS algorithm

MASS is an algorithm [13] for distance profile calculation, which is an important subroutine in STAMP

$$Dist_i = \sqrt{2m\left(1 - \frac{QT_i - m\mu_Q M_{Ti}}{m\sigma_Q \Sigma_{Ti}}\right)} \qquad (6)$$

where Q denotes all subsequences of length m in a time series T, QT_i is a inner product of Q and each subsequence $T_{i,m}$, μ_Q is an arithmetic mean of Q, σ_Q is a standard deviation of Q, M_{Ti} is an arithmetic mean varying i of $T_{i,m}$ and Σ_{Ti} is the standard deviation varying i of $T_{i,m}$.

Algorithm 1 Calculation of InnerProductsSlidingSequence(T, Q) [13]

```
Procedure InnerProductsSlidingSequence($\boldsymbol{T}$, $\boldsymbol{Q}$)
Input:
    $T$ : a user-defined time series sequence
    $Q$: a query of time series $T$
Output:
    $QT$: the inner product between subsequences $T$ and $Q$
$n \leftarrow$ Len($T$) // A time series $T$ of length $n$
$m \leftarrow$ Len($Q$) // A subsequences $Q$ of length $m$
$T_{zeros} \leftarrow$ Append $T$ with $n$ zeros
$Q_{reverse} \leftarrow$ ReverseOfSubsequences($Q$)
$Q_{reversezeros} \leftarrow$ Append $Q_{reverse}$ with $2n - m$ zeros
$Q_{reversezerosf} \leftarrow$ FFT($Q_{reversezerosa}$ )
$T_{zerosf} \leftarrow$ FFT($T_{zeros}$ )
$QT \leftarrow$ InverseFFT(MultiplyElementwise
($Q_{reversezerosf}$, $T_{zerosf}$))
return $QT$
```

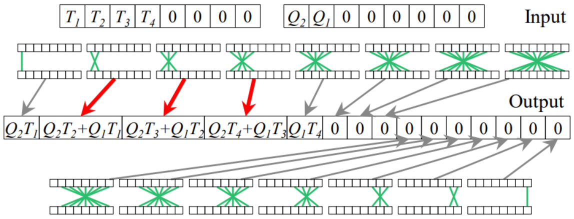
T_1 T_2 T_3 T_4 0 0 0 0
Q_2 Q_1 0 0 0 0 0 0
Input
Output
Q_2T_1 $Q_2T_2+Q_1T_1$ $Q_2T_3+Q_1T_2$ $Q_2T_4+Q_1T_3$ Q_1T_4 0 0 0 0 0 0 0 0 0 0

Figure 5: An example of a convolution operation [6, 14] between subsequences T and Q related to its corresponding the inner product between subsequences T and Q.

Figure 5 illustrates an example of a convolution operation [6, 14] related to the inner product between subsequences T and Q where the inner product is calculated at every sliding window.

Algorithm 2 Mueen's Algorithm for Similarity Search (MASS) [13]

```
Procedure MASS($\boldsymbol{T}$, $\boldsymbol{Q}$)
Input:
    $T$ : a user-defined time series sequence
    $Q$ : a query of time series $T$
Output:
    $Dist$ : a distance profile of the query $Q$
$QT \leftarrow$ InnerProductsSlidingSequences($T$, $Q$)
$M_T, \mu_Q \leftarrow$ CalculateMean($T$, $Q$)
$\Sigma_T, \sigma_Q \leftarrow$ CalculateStandardDiviation($T$, $Q$)
$Dist$CalculateDistanceProfile($T$, $Q$, $QT$, $\mu_Q$, $\sigma_Q$, $M_T$, $\Sigma_T$ )
return $Dist$
```

From ALGORITHM 1 and 2, InnerProductsSlidingSubsequences(T, Q) is a subroutine of MASS algorithm [13], and is calculated by the inner product between T and a query Q in T, which employs Fast Fourier Transform algorithm (FFT) and reverses the subsequence by Inverse Fast Fourier Transform algorithm (IFFT). Then, the mean and standard deviation μ_Q, σ_Q, M_T, Σ_T , as well as the distance profile $Dist_i$ will be calculated and collected in the matrix profile $Dist$.

First, STAMP algorithm [6] starts creating a matrix profile P with an initial value of infinity, and its associated matrix profile index I with a zero vector as its initial value. Then, MASS [13], an anytime algorithm, is used to calculate a distance profile $Dist$ in random order. Each element P and I are updated with the minimum distance and its corresponding index, respectively. The final matrix profile will be returned after completing all iterations.

It is important to note that for any subsequence $T_{i,m}$ of T, the distance profile $Dist_i$ will be zero and almost zero for trivial matches (a few positions just to the left and right of i^{th} location). We could avoid trivial matches by setting values of $d_{i,j}$ to infinity where $i - \frac{m}{2} \leq j \leq i + \frac{m}{2}$.

Another strength of STAMP algorithm is that it is an anytime exact motif discovery algorithm. However, its time and space complexities are quite large, posing some problems for very large datasets.

Algorithm 3 The STAMP algorithm [6]

```
Procedure STAMP($\boldsymbol{T}$, $\boldsymbol{m}$)
Input:
    $T$ : a user-defined time series sequence
    $m$ : length of the given subsequence
Output:
    $P$ : an updated matrix profile
    $I$ : an associated updated matrix profile index
$n \leftarrow$ Len($T$)
$P \leftarrow$ infinity
$I \leftarrow$ zero
$idxes \leftarrow$ range(1, $n - m + 1$)
for $idx$ in $idxes$ // In any order $idx$ in $idxes$
    $Dist \leftarrow$ MASS($Q, T$) // $Q$ is subseq in $T$
    $P$, $I \leftarrow$ UpdateMinElementwise($P$, $I$ , $Dist$, $idx$)
end for
return $P$, $I$
```

2.2.2 *Time Series Motif Discovery Using Approximated Matrix Profile (AMP).*

The core idea of AMP is the reduced number of iterations k [8]. So, only partial matrix profile needs to be accessed.

Time Ordering of computation. AMP is an anytime algorithm that determines the subsequence ordering by randomly selecting the first index sequence to compute each distance profile corresponding to the time series T and each of the subsequences within; then, the next iteration depends on the next sequence of random permutation index. The method will be completed after k iteration.

Number of iteration (k). The main objective is to select the minimum distance between any two-subsequence pair, using the idea from birthday paradox problem [15] defined as follows.

- n people is comparable to subsequence $n - m + 1$
- 365 possible day is comparable to $n - m + 1$ possible nearest neighbor distance

Thus, a probability of two minimum distance subsequences (motif) is assumed to relate to a probability of two people having the same birthday. The number of iteration (k) is determined by FindK algorithm, which is a part of AMP algorithm, while user is to provide the probability input p.

The formula to find k called $CalculateProb$ function is based on 1st derivative of an exponential function approximated by a Taylor Series, assuming independent probability, which is defined by

$$CalculateProb\,(k, n, m) = 1 - exp\left(\frac{-k^2}{2\,(n - m)}\right) \quad (7)$$

AMP algorithm is faster than STAMP in finding the approximated matrix profile, but AMP uses raw time series and k iteration. Therefore, some further refinement and improvement can be made.

3 PROPOSED WORKS

In our proposed work, the whole matrix profile (STAMP [6] and STOMP [7]) and raw time series data are not actually needed for the time series motif discovery task. In particular, we propose to reduce the dimension of the data first, before applying them to our modified approximated matrix profile (with newly proposed $\tilde{k}$ iterations).

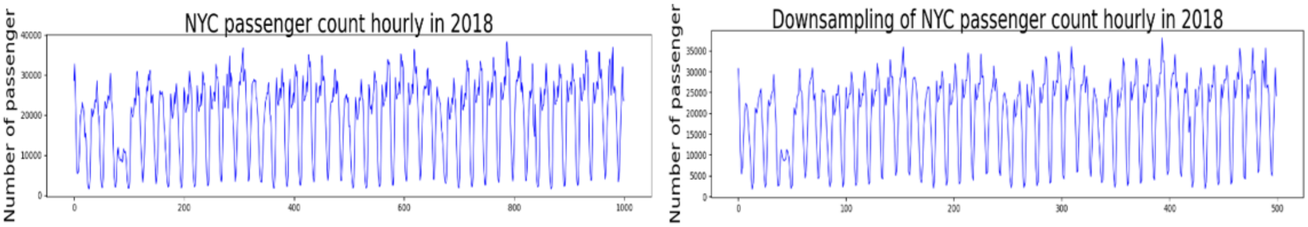

Figure 6: (left) raw data of the New York City yellow taxi hourly passenger count in 2018 of length 1,000. (right) The downsampled data of the same New York City yellow taxi hourly passenger count in 2018 of length 500.

3.1 Dimensionality Reduction

Based on our empirical findings, downsampling the data by half could actually improve the results. Simply, two-point running average is used to reduce noises between time steps and to better expose the signal of the underlying causal process. We use the averages of two points defined by

$$\bar{T}_{\frac{i+1}{2}} = \frac{T_i + T_{i+1}}{2} \tag{8}$$

Index i in the set of *odd* counting numbers where $1 \le i < n$.

Algorithm 4 Pseudocode of downsampling

```
Procedure Downsampling($T$, $m$)
Input:
    $T$ :  a time series sequence
    $M$ : a motif length
Output:
    $\bar{T}$ :  a downsampled time series $T$
    $n$ :  a downsampled time series length
    $m$ : a downsampled subsequence length
1: $n \leftarrow$ Len($\bar{T}$)
2: $idxes \leftarrow$ range(1, $N$)
3: for $i$ in range($N$) :
4:     $\bar{T} \leftarrow (T_i + T_{i+1})/2$
5:     $n \leftarrow$ Len($\bar{T}$)
6:     $m \leftarrow M/2$
7: return $\bar{T}$, $n$, $m$
```

Figure 6 illustrate an example of data downsampling. The raw data of 1,000 data points are downsampled by half to 500 data points that could still maintain the characteristic of the data.

3.2 Our proposed Approximated Matrix Profile using Stirling's Approximation (AMPSA)

A probability of two minimum distance subsequences (motif) is related to a probability of two people having the same birthday, assuming an independent probability. According to the $CalculateProb(k, n, m)$ [8] formula in section 2.2.2, it uses 1^{st} derivative of an exponential function approximated by a Taylor Series, but 1^{st} derivative of exponential function gives k value that is unnecessarily large. To reduce unnecessary number of iteration (k) and achieve some speedup, we propose to use the Stirling's approximation, which is closer to an optimal number of iterations.

A Stirling's approximation gives an approximated value for the factorial $n!$ as follows.

$$n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n \tag{9}$$

An extended Stirling's approximation [16] is defined as follows.

$$\sqrt{2\pi n}\left(\frac{n}{e}\right)^n \le n! \le e\sqrt{n}\left(\frac{n}{e}\right)^n \tag{10}$$

For $n = 1, 2, 3, \ldots$ and e is a mathematical constant approximately equal to 2.71828.

Our modified $CalOptProb(\tilde{k}, n, m)$ to find the number of iteration ($\tilde{k}$) is shown in ALGORITHM 5. It employs evaluation of the integral extended to the double inequality approximated by Stirling's approximation. This will reduce down the value of $\tilde{k}$, and hence speed up the process. Our $CalOptProb(\tilde{k}, n, m)$ is defined by

$$CalOptProb\left(\tilde{k}, n, m\right) = 1 - \left(\frac{\sqrt{2\pi}}{e^{\tilde{k}+1}}\right)\left(\frac{n-m}{n-m-\tilde{k}}\right)^{n-m-\tilde{k}+0.5} \tag{11}$$

Algorithm 5 Finding Optimal K iterations (FindOptimalK)

```
Procedure FindOptimalK($n$, $m$, $p$)
Input:
    $n$ :  a length of time series $T$
    $m$ : a length of a given subsequence
    $p$ :  a user-defined probability
Output:
    $\tilde{k}$ : a maximum number of iterations
1: $\tilde{k} \leftarrow 1$
2: probability $\leftarrow$ 0 // Initial probability is 0
3: while probability < $p$ :
4:         probability $\leftarrow CalOptProb(\tilde{k}, n, m)$
5:         $\tilde{k} = \tilde{k} + 1$
6: end while // End while loop when probability is greater than $p$
7: return $\tilde{k} - 1$
```

In ALGORITHM 6, we first assign new indices for matrix profile calculations using random permutation (line 4), then $\tilde{k}$ iterations is determined (line 5) using FindOptimalK from ALGORITHM 5 Lines 6-10 will calculate matrix profiles for $\tilde{k}$ iterations by using the (current) first index from random permutation and finding each element of the distance profile using the MASS algorithm. $\bar{T}_{idx}$ denotes a query subsequence of $\bar{T}$ with idx index. Then the first index (used index) is removed before updating $\bar{P}$ and $\bar{I}$ when a new

Algorithm 6 Algorithm of our proposed An Approximated Matrix Profile using Stirling's approximation (AMPSA)

```
Procedure AMPSA($\bar{T}$, $n$, $m$, $p$)
Input:
    $\bar{T}$ : a time series $\bar{T}$ of length $n$
    $m$ : a length of a given subsequence
    $p$ : a user-defined probability
Output:
    $P$ : an updated matrix profile
    $I$ : an associated updated matrix profile index
$P \leftarrow$ infinity // initialize all values in $P$ to infinity
$I \leftarrow$ zero // initialize all values in $I$ to zero
$idxes \leftarrow$ range(1, $n - m + 1$)
$new\_idx \leftarrow$ RandomShuffle($idxes$)
$\tilde{k} \leftarrow$ FindOptimalK($n$, $m$, $p$)
for $i$ in range($\tilde{k}$) :
    $idx \leftarrow new\_idx(1)$ // uses 1$^{\text{st}}$ index
    $\bar{D} \leftarrow$ MASS($\bar{T}_{idx}$, $\bar{T}$) // $\bar{T}_{idx}$ is a query subseq of $\bar{T}$ index $idx$
    $new\_idx$.remove($new\_idx(1)$) // remove used index
    $\bar{P}$, $\bar{I} \leftarrow$ UpdateMinElementwise($\bar{P}$, $\bar{I}$, $\bar{D}$, $idx$)
$P \leftarrow \bar{P}_{idx} * 2$, $I \leftarrow \bar{I}_{idx} * 2$
return $P$, $I$
```

minimum Euclidean distance is discovered. Finally, every distance profile value and index of $\bar{P}$ and $\bar{I}$, respectively, are multiplied by two (Approximated real value of P and I) before returning P and I as outputs.

4 EXPERIMENTAL EVALUATION

4.1 Time and Space Complexity

Both STAMP and AMP algorithm consist of two parts, i.e., matrix profile calculation and motif discovery. In terms of computational time, we only need to compare our proposed algorithm with AMP alone as AMP is an approximated algorithm with much smaller time complexity comparing with STAMP. However, we will compare the correctness of the results based on the results from STAMP algorithm as it is an exact algorithm.

In terms of time complexity, our AMPSA algorithm is $O(\tilde{k}n \log n)$, where $\tilde{k} < k \ll n$ when k is AMP's number of iteration, comparing with $O(kn \log n)$ for AMP and $O(n^2 \log n)$ for STAMP. The space complexity of AMPSA, AMP, and STAMP is still $O(n)$.

We set up and experiment using random walk data with variable lengths to illustrate that as the size of time series gets large, our algorithm is significantly faster than STAMP and AMP. Motif pairs are planted into each set of the data according to 'Motif index' in Table 1, 2 and 3 respectively, and the probability p is set to 0.999. Figure 7 (left) compares the computational time among our proposed AMPSA, AMP, and STAMP, and Figure 7 (right) only compares the computation time between our AMPSA and AMP algorithm, which clearly demonstrates the superiority of our proposed work.

4.2 Accuracy of the Motif

In order to quantitatively interpret the experimental results, we will compare the value of motif Overlapping Ratio (OR) [17], which is the overlapping percentage between the exact motif pattern and the discovered pattern.

Let A be a set of an exact motif pattern, and B be a set of a discovered pattern, $OR(A, B)$ is defined by

$$OR\,(A,\ B) = \frac{|A \cap B|}{|A|} \times 100$$

To illustrate the OR calculation, we assume time series T of length n, a given subsequence of the m, a motif indexed at i and the discovered the motif pair of length m at index j.

$$OR\,(A,\ B)\ = \left(\frac{m - |i - j|}{m}\right) \times 100$$

From the results in Table 1, 2 and 3, we demonstrate that our proposed AMPSA is much faster than the rival methods, especially in large data. The OR results reconfirm that our AMPSA could correctly locate the motif. In particular, the discovered motifs have discrepancy less than 3% of the subsequence length, i.e., given a time series of length n, whose exact motif pair of length m is located at position q and r of the time series, AMPSA does discover a motif pair located between positions $[q - \frac{3m}{100},\ q + \frac{3m}{100}]$ and $[r - \frac{3m}{100},\ r + \frac{3m}{100}]$.

4.3 Case Study: Finding Repeated Insect Behavior

We also tested our proposed AMPSA on real-word dataset, i.e., an Electrical Penetration Graph of insect behavior data (EPG), to find a motif pair that represents insects sucking sap from living plants [1] to reconfirm that our result is faster while maintaining high detection accuracy. The experiment's purpose was to understand the insect's behavior [18] by measuring fluctuations in voltage level (EPG) by gluing a thin wire of insect electrode on its back, then completing the circuit using a stiff uninsulated wire through a host plant. The EPG data is shown in Figure 8

We use the EPG of insect behavior data with 33,021 data points. We set a subsequence length to 480 (the same as [1]) and probability $p = 0.999$ to find the pair of time series motif.

We first downsampled the EPG of insect behavior from 33,021 data points to 16,510 datapoints, as shown in Figure 9. Then, our algorithm, AMPSA, located the motif pair using the downsampled version of the data and Stirling's approximation to determine the number of iterations to calculate the distance profile. As a result, AMPSA discovered the motif pair at index 1,778 and 4,462, respectively, as shown in Figure 10 before converting them back to their original dimension with index 3,556 and 8,924, respectively.

Figure 11 and Table 4 show the motif pair results of length 480, representing insect sucking sap from living plants from our AMPSA, AMP, and STAMP algorithms. Our AMPSA and the AMP algorithms provide the motif pairs that are very close to the exact motif results provided by STAMP. However, our AMPSA is statistically much faster. The number of iterations is reduced, and in turn reducing the running time by a large margin.

5 CONCLUSIONS

This paper proposes an approximated motif discovery algorithm for time series data. The algorithm is further modified to discover

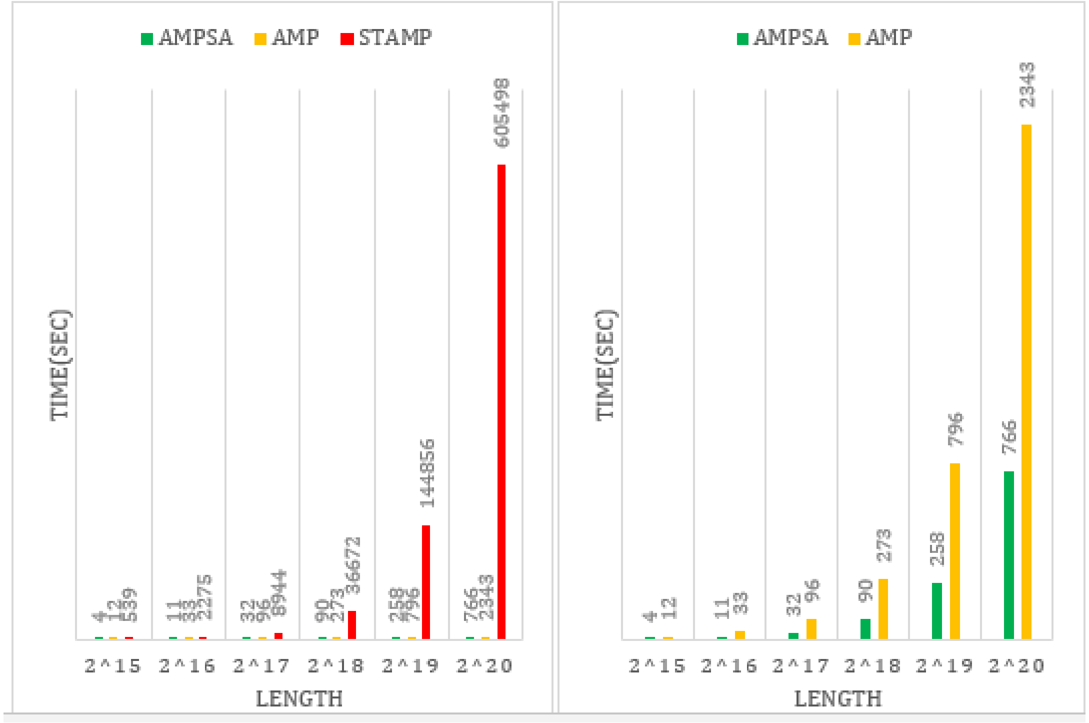

Figure 7: (left) Computation time of AMPSA, AMP and STAMP algorithm. (right) Computation time of AMPSA and AMP algorithm (Table 1, 2 and 3) with varying time series length n and $m = \frac{5n}{100}$

Table 1: Accuracy of motif is measured by Overlapping Ratio (OR) and Computation time of AMPSA algorithm

		AMPSA			
Time Series Length	Motif Index	DiscoveredMotif Index	OR 1stMotif (%)	OR 2ndMotif (%)	Time(Sec)
2^{15}	5,000 | 16,638	5,010 | 16,648	99.85	99.85	4
2^{16}	10,000 | 33,276	10,016 | 33,292	99.88	99.88	11
2^{17}	20,000 | 66,553	20,030 | 66,582	99.89	99.89	33
2^{18}	40,000 | 133,107	40,0082 | 133,188	99.84	99.85	90
2^{19}	80,000 | 266,214	80,148 | 266,362	99.86	99.86	257
2^{20}	160,000 | 532,428	160,356 | 532,784	99.83	99.83	766

Table 2: Accuracy of motif is measured by Overlapping Ratio (OR) and Computation time of AMP algorithm

		AMP			
Time Series Length	Motif Index	DiscoveredMotif Index	OR 1stMotif (%)	OR 2ndMotif (%)	Time(Sec)
2^{15}	5,000 | 16,638	5,002 | 16,640	99.96	99.96	11
2^{16}	10,000 | 33,276	10,013 | 33,289	99.90	99.90	33
2^{17}	20,000 | 66,553	20,038 | 66,591	99.86	99.86	96
2^{18}	40,000 | 133,107	40,095 | 133,202	99.82	99.82	273
2^{19}	80,000 | 266,214	80,133 | 266,347	99.87	99.87	796
2^{20}	160,000 | 532,428	160,359 | 532,787	99.83	99.83	2,343

Table 3: Accuracy of motif is measured by Overlapping Ratio (OR) and Computation time of STAMP algorithm

		STAMP			
Time Series Length	Motif Index	DiscoveredMotif Index	OR 1st Motif (%)	OR 2nd Motif (%)	Time(Sec)
2^{15}	5,000 | 16,638	5,000 | 16,638	100	100	539
2^{16}	10,000 | 33,276	10,000 | 33,276	100	100	2,275
2^{17}	20,000 | 66,553	20,000 | 66,553	100	100	8,944
2^{18}	40,000 | 133,107	40,000 | 133,107	100	100	36,672
2^{19}	80,000 | 266,214	80,000 | 266,214	100	100	144,856
2^{20}	160,000 | 532,428	160,000 | 532,428	100	100	605,498

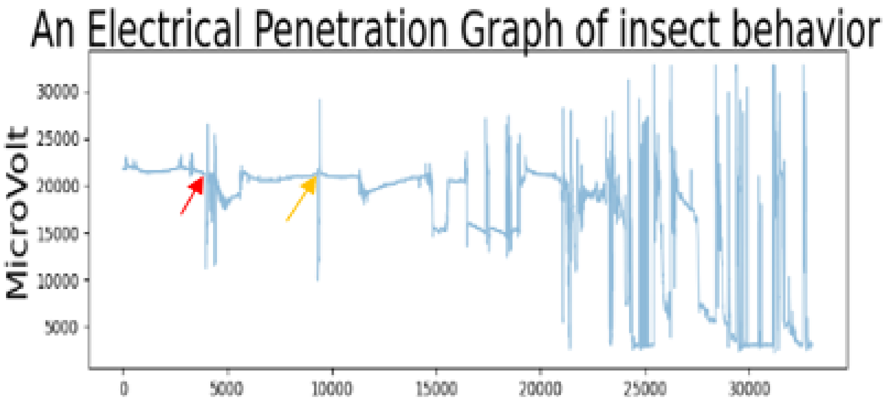

Figure 8: The EPG of insect behavior with 33,021 data points [18]. The red arrow indicates a motif starting at 3,553, and the orange arrow indicates a motif starting at 8,921. This EPG is quite difficult to understand as it consists of highly nonstationary data, noise etc.

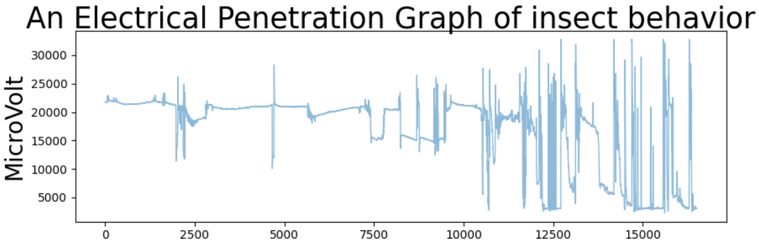

Figure 9: The downsampled version of the EPG of insect behavior data, originally from 33,021 data points [18] down to 16,510 data points.

Table 4: Computation time and motif position of AMPSA, AMP and STAMP algorithms

Algorithm	Number of iterations	Time (Sec)	Discovered Motif	OR 1st motif (%)	OR 2nd motif (%)
AMPSA	462	6	3556 | 8924	99.37	99.37
AMP	666	27	3548 | 8916	98.96	98.96
STAMP	32,542	1,300	3553 | 8921	100	100

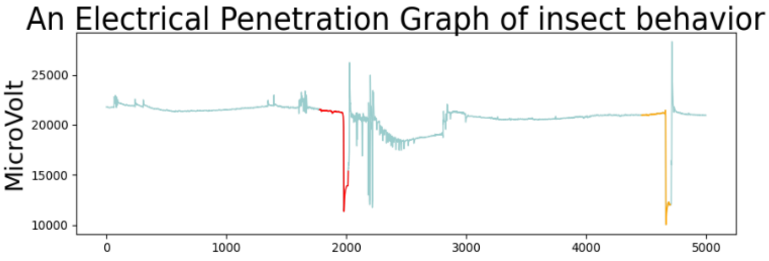

Figure 10: The first 10,000 data points being downsampled to 5,000 data points. Our AMPSA discovered the motif pair of length 240 data points at index 1,778 (in red) and 4,462 (in orange), respectively, before converting them back to their original dimension of length 480 data points with index 3,556 and 8,924, respectively.

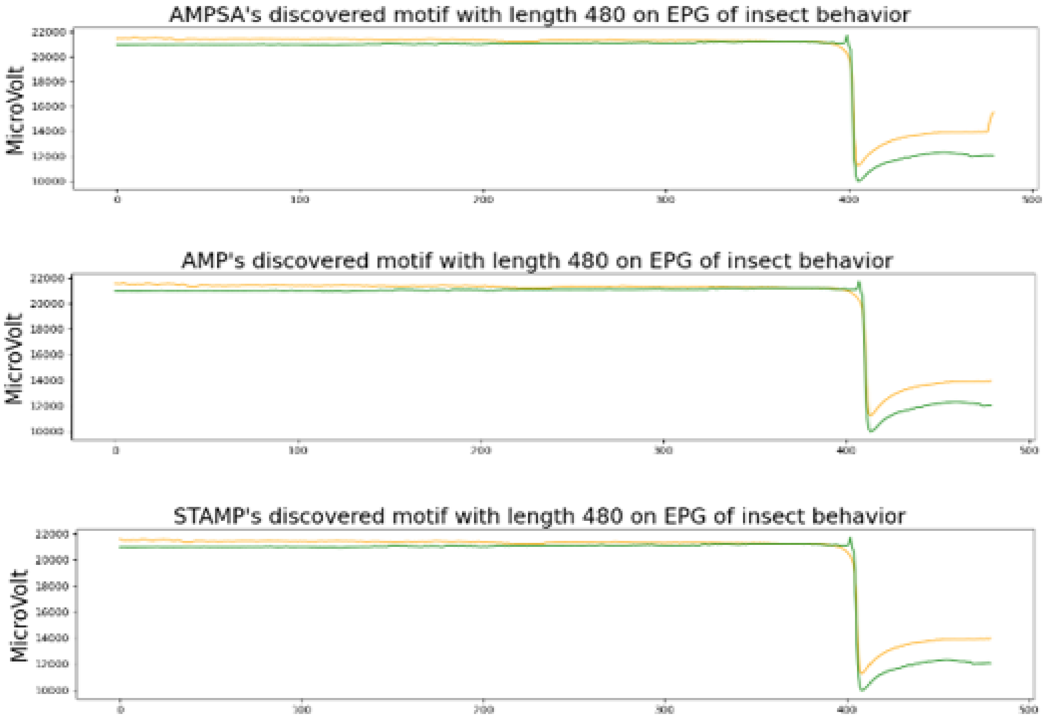

Figure 11: Motif pair of length 480 representing insect sucking sap from living plants from our AMPSA algorithm (top), AMP algorithm (middle) and STAMP algorithm (bottom).

motifs under reduced number of dimension and number of iterations in matrix profile computation. Experiments were performed on random walk data and the cases the Electrical Penetration Graph of insect behavior data. Our modified algorithm is fast, simple and parameter free. The results of our experiments illustrated that our proposed algorithm can significantly achieve large speedup especially in large datasets, while maintaining high detection accuracy.

REFERENCES

[1] Mueen, A., Keogh E., Zhu Q., Cash S. & Westover B. Exact Discovery of Time Series Motifs. *Proceedings of the 2009 SIAM international conference on data mining.* (pp 473-484)

[2] Yoon, C., O'Reilly, O., Bergen, K and Beroza, G. (2015). Earthquake detection through computationally efficient similarity search. *Sci. Adv.*

[3] Silva, D. F., Yeh, C.-C.M., Batista, G. E. D. A. P. A., & Keogh, E. (2016). SiMPle: *assessing music similarity using subsequences joins.* International Society for Music Information Retrieval Conference, XVII.

[4] McGovern, A., Rosendahl, D.H., Brown, R.A., Droegemeier, K.K., (2011) Identifying predictive multidimensional time series motifs: an application to severe weather prediction. *Data Mining Knowledge Discovery* 22(1):232-258.

[5] Yankov, D., Keogh, E., Medina, J., Chiu, B., & Zordan, V. (2007). Detecting time series motifs under uniform scaling. *Proceedings of the ACM SIGKDD.* (pp 844-853)

[6] Yeh, C.-C. M., Zhu, Y., Ulanova, L., Begum, N., Ding, Y., Dau, A., Silva, D., Mueen, A. & Keogh E. (2016). *Matrix Profile I: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets.* 16th ICDM.

[7] Zhu, Y., Zimmerman, Z., Shakibay, N.S., Yeh, C.-C.M., Funning, G., Mueen, A., Brisk, P. & Keogh E. (2016). *Matrix Profile II: Exploiting a Novel Algorithm and GPUs to Break the One Hundred Million Barrier for Time Series Motifs and Joins.* IEEE 16th ICDM.

[8] Pariwatthanasak, K., & Ratanamahatana, C. (2019). Time Series Motif Discovery Using Approximated Matrix Profile: ICICT 2018, London.

[9] Yi, B.K., & Faloutsos, C. (2000) Fast Time Sequence Indexing for Arbitrary Lp Norms. Proceedings of the 26th VLDB'00. 385-394.

[10] Chiu, B., Keogh, E., & Lonardi, S. (2003). *Probabilistic discovery of time series motifs*. Proceedings of the 9th ACM SIGKDD

[11] Zhu, Y., Yeh, C.-C. M., Zimmerman, Z., Kamgar, K. & Keogh, E. (2018). *Matrix Profile XI: SCRIMP++: Time Series Motif Discovery at Interactive Speeds*. IEEE 18th ICDM.

[12] Mueen, A., & Keogh, E., (2010) *Online discovery and maintenance of time series motifs*. Proceedings of the 16th ACM SIGKDD.

[13] Mueen, A., Zhu, Y., Yeh, C.-C.M., Kamgar, K., Viswanathan, K., Gupta, C., & Keogh, E. (2015) The fastest similarity search algorithm for time series subsequences under Euclidean distance. url:www.cs.unm.edu/~mueen/FastestSimilaritySearch.html (accessed 25 August, 2018).

[14] "Convolution - Wikipedia, the free encyclopedia," https://en.wikipedia.org/wiki/Convolution (accessed 21 August, 2018)

[15] Von Mises, R. (1936). Probability, Statistics and Truth, 2nd rev. English ed., New York, Dover, 1981

[16] Herbert, R. (1955). A Remark on Stirling's Formula, *The American Mathematical Monthly*, **62** (1): 26–29 pp

[17] Niennattrakul, V., Wanichsan, D., & Ratanamahatana, C. (2009). Accurate Subsequence Matching on Data Stream under Time Warping Distance. PAKDD Workshop. 156-167.

[18] Stafford, C. & Walker, G. (2009). Characterization and correlation of DC electrical penetration graph waveforms with feeding behavior of beet leafhopper, Circulifer tenellus. Entomologia Experimentalis et Applicata. 130. 113 - 129.