1-2023

PMP: Privacy-Aware Matrix Profile against Sensitive Pattern Inference

Li Zhang

Jiahao Ding

Yifeng Gao
The University of Texas Rio Grande Valley

Jessica Lin

PMP: Privacy-Aware Matrix Profile against Sensitive Pattern Inference for Time Series

Li Zhang* Jiahao Ding† Yifeng Gao‡ Jessica Lin*

Abstract

Recent rapid development of sensor technology has allowed massive fine-grained time series data to be collected and set foundation for the development of data-driven services and applications. During the process, data sharing is often involved to allow the third-party modelers to perform specific time series data mining tasks based on the need of data owner. The high resolution of time series data brings new challenges in protecting privacy. On one hand, meaningful information in high-resolution time series shifts from concrete point values to local shape-based segments. On the other hand, numerous research have found that long shape-based patterns could contain more sensitive information and may potentially be extracted and misused by a malicious third party. However, the privacy issue for time series patterns is surprisingly seldom explored in privacy-preserving literature. In this work, we consider a new privacy preserving problem: preventing malicious inference on long shape-based patterns while preserving short segment information for the utility task performance. To mitigate the challenge, we investigate an alternative approach by sharing Matrix Profile (MP), which is a non-linear transformation of original data and a versatile data structure that supports many data mining tasks. We found that while MP can prevent the concrete shape leakage, the canonical correlation in MP index can still reveal the location of sensitive long pattern information. Based on this observation, we design two attacks named Location Attack and Entropy Attack to extract the pattern location from MP. To further protect MP from these two attacks, we propose a Privacy-Aware Matrix Profile (PMP) via perturbing the local correlation and breaking the canonical correlation in MP index vector. We evaluate our proposed PMP against baseline noise-adding methods through quantitative analysis and real-world case study to show the effectiveness of the proposed method. Our source code is available at https://github.com/lzhang18/PMP.

1 Introduction

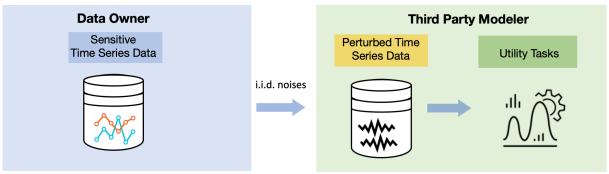
The wide use of sensors in personal devices and other infrastructures have allowed the collection of high-resolution time series and boosted the demand for third-party data-oriented services and applications such as medical monitoring [27], industrial system prognostics [29] and smart homes [26]. For example, a farm owner (referred as **'data owner'** or **'owner'** for short) might collect massive data from their own smart sensor system to monitor the behavior of farm animals such as cows and chicken in real-time [1]. Since the owner does not have the expertise to analyze the data, they might seek some third-party data mining service tasks (aka **'utility task'**) such as activity recognition (e.g., identify eating food or egg laying), or detecting animal sickness through anomaly detection.

While this data sharing service brings benefits, there has been long-time concern for data sharing such as personal information leakage and data breach [2]. Existing solution has been relying on Differential Privacy (DP) [10, 25] to hide the concrete data values associated with sensitive information. However, the high resolution of time series data brings new challenges in privacy protection. The concrete data values become less important as meaningful information shifts to the shape of small subsequences [9]. On the other hand, numerous research [12, 27, 30] have found that long shape-based patterns could contain more sensitive information and may potentially be extracted and misused by a malicious third party. For example, a malicious modeler might detect a day-long pattern that infer farmer's daily work route even though such day-long patterns is unrelated to any utility task mentioned above, which typically relies on minute or hour-long time series segments [8, 23]. However, the privacy issue for time series patterns is surprisingly seldom explored in privacy-preserving literature.

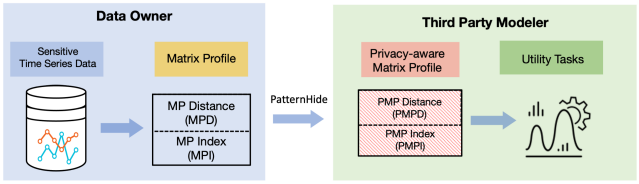
In this work, we consider a new privacy preserving problem: preventing malicious inference on long shape-based patterns while preserving short segment information for the downstream task performance. As shown in Fig. 1(a), most existing privacy-preserving approaches are based on differential privacy [10, 25] to sanitize the raw time series values with independent and identical distributed (i.i.d) noises and then share the perturbed data with the third party modeler. However, as pointed out by Xiao et al. [25], DP methods cannot protect the privacy of pattern, which consists of a set of contagious autocorrelated points. In fact, if we adopt previous DP methods to protect the long pattern, the amount of noises needed to perturb a long pattern region would be more than sufficient to disrupt local segments, essentially making the shared time series useless for the utility tasks (as will be illustrated in Section 8.6).

To address the dilemma between utility of short seg-

*George Mason University, {lzhang18, jessica}@gmu.edu
†University of Houston, jding7@uh.edu
‡University of Texas Rio Grande Valley, yifeng.gao@utrgv.edu



(a) Existing perturbed data sharing pipeline



(b) Proposed Private Matrix Profile (PMP) Data Sharing and Mining pipeline.

Figure 1: Comparison between existing pipeline and our proposed pipeline. PMP is computed from the raw time series and then shared to the third party to protect the shape and location for long sensitive patterns while maintaining the utility on local patterns.

ments and privacy (preventing malicious inference on long shape-based patterns), we investigate an alternative approach by sharing Matrix Profile (MP) instead. MP is recent proposed as an efficient and versatile data structure that records a single distance and the index of its closest match for each subsequence of a given length in time series. There are two advantages to using MP. First, MP is ideal for third-party modelers to use as a versatile intermediate feature to support most fundamental data mining tasks such as motif discovery, anomaly detection, and more complex tasks such as rule discovery, segmentation, and data summarizing. Second, we found that it is difficult for malicious third-party to recover the original time series and the patterns themselves by solely using Matrix Profile due to the use of Z-normalized Euclidean distance (as discussed in Section 5). However, we found that the canonical correlation in MP index can still reveal the location of long patterns. Based on the observation, we design two attacks named Location Attack and Entropy Attack to retrieve the pattern location from MP. To further protect MP from these two attacks, we propose a Privacy-Aware Matrix Profile (PMP) via perturbing the local correlation and breaking the canonical correlation in MP index vector. Our overall framework is shown in Figure 1. Instead of sharing the raw time series or perturbed time series, the third-party modeler would only have access to the PMP. The modeler could still perform the utility task(s) based on PMP, but they would not be able to infer the shapes nor the locations of long sensitive patterns.

In summary, our contributions are listed as follows:

- We consider a new privacy preserving problem: preventing malicious inference on long shape-based patterns while preserving short segment information for the downstream task performance. To the best of our knowledge, this is the first work to investigate this problem, and it cannot be protected by existing approaches.
- We investigate an alternative approach by sharing Matrix Profile (MP), a non-linear transformation of original data and a versatile data structure supporting many data mining tasks. We found that MP can protect sensitive long time series pattern shape, but could still leak the pattern location due to correlation in MP Index (MPI).
- We design two attacks based on location and entropy of MPI to extract sensitive pattern location from MP.
- We further propose a defense algorithm called *PatternHide* to generate a Privacy-aware Matrix Profile (PMP), which can prevent the location leakage of sensitive patterns while keeping the downstream task performance.
- We evaluate our proposed PMP against baseline methods through quantitative analysis and multiple real-world case studies to show the effectiveness of the proposed method.

2 RELATED WORK

In the last two decades, vast research efforts have been put into time series data mining tasks such as time series classification [3], discord discovery [16], motif discovery [22], and segmentation [14]. Different from point-based tasks, pattern-based time series approaches are based on similarity on the subsequence level instead of point values to capture the notion of *shape* information, which goes beyond point-values and reifies human natural understanding and visualization [9]. Pattern-based time series methods handle large scale time series well with excellent performance and interpretation. Recently, Matrix Profile (MP) [27, 30] is proposed as an efficient and effective data representation on subsequence level and support most major fundamental tasks for downstream applications in a broad range of domains applications [27, 30, 28]. Existing work such as [31] typically use the raw data to compute MP as the first, feature generation step, and then design algorithms or models based on the computed MP. None of the work considers the use of MP in the context of data sharing with a third party, nor the privacy issues raised by sensitive patterns.

Most existing approaches for the privacy-preserving release are designed for low-resolution time series based on differential privacy (DP) to protect the events where each event is associated with a single point. For example, Dwork et al. [7] proposed a binary tree based DP algorithm for single events in finite streams. Fan et al. [10] presented FAST for realizing DP on user-based finite streams with a framework of sampling-and-filtering.However, these approaches are designed for low-resolution time series and cannot be used to protect long sequence while maintaining the downstream task performance. In addition, our problem is also different from PatternLDP [24], which is designed to protect point values from malicious attacks while preserving the utility of local patterns, whereas our problem is in the opposite direction to protect sensitive long pattern while keeping the utility of short patterns.

In summary, these time series DP solutions cannot be directly applied to our setting, since these approaches mainly focus on computing the aggregated estimates (e.g., prefix-sum and moving average) of the values under DP, whereas our goal is to release subsequence-level representation that prevents leaking the information of long patterns while maintaining different downstream task performance.

Another line of research focuses on sharing encrypted time-series data. However, these methods can only support aggregation statistics computation and cannot support time series data mining tasks such as motif discovery and anomaly detection. Moreover, they mostly rely on oblivious RAM, secure multiparty computation, and function secret sharing [4, 5, 6], which require a significant number of cryptographic operations and secure communication channels.

To the best of our knowledge, there has been no existing work that offers solution for the data sharing issue or potential leakage of patterns from using Matrix Profile, nor the task to protect from long pattern inference while maintaining downstream task performance on local segments.

3 Background and Preliminaries

In this section, we first review necessary time series related notations.

Time Series $T = [t_1, t_2, \ldots, t_n]$ is a set of observation ordered by time, where t_i is a finite real number and n is the length of time series T.

Subsequence $S^T_{i,l} = [t_i, t_{i+1}, \ldots, t_{i+l-1}]$ of time series T is a contiguous set of points starting from position i with length l. Typically $l \ll n$, and $1 \leq i \leq n - l + 1$.

Previous work such as [27, 16] require subsequence comparison be non-trivial match, which prevents the comparison of subsequences that overlap more than 50% of the length.

Non-trivial Match Given a time series T, a subsequence $S_{i,l}$ with length l is considered a *non-trivial match* of another subsequence $S_{j,l}$ of length l if $|i-j| > l/2$.

In many applications, we are interested in finding similar "shapes" between two subsequences. Z-normalized Euclidean distance is used to achieve scale and offset invariance [17].

Z-normalized Euclidean Distance (Z-norm ED): $d(S_{p,l}, S_{q,l})$ of subsequences $S_{p,l}$, $S_{q,l}$ of length l is computed as $\sqrt{\sum_{m=1}^{l} (\frac{t_{p+m-1}-\mu_{p,l}}{\sigma_{p,l}} - \frac{t_{q+m-1}-\mu_{q,l}}{\sigma_{q,l}})^2}$, where $\mu_{p,l}$, $\sigma_{p,l}$ and $\mu_{q,l}$, $\sigma_{q,l}$ are the means and standard deviations of subsequences $S_{p,l}$ and $S_{q,l}$, respectively.

Z-normalization step is very critical, as noted in previous work — "without normalization time series similarity has essentially no meaning. More concretely, very small changes in offset rapidly dwarf any information about the shape of the two time series in question" [15]. There is additional benefit of preventing data leaking brought by utilizing Z-normalization distance, and we will discuss in details in Sec. 5.

Distance Profile Given a query subsequence Q_l of length l and a time series T, a distance profile $D_T(Q)$ is a vector containing the Z-normalized Euclidean distances between Q and each subsequence of the same length in time series T. Formally, $D(Q_l, T) = [d(Q_l, S^T_1), d(Q_l, S^T_2), \cdots, d(Q_l, S^T_{n-l+1})]$.

Matrix Profile Distance (MPD): Matrix Profile Distance of time series T given subsequence length l is a vector of the Z-normalized Euclidean distances between every subsequence $S_{i,l}$ and its nearest neighbor (most similar) subsequence in time series T. Formally, $MP = [\min(D(S_{1,l}, T)), \min(d(S_{2,l}, T)), \cdots, \min(d(S_{n-l+1,l}, T))]$.

Matrix Profile Index (MPI): Matrix Profile Index of time series T is a vector of indices containing the index of the non-trivial match of nearest neighbor subsequence of subsequence $S_{i,l}$ in time series T. Formally, $\text{MPI} = [\arg\min(d(S_{1,l}, T), \cdots, \arg\min(d(S_{n-l+1,l}, T)]$. Matrix Profile (MP) consists of two vectors, MPD and MPI. MP contains rich information about the data and the pattern location. For example, time series motif [22] can be found by exacting minimum value of MPD and the corresponding MPI.

4 Problem Statement

In practice, companies may utilize sensitive data for data mining in order to provide better services. As shown in Fig. 1(a), some data owner (e.g., factories with smart sensors, e-commerce platforms and hospitals) may capture clients' time series and send to cloud

servers/third party modelers for data analysis. The third party modelers will run Matrix Profile and then design a model for improving the quality of data owners' service or production-related decision making. However, directly transmitting raw time series or perturbed time series to modelers would allow malicious third-party modelers to use the long patterns to infer sensitive information as we explained earlier in Introduction.

Since it is risky to share the data directly, a better protocol (Fig. 1(b)) is to first generate the MPD and MPI with a given length for the utility tasks, and then send it to the external cloud servers/modelers for utility tasks. For convenience, this given length is referred as the **utility length** and denoted as L_{util}. There are two key research problems we should answer to comprehensively evaluate the privacy preserving performance of Matrix Profile:

- *Does sharing MPD and MPI of length L_{util} protect the shapes of long patterns?*
- *Does sharing MPD and MPI of length L_{util} protect the locations for long patterns?*

5 Advantages of Matrix Profile for Privacy Protection

To answer above questions, we conduct a comprehensive study of Matrix Profile in terms of the privacy protection for long patterns.

1) Difficult to recover raw data from MPD and MPI We found that it is very difficult to recover the concrete values of time series solely from the shared Matrix Profile because of Z-normalization. Recall that Z-normalization requires the knowledge of the mean and the standard deviation for both subsequences (for equation, see Z-normalized Euclidean Distance in Section 3). To extract entire the time series from MP, we would need to know the mean and standard deviation for every pair of subsequences. Thus, if we would like to recover the data from the distance, we have to solve for the values of data from pairwise distance between subsequences with the mean $\mu_{i,l}$ and standard deviation $\sigma_{i,l}$ as parameters in every equation. As σ_i is a non-linear function of variable t_i to t_{i+l-1}, it makes the inverse problem ill-defined. One could get infinitely many possible time series data as possible solutions satisfying a given MP, and hence cannot retrieve the original data.

2) Difficulty to infer long pattern location from MPD In addition, as pointed out by previous work [11], "the distance between a pair of short subsequences does not necessarily share similar behavior with the distance between long subsequences if the length difference is large". As a result, MPD naturally suppress the location correlation between small pattern and the long pattern. Thus, it is difficult to recover large pattern location from a short length MPD.

6 Threat Model and Attack Methods

While MP provides these two advantages in protecting privacy, since both MPD and MPI have to be shared to fulfill task utility, we found that attacker still can potentially derive the sensitive pattern location from MPI. We first define the threat model as follows:

Adversary Goal. Given MPD and MPI, the adversary aims to locate a pair of sensitive motifs of a sensitive length and we assume it is much longer than utility length L_{util}. For convinience, we refer to this length as **attack length** and denote it as L_{attack}. Specifically, the adversary goal is to detect frequent patterns in time series, which is similar to motif discovery task [22, 21, 18, 20] with one exception – the attacker could only use the shared Matrix Profile of length L_{util} to detect the long motif.

Following the widely used evaluation criteria [13, 22, 21, 12], we use *the success rate* to measure how accurate the detected pair of patterns match the actual locations.

Adversary Knowledge. We assume a strong adversary (e.g., honest-but-curious modeler), who has no access to the sensitive time series data, but has the white-box access to the Matrix Profile Distance MPD and Matrix Profile Index MPI with short subsequence length L_{util} pre-generated from the sensitive data.

6.1 Attack Methods

We use a real-world time series to demonstrate how MPI can leak long pattern information. Figure 2.top shows a snippet of a dishwasher power consumption time series. The time series contains two long dishwasher cycles. Obviously, at both pattern locations, the index evolution is more smooth and gradual than other region. This is because the consecutive 1-NN locations in the pattern region are likely similar since each subsequence is only offset by one point. Therefore, the characteristic of a MPI block of length L_{attack} may leak pattern locations. In addition, the histograms of the index among the two dish washer cycles are shown in Fig. 3.left and Fig. 3.right. The most frequent indices in both dishwasher cycles indeed correspond to the long pattern location of the other dishwasher cycle in the time series.

Inspired by the above intuition, we introduce our basic framework of the attack method. The algorithm is illustrated in Algorithm 1. Given a sensitive pattern length L_{attack}, the attack algorithm will first scan through the MP time series with a sliding window equal to L_{attack} and assigns the significant score (Line 4). The candidate with the most outstanding score will be identified as the location of the first pattern instance

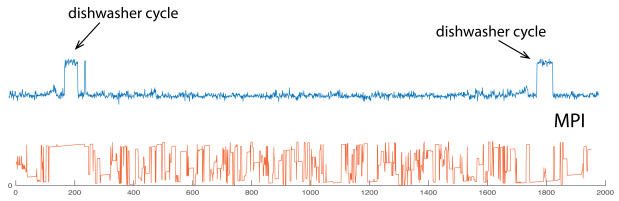


Figure 2: MPI index of a dishwasher time series with two dishwasher cycles

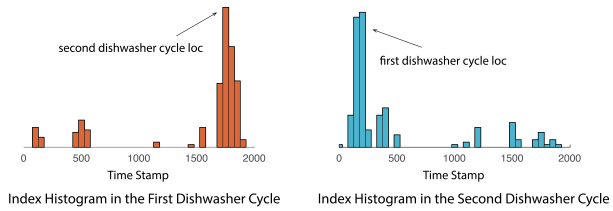


Figure 3: MPI index of a dishwasher time series with two dishwasher cycles

(i.e., $idx1$). Then, the second instance's location is identified by computing the histogram of all indexes belongs MPI$[idx1 : idx1 + L_{attack} - 1]$. The centroid of the highest frequent bin is assigned as the second instance's location (e.g., the peak in Fig. 3) (Line 7-8). Different significant score in Line 4 will result in different attack methods. In this paper, we propose two scores based on MPI for this framework:

- **Location-based Score:** The length of the consecutive index in the sliding window.
- **Entropy-based Score:** The negative entropy value given all indices in the sliding window.

We refer the first strategy as Location based Attack and the second strategy as Entropy-based Attack. Two attack methods could retrieve some degree of long pattern information if MP is not protected.

Note that existing motif discovery algorithms [22, 21, 18, 20] could not be used to detect the long motif of length L_{attack} through the shared Matrix Profile of length L_{util} due to the lack of access to the original time series data. However, our proposed attack methods can still find long motif from a single shared MPI.

7 Defense Strategy

In this section, we first introduce a new concept named Consecutive Index Block (CIB), and a new matrix profile, Masked Matrix Profile (Masked MP), which is highly related to the proposed algorithm. Finally, we introduce our proposed *PatternHide* algorithm.

7.1 Consecutive Index Block and Masked Matrix Profile We introduce Consecutive Index Block (CIB) to capture the gradual changed regions in MPI that highly related to long patterns. Specifically,

Consecutive Index Block (CIB) Given a Matrix Profile index MPI, a Consecutive Index Block (CIB) C_k consists of a set of consecutive index starting from index k where for any index $i \in C_k$, we have $|\text{MPI}(i) - \text{MPI}(i+1)| < L_{attack}$.

Intuitively, the overall defense strategy of the proposed algorithm is perturbing and hiding any outstanding CIB regions in MPI because long CIB potentially aligns with long pattern location and leads to pattern leakage. We next introduce Masked MP, the alternative "fake" MP used to hide real MP, while maintaining compatible functionality.

Masked Matrix Profile Given a mask vector M, Masked Matrix Profile is computed through the same process as Matrix Profile but exclude any masked locations in M during computing MPI and MPD.

Algorithm 1 Proposed Attack Algorithm

1: **Input**: MPD, MPI, L_{attack}
2: **Output**: long pair motif indices $idx1$, $idx2$
3: /* Calculate sum of distances in each sliding window of length L_{attack}*/
4: SumDist = Score(MPD, L_{attack})
5: $idx1 = \arg\min(\text{SumDist})$
6: /* Identify second motif index based 1st motif index*/
7: $idxPool = \text{MPI}[idx1 : idx1 + L_{attack} - 1]$
8: $idx2 = \arg\max$ (GetIntervalFrequency($idxPool$))
9: **return** $idx1$, $idx2$

7.2 Proposed Algorithm The proposed algorithm, *PatternHide*, is described in Algorithm 2. Given a matrix profile MP, the algorithm consists of three steps. First, the algorithm obtains all the CIBs $\mathcal{C}$ in T and forms a set of sensitive segments $\mathcal{S}$ that may leak the pattern information. A segment is sensitive if it is close (index difference is less than L_{attack}) to a long CIB C_i with length greater than L_{perm} (Line 4-8). Then the algorithm will perform the permutation step to generate "fake" 1-NN via masked matrix profile to break down long CIBs into small pieces that is similar to the non-sensitive area for every sensitive segment (Line 10-24). Finally, after checking all the sensitive segments, the algorithm will further examine existing cycles in MPI and modify any conflicting distance values to ensure consistency with the MPI.

1)Breakdown Long CIBs in Sensitive Segments The goal of this component is to modify the indices in any sensitive segments that potentially leak the long pattern, but still keep most of the functionality of the

original matrix profile. To achieve this goal, we replace the MPD and MPI with Masked Matrix Profile (Line 10-23) which is computed with the sensitive area masked to ensure that no CIB of significantly large length exists. In the algorithm, if the consecutive index length is greater than a randomly generated threshold L_{rnd} at time point i, we update the corresponding MPI and MPD with that of mask matrix profile. Specifically, given a sensitive segment S, the algorithm maintains a mask vector M to control the permuted matrix profile (Line 10-11). Every time the condition in Line 14 is met, any subsequence overlapped with MPI(i) is added into mask vector (Line 16) and triggers the permutation process (Line 17-20). In the process, the algorithm computes masked matrix profile with M (Line 17) and replaces the remaining MPD and MPI in the sensitive segments with newly computed mask MP (Line 19-20). Then the algorithm samples another length threshold L_{rnd} to prepare for the next pattern hiding operation (Line 13). The mask M is set to zero vector after examining each sensitive segment. Note that the time complexity of this component is the same as computing a matrix profile because it only require query time series length N times 1-NN queries.

2) Resolve Conflicts in MPD Every MPD value is a distance and MPI may form a 'cycle' (i.e., given two subsequences S_i and S_j, MPI(i) = j and MPI(j) = i. In this case, we need to enforce distance symmetry constraint MPD(j) = MPD(i), otherwise a smart attacker might use this loophole to infer the modified locations. Therefore, we further generate fake symmetric distances (Line 25). Finally, the algorithm identifies any 'cycles' in the MPI and checks if MPD(i) = MPD(j) constraint is violates or not. If the distances are not equal, it adjusts MPD value to be the minimum of the two.

Algorithm 2 Defense Algorithm: *PatternHide*

1: **Input**: T, MPD, MPI, L_{perm}, L_{attack}
2: **Output**: PMP = {MPD, MPI}
3: $\mathcal{C}$ = GetCIB(MPI)
4: **for** Each $C_i \in \mathcal{C}$ and $C_i > L_{perm}$ **do**
5: /* Obtain Sensitive Intervals*/
6: $\mathcal{C}_{neighbor} = \{C| \quad |C.start - C_i.start| < L_{attack}\}$
7: S = concate($C_i \cup C_{neighbor}$)
8: $\mathcal{S}$.add(S)
9: **end for**
10: **for** $S \in \mathcal{S}$ **do**
11: $M = \{\}$
12: **for** idx in [$S.Start$, $S.End$] **do**
13: $L_{rnd} \sim U(0, L_{perm})$
14: **if** ConsecutiveIdxCount(idx) > L_{rnd} **then**
15: /* Compute Masked Matrix Profile*/
16: M.add(OverlappingIntervals(MPI(idx)))
17: MPD$'$, MPI$'$ = MaskMatrixProfile(T, M)
18: /* Replace Original Matrix Profile*/
19: MPD[$idx : S.End$] = MPD$'$[$idx : S.End$]
20: MPI[$idx : S.End$] = MPI$'$[$idx : S.End$]
21: **end if**
22: **end for**
23: **end for**
24: /* Resolve any Symmetric Conflicts in Cycle Link Indexes by In-Place Update*/
25: MPD, MPI = FakeCycleLink(MPD, MPI)
26: **return** MPD, MPI

7.3 Advantage of Proposed Algorithm

Our defense algorithm is carefully designed and has two advantages attributed to the masked MP replacement. First, our algorithm plays a specific defense strategy to the attack method on protecting the location of large motif index, while keeping utility task performance in mind. The replaced values are still meaningful as similarity information can be seen as an approximation to the information recorded by actual MPD and MPI. Second, our strategy only perturb a small amount of index, so there is less information loss compared with the original MP.

8 Experimental Evaluation

In this section, we demonstrate that the proposed defense methods can successfully defend the proposed two attacks while maintaining the utility task performance on both real-world and synthetic data. Unless otherwise specified, the parameter L_{perm} is set to $L_{perm} = L_{util}/4$.

8.1 Detecting Planted Motif while Protected Pattern Leakage

We first evaluate the proposed defense method in motif discovery. Specifically, the utility task in the experiment is detecting motifs of length $L_{utility}$ while keeping the motif of length L_{attack} protected. Following the previous planted motif evaluation experiment setting [13, 12, 21], we test our Privacy-Aware MP in two different scenarios:

1) Independent Scenario: In this scenario, sensitive patterns are independent of non-sensitive patterns. We randomly planted two independent motifs of lengths L_{util} and L_{attack}, each one with two instances, into a long time series.

2) Correlation Scenario: In this scenario, the location of sensitive pattern is correlated with non-sensitive pattern. We randomly planted a motif of length L_{attack} of two instances, and a motif of length L_{util} of three instances into the time series. Different from first experiment, two out of three instances of the short motif overlap with the long motif. Following the setting in large-scale planted motif experiment, the shape of motifs are

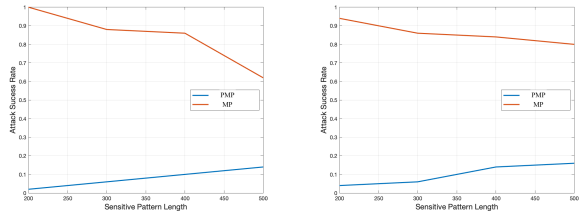
randomly generated by using: $p = \sum_{i=1}^{5} A_i \sin(\alpha_i x + \beta_i)$, with random parameters $A_i \in [0, 10]$, $\alpha_i \in [-2, 2]$ and $\beta_i \in [-\pi, \pi]$. Each instance of a motif, $\pm 5\%$ random noise is added. In both experiments, the length of the random walk time series is 10,000. We test four different $L_{attack} = \{200, 300, 400, 500\}$ while keeping the utility length $L_{util} = 100$. All the experiments were repeated 50 times with different time series and motif shapes.

8.2 Evaluation Criteria There are two evaluation criteria: average utility task performance and average attack success rate. Both criteria are evaluated based on motif detection rate. The overlapping rate is measured by Jaccard similarity index: $J(pred, gt) = \frac{pred \cap gt}{pred \cup gt}$. If the overlapping rate of the detected location and ground truth is greater than 0.25 in both instances, we say the motif is detected. The average attack success rate is computed based on the success of locating motif of length L_{attack} in the 50 experiments for each length. The utility task performance is measured by average motif discovery success rate of length L_{util}.

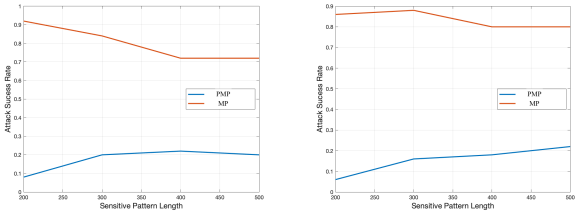
8.3 Baselines To the best of our knowledge, this is the first work that investigates the approach to prevent information leakage from deducing long pattern. None of existing approaches are designed for this problem. Therefore, we compare with two simple baselines: (1) **directly sharing MP** and (2) **adding noise in raw data**. For the second approach, we test three different variations, by adding small, medium, and large amounts of noise, respectively.

8.4 Attack Methods For MP sharing strategies, we evaluate the defense performance based on the success rate of two attack methods introduced in Sec. 6. For the approach that directly shares the matrix profile, we evaluate the performance by the success rate of directly detecting motif of L_{attack} given the shared time series.

8.5 Vs. Sharing Original Matrix Profile We first compare the proposed PMP with the original MP. We apply both attacks described in Section 6 and report the attack success rate. Fig. 4(a)-(b) show the attack success rates in the non-overlap setting. According to the figure, PMP can significantly reduce the attack success rate. When the sensitive pattern length increases, the attack success rate on sharing MP decreases. This is because the correlation between MP of L_{util} and L_{attack} is much smaller when the length increases, making it harder for attackers to retrieve information. However, our attack methods still maintain up to 70% success rate. Compared with sharing the original matrix profile, PMP maintains a stable low attack success rate (less than 0.15). Fig.



(a) Entropy-based Attack (b) Location-based Attack



(c) Entropy-based Attack (d) Location-based Attack

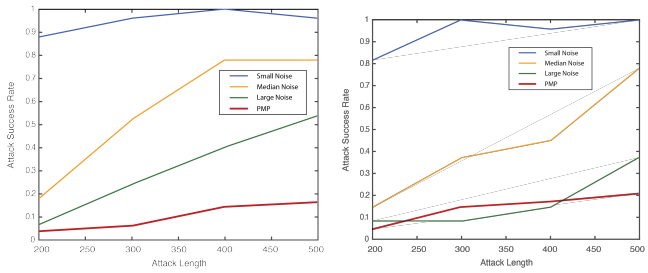
Figure 4: Compared with Sharing MP under Different L_{attack} (a-b) Independent Scenario, (c-d) Correlation Scenario.

4(c)-(d) show the attack success rates in the overlap setting. Similar observation is found when we test the overlapping case. Sharing original matrix profile will lead to at least 0.7 success attack rate while the proposed approach can significantly reduce the chance of success in attack (up to 0.21). Moreover, the utility of PMP is shown in Table 1. From the table, it is about 0.88, which indicates our defense method can successfully defend the attacks while keeping the utility of the shorter segments.

8.6 Vs. Perturbed Raw Series We next test our proposed method with perturbed raw data. Specifically, we compare with sharing raw data after adding Gaussian noise with variance $\sigma^2 = \{0.1, 0.3, 0.5\}$, respectively. The attack success rates for all three cases are shown in Fig. 5(a)-(b) for both experiments. The utility task performance is shown in Table 1.

In both experiments (Independent Scenario and Correlation Scenario as explained in Section 8.1), adding only small amounts of noise will result in high attack success rate and high utility of the data. When more noise is added, both attack success rate and the utility decrease. However, the utility decreases faster than the attack success rate. This is because small utility length pattern is much easier to be affected compared with long pattern. Moreover, compared with noise-adding approach, sharing PMP could maintain a very high level of utility (0.875 success rate on motif detection) while keeping the attack success rate close to the best defense performance of sharing perturbed time se-

ries. The experiment shows that the proposed approach outperforms the perturbed raw time series protocol.



(a) (b)

Figure 5: Compared with Sharing Perturbed Time Series **(a)** Independent Scenario **(b)** Correlation Scenario

Table 1: Shared PMP vs. Perturb Raw Time Series

Setting Method	small σ	median σ	large σ	PMP
Independent (utility)	0.84	0.32	0.115	**0.875**
Correlation (utility)	0.7	0.295	0.08	**0.882**

8.7 Motif Discovery on Electrooculogram Time Series In sleep quality study, Electrooculogram is a popular data type to study the sleep behavior of a subject [19]. In this case study, we test our proposed method on preventing the leakage of long eye blinking activities from the Electrooculogram time Series. We utilized the EOG time series used by Madrid et al. [19]. According to the authors, the time series is captured from a 66- year old healthy male recorded during a sleep study. The time series is shown in Fig. 6(a). Two types of motifs which correspond to two different types of eye blinking activities are highlighted in the time series. Without any perturbation, it is very easy for the attacker to retrieve the type II eye blinking pattern through the location/entropy based attack method. The detected pattern is shown in Fig. 9(b). After the

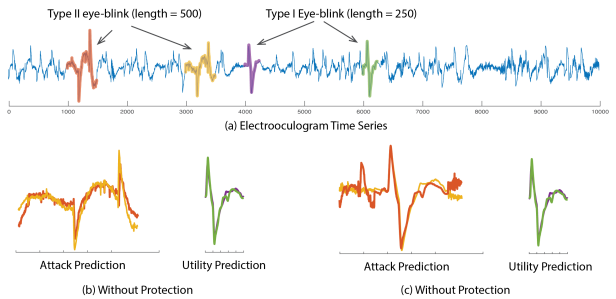


Figure 6: Protected MP successfully Prevent Type II Eye Blink Activity Leak

perturbation, the attack algorithm fails to detect the long pattern. In fact, it locates the pattern somewhere overlapped with the small motif. Therefore, the proposed perturbed Matrix Profile protects the data from the attacker.

9 Conclusion

In this paper, we consider a new type of privacy preserving problem in time series, i.e., preventing malicious inference on long shape-based patterns while preserving short segment information for the downstream task. To deal with the above problem, we introduced a shared Matrix Profile (MP) approach by utilizing the characteristics of MP as a stand-alone and versatile intermediate features. However, we illustrated that MP index sharing can still reveal the location of sensitive long pattern information based on two proposed attack methods. We further proposed a Privacy-Aware Matrix Profile (PMP) based on perturbing the index of matrix profile at sensitive pattern regions. We evaluated our proposed PMP sharing solution with several classic defense methods through quantitative analysis and demonstrated the effectiveness of protecting sensitive information in the real-world case study. This work will be a good first step that will hopefully inspire new interesting research for privacy-aware shape-based time series data mining methods.

References

[1] A. Abdoli, S. Alaee, S. Imani, A. Murillo, A. Gerry, L. Hickle, and E. Keogh. Fitbit for chickens? time series data mining can increase the productivity of poultry farms. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3328–3336, 2020.

[2] S. Avancha, A. Baxi, and D. Kotz. Privacy in mobile technology for personal healthcare. *ACM Computing Surveys (CSUR)*, 45(1):1–54, 2012.

[3] A. Bagnall, H. A. Dau, J. Lines, M. Flynn, J. Large, A. Bostrom, P. Southam, and E. Keogh. The uea multivariate time series classification archive, 2018. *arXiv preprint arXiv:1811.00075*, 2018.

[4] L. Burkhalter, A. Hithnawi, A. Viand, H. Shafagh, and S. Ratnasamy. {TimeCrypt}: Encrypted data stream processing at scale with cryptographic access control. In *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)*, pages 835–850, 2020.

[5] L. Burkhalter, N. Küchler, A. Viand, H. Shafagh, and A. Hithnawi. Zeph: Cryptographic enforcement of end-to-end data privacy. In *15th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 21)*, pages 387–404, 2021.

[6] E. Dauterman, M. Rathee, R. A. Popa, and I. Stoica. Waldo: A private time-series database from function

secret sharing. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 2450–2468. IEEE, 2022.

[7] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum. Differential privacy under continual observation. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 715–724, 2010.

[8] E. Erdemir, P. L. Dragotti, and D. Gündüz. Privacy-aware time-series data sharing with deep reinforcement learning. *IEEE Transactions on Information Forensics and Security*, 16:389–401, 2020.

[9] P. Esling and C. Agon. Time-series data mining. *ACM Computing Surveys (CSUR)*, 45(1):1–34, 2012.

[10] L. Fan and L. Xiong. An adaptive approach to real-time aggregate monitoring with differential privacy. *IEEE Transactions on knowledge and data engineering*, 26(9):2094–2106, 2013.

[11] Y. Gao and J. Lin. Efficient discovery of variable-length time series motifs with large length range in million scale time series. In *2017 IEEE 17th International Conference on Data Mining (ICDM)*, 2017.

[12] Y. Gao and J. Lin. Exploring variable-length time series motifs in one hundred million length scale. *Data Mining and Knowledge Discovery*, 32(5):1200–1228, 2018.

[13] Y. Gao and J. Lin. Exploring variable-length time series motifs in one hundred million length scale. *Data Mining and Knowledge Discovery*, May 2018.

[14] S. Gharghabi, Y. Ding, C.-C. M. Yeh, K. Kamgar, L. Ulanova, and E. Keogh. Matrix profile viii: domain agnostic online semantic segmentation at superhuman performance levels. In *2017 IEEE international conference on data mining (ICDM)*, pages 117–126. IEEE, 2017.

[15] E. Keogh and S. Kasetty. On the need for time series data mining benchmarks: a survey and empirical demonstration. *Data Mining and knowledge discovery*, 7(4):349–371, 2003.

[16] E. Keogh, J. Lin, and A. Fu. Hot sax: Efficiently finding the most unusual time series subsequence. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, pages 8–pp. IEEE, 2005.

[17] J. Lin, E. Keogh, L. Wei, and S. Lonardi. Experiencing sax: a novel symbolic representation of time series. *Data Mining and knowledge discovery*, 15(2):107–144, 2007.

[18] M. Linardi, Y. Zhu, T. Palpanas, and E. Keogh. Matrix profile x: Valmod-scalable discovery of variable-length motifs in data series. In *Proceedings of the 2018 International Conference on Management of Data*, pages 1053–1066, 2018.

[19] F. Madrid, S. Imani, R. Mercer, Z. Zimmerman, N. Shakibay, and E. Keogh. Matrix profile xx: Finding and visualizing time series motifs of all lengths using the matrix profile. In *2019 IEEE International Conference on Big Knowledge (ICBK)*, pages 175–182. IEEE, 2019.

[20] R. Mercer, S. Alaee, A. Abdoli, S. Singh, A. Murillo, and E. Keogh. Matrix profile xxiii: Contrast profile: A novel time series primitive that allows real world classification. In *2021 IEEE International Conference on Data Mining (ICDM)*, pages 1240–1245. IEEE, 2021.

[21] A. Mueen. Enumeration of time series motifs of all lengths. In *13th International Conference on Data Mining (ICDM), 2013*, pages 547–556. IEEE, 2013.

[22] A. Mueen, E. J. Keogh, Q. Zhu, S. Cash, and M. B. Westover. Exact discovery of time series motifs. In *SDM*, pages 473–484. SIAM, 2009.

[23] A. K. Tyagi and D. Goyal. A survey of privacy leakage and security vulnerabilities in the internet of things. In *2020 5th International conference on communication and electronics systems (ICCES)*, pages 386–394. IEEE, 2020.

[24] Z. Wang, W. Liu, X. Pang, J. Ren, Z. Liu, and Y. Chen. Towards pattern-aware privacy-preserving real-time data collection. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pages 109–118. IEEE, 2020.

[25] Y. Xiao and L. Xiong. Protecting locations with differential privacy under temporal correlations. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1298–1309, 2015.

[26] S. Yao, S. Hu, Y. Zhao, A. Zhang, and T. Abdelzaher. Deepsense: A unified deep learning framework for time-series mobile sensing data processing. In *Proceedings of the 26th international conference on world wide web*, pages 351–360, 2017.

[27] C.-C. M. Yeh, Y. Zhu, L. Ulanova, N. Begum, Y. Ding, H. A. Dau, D. F. Silva, A. Mueen, and E. 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, 2016.

[28] Y. Zhu, S. Gharghabi, D. F. Silva, H. A. Dau, C.-C. M. Yeh, S. Senobari, et al. The swiss army knife of time series data mining: ten useful things you can do with the matrix profile and ten lines of code. *Data Mining and Knowledge Discovery*, 34(4):949–979, 2020.

[29] Y. Zhu, M. Imamura, D. Nikovski, and E. Keogh. Matrix profile vii: Time series chains: A new primitive for time series data mining (best student paper award). In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 695–704. IEEE, 2017.

[30] 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 *Data Mining (ICDM), 2016 IEEE 16th International Conference on*, pages 739–748. IEEE, 2016.

[31] M. Zymbler and E. Ivanova. Matrix profile-based approach to industrial sensor data analysis inside rdbms. *Mathematics*, 9(17):2146, 2021.