Privacy Attacks on Matrix Profiles via Reconstruction Techniques

Haoying Zhang
INSA CVL, Univ. d'Orléans, Inria
France
haoying.zhang@inria.fr

Nicolas Anciaux
Inria, INSA CVL, Univ. Paris Saclay
France
nicolas.anciaux@inria.fr

Benjamin Nguyen
INSA CVL, Univ. d'Orléans, Inria
France
benjamin.nguyen@insa-cvl.fr

Fabien Girard
ENSTA
France
fabien.girard@ensta-paris.fr

Jose Maria de Fuentes
Univ. Carlos III de Madrid
Spain
josemaria.defuentes@uc3m.es

Adrien Boiret
INSA CVL, Univ. d'Orléans, Inria
France
adrien.boiret@insa-cvl.fr

Abstract

Matrix Profile (MP) is a data mining structure increasingly used for time series analysis in both academic and industrial contexts. Given its application to sensitive domains such as healthcare or energy monitoring, it is crucial to examine associated privacy risks, especially since MPs are often shared or processed in untrusted environments like the cloud. While recent studies suggest that MPs offer some privacy protection, this assumption remains largely untested. This paper analyzes the privacy risks of MP publication through the lens of EU data protection law, focusing on singling-out, linkability, and inference risks. We introduce a reconstruction technique based on constraint optimization, capable of recovering approximate original time series from their MPs, leading to severe privacy attacks. Experiments on real-world datasets reveal vulnerabilities to all attack types, with reconstructed series reaching up to 0.99 Pearson Correlation with the original.

1 Introduction

Matrix Profile (MP) [66] has emerged as a versatile data mining structure for analyzing and sharing time series data in domains such as healthcare [23, 60], energy management [36, 53], anomaly detection [36, 45], and continuous authentication [28], in both cloud and edge models [35]. It has seen increasing adoption in real-world industrial settings for time series analysis and anomaly detection. For instance, Oracle employs MP techniques to enhance root cause analysis in vehicle manufacturing. By detecting patterns and anomalies in time series data from machines and products, these methods bolster intelligent observability for IoT applications [42]. In the financial sector, TD Ameritrade integrates the open-source stumpy Python library to implement scalable MP-based analytics for time series data, enabling robust and interpretable data mining pipelines for monitoring and investment tools [3]. Similarly, Amazon Web Services (AWS) demonstrate the use of MP for real-time anomaly detection in streaming applications via the Apache Flink service, highlighting MP's value in scalable cloud analytics environments [2]. Importantly, real-world applications of MP often concern personal data, including biomedical data from patients, such as Electro-Cardiograms (ECG) [40, 69] or beat-to-beat intervals [64], smart-home measurements (e.g., for energy consumption [12, 41, 56] and home sensors [7]). [31] lists numerous but non exhaustive additional application examples in research. Also, MP computation libraries are integrated officially into programming languages used for data analysis, such as Python, R, and MatLab (since release R2024B in sept 2024), encouraging their use in academia and beyond.

Among all MP uses, we focus on the essential unsupervised anomaly detection in time series using self-join MP (comparing a time series with itself, see Figure 1). Comparing a time series to another reference/template time series is possible, but is neither always feasible nor unsupervised, hence the need for self-join MP. E.g., the Statewide California Earthquake Center [70] explain that they use the self-join MP to find non cataloged events by matching all sub-windows in the continuous stream with the rest of the stream, which can not be done with a template matching method.

By transforming raw time series into more complex representations using various mathematical techniques, MPs are assumed to obscure the original data sufficiently, offering a perceived level of privacy protection. This perception has fueled claims that MP can act as a privacy-preserving tool. For instance, recent work [68] suggest that the complexity of MP transformations prevents shape leakage and hinders raw data reconstruction. Similarly, [16] propose generating synthetic time series that preserve MPs while altering the shape of the original data, ensuring utility while mitigating sensitive attribute inference risks. In addition, [28] leverages MPs as privacy-enhancing features in deep learning algorithms, achieving high accuracy in continuous authentication tasks.

However, consider a concrete scenario in the healthcare domain: a hospital publishes MPs derived from patients' electrocardiogram (ECG) data for research purposes. An adversary with domain knowledge could use MPs to infer parts of the original ECG data, use it single out or identify patients and deduce medical markers or demographic attributes. This would violate patient privacy and could lead to severe consequences, such as stigmatization, insurance discrimination, or unauthorized profiling.

This example highlights the critical nature of being able to quantify the privacy risks associated with MPs. While the nonlinear transformations involved in MP computation introduce some form

Proceedings on Privacy Enhancing Technologies 2026(1), 65–86

https://doi.org/10.56553/popets-2026-0005

of blurring, they do not inherently prevent adversaries from exploiting auxiliary information to infer sensitive attributes or reconstruct the original time series.

Research question. The central question of this work is: *What are the privacy risks associated with publishing Matrix Profiles, particularly in privacy-sensitive domains?* This question is critical, given the increasing adoption of MPs as a tool for data sharing and analysis, often in unsecured environments such as cloud systems [35]. The assumption that MPs inherently protect privacy must be rigorously evaluated, especially in light of modern privacy threats.

Formally, the problem lies in assessing whether MPs are vulnerable to major privacy risks identified in privacy regulations, such as singling-out, linkability, and inference attacks, as defined by the European Data Protection Board (EDPB) [21, 22, 61]. It has lead us to consider reconstruction techniques to mount attacks that recover complete or partial sensitive raw time series from their MP. These risks reflect realistic threats that could compromise the privacy of individuals whose data is encoded in MPs.

Limits of existing solutions. Anonymeter [22] is a framework developed to quantify privacy risks. Its core contribution is to model and quantify the three GDPR inspired privacy risks (singling-out, linkability and inference) for synthetic tabular data. However, its model is not suitable for time series data (particularly MP data) which requires a privacy risk model that specifically identifies, defines, quantifies, and evaluates the risks of MP publishing, a topic that, to the best of our knowledge, remains unexplored.

Contributions. To answer this research question, the paper makes the following contributions:

- **Formalization of privacy risks in MPs publication.** We adapt the privacy risks of the EDPB to the context of MPs, and formally define the notions of singling-out, linkability, and inference in this context (Section 3).
- **Design of targeted attack strategies.** We develop concrete attack methods corresponding to each risk, with a particular focus on a novel reconstruction technique capable of approximating original time series from MP data. This reconstruction serves as a key building block for privacy attacks (Section 4).
- **Experimental validation on real data.** We evaluate the effectiveness of the proposed attacks on diverse real-world time series datasets, demonstrating that MP publication can lead to significant privacy breaches in practice. All code and data are made available for reproducibility at our repository [67] and will be discussed in Section 5.

Paper organization. Section 2 describes the background and problem formulation. Privacy attack models against *MP* are described in Section 3. The reconstruction technique and its application to attacks is presented in Section 4 and evaluated in Section 5. Related works are presented in Section 6. Future works and conclusion are given in Section 7.

Ethical considerations and responsible disclosure. No company systems were analyzed in this work. All referenced use of MP is based on public documentation or academic publications. Our goal is to inform on the limitations of MP as a privacy mechanism, not to attack specific deployments.

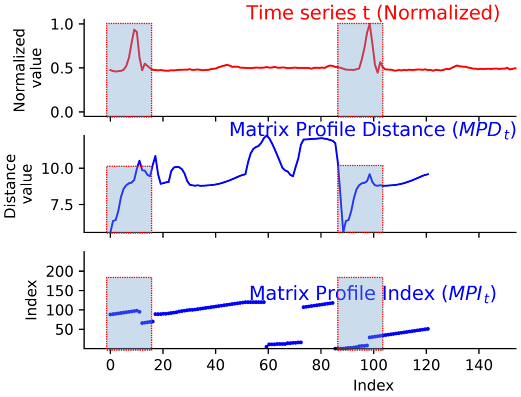


Figure 1: Matrix Profile for a time series t.

2 Problem Statement

We introduce the necessary background and notations concerning Matrix Profile (MP), and formulate the problem addressed.

2.1 Background on Matrix Profile

Matrix Profile (*MP*) [66] is a data mining structure composed of two vectors *MPD* and *MPI*, parameterized by two time series (t, t'), a distance measure *Dist* and a subsequence length m noted $MP_{Dist,m}(t, t') = (MPD_{Dist,m}(t, t'), MPI_{Dist,m}(t, t'))$, as shown in Figure 1. For this paper, we adopt a generalized definition adapted to any distance measure (e.g., Dynamic Time Warping, etc.).

Subsequence Notation. Consider two time series t and t' of lengths n and n'. We denote a subsequence $s^t_{i,m}$ as a continuous subset of values of t of length m, starting at index i. We denote the set of subsequences of length m of t as $Sub^t_m = \{s^t_{0,m}, ..., s^t_{n-m,m}\}$.

We introduce next the 1-NN subsequence, which will be used in the definition of *MP*:

Definition 2.1 (1-Nearest Neighbor (1-NN) subsequence). We say that $s^{t'}_{j,m}$ (the subsequence of t' starting at index j and of length m) is the 1-NN subsequence of a subsequence $s^t_{i,m}$ if $Dist(s^t_{i,m}, s^{t'}_{j,m})$ is the minimum value in $\{Dist(s^t_{i,m}, s^{t'}_{k,m}) | s^{t'}_{k,m} \in Sub^{t'}_m\}$.

Now we give the definitions of Matrix Profile Distance and Matrix Profile Index, which are the two components of a MP:

Definition 2.2 (Matrix Profile Distance). Given two time series t and t' of lengths n and n', the Matrix Profile Distance w.r.t a distance measure *Dist* and a subsequence length m, noted $MPD_{Dist,m}(t, t')$, is a **vector of size n-m+1** of the *distance* between each subsequence $s^t_{i,m} \in Sub^t_m$, and its 1-NN subsequence in t'. Formally,

$$MPD_{Dist,m}(t, t') = [\min_{j \in [0;n'-m]} (Dist(s^t_{i,m}, s^{t'}_{j,m})]_{i \in [0;n-m]}$$

Definition 2.3 (Matrix Profile Index). Using the same notations as in the previous definition, the Matrix Profile Index ($MPI_{Dist,m}(t, t')$) is a **vector of size n-m+1** of integers, where each entry corresponds to the *index* of the 1-NN subsequence in t' of a given subsequence $s^t_{i,m}$ in t. Formally,

$$MPI_{Dist,m}(t, t') = [\operatorname*{argmin}_{j \in [0;n'-m]} (Dist(s^t_{i,m}, s^{t'}_{j,m})]_{i \in [0;n-m]}$$

Symbol	Definition
t	A time series
n	The length of the time series
$\mathcal{D}$	The domain of the values in the time series
m	The subsequence length in the Matrix Profile
$s^t_{i,x}$	Subsequence in t starting from i, of length x
s_i	Subsequence in t starting from i of length $x = m$ (subsequence length in the Matrix Profile)
MP_t	The (self-join) Matrix Profile of t
MPD_t	The (self-join) Matrix Profile Distance vector of t
MPI_t	The (self-join) Matrix Profile Index vector of t
d_M	The Manhattan Distance
d_E	The Euclidean Distance
d_Z	The z-normalized Euclidean Distance
T	The set of time series considered
MP_T	The set of Matrix Profiles associated to the set of time series T

Table 1: Notations used in the paper.

Notation Simplifications. In most cases (and this will be the case in the rest of the article), one calculates the MP of a time series with itself (i.e., *self-join* with $t = t'$, thus $n = n'$), and both $Dist$ and m are omitted; thus the subsequence $s^t_{i,m}$ is simplified as s_i and $MP_{Dist,m}(t,t)$ is noted as MP_t, with the distance and index vectors noted as MPD_t and MPI_t. It is important to note that in this case, the computation of min (resp. argmin) in MPD (resp. MPI) are now typically taken over $j \in [0; n-m]$ *and* $|j-i| > m$, in order to exclude the trivial solutions where j is within the same m length subsequence as i. Thus we define:

$$MPD_t = \left[\min_{j\in[0;n-m]\ and\ |j-i|>m} (Dist(s_i,s_j)\right]_{i\in[0;n-m]}$$

$$MPI_t = \left[\operatorname{argmin}_{j\in[0;n-m]\ and\ |j-i|>m} (Dist(s_i,s_j)\right]_{i\in[0;n-m]}$$

Self-join MP is notably used to detect patterns in an unsupervised way. As illustrated in Figure 1, the first red time series represents an ECG signal. In order to build the MP, we consider the $n = 140$ first points with a window of size $m = 20$, containing two heartbeat phases, which are similar. The second and third graphs show the two components of the MP: the MPD and the MPI. In the MPD, the detection of similar patterns corresponds to the parts of the curve where the distance value is *lowest* (0 and 90 in Figure 1). The corresponding indexes are then read on the MPI. For instance, using the MPD we can see that a pattern beginning at index 0 (resp. 90) is similar to another pattern. Using MPI, we see that this pattern begins at index 90 (resp. 0).

For clarity, other notations concerning MP and used throughout the paper are summarized in Table 1.

2.2 Problem Definition

Matrix Profiles (MPs) are often viewed as offering privacy protection by design: they replace raw time series with derived representations involving z-normalization, distance-based similarity and non invertible operations like argmin. These transformations are believed to protect the underlying shape of the signal, supporting the assumption that MPs enhancing privacy [28, 68] and that synthetic data which preserve MPs can be considered privacy preserving [16].

However, such assumptions remain largely untested. In practice, the abstraction provided by MPs may not prevent adversaries from exploiting auxiliary knowledge to recover sensitive information. Recent advanced reconstruction methods could recover the original time series, enabling re-identification attacks or exposing sensitive attributes.

To study this systematically, we introduce a general model of privacy attacks on MPs, grounded in the categories identified by privacy regulations (e.g., GDPR). This includes singling-out, linkability, and inference, representing distinct risks depending on what the attacker knows and aims to extract. Formally:

Definition 2.4 (Generic Attack on Matrix Profiles). Let

- $U = \{u_1, u_2, \ldots, u_p\}$ represent a set of p users;
- $MP_T = \{MP_{t_1}, MP_{t_2}, \ldots, MP_{t_q}\}$ denotes a public set of q matrix profiles. Each $MP_{t_i} = (MPD_{t_i}, MPI_{t_i})$ is the MP of a time series t_i, and each time series t_i is associated with a single user $u_j \in U$. The user associated with a time series t_i is denoted by $u(t_i)$. A single user in U may have multiple associated time series and corresponding matrix profiles;
- $\mathcal{A}$ represents auxiliary knowledge available to the attacker, which encodes *relations* between users, time series, and other data sources. Formally, $\mathcal{A}$ may contain mappings such as $\mathcal{R} : U \rightarrow E$, where E includes external datasets, partial attributes, fragments of time series, etc.

An attacker seeks to exploit MP_T and $\mathcal{A}$ to define a function $\mathcal{F}_{\mathcal{A}} : MP_T \rightarrow R$, where R represents the outcome domain of the attack. Potential outcomes considered in this article are:

(1) Singling-out: Identifying a specific matrix profile $MP_{t_i} \in MP_T$ corresponding to a single individual $u_j \in U$;
(2) Linkability: Establishing associations between two (or more) matrix profiles MP_{t_i}, MP_{t_k} that correspond to the same user (or group of users);
(3) Inference: Deriving sensitive or personal attributes, such as the shapes or values of all or part of the original time series in T, associated with the matrix profiles about one or more users $u_j \in U$.

This generic attack problem is denoted $\mathcal{MP}_{\mathcal{A},R}$ and depends on the specific definitions of $\mathcal{A}$ and R.

The novelty of this work lies in analyzing the privacy risks posed by MP publication under this attack model. MPs occupy a position between raw data and anonymized representations: they are widely used as feature descriptors, yet implicitly assumed to obfuscate sensitive information. However, this trade-off between utility and privacy has never been rigorously evaluated. Unlike established frameworks for synthetic data (e.g., [22]), MPs lack a dedicated risk assessment methodology. Differential privacy techniques [20] are not directly applicable, and MP-specific protections remain unexplored. Rather than adapting MPs to privacy models, we first aim to understand the extent to which they leak information, especially when auxiliary knowledge is sparse or partial.

The difficulty of the problem lies in several aspects: (i) privacy risks vary by attack type and attacker capability, (ii) MP transformations are nonlinear and complex, impeding intuitive understanding of what is leaked, and (iii) no existing tools quantify these risks. Moreover, in terms of implementation, the attacks highly depend on the auxiliary knowledge $\mathcal{A}$. While an attacker with a lot of auxiliary knowledge may be able to launch certain attacks easily (e.g., an attacker that has full knowledge of a time series with a frequency of 2 minutes may simply interpolate values to predict the values of the 1-minute frequency time series, an example of an inference attack), an attacker with little or no auxiliary knowledge would resort to a reconstruction of the initial time series from the MP.

In the following sections, we instantiate and formalize the three privacy attack types outlined above, then develop and evaluate a reconstruction-based method capable of enabling all three, under limited (or even inexistent) auxiliary knowledge.

3 Modeling Privacy Attacks on Matrix Profile

This section focuses on three foundational principles of privacy risk analysis representing real world attacks: *singling-out*, *linkability* and *inference.* These principles are well established in legal [37, 47], technical [22] and industrial privacy frameworks [29] as they target core aspects of data protection frameworks, such as those enshrined in the GDPR. While non-exhaustive, these principles encapsulate key mechanisms by which attackers can exploit datasets.

We first describe potential auxiliary knowledge of the attacker in the studied scenarios. Then, for each principle, we provide a precise model by instantiating the result R of the generic attack $\mathcal{MP}_{\mathcal{A},R}$ and by providing illustrative examples of the attacker's knowledge $\mathcal{A}$. Algorithms based on reconstruction techniques are described in Section 4, while baseline algorithms, evaluations, and benchmarks are presented in Section 5.

3.1 Auxiliary Knowledge of the Attacker

Beyond access to the MP, attackers may leverage potential auxiliary knowledge to improve the accuracy of their attack, to realistically simulate a wide range of scenarios depending on what the attacker knows about the domain or the individual.

Domain knowledge includes global properties of the dataset that can be inferred from its nature or from public documentation. For example, the attacker may have access to public knowledge such as the generic shape ($\mathcal{A}_{shape}$, e.g., standard ECG waveforms) of specific types of time series. These elements are often easily accessible, e.g., through technical reports or industry standards.

Individual knowledge refers to what the attacker knows about a specific user or target time series. While this may seem like a strong assumption, it is justified by real-world cases where such information is accessible. As detailed in the next sections, practical scenarios of singling-out, linkability and inference attacks may include cases involving knowledge of original time series of given individuals (noted $\mathcal{A}_{ts}$), continuous subsequences ($\mathcal{A}_{subseq}$) or sparse points ($\mathcal{A}_{sparse}$), sum of the values for a given time interval ($\mathcal{A}_{sum}$, e.g., daily energy consumption in a house) or even known ownership of certain MPs ($\mathcal{A}_{link}$).

A summary of the attacker's potential auxiliary knowledge is shown in Table 2, including notations and references to examples.

3.2 Singling-out Attacks

In the context of data privacy, *singling-out* is defined as "*the possibility of isolating some or all records which identify an individual in the dataset*"[1]. Under GDPR Recital 26[2] data that allows singling-out is not considered anonymized but pseudonymized. Several works [13, 22] have formalized this principle. Anonymeter [22] evaluates this principle only for tabular data. In [13], the notion of Predicate Singling Out (PSO) is introduced, where an attacker can uniquely identify an individual from published data. For Matrix Profiles, time series (personal data) are transformed into MPs (published data). While the transformation is not strictly invertible (e.g., all time series modulo a constant share the same MP), an attacker with sufficient but reasonable auxiliary knowledge could still isolate a time series using its MP. This risk depends on the data distribution and available auxiliary knowledge, motivating our definition of singling-out attacks on MPs by taking the auxiliary knowledge into account. For MPs, we define singling-out attacks as follows:

Definition 3.1 (Singling-out Attack on Matrix Profiles). Given a generic attack $\mathcal{MP}_{\mathcal{A},R}$, a singling-out attack assumes that $\mathcal{A}$ includes information about a specific individual $u_j \in U$. Given the set MP_T, the attacker must identify the matrix profile MP_{t_i} such that $u_j = u(t_i)$. The output domain R is $\{1, 2, ..., q\}$, and the attackers' result $r \in R$ is the identifier of the singled-out matrix profile MP_{t_i}. If no matrix profile can be singled out, the attacker outputs $r = \emptyset$ (to avoid false positives).

Auxiliary Knowledge Scenarios. The attacker's success depends on the type of auxiliary knowledge $\mathcal{A}$, which takes various forms:

- $\mathcal{A}_{ts}$: The attacker knows a time series t_{known};
- $\mathcal{A}_{subseq}$: The attacker knows a partial subsequence $s_{i,\ell}^{t_{known}}$ of length ℓ (see example 3.1 below);
- $\mathcal{A}_{sparse}$: The attacker knows only sparse points $\{s_{i,1}^{t_{known}}\}$.

For example, a singling-out attack based on a known subsequence is as follows (for additional examples, see Appendix A):

Example 3.1 (Singling-out from a known subsequence -$\mathcal{A}_{subseq}$-)**.**

> Alice underwent an ECG at a hospital for diagnosis purposes. The hospital published *MP*s, including Alice's, with associated signal classified as either benign or malignant. These MPs were published under the assumption that the identities of the patients were anonymized. However, due to a data leakage, Eve, an attacker, gained access to a part of an original ECG of Alice. Using this auxiliary information, Eve aims to single out Alice's *MP* to determine whether it corresponds to a malignant anomaly.

3.3 Linkability Attacks

In data privacy, *linkability* refers to "*the ability to link, at least, two records concerning the same data subject or a group of data subjects (either in the same database or in two different databases)*"[1]. Privacy risks emerge when attackers associate multiple matrix profiles

[1] Definition from the EU Data Protection Working Party and by the French CNIL.

[2] Excerpt from GDPR, Recital 26: "*(...) To determine whether a natural person is identifiable, account should be taken of all the means reasonably likely to be used, such as singling-out (...)*".

Annotation	Description	Knowledge Type	Justification
$\mathcal{A}_{ts}$	A complete time series in T	Individual	Sections 3.2-3.4, Appendix A and B.
$\mathcal{A}_{subseq}$	A continuous subsequence from a time series in T	Individual	Sections 3.2, 4.3 and Appendix A (A.2).
$\mathcal{A}_{sparse}$	Sparse points in a time series in T	Individual	Sections 3.2, 3.4, 4.3 and Appendix A.
$\mathcal{A}_{sum}$	Sum of the values in a time series in T	Individual	Sections 3.4 and 4.3.
$\mathcal{A}_{link}$	A set of one or more MPs of an individual in MP_T	Individual	Section 3.3 and Appendix B.
$\mathcal{A}_{shape}$	Public knowledge of the time series shape in T	Domain	Sections 3.4, 4.3 and Appendix C.

Table 2: Auxiliary Knowledge Summary

corresponding to the same user across datasets or contexts. We define linkability attacks on MPs as follows:

Definition 3.2 (Linkability Attack on Matrix Profiles). Given the generic attack $\mathcal{MP}_{\mathcal{A},R}$, a linkability attack assumes that $\mathcal{A}$ includes auxiliary information about specific users in U or relationships between them, and at least two MPs in MP_T correspond to the same user $u_j \in U$ (or to the same group of users). The output domain is a set of k integers in $[1; q]$, where $R = \{1, 2, ..., q\}^k$. The attacker must produce $r = \{r_1, r_2, \ldots, r_k\}$, a set of identifiers for matrix profiles $MP_{t_{r_1}}, MP_{t_{r_2}}, \ldots, MP_{t_{r_k}} \in MP_T$ that are linked to the same user $u_j \in U$ (or the same group of users). If no such profiles can be linked, the attacker outputs $r = \emptyset$.

Auxiliary Knowledge Scenarios. The success of linkability attacks depends on the auxiliary knowledge $\mathcal{A}$ available to the attacker. This may include the knowledge of one or more MPs corresponding to a target individual or group ($\mathcal{A}_{link}$), which may potentially be computed from the knowledge of a complete time series ($\mathcal{A}_{ts}$)[3].

An example of a linkability attack scenario is as follows:

Example 3.2 (Linking matrix profiles from ECGs -$\mathcal{A}_{link}$-).

> Two hospitals, H_1 and H_2, share pseudonymized MPs of their patients' ECGs for research purposes. Alice, a patient at both hospitals, has multiple ECGs recorded at each. All these MPs are published in MP_T along with meta-data specific to each hospital. Eve, an attacker, gains knowledge of one or more of Alice's MPs from H_1 (e.g. due to data leakage) and links them to a MP from H_2, which will grant her knowledge of Alice's additional metadata.

The algorithm detailing the implementation of the baseline linkability attack is given in Algorithm 2, Appendix B. It takes as input the attacker's knowledge of one or several MPs ($\mathcal{A}_{link}$) and outputs the closest MP in MP_T that is not already part of $\mathcal{A}_{link}$.

3.4 Inference Attacks

The regulator defines *Inference* as "*the possibility to deduce, with significant probability, the value of an attribute from the values of a set of other attributes*"[1]. In the context of MPs, we consider each value of the original time series t of length n to be an attribute, thus we consider the inference of n attributes.

Definition 3.3 (Inference Attack on Matrix Profiles). Given the generic attack $\mathcal{MP}_{\mathcal{A},R}$, an inference attack assumes that $\mathcal{A}$ *can include (or not)* prior knowledge either about the global characteristics of the data, or about specific target users $u_k \in U$.

Given the set MP_T, the attacker must infer the values t_i associated with each matrix profile $MP_{t_i} \in MP_T$, leveraging the auxiliary knowledge $\mathcal{A}$. The output domain R consists of all possible time series of length n, noted $\mathcal{T}_n$. Specifically, the attacker must produce as a result $r = (r_1, r_2, \ldots, r_n)$, a time series inferred for each user $u(t_i)$. If the attacker cannot infer a specific value at a given timestamp for a given matrix profile, the corresponding entry in r is left empty (e.g., $r_1 = \emptyset$) or if no inferences are possible then $r = \emptyset$.

Auxiliary Knowledge Scenarios. The attacker's ability to infer personal attributes depends on the auxiliary knowledge $\mathcal{A}$, which may include:

- *Some points of the original time series ($\mathcal{A}_{subseq}$ or $\mathcal{A}_{sparse}$)*: For instance by observing a user or obtaining the time series values during a limited duration or by possessing a smart meter read with a lower frequency.
- *Aggregate points of the original time series* ($\mathcal{A}_{sum}$): For instance aggregated points of a given time series e.g., 1 hour aggregates of a smart meter reading instead of a point per minute, or the daily total energy consumption (sum of daily readings) deduced from an electricity bill.
- *General information on the original time series* ($\mathcal{A}_{shape}$): Knowledge of the type of time series allowing to know some characteristics of the time series such as its general shape, amplitude, frequency, etc.

An illustrative scenario is as follows:

Example 3.3 (Inference attack from smart meter readings -$\mathcal{A}_{sum}$-).

> An electricity company suffers a data leak of the 1h frequency time series of their clients. The electricity company also shared the *MPs* of the 1 minute frequency time series with a cloud provider for data analysis with a view to lessen power peaks. A malicious employee of the cloud provider who has retrieved the leaked dataset aims to reconstruct these leaked time series with a 1 minute frequency.

Another possible inference scenario is to infer sensitive label such as age or gender, which is discussed in Appendix D.

4 Exploiting Reconstruction for Privacy Attacks

This section defines the reconstruction problem, models it as a Constraint Satisfaction Problem (CSP), provides a high-level description of the reconstruction algorithm, and demonstrates how this technique can be leveraged to enhance the effectiveness of the three privacy attacks introduced earlier.

[3] See Example B.1 in Appendix B.

4.1 Reconstruction Definition

A reconstruction seeks to recover partial or complete time series data from a published matrix profile. Formally:

Definition 4.1 (Reconstruction from Matrix Profile). Given a $MP_t \in MP_T$ of a user $u_i \in U$. The attacker's objective is to reconstruct the time series t corresponding to the MP_t. The output domain $R = (\mathcal{D})^n$ is a vector of values of length n. The attack outputs the reconstructed vector $r \in R$ or $r = \emptyset$ if no value can be reconstructed.

4.2 Modeling Reconstruction as CSP

A matrix profile, MP_t, computed with subsequence length m and distance measure $Dist$, can be interpreted as two vectors: MPD (Matrix Profile Distances) the distance values, and MPI (Matrix Profile Indices), the indices of nearest neighbors. These vectors form constraints on distances and indices:

- ***Equality constraints*:** The distance between any subsequence s_i of t (starting at index i) and its nearest neighbor s_j (with $j = MPI[i]$) must be equal to the value $MPD[i]$: $\forall i \in (0, n-m), \mathrm{Dist}(s_i, s_{MPI[i]}) = MPD[i]$.
- ***Inequality constraints*:** For each subsequence s_i, the distance to all other subsequences s_j $(j \neq i)$ must be greater than or equal to $MPD[i]$: $\forall (i,j) \in (0, n-m)^2, |i-j| > m, \mathrm{Dist}(s_i, s_j) \geq MPD[i]$.

While MP is widely used and developed for data mining applications, few prior works have explored the use of noisy or partial MPs as a privacy-preserving measure. As such, the exploration of obfuscated or noisy MPs falls outside the scope of this paper and is left as future work (see Section 7). Also, the subsequence length and the distance metric used in the MP computing are usually publicly known, as in practice, the metric used is one of a few classical ones (e.g., (z-normalized) Euclidean, see [16, 36, 66]), and is usually deducible from the kind of data treated. Hence, the attacker assumes access to the MP components (MPD, MPI, m and $Dist$) and public knowledge of the data type (e.g., ECG, energy consumption). Additional information (that can be easily computed) used in our approach is:

- Upper and lower bounds of time series values based on data characteristics;
- The time series length, inferred as $n = |MPI| + m - 1$;
- Normalization of data (e.g., scaled to [0, 1], see Appendix E).

Reconstructing t involves solving a CSP to find a sequence of values $\hat{t}$ in domain $\mathcal{D}^n$, subject to equality and inequality constraints.

Definition 4.2 (CSP for Reconstruction). The CSP seeks to find (ideally a unique) sequence $\hat{t} = [\hat{t}[0], \ldots, \hat{t}[n-1]]$, where each subsequence $\hat{s}_i = [\hat{t}[i], \ldots, \hat{t}[i+m-1]]$ satisfies the following equality and inequality constraints:

$$\begin{cases} \forall i \in (0, n-m), \mathrm{Dist}(s_i, s_{MPI[i]}) = MPD[i] \\ \forall (i,j) \in (0, n-m)^2, |i-j| > m, \mathrm{Dist}(s_i, s_j) \geq MPD[i] \end{cases}$$

Optimized Reconstruction. CSP are potentially exponentially long to solve, thus we adopt an optimization approach : instead of searching for an exact solution, we seek to minimize the following objective function (note that the original time series verifies $O(t) = 0$), while removing the *equality* constraints:

$$O(\hat{t}) = \sum_{i=0}^{n-m} (\mathrm{Dist}(s_i, s_{MPI_{\hat{t}}[i]}) - MPD_{\hat{t}}[i])^2 \tag{1}$$

This presents the advantage of not needing to wait for the complete computation of the CSP solution (thus providing *anytime* solutions), however it may lead to the presence of violated constraints (i.e., incorrect solutions) in any case where $O(\hat{t}) > 0$. While this approach does provide *anytime* solutions, experimental convergence remains slow. Thus, inspired by [16], we use a modified objective function C, which incorporates penalties for constraint violations using the popular Rectified Linear Unit (ReLU) activation function, in order to also remove the *inequality* constraints:

$$C(\hat{t}) = \sum_{i=0}^{n-m} \sum_{j=0}^{n-m} ReLU(MPD_{\hat{t}}[i] - Dist(s_i, s_j)) \tag{2}$$

where ReLU penalizes violations of the inequality constraints. The final combined objective function that we use is thus:

$$O_{opt}(\hat{t}) = \alpha * O(\hat{t}) + \beta * C(\hat{t}) \tag{3}$$

with α and β as weighting parameters (default: $\alpha = \beta = 1$), and $Dist = d_E$ or $Dist = d_Z$, as discussed in Section 5.1 and in Appendix G.

This formulation embeds the constraints into the objective function, which improves experimental efficiency as we no longer try to satisfy them exactly. In practice, reconstructing a time series of length 200 shows a threefold speed-up compared to using the O objective function (See Appendix H).

4.3 Incorporating Auxiliary Knowledge

Auxiliary knowledge can be incorporated as constraints in our reconstruction model, to improve the accuracy of the attack.

Both individual knowledge like $\mathcal{A}_{subseq}$ and $\mathcal{A}_{sparse}$ and domain knowledge (e.g., $\mathcal{A}_{shape}$) can be expressed as hard constraints. For $\mathcal{A}_{subseq}$ and $\mathcal{A}_{sparse}$ (see Section 3.1), let I be the set of indices for which the attacker knows the values. Thus:

$$\forall i \in I, \hat{t}[i] = t_{known}[i] \tag{4}$$

We model $\mathcal{A}_{shape}$ for a function t of period τ (allowing for a relative error η) and/or amplitude $[min, max]$ by:

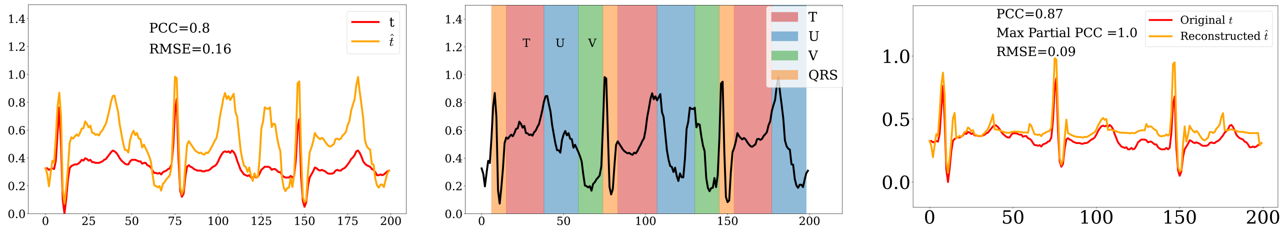
$$\forall i \in [1, n-\tau], (1-\eta).\hat{t}[i] \leq \hat{t}[i+\tau] \leq (1+\eta).\hat{t}[i] \tag{5}$$

$$\forall i \in [1, n], min \leq \hat{t}[i] \leq max \tag{6}$$

Overall modeling. Our reconstruction attack model thus consists of an objective function $O_{opt}(\hat{t})$ (i.e., equation 3) which encodes the soft constraints from the MP and is minimized to approximate the target profile while maintaining performance. When relevant, attacker auxiliary knowledge is incorporated as additional hard constraints (e.g., equations 5 and 6). Any knowledge that cannot be formulated as constraints will be incorporated during the post-processing stage (see Section 4.4).

4.4 Reconstruction Algorithm

The reconstruction attack algorithm, which derives from this model, is as follows (see Appendix C for the pseudo-code):



(a) Before post-processing.

(b) Zone identification.

(c) After post-processing.

Figure 2: Post-processing for ECG reconstruction improves PCC from 0.8 to 0.87 and decreases RMSE from 0.16 to 0.09.

(1) *Initialization*: Given $MP_t \in MP_T$ and m, deduce the length n of the target time series. Deduce MP_t calculated from normalized values (see equation 10 in Appendix E). Then generate a set $\hat{T} = \{\hat{t}_i\}$ of candidate time series of length n. Each $\hat{t}_i$ is built by generating n random values in this normalized range [0,1];

(2) *Objective function evaluation*: For each candidate time series $\hat{t}_i$, evaluate its loss versus MP_t, using objective function O_{opt};

(3) *Candidate selection*: Select the best-performing candidates with the lowest loss for further optimization (in practice, we keep as many candidates as the parallel rate p_{rate});

(4) *CSP solving* (for computational efficiency, this step is performed in parallel on p_{rate} CPUs): For each of the p_{rate} best-performing candidates $\{\hat{t}_i^{best}\}$, run the CSP solver to iteratively refine $\hat{t}_i^{best}$ to minimize O_{opt};

(5) *Post-processing (optional)*: Domain-specific knowledge can be used to further refine each reconstructed time series $\hat{t}_i^{best}$. For example, for ECG data, the attacker can apply standard ratios to adjust waveforms (e.g., QRS, T, U, V patterns);

(6) *Final Output*: Output $r \in \{\hat{t}_i^{best}\}$, by choosing the time series whose MP has the lowest O_{opt} loss value.

Post-Processing (case of ECG) Prior knowledge that can be naturally expressed using variables can be encoded as hard constraints; otherwise, it can be incorporated during the post-processing phase to improve the effectiveness of the reconstruction. In the case of ECG data, this involves leveraging public knowledge of ECG patterns to correct distortions in amplitude, timing, or waveform proportions introduced during reconstruction, as the fundamental QRS patterns (most prominent peaks in the ECG, see Figure 2b and Appendix C for more details) are typically preserved. We propose the following Post-Processing, for ECG Data, as step (5) of reconstruction algorithm above (see Appendix C for detailed description):

(1) *QRS Pattern Identification*: Identify QRS complexes using the distance profile;

(2) *Segmentation of Adjacent Waveforms*: Divide the intervals between successive QRS patterns into three primary segments corresponding to U, T, and V waveforms, based on standard proportional spacing in a typical ECG signal;

(3) *Normalization and Adjustment*: Normalize the amplitude of each segment using conventional ratios relative to the QRS complex, to align with standard clinical ECG proportions.

Figure 2a illustrates a reconstructed ECG signal before post-processing. While the general shape of the signal is captured, amplitude ratios between waveforms are inconsistent, and some distortions are present. Figure 2b shows the segmentation of the ECG signal into QRS, U, T, and V zones, based on proportional spacing. Finally, Figure 2c demonstrates the corrected signal after applying amplitude normalization and proportional adjustments, using public information for a classic ECG.

The application of post-processing techniques significantly improves the quality of the reconstructed signal. In the example presented, which is representative of general results of our approach, the Pearson Correlation Coefficient (PCC [54], see Section 5.1) between the reconstructed ECG and the original signals increases from 0.8 to 0.87 after post-processing. On the contrary, the Root Mean Square Error (RMSE) decreases from 0.16 to 0.08, thus ensuring closer alignment with the raw data.

4.5 Privacy Attacks using Reconstruction

We now show how the reconstruction algorithm developed in the previous section can be used to instantiate concrete privacy attacks, even in the absence of strong auxiliary knowledge. For each of the three categories of privacy risk identified in Section 3 we describe how the attack is implemented using the reconstructed time series, and how its effectiveness can be evaluated.

Inference attack. In the inference setting, the attacker seeks to extract sensitive attributes or behavioral features directly from the reconstructed signal.

Algorithm. The attacker takes as input the matrix profile MP_t of a time series t and produces a reconstruction $\hat{t}$ using the method from Section 4.4. This reconstructed signal is then directly interpreted as the inferred output. No auxiliary knowledge is required. Depending on the context, $\hat{t}$ may itself be the sensitive information (e.g., biomedical signal), or serve as a proxy to infer higher-level properties (e.g., activity, condition).

Evaluation Metrics. We measure inference quality using standard similarity metrics between $\hat{t}$ and t, when the ground truth is available: Root Mean Squared Error (RMSE), Pearson Correlation Coefficient (PCC), and Partial PCC on regions of interest. To assess the quality of inference, we also adopt a conformal-based confidence score, inspired by conformal prediction [34, 44], that reflects the stability of the reconstruction with respect to input variations.

Singling-out attack. Here, the adversary aims to identify which matrix profile in a public set corresponds to a known partial time series from a target user.

Algorithm. Given a collection MP_T of matrix profiles, the attacker first reconstructs a set of estimated time series $\hat{T}$. Provided with a partial original time series t (auxiliary knowledge), they compute the distance between t and each $\hat{t}' \in \hat{T}$. The matrix profile whose reconstruction is closest to t is selected as the predicted match.

Evaluation Metrics. We report the singling-out success rate, defined as the proportion of individuals $u_i \in U$ for whom the correct matrix profile is identified. This is evaluated in a top-1 identification setting over the candidate set, and measures how reconstruction reduces anonymity.

Linkability attack. In this scenario, the attacker attempts to associate multiple MPs to the same user, even when no identity information is provided.

Algorithm. The attacker begins by reconstructing $\hat{T}$ from MP_T, and selects a reference reconstruction $\hat{t}$ known to belong to a given user. Then, they compute similarity scores between $\hat{t}$ and all other $\hat{t}' \in \hat{T}$. Any series sufficiently close to $\hat{t}$ is linked to the same user.

Evaluation Metrics. We measure linkability success as the proportion of users for whom at least one additional MP is correctly linked to the same identity. This task is analogous to recall in a one-vs-rest matching problem, and reflects the ability of reconstruction to compromise unlinkability.

5 Experiments

5.1 Experimental Setting

Datasets. We evaluate our methods using datasets already employed in privacy studies involving matrix profiles [16, 28, 49, 68]:

- **ECG**: The ElectroCardioGram Database [23] contains seven long-duration recordings (14-22 hours, sampled at 128 Hz) with annotations and demographic labels (including age, sex). We extracted the first 4000 points from each record and segmented them into 20 samples of 200 points each.
- **IHEPC**: The Individual Household Electricity Power Consumption Dataset [25], comprises electricity usage recorded every minute for four years. We selected randomly data from 140 days, sampling 200 data points (10:00-13:20).
- **H-MOG**: The dataset [65] provides smartphone accelerometer data, we use the X-axis, sampled at 80-length time series.

Platforms. For the baseline attacks used for comparison with our reconstruction based attacks, evaluations were conducted on an HP laptop equipped with a 13th Gen Intel® Core™ i7-1365U processor and 30 GB of RAM. The reconstruction based attacks are executed on a server equipped with an Intel(R) Xeon(R) E-2276G CPU (3.80 GHz, 12 cores) and 62 GB RAM.

CSP Solvers. OR-Tools [48] was used for preliminary analysis involving an integer programming formulation of the CSP (see Appendix L). IPOPT [59] was used for real-valued optimization, with hyperparameters and coefficients for objective functions carefully tuned. While other solvers were tested, they provided less efficient results (See Appendix G-F-I for details).

Metrics. We list here the metrics that quantify the effectiveness of the reconstruction and of the inference attack. For singling-out and linkability attacks, we use success rate to evaluate.

- **Root Mean Squared Error (*RMSE*)** [27]: measures the average deviation between the original time series t and its reconstruction $\hat{t}$, scaled between 0 and 1. Lower *RMSE* values indicate better reconstruction.
- **Pearson Correlation Coefficient (*PCC*)** [52, 54]: quantifies the linear relationship between t and $\hat{t}$, or their respective matrix profile distances (MPD_t and $MPD_{\hat{t}}$), with values ranging from 0 to 1. Higher *PCC* values indicate stronger linear similarity. As in prior work (see e.g., [52]), reconstructed time series with a *PCC* above 0.7 are considered to have shapes closely matching the original.
- **Partial-PCC** and **Partial-RMSE**: These metrics evaluate subsequences of reconstructed time series. For time series of size n, Partial-*PCC* (or Partial-*RMSE*) is the maximum *PCC* (or the minimum RMSE) value computed for all subsequences of length $2 * m$ in $\hat{t}$. For example, a Partial-*PCC* of 0.7 indicates that at least one subsequence of size $2 * m$ has a $PCC \geq 0.7$ compared to its counterpart in t.

5.2 Reconstruction Efficiency

We evaluate the reconstruction technique introduced in Section 4.4. The experimental process comprises four steps:

(1) Select 140 random normalized time series $\{t\} = T$ from each dataset.
(2) Compute their matrix profiles $\{MP_t\} = MP_T$ (with $MP_t = \{MPD_t, MPI_t\}$)
(3) Reconstruct each time series $\hat{t}$ using the proposed reconstruction attack. We note $\hat{T} = \{\hat{t}\}$
(4) Evaluate the fidelity of reconstructed series ($\hat{T}$ vs. T) and their matrix profiles ($MP_{\hat{T}}$ vs. MP_T) using *RMSE*, *PCC* and Partial-*PCC* and -*RMSE* metrics.

Time series are obtained from the ECG, IHEPC and H-MOG datasets, and are normalized to the range $[0, 1]$ to enable cross-dataset comparison[4]. Matrix profiles are computed using Euclidean distance (d_E) and and Z-normalized Euclidean distance (d_Z) distance measures. The size of time series n and subsequence length m are aligned with configurations used in prior studies (see Table 3).

Reconstruction results without Auxiliary Knowledge.

- *Quantitative Results*: Table 4 summarizes the quality of reconstructed time series $\hat{t}$ compared to original time series t, and $MP_{\hat{t}}$ compared to MP_t, using PCC (and Partial-PCC for subsequences) to evaluate shape similarity, and RMSE (and Partial-RMSE) to assess value accuracy.
- *Results Distributions and Cross-Dataset Comparisons*: Figures 5 and 6 show the distribution of different metrics (i.e., PCC and RMSE values) for both ECG and IHEPC datasets with the two

[4] Normalization of input time series has no impact on our reconstruction algorithm, calculating the normalized MP from the original MP is explained in Appendix E.

Parameter	H-MOG	ECG	IHEPC
Number of time series	140	$140^{(**)}$	140
Length for t (n)	$80^{(*)}$	$200^{(**)}$	$200^{(***)}$
Subseq. length for MP_t (m)	$5^{(*)}$	10	$10^{(***)}$
Max Solver time (in min.)	<5	~60	~60

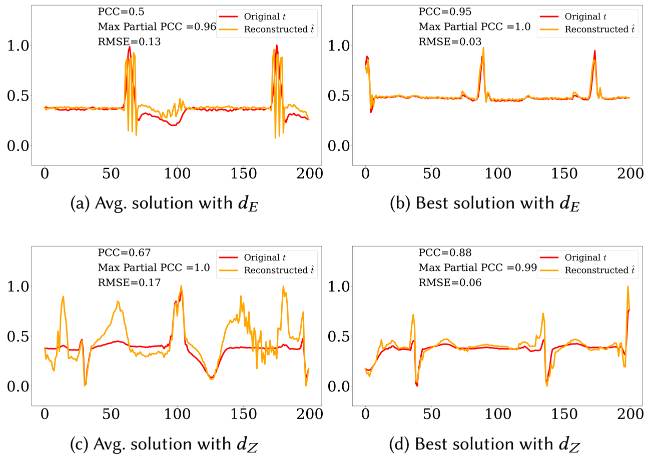
$^{(*)}$ same as [28], is 5 sec. movements
$^{(**)}$ $n = 200$ represents $\tilde{3}$ heart beats, 140 time series to align with [16]
$^{(***)}$ $n = 200$ is 3 hours 20 min. energy consumption, $m = 10$ is 10 min.

Table 3: Parameters for the reconstruction experiments.

distance measures (i.e., d_E and d_Z) to facilitate comparisons across different datasets and MP parameters.
- *Reconstruction Examples*: Representative examples of reconstructed time series versus their original counterparts are shown in Figures 3 (ECG dataset) and 4 (IHEPC dataset), with PCC and RMSE scores to illustrate the reconstruction quality.

Takeaway. The following main conclusions can be drawn from these results about the reconstruction process:

- *Reconstruction quality in shape and values*: Table 4 shows that the average PCC is above 0.71 and the average $RMSE$ is below 0.25 for all the time series considered in the experiments (see "TOTAL" in the last line of Table 4) with both d_E or d_Z distance metrics. Nearly 60% of the complete reconstructed time series have $PCC > 0.7$, and on average 37.1% have $RMSE \leq 0.1$. Partial-PCC is above 0.7 for nearly 100%, meaning all the reconstructed time series have at least one subsequence of significant size with well-preserved shape. Similarly, Partial-RMSE is below 0.1 for almost 90% of the time series encountered in the experiments, indicating that the values are also very well-preserved. In general, time series with constant or perfectly repeated patterns (though this is an unrealistic case) are more resistant to reconstruction, as they admit a larger number of possible preimages that correspond to the same MP.
- *Significant post-processing improvement*: Values marked with a star in Table 4 were obtained after post-processing. While PCC remains mostly unchanged after post-processing (as post-processing involves vertical shifts that do not alter the shape), the $RMSE$ is significantly improved, reducing the average error by almost 30%. This improvement doubles the proportion of reconstructed time series with $RMSE \leq 0.1$.
- *Preservation of Matrix Profiles accuracy*: Table 7 shows that the matrix profiles of the reconstructed time series are very close to their counterparts in MP_T. The average accuracy for indices in MPI is > 0.86 (i.e., 86% of indices in $MPI_{\hat{t}}$ are identical to those in MPI_t), and the average RMSE < 0.1 and PCC > 0.97, indicating that the values and shapes of the reconstructed MPs are very similar to the originals.
- *Limited impact of distance metrics*: Using z-normalized Euclidean Distance rather than simply Euclidean Distance does not provide much significant advantage in average in protection against reconstruction attacks, contrary to what was expected in previous works (see e.g., [28, 68]). Results are similar with both distances for H-MOG, slightly better with d_E for ECG, and slightly better with d_Z for IHEPC (Table 4). Importantly, for the ECG dataset, although d_Z yields a lower average PCC for the total time series, it reconstructs QRS patterns more effectively (see Figure 3), raising significant privacy concerns due to the sensitivity of QRS data in ECG. All metrics show good preservation of MPs (Table 7), confirming the success of the reconstruction. This finding contradicts certain assumptions from previous work [16, 68] that d_Z increases reconstruction difficulty due to its non-linearity.



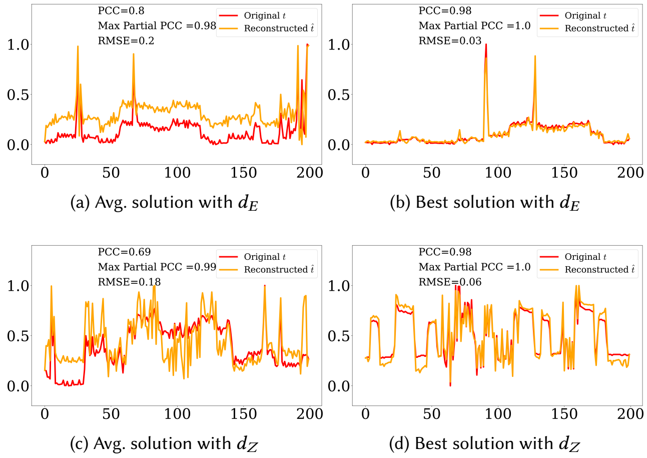
(a) Avg. solution with d_E

(b) Best solution with d_E

(c) Avg. solution with d_Z

(d) Best solution with d_Z

Figure 3: Avg/best quality reconstructions examples (ECG).



(a) Avg. solution with d_E

(b) Best solution with d_E

(c) Avg. solution with d_Z

(d) Best solution with d_Z

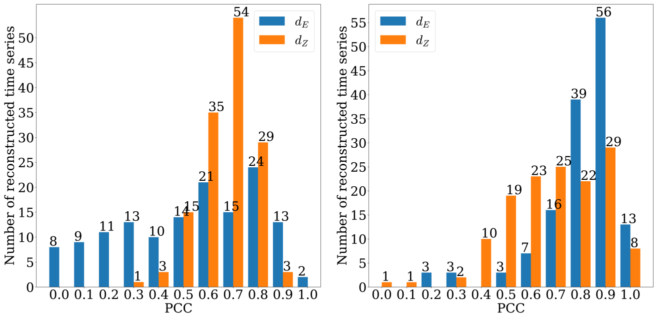
Figure 4: Avg/best quality reconstructions examples (IHEPC).

Reconstruction results with Auxiliary Knowledge. The results presented so far assume that the attacker only has access to the MP. We now evaluate reconstruction performance when the attacker possesses auxiliary knowledge, as discussed in Section 4.3. The selected knowledge types and parameters are inspired by real-world use cases and include: (i) fixing the first 30 points (i.e., knowing the consumption pattern over the first 30 minutes), (ii) downsampling

Dataset	Dist.	**PCC** (Avg - ≥ 0.7 - Max)	**Part-*PCC*** (≥ 0.7)	**RMSE** (Avg- ≤ 0.1 - Min)	**Part-*RMSE*** (≤ 0.1)
H-MOG	d_E	0.83 - 84% - 0.99	100%	0.16 - 58% - 0.03	90%
	d_Z	0.78 - 79% - 0.97	100%	0.23 - 36% - 0.07	80%
ECG	d_E	0.53 - 39% - 0.98	100%	0.15 - 57% - 0.03	99%
	d_Z	0.68 - 61% - 0.86	100%	0.25 - 9% - 0.10	93%
	d_Z*	0.67* - 59%* - 0.92*	100%*	0.18* - 19%* - 0.06*	94%*
IHEPC	d_E	0.81 - 89% - 0.99	100%	0.19 - 49% - 0.03	87%
	d_Z	0.69 - 60% - 0.98	99%	0.34 - 14% - 0.06	75%
TOTAL	d_E	0.72 - 70.7% - 0.99	100%	0.17 - 54.7% - 0.03	92%
	d_Z	0.71 - 64.8% - 0.98	99.8%	0.25 - 19.5% - 0.07	85.5%

* The results on this line were obtained using post-processing.

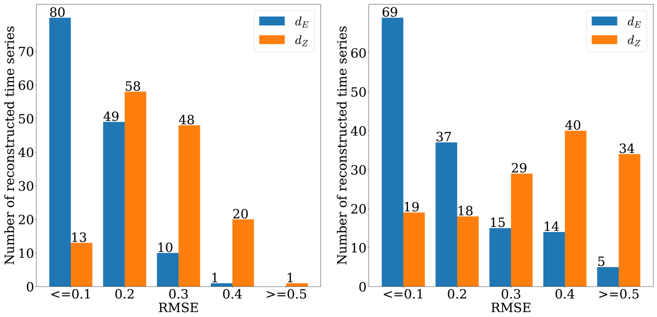
Table 4: Reconstruction fidelity: $\hat{t}$ vs. t.



(a) ECG.

(b) Energy consumption.

Figure 5: *PCC* distribution for $\hat{t}$ after reconstruction.



(a) ECG.

(b) Energy consumption.

Figure 6: *RMSE* distribution for $\hat{t}$ after reconstruction.

at a rate of 1 point per 30 minutes, and (iii) knowing the mean value (i.e., the average consumption per minute).

Table 5 summarizes the results. Both shape and value reconstruction improve with all types of auxiliary knowledge, except for shape reconstruction from *MP* calculated with d_E using the mean value. An example of reconstruction under different auxiliary knowledge conditions is shown in Appendix M. Notably, we observe that real-value reconstruction accuracy can improve significantly with even limited auxiliary information, such as a mean value or a low-resolution sampling (e.g., 3% of the points). This is because such constraints effectively reduce the size of the output domain.

	Distance measure	
Auxiliary Knowledge	d_E	d_Z
$\emptyset$ (Baseline)	0.83 (0.19)	0.69 (0.34)
$\mathcal{A}_{sparse}$ (1 point/30 min.)	0.90 (0.12)	0.76 (0.17)
$\mathcal{A}_{subseq}$ (first 10% time series)	0.89 (0.09)	0.77 (0.21)
$\mathcal{A}_{sum}$	0.79 (0.12)	0.75 (0.14)

Table 5: Average PCC (and average RMSE in parentheses) between reconstructed and original time series under different attacker auxiliary knowledge and distance measures, using the energy consumption dataset.

5.3 Privacy Attacks Baselines and Evaluation

In this section, we first define baselines for the three attacks. We then present the evaluations with reconstruction techniques and compare them with these baselines. Table 6 summarizes the results of the attacks using reconstruction, compared to baseline attacks (see Section 5.1), with high effectiveness highlighted in red. For detailed results, please refer to Appendices A and B.

Baseline attacks. We consider several simple baselines that do not use reconstruction, for comparison purposes (see repository git and Appendices A-B for details).

- **Singling-out baseline**: This baseline attack exploits auxiliary knowledge and the known function and parameters of MP. If the attacker knows an entire time series t_{known}, they can compute its matrix profile and directly match it to an identifier in MP_T. With only a subsequence (partial time series), the attacker compares the matrix profile of the subsequence against the published MP_T, looking for alignment in *MPD* (distance) and *MPI* (index) values (an example is given in Appendix A). If the auxiliary information is sparse or insufficient, we compute the missing values using linear interpolation to form an estimated time series t'_{known}. Then the published profile with a most similar *MPD* (i.e., by measuring the correlations) is returned.

- **Linkability baseline**: The linkability baseline attack considers that the attacker has complete knowledge of y different MPs of a target user ($\mathcal{A}_{link}$). In this baseline attack, the y known MPs are taken from MP_T (as each user in the dataset has 10 MPs, we test values of $y < 10$). The goal of the attack is to find *another* MP in MP_T that belongs to the same user. This baseline attack simply computes the distance between the known MPs and all the MPs of MP_T (i.e., we take the MPD component of the MPs, and view them here as vectors of length n) and chooses the one that is the closest to a MPD in $\mathcal{A}_{link}$. If this chosen MP belongs to the same user, the attack is deemed successful. Any distance between vectors can be used in the baseline attack algorithm (e.g., the Euclidean Distance), however in our experiments we chose to use the *minimal value of the MPD between these vectors*, as it is more general, and allows for comparing smaller subsequences. It is tailored to the characteristics of the application, such as ECG, using a specific length (e.g., $\ell = 100$ for the results in Table 6) of the vectors rather than only being able to compare the whole vectors (of length n). Note that if we chose $\ell = n$ this is equivalent to simply computing their Euclidean Distance. The baseline algorithm is detailed in Appendix B (Algorithm 2).
- **Inference baseline:** Inferring original time series from their matrix profiles is a difficult task. Traditional basic methods such as interpolation [50], moving average smoothing [9], and nearest-neighbor search [32] cannot be directly applied, as the matrix profile represents a non-linear transformation of the time series. Additionally, inference capacity cannot be fully captured by RMSE or PCC alone. Indeed, given a set of N time series T and their corresponding MPs, a low RMSE or a high PCC between a reconstructed time series $\hat{t}$ and its true antecedent (the original time series t) does not guarantee successful inference by an adversary. We adapt to our context a baseline metric inspired by conformal prediction: for each $\hat{t}$, we rank the time series in T from closest to furthest (using PCC or RMSE) and record the rank of the true time series. We then report the percentage of cases where the true series is ranked top-1. A value of 100% indicates a highly successful attack, where each $\hat{t}$ is closest to its true source t.

Singling-out. We evaluate two types of auxiliary knowledge: $\mathcal{A}_{subseq}$ and $\mathcal{A}_{sparse}$. For $\mathcal{A}_{sparse}$ (see first line of Singling-out in Table 6), the baseline attack fails with only one point and remains largely ineffective at low sampling frequencies (e.g., one point every four), achieving at most a 24% success rate. The baseline achieves its maximum at 72% with one in two points. In contrast, reconstruction achieves 80% with every eighth point, then sharply increases to 92% success when given every fourth point, and achieves 95% with one in two points. For $\mathcal{A}_{subseq}$ (see the second line of Singling-out), our experiments show that reconstruction can successfully single out 57% of matrix profiles using just m known points (5% of the time series), compared to 0% with the baseline attack. With $2m$ and $3m$ known points, reconstruction achieves 80% and 82% success rates, respectively, while the baseline only reaches 21% and 60%. When more than $3m$ points are known (see the last two lines of Singling-out), both baseline and reconstruction attacks are highly effective (over 96% accuracy, up to 100% for reconstruction).

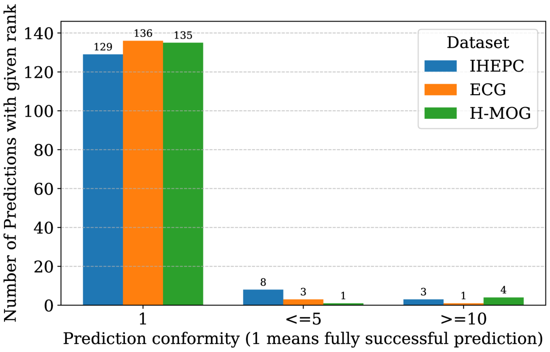


Figure 7: Ground truth prediction ranking distribution.

Linkability. The successful rate of a linkability attack by considering a weak attacker is low, with success rate of 67% (see the first line of Linkablity in Table 6). Success rate increases with the attacker's knowledge, with 2 to 3 known links, it achieves around 70% and with 4 to 6 known links it reaches 76%. However, with more prior knowledge (more than 6), it stays within a narrow band of 67% to 76%. More detailed results are provided in Appendix B. This result is slightly improved, with an increased success rate of 77% when using the reconstructed time series, by considering the same weak attacker (first line of last column for Linkability). When the attacker knows a pair of MPs, the success rate increases to 94.3%, and remains high (between 85% and 95.7%) with more prior knowledge, indicating a high risk of linkability.

Inference. The results are shown in Figure 7. Over 92% of predictions (129 out of 140) correctly identify the true time series as the top-1 prediction, signifying fully successful prediction, and over 97% place the true time series within the top-5 across all datasets. The outliers (outside the top-10) in the energy consumption dataset correspond to days with near-zero consumption (i.e., when no one is home), resulting in matrix profiles heavily affected by noise. For the H-MOG dataset, outliers arise due to insufficient resolution, as the algorithm was run for a shorter duration, as detailed in Table 3. Results of another type of inference attack (i.e., label inference) are proposed and discussed in Appendix D.

5.4 Conclusion and Limitations

Experimental results demonstrate that sharing MPs does not ensure privacy. Firstly, MP does not preserve privacy against singling-out, linkability and inference attacks. Secondly, it is not secure against reconstruction attack, considering an attacker with moderate power and limited prior knowledge.

There are some limitations to our evaluations. Firstly, we evaluate our attacks on three time series datasets to validate their effectiveness. Additional datasets could be explored to draw broader conclusions about the attack utility based on high-level dataset characteristics, such as value distribution.

Secondly, due to computational constraints, we used six "good enough" starting points to solve the constraint systems and yielding six candidate solutions per reconstruction. This limits solution space exploration, but more sophisticated mechanisms could be

Attack	**Knowledge**	**Dataset**	**Baseline attack**	**Reconstruction-based attack**
(Algorithm)	($\mathcal{A}$)	(Section 5.1)	(Accuracy)	(Accuracy)
Singling-out	$\{t_{[i,1]}\}$, at least 1 element	IHEPC	[0%, 72%]	[1%, 95%]
	$t_{[i,x]}$ with $m \leq x \leq 3m$	IHEPC	$\leq$ 60%	[57%, 82%]
	$t_{[i,x]}$ with $4m \leq x \leq 5m$	IHEPC	[87%, 97%]	[85%, 96%]
	$t_{[i,x]}$ with $x > 5m$	IHEPC	$\geq$ 97%	[96%, 100%]
Linkability	$y = 1$, at least 1 MP	ECG	67%	77%
	$2 \leq y \leq 3$ MPs linked	ECG	$\approx$ 70%	[85%, 95.7%]
	$y \geq 4$ MPs linked	ECG	[72%, 76%]	95%

Table 6: Summary of Singling out (SO) and Linkability attacks and their effectiveness.

Dataset	Dist.	**RMSE** - **PCC** for $MPD_{\hat{t}}$	**Accuracy** for $MPI_{\hat{t}}$
		(Avg. - Avg.)	(Avg.)
H-MOG	d_E	0.05 - 0.97	0.95
	d_Z	0.04 - 0.97	0.95
ECG	d_E	0.02 - 0.99	0.90
	d_Z	0.13 - 0.97	0.88
IHEPC	d_E	0.03 - 0.99	0.86
	d_Z	0.09 - 0.98	0.93
TOTAL	d_E	0.03 - 0.98	0.90
	d_Z	0.09 - 0.97	0.92

Table 7: Reconstruction: $MP_{\hat{t}}$ vs. MP_t.

developed to dynamically or statically identify the true solution among the candidates, based on the nature of the time series.

Finally, a remaining limitation of our reconstruction attack stems from the restricted modeling of the attacker's knowledge within the CSP framework. Specifically, certain types of knowledge (such as general, approximate, or statistical prior) are not easily expressible using the standard constraint formulations considered in this work. While our approach accounts for explicit and deterministic constraints, it does not incorporate more advanced inference techniques, such as those offered by machine learning, particularly reinforcement learning, which could enable an attacker to identify and exploit complex patterns in the data or solver behavior. The absence of such methods limits the generality of our model and potentially underestimates the capabilities of a more sophisticated adversary employing data-driven strategies.

6 Related Work

In this section, we review existing privacy attacks and countermeasures designed for and applied to personal data, and discuss existing studies on using MP for privacy protection.

6.1 Privacy Attacks on Time Series Data

We summarize the state-of-the-art privacy attacks on personal data, in particular time series data, in Table 8. We provide also a description of each attack in the following subsections. It is important to note that MP contains a non-linear transformation that discards explicit temporal alignment and compresses time series into subsequence distance and index profiles. It thus removes much of the direct attribute-record mapping structure that many privacy attacks exploit, indicated in the *Compatible with MP* column.

Singling-out/Re-identification: Anonymeter [22] was the first to quantify the risks of singling-out in synthetic datasets. [46] propose using Generative Adversarial Networks (GANs) to generate synthetic tabular data and evaluate re-identification risks. However, they focus on synthetic tabular data. Simply treating time series data as tabular data would trivially show high singling-out risk (due to the uniqueness of TS) and no linkability risk (due to the non-linear nature of MP transformation). [13] translates the singling-out concept in GDPR into a mathematical framework, through the concept of Predicate Singling Out (PSO), and illustrates it using Differential Privacy and k-anonymity. However, they assume no prior auxiliary knowledge, which is different from our definition. [1] and [58] quantify re-identification risk by assessing subsequence unicity in synthetic time series, both differ from our perspective of singling-out. For human mobility data, [15] shows that four spatio-temporal points can identify 95% of individuals. However, that analysis is based on measuring uniqueness in raw data, without considering any anonymization or transformation like MP.

Linkability: [47] identifies linkability as a key risk in anonymization, and [5, 22, 46] evaluate this risk using tabular or graph data, but not time series data. [6] demonstrates linkability attacks on *MicroRNA* patterns to infer health characteristics, while [11] targets GPS data of the same individual. [14] further shows that even pseudonymized interaction histories remain linkable over extended time periods. However, none of these works investigate linkability risks when time series data is transformed using MP.

Inference: Prior works have examined original data inference in time series by reconstructing perturbed datasets using filtering techniques and leveraging known values or approximate shapes [43]. Shape extraction attacks under Local Differential Privacy (LDP) has been proposed [38], as well as original location inference under DP [24, 63]. Other work mathematically quantifies the risk of predicting future points in continuous DP releases [8]. To our knowledge, no prior work analyzes attribute inference with MP.

Other inference attacks such as sensitive label and membership inference on time series data have also been explored in several studies. [16] predict label like age and sex from ECG data, while [68] propose a location inference attack on energy consumption data, and [57] analyze privacy risks in IoT networks. Only [16] and [68] consider inference attacks on data after a Matrix Profile transformation, which we will compare in Section 6.3.

Membership inference attacks [55] involve determining if an individual's data is part of a dataset (mainly used for training AI

Ref.	Singling-out/ Reidentification	Linkability	Inference attack type	Applicable to TS Data	Compatible with MP	Reconstruction technique
[22, 46, 47]	✓	✓	Sensitive label & Membership	✗	✗	✗
[13]	✓	✗	✗	✗	✗	✗
[1, 58]	✓	✗	Membership	✓	✗	✗
[15]	✓	✗	✗	✓	✗	✗
[6, 11]	✗	✓	✗	✓	✗	✗
[5]	✗	✓	✗	✗	✗	✗
[57]	✗	✗	Original data & Sensitive label	✓	✗	✗
[33, 51]	✗	✗	Membership	✓	✗	✗
[8, 24, 38, 43, 63]	✗	✗	Original data	✓	✗	✓
[16, 68]	✗	✗	Sensitive label	✓	✓	✗
ours	✓	✓	Original data & Sensitive label	✓	✓	✓

Table 8: Privacy attacks in state-of-the-art personal data releasing.

models). MIA efficiency is assessed using confidence scores [1, 51] or unsupervised methods [58], and can be enhanced by leveraging time series features [33]. MIA attacks are beyond the scope of this work but could be explored in future research.

6.2 Differential Privacy

Differential Privacy (DP) is a well known privacy-preserving technique, initially tailored for tabular data. Numerous studies have sought to adapt DP principles to time series data. For instance, [30] introduces *event-level* DP, which safeguards a single element within the TS, and *user-level* DP, which protects all elements associated with an individual. A recent survey [39] outlines state-of-the-art DP mechanisms for various TS processing tasks.

Despite these advancements, DP techniques encounter notable limitations. Ensuring user-level privacy often comes at the expense of utility [39]. Furthermore, attacks leveraging temporal correlations present significant challenges [8, 24, 39, 63], particularly for time series data with strong temporal dependencies, thus contrary to tabular data, DP may not be as good a *silver bullet* [17].

6.3 Claims on MP as Privacy Preserving

Some prior works have used MP to claim some degree of privacy protection. [16] claim that synthetic data preserving the same *MP* can prevent sensitive attribute inference attacks. However, we showed that all of our attacks can be applied to the solution they propose, when both MPD_t and MPI_t are well-preserved. If only MPD_t is preserved, all our attacks except for inversion can always be applied. [68] shows that index vector *MPI* is vulnerable to long-pattern location inference, while MPD_t combined with slightly modified MPI_t is robust against reconstruction, and can be used as privacy-preserving data sharing technique. However, all of our attacks can be applied since the *MP* is well-preserved in their proposition: the slight modification of MPI_t does not have enough impact to foil our proposed inversion attack, which tolerates an average error of 10-12% of MPI_t indexes, which is comparable to the expectation of the number of long patterns (See Section 5).

As shown in Table 8, our work is the first to address and propose singling-out, linkability, inference, and reconstruction attacks in the context of time series data transformed using MP.

7 Conclusion and Future Work

Matrix Profile (MP) has established itself as a valuable tool for time series analysis due to its versatility and efficiency. However, this study highlights significant privacy vulnerabilities associated with using MPs in time series data analysis. By formalizing generic privacy risks and proposing targeted attacks (singling-out, linkability, and inference), our work highlights the shortcomings of MP as a standalone privacy-preserving technique. By incorporating reconstruction techniques, our evaluations show over 90% success rate for singling-out, even when considering an attacker knowing only a small subsequence in the original time series (only a few points). Linkability attacks achieved up to 95% accuracy. Inference attacks are also very efficient, even without any auxiliary knowledge, yielding an average PCC of 0.71, and up to 0.99. Notably, 99.8% of cases included at least one strongly reconstructed subsequence, indicating robust shape recovery. The average RMSE was 25%, with 20% of cases below 10% (low error rate), and 85.5% featured at least one highly accurate subsequence, demonstrating strong value reconstruction. By considering auxiliary knowledge in reconstruction, we further amplify the risk of value inference. To our knowledge, this is the first work to address privacy concerns associated with sharing MPs. We believe our study paves the way for future research on privacy-preserving methods for MP.

Future work. For the reconstruction technique, we have demonstrated that it is possible to invert the *MP* function on time series of up to 200 points, using small subsequence lengths, which are common parameters in many MP applications [28, 68]. Longer time series (over 1,000 points) with larger subsequence lengths (e.g., $m = 75$, 150, or 256) should be considered for alignment with anomaly detection in long ECG sequences using MPs [66]. A more comprehensive analysis of adversarial knowledge in the context of reconstruction attacks remains an important direction for future work, for instance, by considering fuzzy or incomplete MPs. Another important direction is the implementation of countermeasures, such as adding Laplacian noise to perturb the *MPD*, which could offer privacy protection under event-DP. More advanced countermeasures [8] should also be explored and evaluated to strengthen the privacy of MP.

Acknowledgments

This work was supported by grants ANR-22-PECY-0002 (IPoP project), ANR-23-CMAS-0019 (CyberINSA project) funded by the ANR, and PID2023-150310OB-I00 (MORE4AIO project) funded by MCIU, AEI and FEDER.

References

[1] Tristan Allard, Hira Asghar, Gildas Avoine, Christophe Bobineau, Pierre Cauchois, Elisa Fromont, Anna Monreale, Francesca Naretto, Roberto Pellungrini, Francesca Pratesi, et al. 2024. Analyzing and explaining privacy risks on time series data: ongoing work and challenges. *ACM SIGKDD Explorations Newsletter* 26, 1 (2024), 49–58.

[2] Amazon Web Services. 2023. Real-time time series anomaly detection for streaming applications on Amazon Managed Service for Apache Flink. https://aws.amazon.com/fr/blogs/big-data/real-time-time-series-anomaly-detection-for-streaming-applications-on-amazon-managed-service-for-apache-flink. Accessed April 2025.

[3] TD Ameritrade and STUMPY contributors. 2020. STUMPY: A powerful and scalable time series data mining library. https://github.com/TDAmeritrade/stumpy. Accessed April 2025.

[4] Zachi I. Attia, Paul A. Friedman, Peter A. Noseworthy, Francisco Lopez-Jimenez, Dorothy J. Ladewig, Gaurav Satam, Patricia A. Pellikka, Thomas M. Munger, Samuel J. Asirvatham, Christopher G. Scott, Rickey E. Carter, and Suraj Kapa. 2019. Age and Sex Estimation Using Artificial Intelligence From Standard 12-Lead ECGs. *Circulation: Arrhythmia and Electrophysiology* 12, 9 (2019), e007284. https://doi.org/10.1161/CIRCEP.119.007284 arXiv:https://www.ahajournals.org/doi/pdf/10.1161/CIRCEP.119.007284

[5] Michael Backes, Pascal Berrang, Oana Goga, Krishna P. Gummadi, and Praveen Manoharan. 2016. On Profile Linkability despite Anonymity in Social Media Systems. In *Proceedings of the 2016 ACM on Workshop on Privacy in the Electronic Society* (Vienna, Austria) *(WPES '16)*. Association for Computing Machinery, New York, NY, USA, 25–35. https://doi.org/10.1145/2994620.2994629

[6] Michael Backes, Pascal Berrang, Anna Hecksteden, Mathias Humbert, Andreas Keller, and Tim Meyer. 2016. Privacy in Epigenetics: Temporal Linkability of {MicroRNA} Expression Profiles. In *25th USENIX security symposium (USENIX Security 16)*. 1223–1240.

[7] Nivedita Bijlani, Ramin Nilforooshan, and Samaneh Kouchaki. 2022. An unsupervised data-driven anomaly detection approach for adverse health conditions in people living with dementia: Cohort study. *JMIR aging* 5, 3 (2022), e38211.

[8] Yang Cao, Masatoshi Yoshikawa, Yonghui Xiao, and Li Xiong. 2017. Quantifying differential privacy under temporal correlations. In *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*. IEEE, 821–832.

[9] C. Chatfield. 2003. The Analysis of Time Series: An Introduction. (2003).

[10] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research* 16 (2002), 321–357.

[11] Wei Chen, Hongzhi Yin, Weiqing Wang, Lei Zhao, Wen Hua, and Xiaofang Zhou. 2017. Exploiting spatio-temporal user behaviors for user linkage. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 517–526.

[12] Roberto Chiosa, Marco Savino Piscitelli, Cheng Fan, and Alfonso Capozzoli. 2022. Towards a self-tuned data analytics-based process for an automatic context-aware detection and diagnosis of anomalies in building energy consumption timeseries. *Energy and Buildings* 270 (2022), 112302.

[13] Aloni Cohen and Kobbi Nissim. 2020. Towards formalizing the GDPR's notion of singling out. *Proceedings of the National Academy of Sciences* 117, 15 (2020), 8344–8352.

[14] Ana-Maria Creţu, Federico Monti, Stefano Marrone, Xiaowen Dong, Michael Bronstein, and Yves-Alexandre de Montjoye. 2022. Interaction data are identifiable even across long periods of time. *Nature Communications* 13, 1 (2022), 313.

[15] Yves-Alexandre De Montjoye, César A Hidalgo, Michel Verleysen, and Vincent D Blondel. 2013. Unique in the crowd: The privacy bounds of human mobility. *Scientific reports* 3, 1 (2013), 1–5.

[16] Audrey Der, Chin-Chia Michael Yeh, Yan Zheng, Junpeng Wang, Huiyuan Chen, Zhongfang Zhuang, Liang Wang, Wei Zhang, and Eamonn Keogh. 2023. Time series synthesis using the matrix profile for anonymization. In *2023 IEEE International Conference on Big Data (BigData)*. IEEE, 1908–1911.

[17] Josep Domingo-Ferrer, David Sánchez, and Alberto Blanco-Justicia. 2021. The limits of differential privacy (and its misuse in data release and machine learning). *Commun. ACM* 64, 7 (June 2021), 33–35. https://doi.org/10.1145/3433638

[18] David L Donoho. 1995. De-noising by soft-thresholding. *IEEE transactions on information theory* 41, 3 (1995), 613–627.

[19] David L Donoho and Iain M Johnstone. 1995. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the american statistical association* 90, 432 (1995), 1200–1224.

[20] Cynthia Dwork. 2006. Differential Privacy. In *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 4052)*, Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener (Eds.). Springer, 1–12.

[21] Khaled El Emam and Cecilia Alvarez. 2015. A critical appraisalto of the Article 29 Working Party Opinion 05/2014 on data anonymization techniques. *International Data Privacy Law* 5, 1 (2015), 73–87.

[22] Matteo Giomi, Franziska Boenisch, Christoph Wehmeyer, and Borbála Tasnádi. 2022. A unified framework for quantifying privacy risk in synthetic data. *arXiv preprint arXiv:2211.10459* (2022).

[23] Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. 2000. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. *circulation* 101, 23 (2000), e215–e220.

[24] Mehmet Emre Gursoy, Ling Liu, Stacey Truex, and Lei Yu. 2018. Differentially private and utility preserving publication of trajectory data. *IEEE Transactions on Mobile Computing* 18, 10 (2018), 2315–2329.

[25] Georges Hebrail and Alice Berard. 2012. Individual Household Electric Power Consumption. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C58K54.

[26] Matthieu Herrmann and Geoffrey I. Webb. 2020. Early Abandoning PrunedDTW and its application to similarity search. arXiv:2010.05371 [cs.LG]

[27] Rob J Hyndman and Anne B Koehler. 2006. Another look at measures of forecast accuracy. *International journal of forecasting* 22, 4 (2006), 679–688.

[28] Luis Ibanez-Lissen, Jose Maria De Fuentes, Lorena Gonzalez-Manzano, and Nicolas Anciaux. 2024. Continuous Authentication Leveraging Matrix Profile. In *Proceedings of the 19th International Conference on Availability, Reliability and Security*. 1–13.

[29] International Association of Privacy Professionals (IAPP). 2023. A Practical Guide to Anonymization Standards Across the EU and UK. https://iapp.org/news/a/a-practical-guide-to-anonymization-standards-across-the-eu-and-uk Accessed: 2025-04-24.

[30] Georgios Kellaris, Stavros Papadopoulos, Xiaokui Xiao, and Dimitris Papadias. 2014. Differentially private event sequences over infinite streams. (2014).

[31] Eamonn Keogh. 2024. Matrix Profile. https://www.cs.ucr.edu/~eamonn/MatrixProfile.html. Accessed: 2025-08-06.

[32] E. Keogh and S. Kasetty. 2003. On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration. *Data Mining and Knowledge Discovery* 7, 4 (2003), 349–371.

[33] Noam Koren, Abigail Goldsteen, Ariel Farkash, and Guy Amit. 2024. Membership Inference Attacks Against Time-Series Models. *arXiv preprint arXiv:2407.02870* (2024).

[34] Jing Lei and Larry Wasserman. 2014. Distribution-free prediction bands for non-parametric regression. *Journal of the Royal Statistical Society Series B: Statistical Methodology* 76, 1 (2014), 71–96.

[35] Hailin Li and Manhua Chen. 2023. Time series clustering based on normal cloud model and complex network. *Applied Soft Computing* 148 (2023), 110876.

[36] Yue Lu, Renjie Wu, Abdullah Mueen, Maria A Zuluaga, and Eamonn Keogh. 2022. Matrix profile XXIV: scaling time series anomaly detection to trillions of datapoints and ultra-fast arriving data streams. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*. 1173–1182.

[37] Elaine Mackey. 2020. A best practice approach to anonymization. *Handbook of Research Ethics and Scientific Integrity* (2020), 323–343.

[38] Yulian Mao, Qingqing Ye, Haibo Hu, Qi Wang, and Kai Huang. 2024. PrivShape: Extracting Shapes in Time Series Under User-Level Local Differential Privacy. In *40th IEEE International Conference on Data Engineering, ICDE 2024, Utrecht, The Netherlands, May 13-16, 2024*. IEEE, 1739–1751. https://doi.org/10.1109/ICDE60146.2024.00141

[39] Yulian Mao, Qingqing Ye, Qi Wang, and Haibo Hu. 2023. Differential Privacy for Time Series: A Survey. *Data Engineering* (2023), 68.

[40] Hieu X Nguyen, Duong V Nguyen, Hieu H Pham, and Cuong D Do. 2024. MPCNN: a novel matrix profile approach for CNN-based single lead sleep apnea in classification problem. *IEEE Journal of Biomedical and Health Informatics* 28, 8 (2024), 4878–4890.

[41] Cristina Nichiforov and Miltiadis Alamaniotis. 2023. Learning matrix profile method for discord-based attribution of electricity consumption pattern behavior. *Cogent Engineering* 10, 1 (2023), 2199518.

[42] Oracle. 2023. SAX and Matrix Profile techniques for root cause analysis. https://blogs.oracle.com/ai-and-datascience/post/sax-and-matrix-profile-techniques-for-root-cause-analysis. Accessed April 2025.

[43] Spiros Papadimitriou, Feifei Li, George Kollios, and Philip S Yu. 2007. Time series compressibility and privacy. In *Proceedings of the 33rd international conference on Very large data bases*. 459–470.

[44] Harris Papadopoulos, Kostas Proedrou, Volodya Vovk, and Alex Gammerman. 2002. Inductive confidence machines for regression. In *Machine learning: ECML 2002: 13th European conference on machine learning Helsinki, Finland, August 19–23, 2002 proceedings 13*. Springer, 345–356.

[45] John Paparrizos, Yuhao Kang, Paul Boniol, Ruey S Tsay, Themis Palpanas, and Michael J Franklin. 2022. TSB-UAD: an end-to-end benchmark suite for univariate time-series anomaly detection. *Proceedings of the VLDB Endowment* 15, 8 (2022), 1697–1711.

[46] Noseong Park, Mahmoud Mohammadi, Kshitij Gorde, Sushil Jajodia, Hongkyu Park, and Youngmin Kim. 2018. Data synthesis based on generative adversarial networks. *arXiv preprint arXiv:1806.03384* (2018).

[47] Article 29 Data Protection Working Party. 2014. Opinion 05/2014 on Anonymisation Techniques.

[48] Laurent Perron and Vincent Furnon. 2024. *OR-Tools*. Google. https://developers.google.com/optimization/

[49] Adrien Petralia, Philippe Charpentier, Paul Boniol, and Themis Palpanas. 2023. Appliance Detection Using Very Low-Frequency Smart Meter Time Series. In *Proceedings of the 14th ACM International Conference on Future Energy Systems*. 214–225.

[50] D. K. Prasad and S. Rao. 2012. A Review of Interpolation Techniques and Their Applications. *International Journal of Computer Applications* 42, 15 (2012), 23–29.

[51] Apostolos Pyrgelis, Carmela Troncoso, and Emiliano De Cristofaro. 2020. Measuring Membership Privacy on Aggregate Location Time-Series. *Proc. ACM Meas. Anal. Comput. Syst.* 4, 2, Article 36 (June 2020), 28 pages. https://doi.org/10.1145/3392154

[52] Helmy Rahadian, Steven Bandong, Augie Widyotriatmo, and Endra Joelianto. 2023. Image encoding selection based on Pearson correlation coefficient for time series anomaly detection. *Alexandria Engineering Journal* 82 (2023), 304–322.

[53] Nima Sarajpoor, Zohreh Parvini, and Ali Jahanbani. 2023. Time Series Aggregation in Power System Studies in the Presence of Wind Energy: A Matrix-Profile Perspective. In *2023 North American Power Symposium (NAPS)*. IEEE, 1–5.

[54] Philip Sedgwick. 2012. Pearson's correlation coefficient. *Bmj* 345 (2012).

[55] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership Inference Attacks Against Machine Learning Models. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*. IEEE Computer Society, 3–18.

[56] Grigore Stamatescu, Radu Plamanescu, Irina Ciornei, and Mihaela Albu. 2022. Detection of anomalies in power profiles using data analytics. In *2022 IEEE 12th International Workshop on Applied Measurements for Power Systems (AMPS)*. IEEE, 1–6.

[57] Meng Sun and Wee Peng Tay. 2017. Inference and data privacy in IoT networks. In *2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. 1–5. https://doi.org/10.1109/SPAWC.2017.8227701

[58] Muhang Tian, Bernie Chen, Allan Guo, Shiyi Jiang, and Anru R Zhang. 2024. Reliable generation of privacy-preserving synthetic electronic health record time series via diffusion models. *Journal of the American Medical Informatics Association* 31, 11 (2024), 2529–2539.

[59] Andreas Wächter and Lorenz T Biegler. 2006. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming* 106 (2006), 25–57.

[60] Rutuja Wankhedkar and Sanjay Kumar Jain. 2021. Motif discovery and anomaly detection in an ECG using matrix profile. In *Progress in Advanced Computing and Intelligent Engineering: Proceedings of ICACIE 2019, Volume 1*. Springer, 88–95.

[61] Martin A Weiss and Kristin Archick. 2016. US-EU data privacy: from safe harbor to privacy shield.

[62] Dennis L Wilson. 1972. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics* 3 (1972), 408–421.

[63] Yonghui Xiao, Li Xiong, Si Zhang, and Yang Cao. 2017. Loclok: Location cloaking with differential privacy via hidden markov model. *Proceedings of the VLDB Endowment* 10, 12 (2017), 1901–1904.

[64] James J Yang and Anne Buu. 2024. Efficient matrix profile computation with Euclidean distance using Eigen transformation: Performance evaluation based on beat-to-beat interval (BBI) data. *Statistics in medicine* 43, 16 (2024), 3051–3061.

[65] Qing Yang, Ge Peng, David T Nguyen, Xin Qi, Gang Zhou, Zdeňka Sitová, Paolo Gasti, and Kiran S Balagani. 2014. A multimodal data set for evaluating continuous authentication performance in smartphones. In *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*. 358–359.

[66] 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.

[67] Haoying Zhang and Fabien Girard. 2025. Artifacts for: Privacy Attacks on Matrix Profiles via Reconstruction Technique. https://gitlab.inria.fr/haoying.zhang/attack_matrix_profile.

[68] Li Zhang, Jiahao Ding, Yifeng Gao, and Jessica Lin. 2023. PMP: Privacy-aware matrix profile against sensitive pattern inference for time series. In *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*. SIAM, 891–899.

[69] Puyang Zhao, James J Yang, and Anne Buu. 2025. Applied statistical methods for identifying features of heart rate that are associated with nicotine vaping. *The American journal of drug and alcohol abuse* 51, 2 (2025), 165–172.

[70] Yan Zhu, Abdullah Mueen, and Eamonn Keogh. 2012. Efficient Discovery of Time Series Motifs under Uniform Scaling. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1255–1263. https://doi.org/10.1145/2339530.2339727

A Singling Out Case Study

Several concrete scenarios of singling-out attacks, characterized by distinct types of auxiliary knowledge, are presented below:

Example A.1 (Singling out from a known time series)**.**

> Alice underwent an ECG at a hospital for diagnosis purposes. The hospital published *MP*s, including Alice's, with associated anomaly classification, where anomalies are classified as either benign or malignant. These MPs were published under the assumption that the identities of the patients were anonymized. However, due to a data leakage, Eve, an attacker, gained access to the original ECG of Alice. Using this auxiliary information, Eve aims to single out Alice's *MP* to determine whether it corresponds to a malignant anomaly.

In this scenario, the attacker knows the time series t_{known} of an individual u_j, i.e., $\mathcal{A} = \{t_{known}, u_j = u(t_{known})\}$. To realize the singling-out attack, all the attacker needs to do is compute $MP_{t_{known}}$ and identify $r = i$ such that $MP_{t_i} = MP_{t_{known}}$. This is straightforward and very efficient.Note that although the Matrix Profile function *MP* is non-injective (e.g., two time series up to a constant have the same *MP*), the probability of multiple time series mapping to the same *MP* is negligible due to the high variety, precision and length of real time series datasets.

Example A.2 (Singling out from a known subsequence -$\mathcal{A}_{subseq}$-)**.**

> Eve, a taxi driver, takes Alice on a trip and thus knows her exact trajectory for that specific journey. The taxi company publishes weekly trajectory *MP*s for all users, including Alice's, to analyze travel patterns and predict rush hours. These MPs are assumed to be anonymized. However, using the trajectory information from Alice's trip as auxiliary knowledge, Eve seeks to single out Alice's *MP* within the dataset. His goal is to deduce sensitive details, such as Alice's typical absence hours, identify her workplace, or exploit this information to rob her house during her absence.

Example A.3 (Singling out from a known set of points -$\mathcal{A}_{sparse}$-)**.**

> Alice uses Strava to track her running trajectory. One day, Eve observes her running at noon in Central Park and near a company building 20 minutes later, giving him two specific points along Alice's trajectory. Strava later publishes anonymized *MP*s of users running trajectories to analyze exercise habits. Using his knowledge of these discard points as auxiliary information, Eve seeks to single out Alice's *MP* within the dataset. His goal is to infer her repeated running trajectories or even reconstruct her raw trajectory to identify the places she frequently visits.

Algorithm. The general singling-out algorithm operates in two main steps (see Algorithm 1). First, the algorithm classifies the type of auxiliary information into one of three categories according to the three examples (line 4). Second, based on classification, the algorithm applies a tailored attack strategy. For an integral time series or sufficiently large subsequences, it computes the matrix

Algorithm 1 SO_basic

```
Input: Attacker knowledge object A, published MP dataset $\mathbf{MP_T}$, MP subsequence length m
Output: One identifier of $MP$ in $MP_T$
$r \leftarrow \emptyset$
$type \leftarrow$ TypeClassifier(A)        ▷ Identify type: entire series ("integral"), subsequence ("subsequence"), points ("points")
if $type$ = "integral" then
    $MP' \leftarrow MP(A.ts, m)$
    for each $MP_{t_i} \in MP_T$ do
        if $MP_{t_i} = MP'$ then
            $r \leftarrow i$
            break
        end if
    end for
else if $type$ = "subsequence" then
    $MP' \leftarrow MP(A.ts, m)$
    $index \leftarrow A.index$   ▷ Indices of known subsequence in $t \in T$
    $match \leftarrow$ False
    for each $MP_{t_i} \in MP_T$ do
        for each $i$ such that $MP_{t_i}.MPI[i] \in$ index do
            $match \leftarrow$ True
            if $MP_{t_i}.MPD[i] \neq MP'.MPD[i]$ or $MP_{t_i}.MPI[i] \neq MP'.MPI[i]$ then
                $match \leftarrow$ False
                break
            end if
        end for
        if $match$ then
            $r \leftarrow i$
            break
        end if
    end for
else
    $T' \leftarrow Interpolation(A.ts)$
    $MP' \leftarrow MP(T', m)$
    $dists$ = []
    for each $MP_{t_i} \in MP_T$ do
        dists.append($Dist(MP_{t_i}.MPD, MP'.MPD)$)
    end for
    $r \leftarrow argmin(dists)$
end if
return $r$
```

profile MP' of the known data and compares it directly with the profiles in MP_T to identify a match ((lines 5 to 12)). A match occurs if, for indices within the known subsequence (or the integral time series), the distance values in MPD and index values in MPI match those in MP'. If a match is found, the corresponding matrix profile is singled out. If no match is found or if the auxiliary information is insufficient (e.g., sparse points), the algorithm returns an empty set, indicating that no profile could be singled out.

If the attacker knows one or more intervals, as illustrated in Figure 8 with two intervals of length 40 from the original time series (Additional examples are provided in our artifact repository). The attacker can then compute the matrix profile of the known subsequences and single out a matrix profile in the published dataset matching all the points that fall in the known subsequence (see lines 13 to 19). Precisely, the attacker can firstly compute the $MPI_{t_{known}}$ and $MPD_{t_{known}}$ of the known subsequence. By comparing these values to the published MPI and MPD, the attacker can confidently identify ID = i, as all MPI points within the window match $\mathcal{A}$ (points matching $\mathcal{A}$ are highlighted in red).

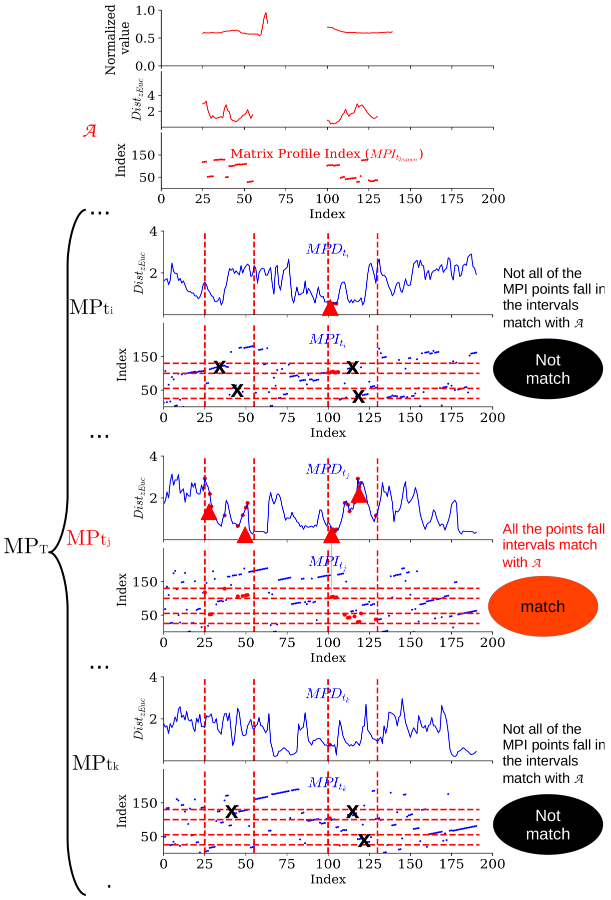


Figure 8: Singling out from two known subsequences.

Experimental Result. We applied the *SO_basic* algorithm to electricity consumption time series from the IHEPC dataset [25] (details in Section 5.1). We extract series of $n = 200$ points, with MPs computed using $m = 10$. We vary the known subsequence length or number of sparse points, denoted as ℓ, to assess the success rate of singling-out. For the known subsequence use case, we suppose the attacker knows the first ℓ consecutive points. And for the known sparse points use case, we form the attacker knowledge by downsampling the time series. Figure 9 shows the results, with the blue curves corresponding to known subsequences and the green curve to known sparse points. The zoomed-in portion of the plot focuses on ℓ values between 40 and 46, the point of transition for success rate across 90%, for the known subsequence scenario, since the transition does not exist in the known sparse point scenario. Results for other datasets are consistent (hence are not shown).

Algorithm 2 Link_basic

```
1:  Input: Attacker knowledge object $\mathcal{A}_{link}$ (a set of MPs linked to the same individual), published MPs dataset $\mathbf{MP_T}$, pattern length $\ell (= \mathbf{n})$ by default
2:  Output: Set of identifiers of $MP$ linked to $\mathcal{A}_{link}$ in $MP_T$
3:  $MPs' \leftarrow \mathcal{A}_{link}.mp$                                  // Known MPs
4:  $r \leftarrow$ {Identifier of the known MPs}
5:  $candidats \leftarrow \emptyset$
6:  for each $MP' \in MPs'$ do
7:      $1NNDists \leftarrow$ empty list
8:      for each $MP_{t_i} \in MP_T$   // Compute the 1NN distance between $MPD_{t_i}$ and target $MPD$ do
9:          $d \leftarrow \min\left(MP_{Euc}(MP'.MPD, MP_{t_i}.MPD, \ell).MPD\right)$
10:         Append $(MP_{t_i}.id, d)$ to $1NNDists$
11:     end for
12:     Sort $1NNDists$ by distance values
13:     $(r', d_{\min}) \leftarrow 1NNDists[0]$
14:     Append $(r', d_{\min})$ to $candidats$
15: end for
16: $res \leftarrow$ Indexes with minimum distance values in $candidats$
17: Append $res$ into $r$
        return $r$
```

Attacker knowledge size (y)	Min	Avg.	Max
1	0.67	0.67	0.67
[2, 3]	0.69	0.7	0.71
[4, 6]	**0.72**	**0.74**	**0.76**
[6, 8]	**0.72**	**0.73**	**0.74**

Table 9: Success rate with various attacker knowledge y.

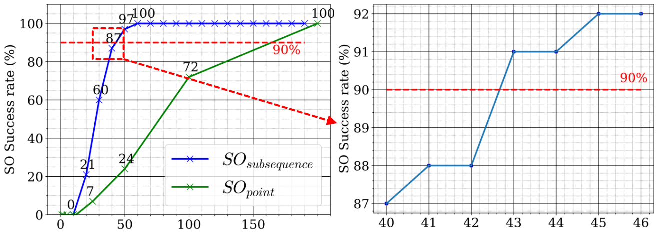


Figure 9: Success rate of SO in function of ℓ.

Takeaway. The risk of singling-out a matrix profile is limited when the known subsequences in the original time series are short ($\ell < m + 20$). However, as the length of the known subsequence increases ($\ell > m + 32$), the risk escalates rapidly, exceeding 90%. The risk associated with known sparse points becomes relatively high only when the downsampling frequency is high (e.g., every other point). Access to the full time series or sufficiently long subsequences ($\ell > m + 50$) results in very high privacy risks.

B Linkability Attack Scenario and Algorithm

B.1 Algorithm and Extended Experiments

Context. We consider an attacker with knowledge of one or more MPs that belong to a given individual ($\mathcal{A}_{link}$). The baseline linkability attack tries to link this attacker knowledge to MPs present in MP_T. This is done by taking the matrix profile(s) either (i) directly from $\mathcal{A}_{link}$ (Example 3.2), or (ii) after building $\mathcal{A}_{link}$ by calculating the MP of the known time series $\mathcal{A}_{ts}$ (see below, Example B.1).

Algorithm. Algorithm 2 gives the pseudo-code of this attack. The algorithm work as follows: it first computes the matrix profile distances (with Euclidean distance measure) between the target MPD and the MPD vectors in MP_T, storing the minimum distances from these profiles (lines 4-6); then, as these distances represent the distances of the closest matching patterns of length ℓ. This value is defined by the attacker. If the attacker has no knowledge then this can simply be $\ell = n$ the total length of the vector (in which case this is equivalent to using simply the Euclidean distance between both vectors), or if the attacker has some knowledge of the application domain, ℓ can be tailored more finely e.g., ℓ can be the interval size between two QRS in a normal ECG, the algorithm identifies the matrix profiles with the smallest distance (line 7). If several matrix profiles have exactly the same smallest distance (this never occurred experimentally), all of them would be returned. Note that other measures may be used, such as Pearson correlation, or even simply using vector Euclidean distance between known MPs and the other MPs. However, measures like DTW [26] are less effective because warping disregards the critical length of patterns, which reflects cardiac rhythm in this context.

Experimental Result. We evaluate *Link_basic* algorithm on a public ECG dataset [23] containing 70 recordings from 7 individuals (details in Section 5.1). The experimental setup is as follows: each individual contributes 10 non-overlapping ECG time series of length 200, aligning with Example 3.2. The attacker knows $y < 10$ MPs of a single individual and attempts to link them to other MPs of the same individual in the published dataset MP_T. The success rate is defined as the ratio of correctly identified MPs linked to the target individual. A success rate of around 1/7 corresponds to random guessing. To avoid bias, each experiment was repeated at least 5 times, and the average success rates are reported in Table 9 in Appendix B. The results demonstrate the influence of the attacker's knowledge size (y) on linkability risks.

Takeaway. The obfuscation introduced by matrix profiles is insufficient to prevent linkability attack under certain scenarios. As shown in Table 9, the success rate increases with the attacker's knowledge, ranging from 67% with a single known link to 76% with 4–6 known links. Even with 6–8 known links, the success rate remains high. Although the risk rises with more prior knowledge, it stays within a narrow band of 67% to 76%, indicating that even limited attacker knowledge poses a significant privacy threat.

B.2 A Linkability Scenario Using $\mathcal{A}_{ts}$

Another scenario of linkability attack, with a different attacker knowledge definition is described below:

Example B.1 (Linkability attack from known subsequence -$\mathcal{A}_{ts}$-)**.**

> Alice uses a ride-sharing service and is driving Marvin as a passenger. Marvin activates his ride-sharing app during all the trip, thus he records the time series of the trajectory in real time. The

Algorithm 3 Reconstruction

```
1:  Input: Published MP dataset $\mathbf{MP_T}$, distance measure in $MP$ $Dist$, subsequence length used in $MP$ $m$, number of parallel processors $x$, CSP solver $solver$.
2:  Output: Original time series dataset $T$
3:  $T \leftarrow []$
4:  for each $MP_{t_i} \in MP_T$ do
5:      $n \leftarrow m + \mathrm{len}(MP_{t_i}) - 1$ // Deduce the length of the time series from the lengths of subsequence and the $MP$
6:      $T_{initialized} \leftarrow \{t_0, t_1, \ldots\}$ // Initialize a population of time series of length $n$ with random float values in [0,1].
7:      $Loss_{initialized} \leftarrow []$
8:      for each $t_j \in T_{initialized}$ do
9:          $loss \leftarrow O^{optimized}(t_j, MP_{t_i})$
10:         Append $(t_j, loss)$ to $Loss_{initialized}$
11:     end for
12:     Sort $Loss_{initialized}$ by increasing order of loss values
13:     $T_{initialized} \leftarrow Loss_{initialized}[0 : x][:, 0]$ // Retain only the $x$ series with the lowest losses, where $x$ is the number of processors
14:     $Loss_{final} \leftarrow []$
15:     for each $t_j \in T_{initialized}$ do
16:         $loss \leftarrow Solve(t_j, MP_{t_i}, Dist, m, solver, O^{optimized})$ // Solve the CSP problem defined in 4.2
17:         Append $(t_j, loss)$ to $Loss_{final}$
18:     end for
19:     Sort $Loss_{final}$ by increasing order of loss values
20:     $t_i \leftarrow Loss_{final}[0][0]$ // Keep the reconstructed time series with the minimum loss value
21:     Append $t_i$ to $T$
22: end for
        return $T$
```

Algorithm 4 Post_processing_ECG

```
Input: A time series $t$, standard ratios of U, T, and V relative to the QRS pattern $\sigma_U, \sigma_T$ and $\sigma_V$, a set of exemplar QRS patterns $QRS_{patterns}$, exemplar portions of U, T, and V $p_U$, $p_T$, and $p_V$, similarity threshold $\epsilon$.
$QRS, T, U, V \leftarrow [], [], [], []$
for each $QRS' \in QRS_{patterns}$ // Detect QRS patterns do
    $QRS\_detection\_mp \leftarrow MP(t, |QRS'|)$
end for
for each $i$ where $QRS\_detection\_mp.MPD[i] < \epsilon$ do
    $QRS_{start} \leftarrow QRS\_detection\_mp.MPI[i]$
    Append $(QRS_{start}, QRS_{start} + |QRS'|)$ to $QRS$
end for
for $i \leftarrow 1$ to $|QRS| - 1$ // Add the U, T, and V intervals according to their standard proportional spacing in a typical ECG do
    $start_T \leftarrow QRS[i][1]$
    $end_V \leftarrow QRS[i+1][0]$
    $length \leftarrow end_V - start_T$
    Append $(start_T, length \times p_T)$ to $T$
    Append $(length \times p_T + 1, length \times (p_T + p_U) + 1)$ to $U$
    Append $(length \times (p_T + p_U) + 2, end_V)$ to $V$
end for
for each $interval \in T$ // Normalize each pattern with their scaling factor in a typical ECG do
    $t[interval] \leftarrow Normalize(t[interval], \sigma_T)$
end for
for each $interval \in U$ do
    $t[interval] \leftarrow Normalize(t[interval], \sigma_U)$
```

> ride-sharing service publishes anonymized matrix profiles (MPs) of its drivers' trajectories for the whole week, including those associated with Alice's trips. These matrix profiles summarize the patterns in the trajectories while preserving anonymity. Leveraging the time series he observed during his ride with Alice, Marvin conducts a linkability attack: he compares it with the pseudonymized MPs published by the ride-sharing service, seeking to identify which published profiles correspond to Alice's trip for the whole week.

In this scenario, the attacker knows a time series t_{known} for individual u_j, i.e., $\mathcal{A}_{ts} = \{t_{known}\}$ and can thus compute its MP in order to build the knowledge needed to run the linkability attack, i.e. $\mathcal{A}_{link} = \{MP(t_{known})\}$. The objective is to link at least one other matrix profile in MP_T that corresponds to the same individual using this auxiliary knowledge.

Note that attacks can be combined: the attacker (Marvin) could first perform a linkability attack to link his own subsequence with the week MP of Alice, and then conduct a reconstruction attack on Alice's MP to obtain trajectory details.

C Reconstruction Algorithms

Based on the optimized reconstruction of MPs as a CSP problem, we derive a reconstruction attack algorithm, which proceeds as follows: For each MP in MP_T, we first initialize a population of time series with random float values in [0,1] (lines 3-4). Only x series with the lowest loss value calculated with the function $O^{optimized}$ will be kept, where x can be defined as the number of processors that the attacker possesses to execute the resolution in parallel (lines 5-10). This step is crucial as the non-convex objective function has multiple minima, and a larger population boosts attack success. Then we solve the system presented in definition 4.2 by using a CSP solver and each initialized series in the population in parallel (line 15). Finally, we append the best solution (with the minimum loss value) to the result list (lines 16-17).

This algorithm can be followed by appropriate post-processing. The specifics of the post-processing depend on the context and the nature of the data (e.g., ECG, electric consumption), as well as any additional knowledge the attacker may have.

Post-processing for ECG Reconstruction. After reconstructing the time series by solving the CSP system, post-processing methods can be applied to improve the utility of the reconstructed time series.

	Classifier	Original TS	Matrix Profile
		(F-score - Acc.)	(F-score - Acc.)
Age	SVM	0.84 - 0.84	**0.87 - 0.88**
	RF	0.86 - 0.87	**0.88 - 0.89**
	kNN	0.86 - 0.87	**0.88 - 0.89**
Sex	SVM	0.92 - 0.92	0.83 - 0.83
	RF	0.92 - 0.92	0.79 - 0.80
	kNN	0.92 - 0.92	0.86 - 0.86

Table 10: F-score and Accuracy for Age/Sex attribute inference using original time series TS and MPs MPD.

	Classifier	Original TS	Reconstructed TS
		(F-score - Acc.)	(F-score - Acc.)
Age	CNN	0.86 - 0.87	**0.73 - 0.73**
Sex	CNN	0.90 - 0.87	**0.90 - 0.84**

Table 11: F-score and Accuracy for Age/Sex label inference using original time series TS and reconstructed ones $\hat{TS}$.

For instance, filtering methods can be applied to separate the noise from the true signal [18, 19, 43]. Assuming the attacker knows the data type (e.g., ECG, energy consumption), they can apply transformation processes to improve the utility of the reconstructed time series.

For ECG data, we present here a specific post-processing leveraging public knowledge of ECG patterns. In a normal heartbeat, four primary waveforms are present: V, QRS, T, and U. The knowledge of this specific "shape" constitutes the attacker's auxiliary knowledge that we term $\mathcal{A}_{shape}$. The reconstructed time series often show slight translations and dilatations, but the QRS patterns are usually preserved. Thus, the attacker can identify the QRS patterns (lines 2-6) and deduce the patterns T, U, and V (lines 7-13) and transform them using the corresponding ratios (lines 14-19). The detailed algorithm is described in Algorithm 4.

D Label Inference Attacks with Matrix Profile

In Section 4.5, we proposed an inference attack where the goal is to infer the original attributes (data) from a MP. We now propose another possible inference, which is to infer sensitive label such as age and gender that can be possibly lead to discrimination. We discuss this attack under different scenarios and auxiliary knowledge and give experimental results.

Example D.1 (Inference from a public dataset)**.**

> A hospital publishes a database of Matrix Profiles derived from patients' ECGs. Eve, an attacker, aims to infer the patients' age and gender from this data.

The general example that we consider here is described in Example D.1, that we will then precise the auxiliary knowledge. We suppose firstly an auxiliary knowledge that includes public matrix profiles or time series with sensitive labels, from a public annotated dataset with data from the same distribution for example. The algorithm trains a model where MPD vectors serve as inputs and attribute labels as outputs (though it could be extended to include MPI values). It then applies the trained model to the MPs in MP_T to predict and return inferred attribute values.

Experimental Results. We evaluate the inference attack using the ECG dataset [23] described in Section 5.1. For each patient, we extract 140 samples of 200 points, and calculate MPs using the same subsequence length and distance measure as in the linkability attack experiments. We used 10-fold cross validation with 50% of the data used for training and 50% for testing, ensuring no overlap and divided randomly, and for each 14, 000 samples of 100 points over sliding windows, which aligns with the same settings used in previous work [16]. We apply SMOTEENN [10, 62] for gender inference, to mitigate issues arising from the imbalanced class distribution. Support Vector Machine (SVM), Random Forest (RF) and k-Nearest Neighbors (kNN) are used for label inference, with parameters tuned for each classifier (default settings for SVM, maximum depth of 20 for RF, and $k = 10$ for kNN). Table 10 summarizes the F-score and accuracy for age and sex inference using original time series TS and matrix profiles MPD. Attributes were labeled as binary for age (over or under 60 years) and sex (male or female).

Takeaway. The results highlight the effectiveness of matrix profiles for label inference. For age, accuracy and F-score using matrix profile exceed those obtained from the original time series (see values in bold). For sex inference, while matrix profiles leads to a slight degradation (10%) in performance compared to the original time series, they remain highly effective for inference. These findings challenge the assumption that matrix profiles provide sufficient protection against inference attacks as considered in [16]. High-risk scenarios arise when attackers have access to auxiliary datasets with similar time series distributions and corresponding attribute labels.

We now consider a black-box attacker scenario, where auxiliary knowledge is limited to an inference model trained on a dataset drawn from the same distribution as the original time series, but without access to the actual training data (e.g., when the dataset is private). In this setting, the attacker must first reconstruct the time series from published MPs before attempting to infer sensitive labels.

Experimental Results. We use the same ECG dataset as in the previous case study and a streamlined version of the CNN model proposed in [4]. To mitigate class imbalance in the gender inference task, we apply the same SMOTEENN mechanism. Each model is trained for 100 epochs, and the averaged results are reported in Table 11.

Takeaway. As shown in Table 11, reconstructed time series remain highly effective for both age and sex inference. For age inference, the F-score drops from 0.86 to 0.73 (accuracy from 0.87 to 0.73), a degradation of about 13–14%, yet performance remains strong (above 70%). For sex inference, reconstructed time series achieve virtually the same F-score as the original data (0.90) and only a slight drop in accuracy (0.87 to 0.84), confirming that MP reconstruction can preserve sensitive information to a degree that poses a privacy risk.

E Normalized Matrix Profile

Suppose a vector (time series) denoted T, we note min and max as the minimum and maximum values in T, the normalized vector $\overline{T}$ where $\overline{T_i} = \frac{T_i - min}{max - min}$. Suppose a Matrix Profile MP, with a distance measure between Manhattan distance, Euclidean distance and z-normalized Euclidean distance, and the subsequence length m. MP is composed by MPI and MPD, we denote $MP = (MPD, MPI)$ as the Matrix Profile calculated from T and $\overline{MP} = (\overline{MPD}, \overline{MPI})$ the Matrix Profile calculated from $\overline{T}$, now we demonstrate how to calculate $\overline{MP}$ from MP.

The MPI corresponds to the indices of the nearest subsequences, the normalization of the values have no impact on this information, so we have $MPI = \overline{MPI}$ for any distance measure.

For the MPD vector, the transformation depends on the distance measure used (which is an information public to the attacker when sharing the Matrix Profile).

If the distance measure is z-normalized Euclidean distance, we have $MPD = \overline{MPD}$, since the z-normalization will be applied on each subsequence before calculating the distances, which erase the difference of T and $\overline{T}$ values.

If the distance measure is Euclidean distance, we have

$$MPD_i^2 = \sum_{k=0}^{m} (T_{i,k} - T_{j,k})^2 \tag{7}$$

$$\overline{MPD_i}^2 = \sum_{k=0}^{m} (\frac{T_{i,k} - min}{max - min} - \frac{T_{j,k} - min}{max - min})^2 \tag{8}$$

$$\frac{1}{(max - min)^2} \sum_{k=0}^{m} (T_{i,k} - T_{j,k})^2 = \frac{MPD_i^2}{(max - min)^2} \tag{9}$$

By inference from equations 8 and 9, we get

$$\overline{MPD} = \frac{MPD}{max - min} \tag{10}$$

If the distance measure is Manhattan distance, the proof is similar to that with Euclidean distance, we can easily get $\overline{MPD} = \frac{MPD}{max - min}$.

F Hyperparameters in IPOPT

The hyperparameter tuning has been effected using optuna, with randomly generated time series of size 120 and matrix profile calculated with m=20 and z-normalized Euclidean Distance. Each proposition of hyperparameters are evaluated with 10 samples, with solving time to each time series limited to 70 seconds. The best hyperparameters are as following, for the explications of the hyperparameters, please refer to the official site of IPOPT [5]:

- linear_solver : ma57
- alpha_for_y : full
- mu_strategy : adaptive
- hessian_approximation : limited-memory
- nlp_scaling_method : none
- fast_step_computation : no
- limited_memory_max_history : 30
- recalc_y : yes
- sens : FD

Here, two parameters are worth a little explanation:

The choice of ma57 as the linear solver. As indicate in [59], this solver is best used on small/medium-sized problems. Therefore, its choice might be due to the small size of the problem that we did the tuning on. For bigger a bigger n, it could be better to choose ma86.

The choice of FD (Forward Difference) as the gradient approximation method. Among the other options were the exact gradient computation. When comparing both, we observe that the exact formula leads to a slightly faster convergence in terms of iterations, but at the cost of a lot more time to compute. Therefore, we will let this option behind as proposed by the process of tuning, but if one is more interested in accuracy than efficiency he might find it useful.

G Coefficients in Objective Function

There are two coefficients, α and β, in the objective function O_{opt} (see section 4.2). We tuned the coefficients ranging from 1 to 5 (1,1.57,2.14,2.71,3.29,3.86,4.43,5), over time series of n=200 and m=10, solving for each in 15 minutes and calculating a mean with 10 time series. The best coefficients according to the experiments are $\alpha = 4.43$ and $\beta = 2.14$ for d_E and $\alpha = \beta = 1$ for d_Z.

H Soft vs Hard Constraints

When reconstructing a time series of length 200 with $Dist = d_E$ and $m = 10$, and the O_{opt} objective function, the model yields 400 non-zero entries in the equality constraint Jacobian (down from $191^2 = 36K$ with the CSP model), 0 entries in the inequality constraint Jacobian (down from around $191^3 = 7M$ with the initial O function or the CSP model). In 200 seconds, the solver is able to complete 30 iterations using O_{opt}, compared to 10 when using objective function O and inequality constraints, and no result produced for the CSP model.

I Benchmark of Solvers

Apart from IPOPT and Or-Tools that we used to solve the CSP system, we also tried SLSQP, it stops to iterate when finding a local minimum point, which is harmful for the utility of the attack.

Another popular gradient-descent-based solver is Adam. However, as shown in Figure 10, our benchmark demonstrates that IPOPT achieves faster convergence.

J Parameters Influence in IPOPT

We first study the influence of the parameters n and m in the solver IPOPT. In each plot in Figure 12, we show the evaluation of the loss value of the objective function in equation 3 and the PCC between the solution and the ground-truth time series varying with iterations. Figures 12a and 12b (resp., 12c and 12d) show the influence of parameter m (resp. n), with n fixed to 100 (resp., with m fixed to 10) for euclidean distance and z-normalized euclidean distance. Each point in the figures is the mean of 10 runs of experiments. The faster the loss decreases, the more efficient the resolution, and the higher the PCC value, the more effective the attack (that we reconstruct a more useful time series).

In general, the loss decreases all quite quickly at the start of the iterations and slows down after, with any value of n or m. Specially for n, the bigger the value of n is, the slower the loss decrease, because the number of variables and the number of equations in

[5] COIN-HSL: A Collection of Linear Algebra Libraries. 2024. https://licences.stfc.ac.uk/product/coin-hsl.

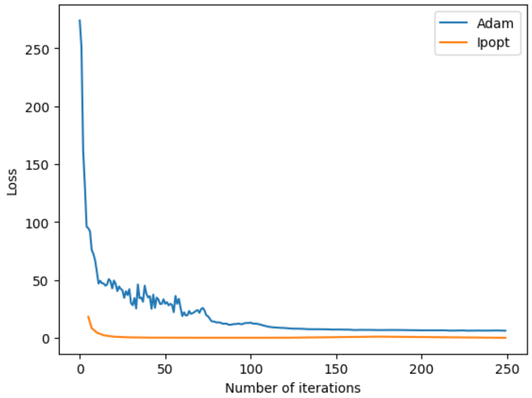


Figure 10: Adam v.s. IPOPT.

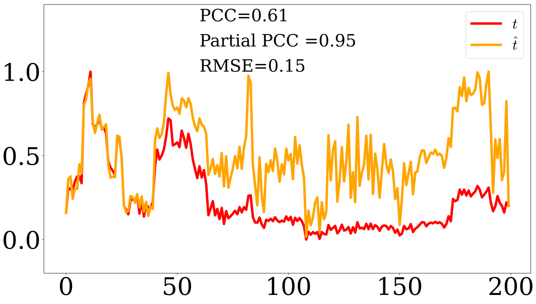


Figure 11: Avg. quality solution with d_Z.

the objective function increase (which is equals to $(n - m + 1)^2$), leading to a higher skewness function and slower convergence. The experimental time complexity is $O(n^3)$. However, the *PCC* increases faster with larger value of n (see figures 12c,12d), because little change in loss brings great improvement in the utility when n is big. For another parameter m, the loss decreases faster with bigger value of m, with the same reason explained with n. The *PCC* increases faster with smaller value at the start of the iterations (see figures 12a and 12b). Additionally, by comparing the values of *PCC*, the performance is better with smaller values of m (with $m \leq 18$ for d_E and $m \leq 18$ for d_Z).

In conclusion, the attack is efficient when n is small and m is large, but becomes more effective with a small m, regardless of n.

K Example of Real Matrix Profile Inversion

We use the maximum partial PCC as one of the metrics to quantify the reconstruction attack's ability to reconstruct at lease one subsequence with high fidelity. As illustrated in Figure 11, the overall PCC is 0.61, showing a moderate performance, yet the first pattern of around 50 points is reconstructed very accurately, yielding a high partial PCC of 0.95. This gap reveals a substantial privacy risk that the global PCC alone would understate.

L Integer Matrix Profile Inversion

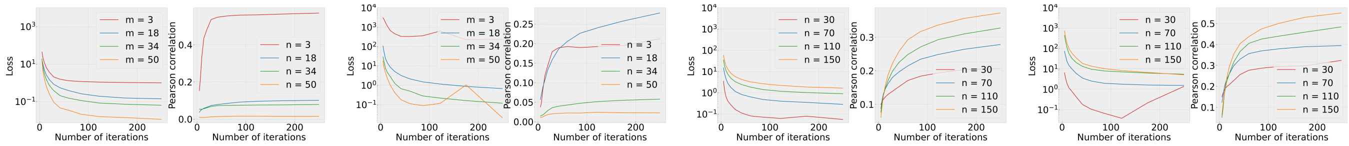
We also implement the reconstruction attack using values from a reduced domain (i.e., integer numbers) as preliminary work. The results showed perfect resolution.

The energy consumption dataset was used to evaluate the reconstruction, which contains real numbers. We then digitized the values by dividing the real domain into several intervals. Each point is assigned an integer value based on the interval it belongs to, as shown in Figure 13a.

After digitization, we compute the *MP* with d_M(ensuring the matrix profile values remain integers) and set m to 10, which is suitable for applications such as detecting activities lasting at least 10 minutes. Using Or-Tools as the solver, we obtain four solutions. All solutions share the same *MP* as the original time series, achieving perfect resolution. The original time series is reconstructed (represented by the green curve in Figure 13b), while the other solutions differ only by one or two points each.

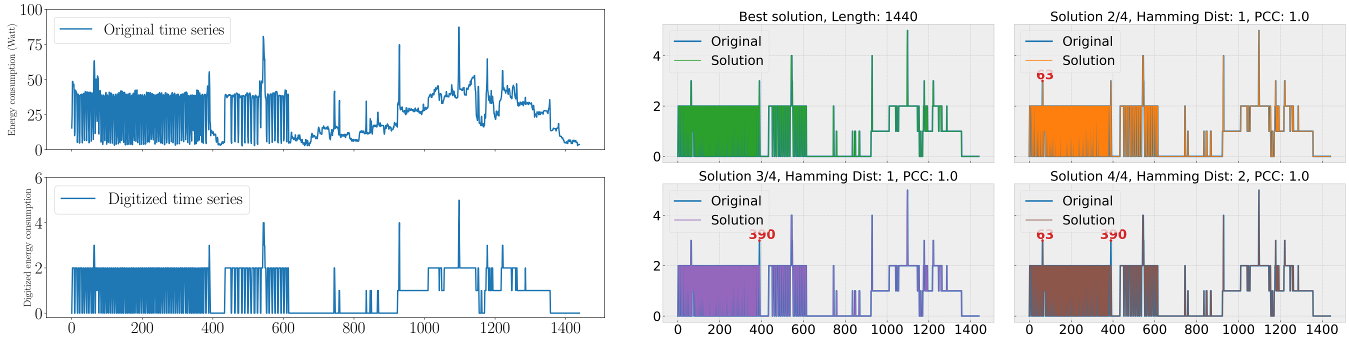
M Reconstruction with Individual Knowledge

Figure 14 illustrates the results of a reconstruction attack on an example using different types of auxiliary knowledge, as described in Section 4.3. The effectiveness of the attack increases with all types of auxiliary knowledge. Among them, using the mean value and downsampling (one point every 30) proves to be the most effective in enhancing reconstruction accuracy with minimal information. In contrast, fixing the first 30 points yields less improvement in this example because the known pattern does not recur in the remainder of the time series. This yields a limitation of using fixed, continuous points as auxiliary knowledge in such cases.



(a) m with d_E. (b) m with d_Z. (c) n with d_E. (d) n with d_Z.

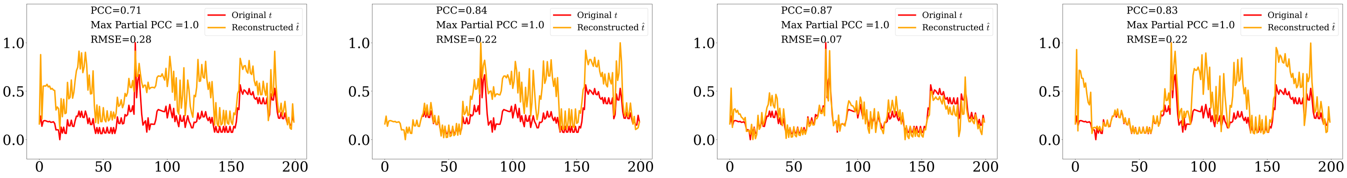
Figure 12: Results for parameters m and n with different distance measures.



(a) Domain reduction of an energy consumption time series of size 1440 over 6 values.

(b) *IMPI* ran over a digitized energy time series Matrix Profile with Manhattan distance and $m = 10$, in 43.076 seconds.

Figure 13: Integer Matrix Profile Inversion



(a) No auxiliary knowledge. (b) $\mathcal{A}_{subseq}$ (c) $\mathcal{A}_{sum}$ (d) $\mathcal{A}_{sparse}$

Figure 14: Results of reconstruction with auxiliary knowledge (using d_Z).