Privacy Preservation for Publishing Medical Time Series: *k*-anonymization of Ngram

Mohammad-Reza Pajoohan[1]*

RECEIVED: 2015/10/9 ACCEPTED: 2016/2/6

Abstract

Healthcare providers may need to publish their operational data for consultation as well as to allow more researches. Consequently, a lot of personal specific data with high level of details are publicly available. This data may contain time series, such as ECG. De-identification of time series is not enough to provide the requirement of privacy preservation. It is because, if a few numbers of time series are published, then appearing specific anomalies in them may reveal the sensitive information of an individual. The problem of privacy preserved time series publication is somewhat studied, but the issues of publishing the Ngrams of the time series, especially that of extracted from a small set of time series, are not considered well before.

In this paper, we address this problem and define the k-anonymity principle for the Ngram. The proposed schema aims to provide the k-anonymization by repeating the rare n-grams to hide them in the crowd of frequent n-grams. We evaluate our method by using two datasets. Results of experiments show that our method can provide the requested anonymity level with low probability and entropy information loss.

Keywords: **Privacy Preservation, k-anonymization, Medical Data, Time Series, Ngram.**

1. Introduction

Nowadays there are a lot of demands for publishing of personal and general data for applications such as data mining, decision support systems, fraud detection, health care researches and so forth. Many organizations and professionals want or need to publish their operational data in the hope of business visibility as well as to allow more researches. Consequently, a lot of personal specific data with high level of details are publicly available. This can jeopardize the privacy of individuals.

Data privacy and data utility are sides of a same coin, i.e. revealing more data values increases data utility, but it decreases data privacy. Hence, there is a need to find solutions that best address both utility and privacy requirements for data publication. This is what privacy preservation data publishing targets to publish modified data without compromising the privacy of individuals reflected in the data, meantime keeping details of data as much as possible. As a kind of important information, health information of people may be published for various purposes and in different styles. For example, the Telegraph revealed that medical records of all people whom serviced by NHS hospitals are sold to insurers [1]. In another case, El Emam et al. [2] investigated the case of privacy concerns in publishing the surveillance data by family physicians.

So far, the privacy concerns in relational databases are addressed using various solutions. Since other types of data also includes sensitive information, publishing them raises privacy concerns. For example, suppose that an individual shares its genome with a third party health institute to determine the genetic relatedness; the person would like to ensure that his/her genetic information would not be published or misused [3]. In a similar empirical case, a physician would like to share the ECG of a patient with colleagues for consultation purposes; then he need to be concerned regarding the disclosure of the identity of the patient. Gene sequences and medical time series are types of data other than relational data that must be taken into account. Giving the original time series to the third parties can be jeopardized. Time series data such as EEG and ECG, are very sensitive since they contains confidential information of individuals. Therefore, data owners do not prefer to publish the original time series and they need to modify them before publishing as well.

Time series mining is utilized in various applications in information systems. Zhu et al. [4] enumerate five sensitive characteristics of the time series including, amplitude, average, peak and trough, trend, and periodicity. Revealing any of this information can threaten the business of the data owner. Privacy preserving schemas provide two major approaches for publishing and analyzing time series as, (1) perturbation, and (2) partitioning.

Perturbation techniques modify the time series by methods such as adding noise, compression and randomization to obtain a private time series. For example, Rastogi and Nath [5] introduced the PASTE framework which includes algorithms addressing differential privacy in distributed time series data. The PASTE contains FPA algorithm which perturbs the Discrete Fourier Transform (DFT) of query answers in a time series retrieval system and DLPA algorithm which adds noise to the perturbed time series.

1. Assistant Professor, School of Electrical and Computer Engineering, Yazd University, Yazd, Iran. mzare@yazd.ac.ir

The second approach called partitioning, as might guess from its name, divides the time series into multiple parts and distributes the shares between different data analyzing agents. Each analyzer has only access to its share. The results of distributed processes are passed to a data aggregator to achieve the final results. Privacy preservation under partitioning uses encryption over communications and secures aggregation methods simultaneously.

Fan and Xiong [6] proposed FAST, an adaptive system which dynamically samples the time series and aggregate the results under differential privacy. Joye and Libert in [7] proposed a privacy preserving schema which efficiently aggregates and evaluate the sum of user's private input. This method has no limitation on number of the users and the data volume.

Moon et al. in [8] consider the noise coming from piecewise aggregate approximation (PAA) and proposed a solution for randomization of the time series symbolized using PAA in order to provide the acceptable level of privacy in a space with distance order.

Shang et al [9] defined the (*k,P*)-anonymity for pattern preserving time series publishing for the first time. They mentioned the limitations of *k*-anonymity in pattern preserving time series retrieval which leads to pattern loss. As the pattern loss problem would be the most important defect in sharing pattern-rich time series, thus their method publishes both time series patterns and feature values together but in different representation form. Shou et al. [10] continues the trend by proposing two algorithms to enforce (*k,P*)-anonymity on time-series data. This anonymity model customizes the data to publish a certain part of the values simultaneously with the publishing of a different part of the pattern of the anonymized time series.

Zhu et al. [11] defined three threat models regarding the level of trust between the data owner and the data analyzer. They, then, introduced three different privacy preserving schemas for each level of the trust. Accordingly, they proposed a method working by engaging in the discretization process. In this method, domain of time series are segmented into a set of ranges and the representative value for each range is selected to be used in the modified time series. Mapping a range of values into one representative value is similar to the idea of generalization for k-anonymization, thus this method perturbs the time series while preserving its trend as much as possible.

Privacy preserving publication of Ngrams is understudied while Ngrams are one of the most applicable models in the sequential predictive analysis. A sequence of symbols, as an output of symbolized time series, is segmented into subsequences of length less than or equal to n to generate an Ngram model. Since the time series is broken, their trends are not exploitable; thus, one may consider that privacy preservation of them is not necessary. However, as we will show later, some rare n-grams can reveal observations of sensitive trends despite of time series segmentation. Therefore, they should be considered and their leakage should be prevented.

As the only study addressing Ngram private publishing, to the best of our knowledge, Chen et al. [12] proposed the Differentially Private Sequential data publishing schema for variable-length NGrams. They utilized Ngrams extracted from databases of variable-length time series to support private publishing, while providing range query retrieval. Their method achieves the differential privacy by adding Laplace noise. Using variable–length Ngrams, they balance the trade-off between the added noise and sensitive information retained from the original time series. According to the evaluative experiments, implementing this method on databases with a large number of Ngrams leads to its ability to resist against Laplace noise coming from sanitization.

What we are addressing in this paper is different from all works mentioned above from privacy preservation point of view, even it is similar to Chen et al. [12] somehow. We consider a situation that data owner, for example a physician, wants to publish n-grams of time series to the third party for statistical analysis or predictive purposes, e.g. [13]. If the time series had been contained a discriminant characteristic, it is illustrated in the n-grams by the appearance of a rare pattern. This pattern can be an observation for a specific anomalous behavior in time series of an individual. Accordingly, data publisher should hide this rare pattern in the published n-grams. Otherwise, an adversary can identify the n-grams owner which is supposed to be preserved from disclosure.

Our proposed method preserves the privacy of a person but not in the same manner done by the schema which hides the presence of an individual. This method hides the observations enabling the adversaries to reveal sensitive information of a patient. It does the task by hiding a rare pattern in a crowd of patterns. Subsequently, this task is an association privacy preserving instead of a presence privacy preservation task. This method aims to overcome privacy leakage by modifying the n-grams to conform the defined privacy principle level (say *k* of *k*-anonymity principle). We increase the frequency of rare patterns to become at least k by adapting the *k*-anonymization principle. We do the process in such a manner to keep infor-

mation loss as low as possible.

An important question may be raised here, that is "why Ngram k-anonymization is critical for medical applications?" Applications such as medical monitoring, financial analysis, as well as traffic monitoring have potential to violate the privacy of individuals and anonymization of these time series is a key problem. However, there is a distinctive property in medical application distinguishing it from others, which is the volume of data. In most of above-mentioned applications, the data owner publishes a massive amount of data for further analysis rather than a small set. Thus, hiding the identity of an individual with a distinguishing behavior is not a challenge in them. Nevertheless, in medical applications, data may not be as massive as others. For example, only a few numbers of patients may have ventricular tachycardia in a hospital. Therefore, there exists a greater chance for a patient with a special ventricular tachycardia to be identified. Therefore, Ngram datasets in medical data mining applications are usually small and specific; and the potential of revealing rare patters is high. Hence, publishing of medical Ngrams is mostly benefited from k-anonymization.

The rest of this paper is organized as follows. We illustrate our contributions in Section 2. Then, we briefly introduce symbolized time series and their Ngram in Section 3. In Section 4, notions of Ngram anonymization principle are introduced and it continues with their definitions. This section is finished with a distinctive example, which highlights differences between proposed method and the naïve method. Details of the algorithm and steps of k-anonymization schema are discussed in Section 5. Section 6 includes evaluative experiments and comprehensive discussion on the results. Finally, section 7 concludes the paper with remarks on extensions in future.

2. Contributions

The major contributions of this paper are as follows:

(1) We propose the notion of k-anonymity for the Ngram model of time series. To the best of our knowledge this is the first time one targets constant-length Ngram model anonymization instead of that of for the time series itself. As n-grams are required to be published for some applications such as predictive analysis, we believe that their publication raises privacy concerns.

(2) Our model targets anonymization of a few number of time series to be published. The most similar work to ours is the schema suggested by Chen et al. [12] which aims private publication of Ngrams in a database of variable length time series. However, our method is able to publish an Ngrams of a few number of time series (even one) protecting the privacy of its individual. Consequently, it can protect the privacy from threats posed by occurring local as well as global anomalies.

(3) Although anomalies include rare patterns and they need to become hidden, our method is not involved in complex process of anomaly detection. Then, our method is easy to implement, has low complexity and give low information loss.

3. Preliminaries

3.1 Symbolic Representation of Time Series

Time-related sequential data are seen in different kinds of data mining and analysis applications. These applications range from medical time series, e.g. ECG and EEG to stock prices and meteorological data. Furthermore, various instances of log data analysis in modern web based systems are of this kind. Time series similar to other types of data are stored, indexed and analyzed in data mining frameworks. Most of all time series are high dimensional data and consequently dimensionality reduction would be unavoidable for storing them. An ultimate solution to this problem is to obtain a reduced representation that uses fewer number of distinct values while shorten the length of the sequence.

These representation methods reduce the needed storage space by decreasing the number of bits used for encoding the values; furthermore, some of these methods reduce the length of the underlying data by replacing symbols/values with a window of time series samples.

3.2. Time Series Ngram

Probabilistic models, e.g. Markov chain and Hidden Markov Model, have been utilized so far in predicting future symbols of sequences in natural language processing and biological sciences. Such stochastic models predict future symbols of the sequence regarding a short history of its past symbols. Markov processes make use of the Markov property which states that if a sequence is time invariant, then the probability of a symbol depends only on the occurred symbols in preceding time. One of prediction models that are used under the Markov assumption is Ngram model. This model is one of the most widely used models for sequential prediction used in computational linguistics and gene processing. The value n in Ngram model, called order, determines its memory size. Ngram model, as a Markov model of order n, predicts the n^{th} symbol regarding the last (n-1) symbols of the sequence.

Ngram model of higher order adopts longer se-

quence of symbols to achieve high accuracy. Since the higher order Ngram model predicts the future more accurately, it could be an empirical challenge to provide and store an enough large set of subsequences of length n to train the model. Furthermore, this model needs to store all subsequences of length *n, (n-1)…, 2, 1* and their frequencies. For a time series represented by p symbols, the total number of happened subsequences with length less than or equal to n is,

$$\sum_{i=1}^{n} p^i \qquad (1)$$

Consequently, the Ngram model includes all occurred subsequences with their frequencies called Ngram table. This table can be exploited for training by predictive models, and then publishing this table is useful in many applications, e.g. [14]–[17]. As we are targeting the privacy preserved publishing of Ngram table, then we continue with the basic notation utilized in the definition of this kind of data.

3.3 Notations

Before formalizing the problem of publishing *k*-anonymous Ngrams, we give the basic notations and definitions.

Definition 1 (Symbolized Time Series).
Let there is a set of p symbols, denoted by $\sum$. A symbolic representation of a time series T is a sequence of symbols of length l as $T = \langle s_1, s_2, ..., s_l \rangle$, where $s_i \in \Sigma$ is generated by a symbolization preprocess.

Definition 2 (n-gram).
Let n be a positive integer number. The n-gram is an ordered set of n symbols, as $G_n = \langle s_1, s_2, ..., s_n \rangle$ and $|G_n| = n$.

Definition 3 (n-gram Frequency).
Let G_n be an n-gram. The number of occurrences of the G_n through the symbolized time series, T, is its frequency and denoted by $f(G_n)$.

Definition 4 (Ngram Table)
Ngram table is a table which is generated by union of all grams with length less than or equal to n. That is $G_n \bigcup G_{n-1} \bigcup ... \bigcup G_1$.

Definition 5 (k-anonymous Ngram Table).
Given an integer k ($k>1$), an Ngram table extracted from T is called k-anonymous Ngram table if it is impossible to find an n-gram G_n such that $f(G_n) < k$. Subsequently, G_n called non-k-anonymous if $f(G_n) < k$.

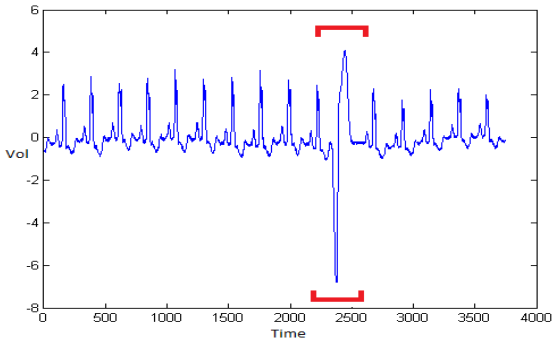

Fig. 1. Plot of ECG "chfdb_chf01_275" from [18] Dataset. The red-indicated region denotes the anomaly in the ECG.

4 .Time series *k*-anonymization

4.1 The Idea

In many applications such as computational linguistics, Ngrams are extracted from a large collection of data. These collections are large enough to avoid disclosure of outliers and rare patterns, but, ***what if a data owner decide to publish a specific small set of sequences, say symbolized ECGs of a few patients***?

For example, suppose that a care provider decides to share Ngrams of a few patients. Clearly, the rare patterns can disclose the identity or presence of an individual. To the best of our knowledge, in most of the studies on privacy preserving Ngram publishing, e.g. [16], [17], Ngrams are generated using large collections of sequences. In contrast to them, we aimed to provide a privacy preserving framework for publishing the Ngram table of a limited set of time series, such as ECGs of some specific patients. In this case, it makes sense to anonymize the Ngrams of the time series by hiding the rare patterns. Now, the question is ***"Is there any schema that can hide Ngrams of an anomaly in the Ngram table?"***

A naïve method for hiding an anomaly can be done by copying and appending it to the time series. By copying an anomaly one time, frequencies of its Ngrams are increased by one. This step can be repeated until all Ngrams of the anomaly meet the anonymity requirement, say k. This approach is very simple but has two disadvantages. First, it requires an anomaly detection step before anonymization. Second, it causes high information loss.

According to [19] most of the anomaly detection methods limit their task to specific problem formulation. Type of anomalies, application domain, the availability of labeled data and output type are major factors in limiting the domain of anomaly detection. Thus, the anomaly detection itself, with all these limitations, is the main challenge of the naïve method. In addition to high cost of anomaly detection, this method causes high information loss. Obviously, the larger the k value, the more the information loss.

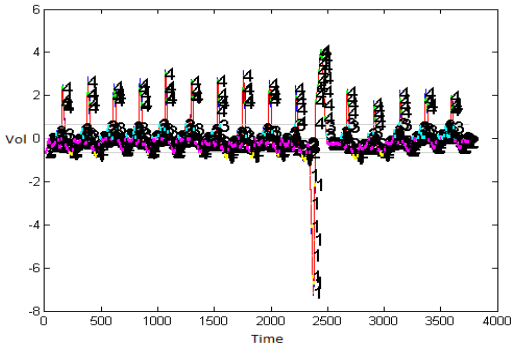

Fig. 2. Plot of ECG "chfdb_chf01_275" from [18] Dataset. This time series is symbolized with SAX symbolization schema.

In this section, we illustrate the idea behind our proposed algorithm using a brief example. Suppose that an ECG time series must be published k-anonymously. Given ECG is illustrated in Fig. 1.

The indicated subsequence of the ECG is an anomaly. By symbolizing this ECG using 4 symbols, denoted by 1, 2, 3, and 4, we get symbolized representation as in Fig. 2.

In the anomalous region, the time series has a sub

sequence of ⟨1 1 1 … 1⟩ and ⟨4 4 4 … 4⟩. For example, the 5-gram of ⟨1 1 1 1 1⟩ is an instance of rare Ngrams which is not repeated in any other region of the sequence. This 5-gram can reveal the presence of a patient with a specific ECG anomaly.

In naïve method, an anomaly is detected and repeated to provide anonymity for rare Ngrams. Intuitively, 5-gram ⟨4 4 4 4 4⟩ does not need to be repeated, because it is appeared in the periodic region of the sequence as well as in anomalous regions. Now the important question is that ***which Ngrams should be repeated to provide k-anonymity?***

While naïve method finds the anomaly and increases the frequency of all of its n-grams by repeating it, a better solution is to repeat only rare n-grams. Then the k-anonymization schema does not need to detect the anomalies. Instead, only rare n-grams can be selected, which is obtained by simply comparing their frequencies with k. Besides, it leads to lower information loss as only some n-grams of anomaly are repeated instead of that of the whole anomaly.

In Fig. 2, n-grams that are common between the anomaly region and the remaining of the sequence, e.g. ⟨4 4 4 4 4 4⟩ have been repeated more than Ngrams occurred only in the anomaly region, i.e. ⟨1 1 1 1 1 1⟩. Therefore, only rare n-gram, ⟨1 1 1 1 1 1⟩, is repeated. Consequently, the anomaly detection preprocess is omitted and the information loss in this schema is lower as well.

Although, just adding frequency of rare n-grams

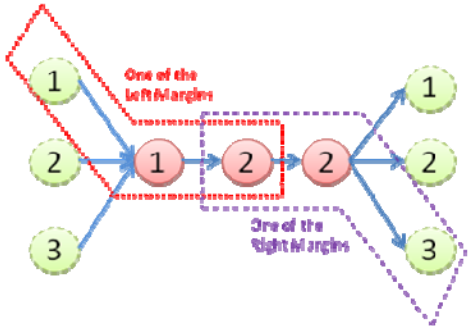

Fig. 3. One of the left margins as well as one of the right margins of the n-gram ⟨1 2 2⟩.

achieves the requirement of the k-anonymization with the lowest possible information loss, it is not applicable as it destroys the sequentiallity of the time series. This is happened because it injects an n-gram into the sequence without considering its neighbors. Therefore, we need to take the neighbors of the rare n-grams into consideration. Example 1 will illustrate this issue.

Example (1): Suppose that symbolic representation of a time series is as follows:

T= ⟨1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 1 1 1 1 1 2 3 4 1 2 3 4⟩

We call the ⟨1 1 1 1 1 1⟩ subsequence the anomalous window which takes place at 17^{th} place and continues to 22^{nd} place of this sequence.

If we set $n=5$ then this region contains ten 5-grams, starting from 4 symbols before the anomalous window and lasts to 4 symbols after that, as follows,

⟨1 2 3 4 1⟩ , ⟨2 3 4 1 1⟩ , ⟨3 4 1 1 1⟩ , ⟨4 1 1 1 1⟩ , ⟨1 1 1 1 1⟩ , ⟨1 1 1 1 1⟩ , ⟨1 1 1 1 2⟩ , ⟨1 1 1 2 3⟩ , ⟨1 1 2 3 4⟩, ⟨1 2 3 4 1⟩

The first n-gram which has at least one common symbol with the anomalous window is occurred in *(n-1)* place before the anomalous window. Analogously, the last n-gram with common symbol is placed at *(n-1)* symbols after the window. Thus, an anomaly of length l is included in a window of length $((n-1)+l+(n-1))$. In the Ngram model, $(l+n-1)$ n-grams can be extracted from a window of length $(2n+l-2)$. Some of the extracted n-grams might be repeated in regions other than anomalous window. This common n-grams are typically placed on boundaries, e.g. ⟨1 *2 3 4 1*⟩ in Example (1). These n-grams may be frequent and need not to be anonymized. The rare n-grams which are usually placed at the center of the anomalous window must be repeated to provide the k-anonymization requirement.

As can be seen in the Example (1), n-grams in windows of length $(n-1)$ at both ends of the anomalous window have common subsequences with this window. We call such n-grams the neighbor n-grams. Since neighbor n-grams have common subsequences,

if an n-gram is repeated, then the frequency of its neighbors must be incremented as well. Consequently, neighbor n-grams of the anomalous window must be considered in anonymization schema.

If the anomaly has been detected, generating the neighbor n-grams would be easy; the neighbors are formed by sliding a window through the time series. But, what if the time series is already broken into n-grams?

Given set of n-grams, the neighbor n-grams should be constructed by estimation.

Example (2): Suppose, ∑={1,2,3,4} and let g=⟨ 2 4 1 1⟩ be a rare 4-gram (probably part of an anomalous window). The 4-gram g has eight 4-gram neighbors. Four neighbors are located before the g and four of them after the g, as follows (unknown symbols are marked by question mark).

{ ⟨? ? ? 2⟩ , ⟨? ? 2 4⟩ , ⟨? 2 4 1⟩ , ⟨2 4 1 1⟩ , ⟨4 1 1 ? ⟩ , ⟨1 1 ? ? ⟩ , ⟨1 ? ? ? ⟩}

Each unknown symbol can be replaced by one symbol from ∑. For example, ⟨? 2 4 1⟩ can be any of the following 4-grams:

{ ⟨1 2 4 1⟩ , ⟨2 2 4 1⟩ , ⟨3 2 4 1⟩ , ⟨4 2 4 1⟩}

Therefore, the frequencies of these 4-grams should be incremented. However, they should be incremented with different values as the probability of each neighbor is different; that is, they are not equally likely to be replaced. Therefore, the frequencies of neighbors must be incremented proportional to the probability of placing at the neighborhood of the n-gram. Suppose that the frequencies of neighbors in Example (2) are 2, 10, 5, 3, respectively, then the second neighbor is two times more likely than the third one, for example.

Recall that Ngram model is a Markov model of order *n* and its Ngram table includes all subsequences of length *n, (n-1), (n-2),…, 1*. Consequently, any increments in the number of occurrences of n-grams affect the frequencies of all of their subsequences in lower level of Ngram table. An n-gram have 2 subsequences of length *(n-1)*, called *(n-1)*-gram, 3 subsequences of length *(n-2)*, 4 subsequences of length *(n-3)* and so forth. Therefore, each n-gram has

$$\sum_{i=2}^{n} i = \frac{n^2+n}{2} - 1 \qquad (2)$$

subsequences. Thus, our proposed method needs to take all these subsequences of the n-grams into account.

In sum, in this paper, we propose a schema to anonymize the Ngram table of a symbolized time series and as we will see, it takes into consideration the neighbors of n-grams and their subsequences as well.

4.2 Definitions for Time series k-anonymization

In this section, we start with the basic definitions of our method. Let *T* be a symbolized time series and *k* be the anonymization parameter. The proposed *k*-anonymization method increments the frequency of rare n-grams to achieve a *k*-anonymous Ngram table in which all n-grams satisfy *k*-anonymity requirement.

To anonymize an n-gram, its frequency must be incremented and this leads to incrementing the frequency of its neighbors. We can adopt two different approaches respecting the effect of neighbors, (1) ignoring the neighbors and (2) considering all possible neighbors in a chain manner.

Considering all possible neighbors requires forming a tree with the branching factor of p (|Σ|) to generate neighbors. In such schema, the farther neighbors receive a small portion of the added values. The small values, added to the neighbors in the second and third levels of the tree, do not provide benefits to the k-anonymizing model; moreover, they increase the information loss and computational complexity of the model. On the other hand, ignoring the neighbors is not reasonable. In this way, limiting the window of neighborhood by a constraint is a compromise between these two extreme options.

Without loss of generality, we limit the number of neighbors by two; one neighbor from each side. By this constraint, underlying n-gram and its neighbors have a common subsequence of length *(n-1)*. To distinguish neighbors in this narrower window from all possible neighbors, we define margins. As neighbors can be generated from both left and right sides of the n-grams, we define left and right margins for each n-gram as follows.

Definition 6 (n-gram Left Margin).

Let $G_n = \langle s_1, s_2, \ldots s_n \rangle$ (n≥2) be an n-gram. Left margin of G_n denoted by G_n^- is $\langle x, s_1, \ldots s_{n-1} \rangle$, where x∈Σ. Obviously, G_n^- is a set of subsequences where all of the members have same last (n-1) symbols.

Definition 7 (n-gram Right Margin).

Let $G_n = \langle s_1, s_2, \ldots s_n \rangle$ (n≥2) be an n-gram. Right margin of G_n denoted by G_n^+ is $\langle s_2, s_3 \ldots s_{n-1}, y \rangle$, where y∈Σ. Similarly, G_n^+ is a set of subsequences where all of the members have same first (*n-1*) symbols.

The rare n-grams and their left and right margins form a set of sequences of length *n*, called candidate set. Fig. 3 illustrates one left margin of an n-gram as well as one of its right margins.

Definition 8 (n-gram Candidate set).

Let $N = \{G_n^1, G_n^2, \ldots, G_n^r\}$ be set of non-anonymous n-

grams, where $f(G_n^i) < k$. Suppose that $L = \{G_n^{-1}, G_n^{-2}, ..., G_n^{-r}\}$ be the set of left margins of the members of N and $R = \{G_n^{+1}, G_n^{+2}, ..., G_n^{+r}\}$ be that of for right margins, where G_n^- and G_n^+ are sets of left and right neighbors for G_n, respectively.

The set of n-grams that their frequencies are needed to be incremented is defined by $I = N \bigcup L \bigcup R$ as candidate n-grams. All members of *I* are n-grams and they have a number of subsequences of length *(n-1), (n-2) ,…1*. We define subsequences of an n-gram and its margins namely core *m*-subgram and margin m-subgram, respectively. We need to consider frequency of these subsequences.

Definition 9 (Core m-Subgram).

Let $G_n = \langle s_1, s_2, ... s_n \rangle$ be an Ngram. A set of sequences C_n^m is called m-subgram of n-gram, if it includes all subsequences of length m (n>m≥1), i.e.

$$C_n^m = \bigcup_{i=1}^{n-m+1} \langle s_i, s_{i+1}, ..., s_{i+m-1} \rangle$$

Definition 10 (Margin m-Subgram).

Let $G_n^- = \langle x, s_1, ... s_{n-1} \rangle$ be the left margin of G_n and $G_n^= = \langle s_2, s_3 ... s_{n-1}, y \rangle$ be the right margin of G_n. Margin m-subgram is a subgram of margins if it includes unknown symbol (that is, x or y) placed at the boundaries. Therefore, M_n^m is defined as follows:

$$M_n^m = \{\langle x, ..., \ s_{m-1} \rangle\} \bigcup \{\langle s_{n-m+2}, ..., y \rangle\}$$

For example, suppose g=⟨2 4 1 1⟩. It has two core 3-subgrams, {⟨2 4 1⟩ , ⟨4 1 1⟩}, three core 2-subgram, {⟨2 4⟩ , ⟨4 1⟩ , ⟨1 1⟩}, and four core 1-subgram, {⟨2⟩ , ⟨4⟩ , ⟨1⟩ , ⟨1⟩}. The *g* also has g^+ and g^- margins as ⟨x 2 4 1⟩ and ⟨4 1 1 y⟩, then margin subgrams are ⟨x⟩ , ⟨x 2⟩ , ⟨x 2 4⟩ , ⟨1 1 y⟩ , ⟨1 y⟩ and ⟨y⟩.

When an n-gram is repeated *t* times, its frequency is incremented exactly by *t* and frequencies of its core subgrams are incremented by *t* as well, because core subgrams tend to be embedded in the n-gram. As noted before, the added value for margins is computed by taking into account of their probabilities, thus frequencies of margins of the mentioned n-grams are increased by a fraction of *t*. Analogously, the frequencies of margin subgrams are incremented by the same value which is added to the frequency of margins. We explain the details of the algorithm using a distinctive example in the next section.

4.3 Distinctive Example

We explain the differences of our method and Ngram-repeating naïve method through the use of a distinctive example.

Table 1. Some of n-grams repeated in anomaly region which is added to the end of the time series t.

5-gram	*f*	5-gram	*f*
⟨2 3 4 5 ***4***⟩	1	⟨***4 4 4 5 4***⟩	3
⟨3 4 5 ***4 4***⟩	1	⟨***4 4 5 4 4***⟩	4
⟨4 5 ***4 4 5***⟩	1	⟨***4 5 4 4 4***⟩	3
⟨5 ***4 4 5 4*** ⟩	1	⟨***5 4 4 4 4***⟩	3
⟨***4 4 4 4 5***⟩	3		

Example (3): Let ∑={1,2,3,4,5} and the order of Ngram model is 5. Suppose the symbolized time series, i.e. *T*, is as follows[2]:

T=⟨ 1 2 3 4 5 ,1 2 3 4 5 ,1 2 3 4 5, 1 2 3 4 5, **4 4 5 4 4**, 1 2 3 4 5, 1 2 3 4 5⟩, by an anomaly pattern <4 4 5 4 4>.

By adopting the naïve method, a modified time series is generated by copying and appending the anomaly region to time series, say at the end of *T*, obtaining time series *TT*, as follows[3]:

TT=⟨1 2 3 4 5, 1 2 3 4 5,1 2 3 4 5, 1 2 3 4 5, **4 4 5 4 4**, 1 2 3 4 5, 1 2 3 4 5, ***4 4 5 4 4 4 4 5 4 4 4 5 4 4 4 4 5 4 4***⟩

As we already mentioned, this method has two disadvantages, (1) it requires the pre-process of anomaly detection which is a challenging task, (2) it destroys the sequentiallity of the time series, resulting to high information loss. Moreover, it cannot increase frequencies of some n-grams placed at the boundaries of the anomaly region. We continue the example to show this. The anomaly window contains a subsequence of length 5. The time series from *(n-1)* symbols before the anomaly to (n-1) symbols after it is as follows:

⟨2 3 4 5 **4 4 5 4 4** 1 2 3 4⟩

This region includes 5-grams such as,

⟨2 3 4 5 **4**⟩ , ⟨3 4 5 **4 4**⟩ , ⟨4 5 **4 4 5**⟩ , ⟨5 **4 4 5 4**⟩, ⟨**4 4 5 4 4**⟩, ⟨**4 5 4 4** 1⟩, ⟨**5 4 4** 1 2⟩ , ⟨**4 4** 1 2 3⟩, ⟨**4** 1 2 3 4⟩

and for all *f=1*; all of the mentioned 5-grams are occurred just one time in the Ngram table. Ngram table of the time series TT (shown in Table 1.), includes following 5-grams in addition to 5-grams occurred in T.

As can be seen, although 5-grams placed at the middle of the anomalous window are repeated, the 5-grams placed at the both boundaries, such as ⟨3 4 5 4 4⟩, are still have low frequency. It may seem to widen the sliding window is a solution to include the neighbors. But, we cannot do this until no more low-frequency n-grams can be found. Thus, the naïve method cannot fully anonymize the Ngram table of the

[2]. Symbols in anomaly region are indicated in bold typeface.
[3]. Repeated instances of anomaly region are denoted by bold-Italic type face.

time series at all.

In contrast to the naïve method, our proposed schema does not require the original time series and to detect its anomalies. Instead, it is inputted with Ngram table and tries to anonymize the Ngrams independently. Under a basic assumption, it tends to anonymize all rare n-grams regardless of their source. In our proposed method, an anonymity level parameter, namely k, is defined to evaluate the rareness of the n-grams. According to the definition, all n-grams having a count less than k should be anonymized. Therefore, it finds the non-anonymous n-grams and repeats them until satisfying the requirement.

To illustrate what the proposed algorithm does, we add the anonymity level of $k=3$, as a constraint, to the last example. Accordingly, all n-grams with $f \leq 3$ are rare and consequently not anonymized. It should be noted that n-grams such as ⟨5 4 4 4 4⟩ with higher frequencies are not considered as non-anonymous n-grams and are not required to be repeated even it comes from the anomalous region. Since fewer numbers of n-grams may not satisfy the anonymity requirement, our decision to leave out the majority of n-grams occurred in the anomalous region as well as in the periodic region decreases the information loss.

The proposed method picks the non-anonymous n-grams one by one and increases their frequencies by the difference between their count and the anonymity level parameter, i.e. k, to achieve the baseline requirement. Then, it finds their left and right margins by replacing the first and the last symbols, respectively. For example, the non-anonymous n-gram g=⟨2 3 4 5 **4**⟩, where f=1, is repeated two times to be anonymized; consequently, its frequency is incremented by 2. The n-gram g, has five left margins[4]:

{⟨**1** 2 3 4 5⟩, ⟨**2** 2 3 4 5⟩, ⟨**3** 2 3 4 5⟩,⟨**4** 2 3 4 5⟩, ⟨**5** 2 3 4 5⟩}

, as well as five right margins as follows:

{⟨3 4 5 4 **1**⟩, ⟨3 4 5 4 **2**⟩, ⟨3 4 5 4 **3**⟩,⟨3 4 5 4 **4**⟩ , ⟨3 4 5 4 **5**⟩}

Frequency of these margin neighbors are increased by a fraction of 2, regarding their probabilities. Suppose that neighborhood probabilities for the left margins are 0.1, 0.2, 0.45, 0.12 and 0.13, respectively. Then their frequencies are increased by 0.1∗2, 0.2∗2, 0.45∗2, 0.12∗2, and 0.13∗2 accordingly. The process of selecting n-grams and repeating them continues until no more non-anonymous n-gram can be found. Clearly, some n-grams are appeared or generated more than one time.

4. Unknown symbols of margins replaced by members of symbol set are marked in bold-underline type face.

Algorithm 1: *k-Anonymization of Frequency Matrix of Ngrams.*

Input:

Ngram Table : $G_n \cup G_{n-1} \cup \ldots \cup G_1$

Anonymization level: K

Output:

Updated Ngram Matrix: $G'_n, G'_{n-1}, \ldots, G'_2$

Function Anonymization

```
UList ← Find (G_n < k);
While not (is_Empty(Updates))
    i ← UList [1,1] ;
    %index of i^th non-anonymized Ngram
    G ← G_n[i, 1:n] ;
    % sequence of non-anonymized Ngram
    f ← G_n[i,n+1] ;
    % freq. of the non-anonymized Ngram
    G'_n[i,n+1] ← k ;
    diff ← k-f ;
    Updates_Ngram (G_n , diff , n , i ) ;
    UList ← Find(G'_n<k) ;
End
End
```

Fig. 4. The k-anonymization algorithm which inputted with Ngram table. This algorithm collects non-k-anonymous n-grams and updates their frequencies in all levels less than n.

5 . Proposed Algorithm

Now, we explain our proposed algorithm for Ngram anonymization. As noted before, k-anonymity of an Ngram model would be guaranteed if all of the n-grams are repeated at least k times. The basic idea behind the proposed algorithm is injecting more numbers of rare n-grams to non-anonymous Ngram model to increase the frequency of n-grams with $f(G_n) < k$. As a consequence, the frequency of neighbors and subgrams of the non-anonymous n-grams should be incremented as well. However, neighbors and non-anonymous n-grams are not treated the same.

This algorithm has six steps, as follows:

1. Input time series are symbolized.
2. Ngram table of the time series are generated.
3. Non-anonymous n-grams (i.e. N) are determined.
4. Margin n-grams (i.e. $L \cup R$) of non-anonymous n-grams are generated.
5. Core and margin subgrams of the candidate set (i.e. $N \cup L \cup R$) are extracted.
6. The frequencies of n-grams and their subgrams are incremented.

Pseudo code of the algorithm has been illustrated in Fig. 4 and 5 showing main parts of the steps, brought from [20].

5.1 Pre-Processing (Steps 1 and 2)

In this paper, we use SAX representation to symbolize the time series using p symbols just same as what introduced in [18]. After symbolizing the time series, a window slides through the time series. Started from

Algorithm 2: Updating Frequency Matrices of Subgrams of Parent Ngram.

Input:
Matrices of Ngrams: $G_n, G_{n-1}, \ldots, G_1$
Symbol Set: S
Parent Ngram: G_n
Frequency Difference of Parent Ngram: *diff*
Level of Ngram Model: n
Index of Ngram: j

Output:
Updated Ngram Matrices: $G'_n, G'_{n-1}, \ldots, G'_2$

```
Function Updates_Ngram
    m ← n-1 ;
    Left  ← Left_Margin(G_n,S) ;
    Right ← Right_Margin(G_n,S) ;
    if (n>1)
        Core_S ← Core_Subgram(G_n,m) ;
        For i ← 1 to Core_S.Length
            G'_n[i , n] ← G_n[i , n]+diff ;
            GC=Core_Subgram(Core_S[i,:] ,m ) ;
            Updates_Ngram (GC , diff , n-1 , i ) ;
        End
    End
    For i ← 1 to Left.Length
        p ← (G_n[i,n])/(G_n-1[j,n])
        G'_n [i , n] ← G_n[i , n]+diff *p;
        if (m>1)
            GL=Margin_Subgram (Left[i,:] ,m ) ;
            Updates_Ngram (GL , diff *p , m , i ) ;
        End
    End
    For i ← 1 to Right.Length
        p ← (G_n[i,n])/(G_n-1[j,n]);
        G'_n [i , n] ← G_n [i , n]+diff *p ;
        if (m>1)
            GR=Margin_Subgram (Right[i,:] ,m ) ;
            Updates_Ngram (GR , diff *p , m , i ) ;
        End
    End
End
```

Fig. 5. The Updates_Ngram function updates the frequency of left margins of an n-gram and that of right as well as their core and margin subgrams.

the first symbol, a subsequence of length n which is placed in the window, namely n-gram, is outputted and the process is continued until reaching the $(n-1)$ symbols before the last symbol. The unique instances of n-grams are counted and added to the Ngram table. As the Ngram model includes all occurred subsequences of the length less than n, then the length of the window is reduced one by one and the process is repeated to find all occurred subsequences. The Ngram table is filled with the discovered subsequences and their frequencies.

5.2 Generating Candidate Set (Steps 3, 4 and 5)

Ngram model anonymization is done in different levels, from 1 to n. In the first level candidate n-grams must be selected and anonymized. Candidate set contains three sets of n-grams, i.e. non-k-anonymous n-grams and that of left margins as well as right margins. According to the definition of the anonymity in the Ngram model, all of n-grams which are not occurred at least k times are not k-anonymous. Thus, non-k-anonymous n-grams are easily determined by comparing the frequencies with anonymization parameter k. The non-k-anonymous n-grams are added to the set N and then for each member of the N, left and right margins are generated using a tree-structured method. In this method for each n-gram denoted by G_n, the margins denoted by G_n^- and G_n^+ are generated by replacing the unknown symbols with each of the symbol set members to generate L and R sets. Thus, for each n-gram, we have p left margins and p right margins. Fig. 6 shows the process of generating margins for two non-k-anonymous n-grams of a time series, where $\sum=\{1,2,3,4\}$.

This process continues until margins of all of the non-k-anonymous n-grams are generated. Margins and their ancestor n-grams are same in $(n-1)$ symbols. Clearly, some of margins are repeated more than one because they can be considered as left or right margins of some other n-grams. For example, ⟨1 2 3⟩ can be constructed by replacing x in ⟨x 2 3⟩ or y in ⟨1 2 y⟩. In addition, the ⟨x 2 3⟩ can be left margin extracted from non-anonymous ⟨2 2 3⟩ or n-gram ⟨4 2 3⟩.

It may be asked ***"Is it necessary to consider a margin more than one time?"*** The answer is yes—we repeat the margins to take into consideration the effect of repeating an n-gram. Meanwhile, the occurrence of one n-gram is independent of the occurrence of others, thus we repeat margins of each n-grams independent of the other occurrences of it, as margins of other n-grams.

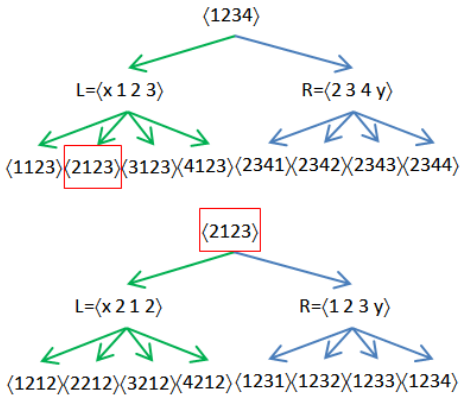

Fig. 6. Two n-grams ⟨1 2 3 4⟩ and ⟨2 1 2 3⟩ are selected and their left and right margins are generated. As can be seen, the second left margin of ⟨1 2 3 4⟩ is ⟨2 1 2 3⟩ which is a non-k-anonymous n-gram required to be anonymized.

5.3 Anonymization of the candidate set (Step 6)

As the next step and after extracting sets L and R, we would like to compute the new frequency values. For each non-anonymous n-gram, where $f(G_n) < k$ we compute the difference between the k and the $f(G_n)$, as follows:

$$d(G_n) = k - f(G_n) \quad (3)$$

The difference, denoted by d, determines the value that should be added to the frequency of the n-gram. To compute the proportion of d to be added to the frequencies of the margins, we estimate their probabili-

Table 2. Detail information of time series of Discord Detection Dataset [18] (Dataset I).

Series	File Name	Application Area	Count of Samples
data1	chfdb_chf01_275	ECG	3751
data2	chfdb_chf13_45590	ECG	3750
data3	chfdbchf15	ECG	15000
data4	ltstdb_20221_43	ECG	3750
data5	ltstdb_20321_240	ECG	3750
data6	mitdb__100_180	ECG	5401
data7	mitdbx_mitdbx_108	ECG	21600
data8	nprs43	Respiration	18052
data9	nprs44	Respiration	24125
data10	power_data	Power Demand	35040
data11	qtdbsel102	ECG	45000
data12	qtdbsele0606	ECG	15000
data13	stdb_308_0	ECG	5400
data14	TEK14	Space Shuttle	5000
data15	TEK16	Space Shuttle	5000
data16	TEK17	Space Shuttle	5000
data17	xmitdb_x108_0	ECG	5400

ties. Every n-gram has left and right margins, G_n^- and G_n^+, which are sets of p different n-grams denoted by $\{g_1^-, g_2^-, ..., g_p^-\}$ and $\{g_1^+, g_2^+, ..., g_p^+\}$, respectively. The probability of being chosen for each of right margins is computed as,

$$P(g_i^+) = \frac{f(g_i^+)}{\sum_{i=1}^{n} f(g_i^+)} \quad (4)$$

Where $f(g_i^+)$ is the frequency of the right margin which is generated by replacing the unknown symbol with the i^{th} symbol of the symbol set. The value of $p(g_i^-)$, for left margins, can be computed in a similar manner.

Obviously, the sum of all probabilities above is 1 and consequently, the sum of the values that are added to the frequencies of the margins are d as well. Intuitively, when an n-gram repeated in a time series, it has only one definitive neighbor with same (*n-1*) symbols at its right end and one at its left end that are repeated one. When we use Ngram table to reconstruct the margins, we divide the added value, i.e. 1, to small fractions between margins. Thus, it ensures that values added to the frequencies in our method are not more than that of the naïve method.

For each non-anonymous n-gram G_n, the value of $d(G_n)$ is added to its frequency and for each of its left and margins denoted by g_i^- and g_i^+, $d(G_n) \times p(g_i^-)$ and $d(G_n) \times p(g_i^+)$ are computed and added to the frequency of them, respectively.

Table 3. Detail information of 10 time series of length 40-seconds selected from different start points through the part-106 ECG of MIT-BIH Arrhythmia Dataset [21] (Dataset II).

#	Start point – End point (Sec)	1st Anomaly location (Sec)	Count of Samples
T1	80 –120	90.74	14400
T2	430 – 470	445.78	14400
T3	700 – 740	710.89	14400
T4	960 – 1000	965.98	14400
T5	1040 – 1080	1048.75	14400
T6	0 – 40	na	14400
T7	200 – 240	na	14400
T8	600 – 640	na	14400
T9	1320 – 1360	na	14400
T10	1380 – 1420	na	14400

Subgrams are generated by sliding window of length m through the n-gram. Core subgrams which are extracted from n-grams are repeated with the same number of their n-grams, thus the values of frequencies of the core subgrams are incremented by $d(G_n)$. Margin subgrams are treated the same as their ancestors. This means, if an n-gram as a right margin of G_n is likely to be occurred with the probability of $p(g_i^+)$ then its subgrams are likely to be occurred with the same probability. Then, we increment the frequencies of the right margin subgrams by the $d(G_n) \times p(g_i^+)$ and that of the left margins by $d(G_n) \times p(g_i^-)$.

6 Experiments

We run various experiments to evaluate the ability of our method in preserving the privacy of the individual in case of publishing a few numbers of time series. We also consider an n-gram sanitization method, that is [12], for comparative experiments; even though it is belonging to a different class of methods. In the next sections, the utilized datasets and details of the experiments will be discussed.

6.1 Experimental Setup

We conduct experiments using two datasets to evaluate the ability of our method. The first dataset is Discord Dataset [18] provided by Keogh (ref. Dataset I). The second dataset is a subset of MIT-BIH arrhythmia dataset [22] (ref. Dataset II). The dataset I includes 17 time series selected from various sources such as medical health records and the NASA Space Shuttle datasets and

Table 4. error and entropy of 3-grams and 2-grams of the Dataset [18] (Dataset I). provided level of anonymity for each record in different orders of Ngram Model are included as well.

#	2-gram APIL	3-gram APIL	K for 2-grams	K for 3-grams
data1	1.98	0.46	1	1
data2	1.81	0.47	1	1
data3	0.3	0.12	1	1
data4	0.72	0.33	1	1
data5	1.49	0.37	1	1
data6	1.7	0.25	1	1
data7	1.96	0.26	3	1
data8	1.56	0.31	5	1
data9	2.31	0.14	3	1
data10	2.05	0.33	2	1
data11	1.97	0.25	9	1
data12	0.76	0.28	1	1
data13	1.07	0.46	1	1
data14	1.72	0.44	1	1
data15	0.06	0.03	1	1
data16	0.73	0.36	1	1
data17	0.84	0.33	1	1

the second dataset contains selected ECG time series of the MIT-BIH dataset. Since anomalies and discords are parts of time series that are more likely to contain rare n-grams, we choose datasets that contain anomalies.

Keogh et al. [18] published the mentioned dataset as a baseline dataset for discord detection in time series[5]. Since each time series has at least one implicit anomalous pattern, it can be an ideal dataset for evaluating our proposed schema. Table 2 shows the detail information of time series of this dataset.

The second dataset includes 10 subsequences of part-106 ECG of the MIT-BIH dataset selected according to the references of Chuah and Fu [23]. They proposed an anomaly detection schema and accordingly they publish a list of anomalies has happened in part-106 of the MIT-BIH ECG. We use these 10 subsequences of part-106 to generate the second dataset. Half of the subsequences include an anomaly and the other half does not. We set the start and the end time of the intervals according to the indices referenced by Chuah and Fu in [23]. Table 3 shows the details of selected intervals.

Each time series is symbolized using SAX representation with window size of 10 samples. Thus, time series of length *c* are symbolized and converted to the time series of length c∗0.1. We fix the symbol set to {1, 2, 3, 4}. Order of the Ngram model for experiments is set to 3 for the sake of simplicity. Therefore, 3-grams are selected to be anonymized.

In order to show the ability of our method, we select the method proposed by Chen et al. [12], as the most similar method, to compare with ours. In addition, we investigate our method in case of probabilistic information loss and entropy information loss criteria to show the efficiency of our method.

Table 5: error and entropy of 3-grams and 2-grams of the MIT-BIH Arrhythmia Dataset [21] (Dataset II). provided level of anonymity for each record in different orders of Ngram Model are included as well.

#	2-gram Error	3-gram Error	K for 2-grams	K for 3-grams
T1	1.7	0.29	1	1
T2	1.71	0.38	1	1
T3	1.35	0.36	1	1
T4	0.75	0.33	1	1
T5	1.37	0.35	1	1
T6	1.14	0.37	1	1
T7	1.58	0.3	1	1
T8	1.94	0.41	1	1
T9	1.49	0.3	1	1
T10	1.38	0.24	1	1

In tuning the Chen's method, we set the l-max and n-max to 3 and α to 0.1. To make ready the dataset as Chen's method required, we repeat the n-grams in times equal to the frequency of them. We should note that the lengths of all of n-gram are constant and equal to 3. The implementation provided by Chen[6] is exploited in tests. The reconstructed set of n-grams is added to the time series Ngram table and n-grams are re-counted to determine the anonymity level provided by this method. Since it is not an anonymization method, then it does not need the anonymity level parameter. Thus, the anonymity level, which is pro vided by this method is the minimum value of the frequency of the n-grams.

6.2 Probabilistic Information Loss Analysis

Information loss can be defined differently in various application domains. In Ngram anonymization we define the information loss according to the application of the Ngram models. Since probabilities are used in the predictive model, we take into consideration the probabilities instead of frequencies in defining our information loss. We define Average Probabilistic Information Loss (APIL) criterion as follows:

$$APIL_N = \frac{\sum_{i=1}^{p^N} \left| P_{G_i} - P'_{G_i} \right|}{p^N} \quad (5)$$

where P_{G_i} and P'_{G_i} are probabilities of n-gram G_n, before and after anonymization, respectively and p denotes the count of symbols in symbolized time series.

[5] http://www.cs.ucr.edu/~eamonn/discords/

[6] https://team.inria.fr/privatics/private-big-data-publication/

6.3 Entropy Information Loss Analysis

Subsequences of time series can be used in analysis other than predictive methods. Some of these analyses use entropy of subsequences of time series, as in [24]. Thus, we take into consideration the entropy of n-grams by redefining the information loss to obtain Average Entropy Information Loss, as follows:

$$AEIL_N = \frac{\sum_{i=1}^{p^N} \left| E_{G_i} - E'_{G_i} \right|}{p^N} \qquad (6)$$

where E_{G_i} and E'_{G_i} are entropy values of n-gram G_n, before and after updating, respectively and p denotes the count of symbols in symbolized time series.

6.4 Results and Discussion

First, we evaluate the information loss of the Chen et al. [12] for all of the time series. The Tables 4 and 5 show the results of the experiments. The columns of the tables show the APIL values and the provided anonymity level of the method. We should note that we evaluate the Chen's method in 3-gram and 2-gram level separately. Since Chen's method does not add enough n-grams to provide preferred anonymity level, the values of APIL for this method are small. Moreover, this method is not able to provide a predefined anonymity level for Ngrams extracted from a few numbers of time series.

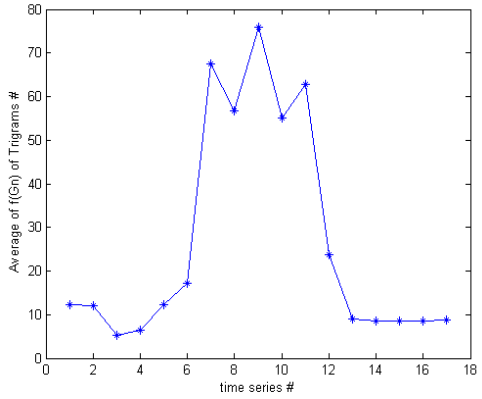

Fig. 7 Average values of frequencies of 3-grams in 17 time series of dataset I. The outlier behavior is happened in the region started from "data7" to "data11".

The outlier behavior of this method for time series "data7" to "data11" can be justified by doing more investigations on statistical properties of these time series. The anomalous regions in these series are short and they contain a few numbers of rare n-grams. Thus, the Ngram table includes two groups of n-grams, (1) a large set of frequent n-grams, (2) a small set of rare n-grams. The differences in distribution of rare n-grams lead to differences in average of n-gram frequencies. Fig. 7 shows the average of 3-grams in time series of the first dataset. The set of rare n-grams of these time series are very small and the average of the frequencies is skewed to the higher frequencies. The Chen's method tries to reconstruct the longest possible n-grams to anonymize the rare pattern, thus it does the best in mentioned time series, which have higher rate of frequent n-grams and consequently, higher frequency average.

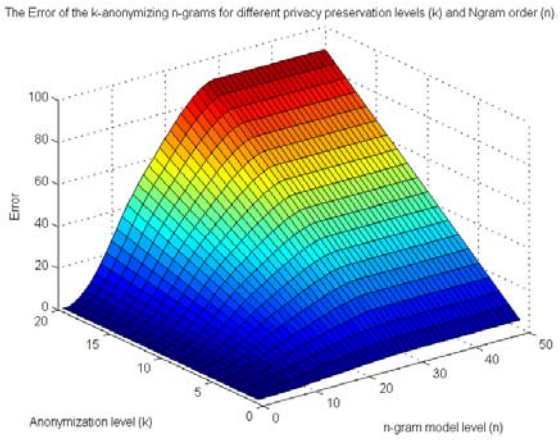

Fig. 8. APIL values for k-anonymizing "data1" with various k values and different orders of Ngram model.

Accordingly, it is not required to evaluate the Chen's method [12] by using AEIL criterion. We continue our test with our method to investigate its ability in anonymizing the Ngram models of time series. We repeat the proposed *k*-anonymization schema with different k values, where $k \in [5,100]$. The results of the average probability and average entropy information loss analysis are illustrated in Fig. 8 and 9. We should note that, incrementing the k leads to more changes in f values, because the higher the k value, the more n-gram are considered as non-anonymous. Considering this effect, the incremental behavior of the trend of APIL can be justified, but *what happened to the tail of the trend?*

In the APIL plot, trend is slowed down or stopped at k, where $k>50$. At the start of APIL plots, the trend is increasing that can be justified by taking into account the effect of appearing more non-anonymous n-grams, as mentioned already. However, this effect does not continue for the bigger k values. It happens because after specific k values, only few n-grams have frequencies less than the k and need to be anonymized. As we explained before, we are here encountering the second group of n-grams; that is they are frequent n-grams appeared at the periodic regions having higher frequencies even than k and they are not required to be anonymized. The similar behavior can be reported for AEIL trends (for the sake of space we skip the illustrations of the AEIL which are similar to APIL plots) for

another dataset, with similar justification.

The APIL and AEIL have similar behavior from macroscopic point of view. However, they have different microscopic behavior. The value of k in which the growth of AEIL is stopped is less than that of k for APIL. The major reason for such a behavior is apparent and comes back to the mechanism we used for incrementing the frequency of the margins. In our method, the frequencies of margins are incremented proportional to their neighborhood probability, therefore the entropy of the margins of an n-gram are constant.

For small k values, the AEIL grows gradually, because the frequencies of margins may be incremented more than one. However, for bigger k values, some of margins are already anonymized, then their frequencies are not required to be increased.

During the experiments, we also investigate the effect of the order of the Ngram model (denoted by the n) to evaluate the effectivity of the proposed method. Fig. 10 shows the results of experiment using the signal 1 (namely chfdb_chf01_275) from Dataset I. Results show that incrementing the order of the Ngram model leads to increasing the APIL. The higher the order of the model, the more rare n-grams, and thus the less n-grams satisfy the k-anonymity. In other words, we can say that the optimum k value must be determined by taking into account the order of the Ngram model. Intuitively, higher Ngram orders provide lower privacy preservation levels. Fortunately, the learning models using Ngram, are commonly trained by low orders; therefore, providing the appropriate level of preservation with acceptable error is achievable.

7. Conclusion

Publishing large datasets of time series raises concerns about revealing the sensitive information. These concerns are addressed through the privacy preservation for publishing time series. However, the problem is somehow different in ad hoc applications when data owner needs to publish a few numbers of time series, especially for medical time series. In contrast to the well studied time series privacy preserving publication, publication of Ngrams is under-studied, while Ngrams are one of the most applicable models in the sequential predictive analysis.

A sequence of symbols, as an output of symbolized time series, is segmented into subsequences of length less than or equal to n to generate an Ngram model. Since the time series is broken, their trends seem not to be exploitable. However, some rare n-grams can reveal observations of sensitive trends. Therefore, they should be considered and their leakage should be prevented.

In this paper, the problem of privacy preserved publishing of Ngrams extracted from a few numbers of medical time series was studied. We defined the k-anonymity principle for n-grams for the first time and accordingly, we introduced a schema, which anonymizes the n-grams by repeating rare n-grams. This method takes into account the indirect effects of repeating of n-grams by considering their neighbors and subgrams. Our method is not involved with the challenging task of anomaly detection. It is also easy to implement, has low complexity and provides low information loss.

As a future work, we plan to extend the proposed schema to suggest the appropriate k value trading off between the information loss and the necessary level of anonymity. Furthermore, we would like to investigate the impact of different symbolization methods on information loss of the schema.

References

[1] L. Donnelly, “Hospital records of all NHS patients sold to insurers,” *The Telegraph*, 23-Feb-2014.

[2] K. El Emam, J. Mercer, K. Moreau, I. Grava-Gubins, D. Buckeridge, and E. Jonker, “Physician privacy concerns when disclosing patient data for public health purposes during a pandemic influenza outbreak,” *BMC Public Health*, vol. 11, p. 454, Jun. 2011.

[3] F. Hormozdiari, J. W. J. Joo, A. Wadia, F. Guan, R. Ostrosky, A. Sahai, and E. Eskin, “Privacy preserving protocol for detecting genetic relatives using rare variants,” *Bioinformatics*, vol. 30, no. 12, pp. i204–i211, 2014.

[4] Y. Zhu, Y. Fu, and H. Fu, “On Privacy in Time Series Data Mining,” in *Advances in Knowledge Discovery and Data Mining*, vol. 5012, T. Washio, E. Suzuki, K. Ting, and A. Inokuchi, Eds. Springer Berlin Heidelberg, 2008, pp. 479–493.

[5] V. Rastogi and S. Nath, “Differentially Private Aggregation of Distributed Time-series with Transformation and Encryption,” in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, 2010, pp. 735–746.

[6] L. Fan and L. Xiong, “Real-time Aggregate Monitoring with Differential Privacy,” in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, 2012, pp. 2169–2173.

[7] M. Joye and B. Libert, “A Scalable Scheme for Privacy-Preserving Aggregation of Time-Series Data,” in *Financial Cryptography and Data Security*, vol. 7859, A.-R. Sadeghi, Ed. Springer Berlin Heidelberg, 2013, pp. 111–125.

[8] Y.-S. Moon, H.-S. Kim, S.-P. Kim, and E. Bertino, “Publishing Time-series Data Under Preservation of Privacy and Distance Orders,” in *Proceedings of the*

21st International Conference on Database and Expert Systems Applications: Part II, 2010, pp. 17–31.

[9] X. Shang, K. Chen, L. Shou, G. Chen, and T. Hu, "(K,P)-anonymity: Towards Pattern-preserving Anonymity of Time-series Data," in *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, 2010, pp. 1333–1336.

[10] L. Shou, X. Shang, K. Chen, G. Chen, and C. Zhang, "Supporting Pattern-Preserving Anonymization for Time-Series Data," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 4, pp. 877–892, Apr. 2013.

[11] Y. Zhu, Y. Fu, and H. Fu, "Preserving Privacy in Time Series Data Classification by Discretization," in *Proceedings of the 6th International Conference on Machine Learning and Data Mining in Pattern Recognition*, 2009, pp. 53–67.

[12] R. Chen, G. Acs, and C. Castelluccia, "Differentially Private Sequential Data Publication via Variable-length N-grams," in *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, 2012, pp. 638–649.

[13] Y.-C. Huang, H. Lin, Y.-L. Hsu, and J.-L. Lin, "Using n-gram analysis to cluster heartbeat signals," *BMC Med. Inform. Decis. Mak.*, vol. 12, no. 1, 2012.

[14] P. Tonella, R. Tiella, and C. D. Nguyen, "Interpolated N-grams for Model Based Testing," in *Proceedings of the 36th International Conference on Software Engineering*, 2014, pp. 562–572.

[15] S. Sekine, "A Linguistic Knowledge Discovery Tool: Very Large Ngram Database Search with Arbitrary Wildcards," in *22Nd International Conference on on Computational Linguistics: Demonstration Papers*, 2008, pp. 181–184.

[16] M. N. Team, "Microsoft Web N-gram Services," *Microsoft*. [Online]. Available: http://research.microsoft.com/en-us/collaboration/focus/cs/web-ngram.aspx.

[17] M. Google Team, "Google Book Ngram," *Google*. [Online]. Available: https://books.google.com/ngrams.

[18] E. Keogh, J. Lin, and A. Fu, "HOT SAX : Finding the Most Unusual Time Series Subsequence : Algorithms and Applications," in *Fifth IEEE International Conference on Data Mining*, 2005, p. 8.

[19] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly Detection: A Survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 15:1–15:58, 2009.

[20] M.-R. Zare-Mirakabad, F. Kaveh-Yazdy, and M. Tahmasebi, "Privacy Preservation by k-anonymizing Ngrams of Time series," in *The 10th International ISC Conference on Information Security and Cryptology (ISCISC'13)*, 2013.

[21] "MIT-BIH Arrhythmia Database." [Online]. Available: http://www.physionet.org/physiobank/database/mitdb/.

[22] G. B. Moody and R. G. Mark, "The impact of the MIT-BIH arrhythmia database," *IEEE Eng. Med. Biol. Mag.*, vol. 20, no. 3, pp. 45–50, 2001.

[23] M. Chuah and F. Fu, "ECG Anomaly Detection via Time Series Analysis," in *Frontiers of High Performance Computing and Networking ISPA 2007 Workshops*, vol. 4743, P. Thulasiraman, X. He, T. Xu, M. Denko, R. Thulasiram, and L. Yang, Eds. 2007, pp. 123–135.

[24] M. Costa, A. L. Goldberger, and C. K. Peng, "Multiscale entropy analysis of complex physiologic time series," *Phys. Rev. Lett.*, vol. 89, no. 6, 2002.

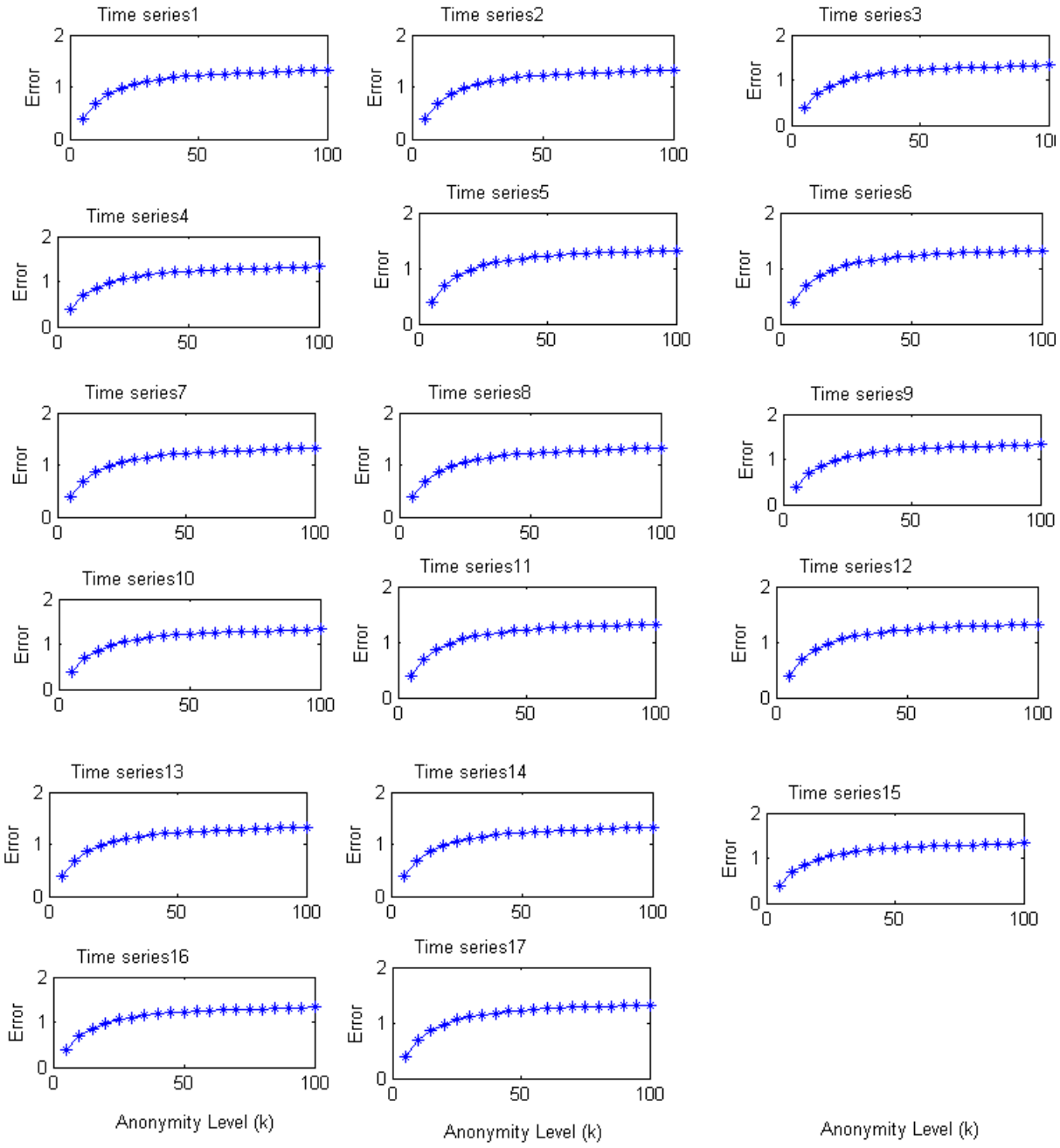

Fig. 9. Average Probability Information Loss (APIL) of anonymizing 3-grams of each times series of Dataset I.

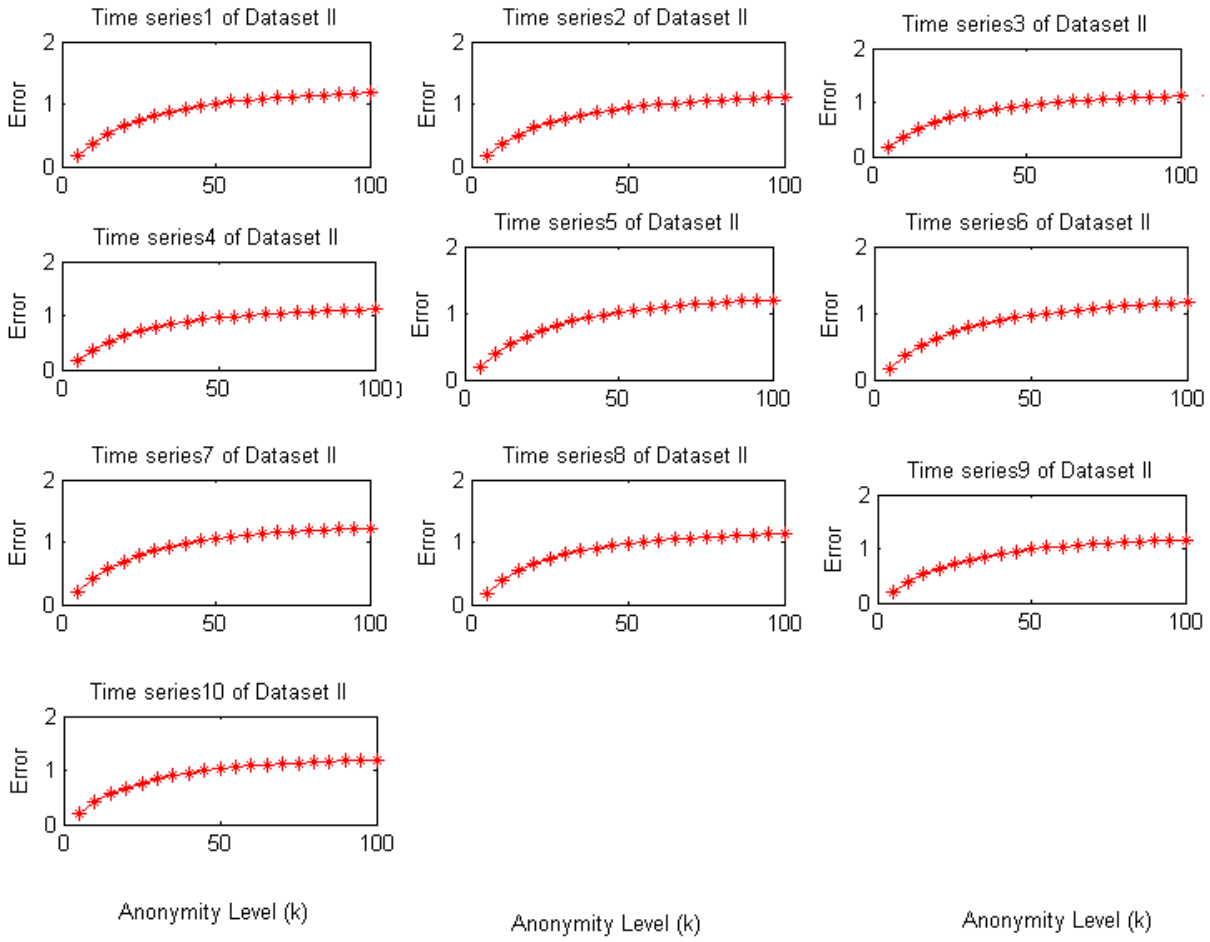

Fig. 10. Average Probability Information Loss (APIL) of anonymizing 3-grams of each times series of Dataset II.