2010

A New Class of Attacks on Time Series Data Mining

Ye Zhu
Cleveland State University, y.zhu61@csuohio.edu

Yongjian Fu
Cleveland State University, y.fu@csuohio.edu

Huirong Fu
Oakland University

Original Citation

Y. Zhu, Y. Fu and H. Fu, "A new class of attacks on time series data mining\m{1}," Intelligent Data Analysis, vol. 14, pp. 405-418, 06, 2010.

A new class of attacks on time series data mining[1]

Ye Zhu[a,*], Yongjian Fu[a] and Huirong Fu[b]
[a]*Department of Electrical and Computer Engineering, Cleveland State University, Cleveland, OH, USA*
[b]*Department of Computer Science and Engineering, Oakland University, Rochester, MI, USA*

Abstract. Traditional research on preserving privacy in data mining focuses on *time-invariant* privacy issues. With the emergence of time series data mining, traditional *snapshot-based* privacy issues need to be extended to be multi-dimensional with the addition of *time dimension*. We find current techniques to preserve privacy in data mining are not effective in preserving time-domain privacy. We present the data flow separation attack on privacy in time series data mining, which is based on blind source separation techniques from statistical signal processing. Our experiments with real data show that this attack is effective. By combining the data flow separation method and the frequency matching method, an attacker can identify data sources and compromise time-domain privacy. We propose possible countermeasures to the data flow separation attack in the paper.

Keywords: Privacy, time series data mining, blind source separation

1. Introduction

With the popularity of data mining, privacy issues have been a serious concern. Most research on privacy issues in data mining focuses on privacy preserving data mining, i.e., how to mine data while protecting the identity of data owners. Various approaches have been proposed to conduct data mining without breaching of privacy [1–6]. However, privacy issues studied in previous research are on time-invariant data which do not change over time. In other words, the data can be viewed as a snap-shot of objects.

Time series data mining has become popular recently. The goal of time series data mining is to find pattens contained in time series data [7–16]. In time series data mining, the data to be mined is labeled with timestamps. One example is the daily stock price. For time series data, because of the temporal information contained in the data, its privacy goes beyond the protection of data. In this paper, when the meaning of privacy is unclear from context, we call the privacy in time-invariant data mining *snap-shot privacy*, and the privacy in time series data mining *time series privacy*.

We focus on time series privacy issues in this paper. As snap-shot privacy issues arise from snap-shot based data mining, time series privacy issues arise from time series data mining. Time series privacy issues concern about changes in data over time. We need to protect data, as well as its properties in time

[1]A preliminary version of the paper appeared in the 2008 Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2008).
[*]Corresponding author: Ye Zhu, Department of Electrical and Computer Engineering, Cleveland State University, Cleveland, OH 44115, USA. Tel.: +1 216 875 9749; Fax: +1 216 687 5405; E-mail: y.zhu61@csuohio.edu.

and frequency domains. For example, sales data on a car model changes over time, but the manufacturer of the car model will worry about sharing the sales data with data miners because the sales data may indicate changes in financial situation or marketing strategies of the manufacturer over time. Another example is that a store may not be willing to share its sales data because a data miner may find out promotion periods of the store by checking periodicities contained in the data provided by the store. We argue that privacy in time series data involves protection of properties in time domain such as peak, trough, and trend, and properties in frequency domain, such as periodicity. Such properties reveal lots of information, even though they do not reveal data.

Two common approaches have been proposed to preserve snap-shot privacy in data mining. One approach is data perturbation in which data to be mined is modified to protect privacy. The other approach is data partitioning in which data is split among multiple parties and each party only see its share of the data. One method in data perturbation approach is aggregation in which time series data from different sources are aggregated and given to data miners. This can prevent data miners from finding private information about individual sources. For example, auto manufacturers usually do not want to publish daily, monthly or yearly sales data of individual car model because too much sensitive information is contained in the time series data. Instead, trusted market research companies aggregate sales data of different car models made by different auto manufacturers and publish these aggregated data for data mining or market study. These time series data can be aggregated in different ways such as according to vehicle types or vehicle features for different purposes.

In this research, we found that current techniques to protect snap-shot privacy were largely ineffective under data flow separation attacks, which can separate aggregated data and separate noise from original data. The data flow separation attack employs the *blind source separation* model [17], which was originally defined to solve the *cocktail party problem*: blind source separation algorithms can extract one person's voice signal given the mixtures of voices in a cocktail party.

Our experiments show that data flow separation can separate independent time series data generated from different sources.

The contributions of this paper can be summarized as follows:

- We introduce the concept of privacy in time series data mining. Because of the nature of time series data, privacy issues in time series data mining go beyond these in snap-shot data mining, especially privacy in time and frequency domains. We believe it is important to preserve privacy in time series data as well as in snap-shot data.
- We present *data flow separation attack* and show that aggregation is not always enough to protect time series privacy. We use experiments on real data to show that data flow separation attacks are effective.
- We present the *frequency matching attack*, a further attack based on data flow separation attacks, which can fully disclose sensitive information of data sources.
- We discuss the pros and cons of countermeasures to data flow separation attacks.

The rest of the paper is organized as follows: Section 2 reviews the related work in privacy preserving data mining and time series data mining. We list time series privacy issues in Section 3. Section 4 outlines the threat model. In Section 5, we introduce the data flow separation attack. We will also describe frequency matching that can be used as further attacks. In Section 6, we use experiments on real stock data to show the effectiveness of the data flow separation attack. Section 7 discusses countermeasures for the data flow separation attack. We conclude this paper in Section 8, with remarks on extensions of this work.

2. Related work

2.1. Privacy preserving data mining

The main approaches to privacy-preserving data mining can be divided into two categories: data perturbation and data partitioning.

In data perturbation approaches, original data is modified by data obscuration or by adding random noises. An example of data obscuration consists on replacing values of a continuous variable with ranges. Distributions of random noises are usually known, such as the even distribution or normal distribution. The modified data is given to data miners. Algorithms have been developed to mine decision trees [1] and association rules [2] in data with noise. Techniques for improving randomization are also proposed [3, 18,19].

In data partitioning approaches to privacy preserving data mining, the original data is distributed among multiple parties, either by the partitioning of centralized data or by the nature of the data collection. The data mining process is split into local computation at individual sites and global computation. During the process, each party does not see other party's data, but cooperates to find global patterns. In almost all cases, secure multi-party computation [20] and encryptions are employed. Secure algorithms for decision tree construction [4], association rules mining [5,6], k-means clustering [21,22], and Bayesian network learning [23] have been proposed. In these algorithms, all parties were assumed to be semi – honest. That is, every party would faithfully follow the protocol or algorithm, but tried to learn as much as possible about others.

As discussed above, past research in privacy preserving data mining focuses on privacy of raw data. Though privacy of derived information has been mentioned [18], we are not aware of any research in time series privacy. We hope to raise the awareness of time series privacy issues by this paper.

2.2. Time series data mining

Because time series data is usually large and noisy, direct application of data mining algorithms on raw data is time-consuming and gives unreliable results. A lot of attention has been paid on preprocessing techniques that facilitate data mining tasks. Research in time series data mining mostly focuses on data preprocessing techniques, such as discretization and transformation [13,16,24], feature extraction and feature reduction [15]. Work has also been done in related techniques such as data representation [14] and similarity metrics [25,26].

The data mining tasks studied by researchers include subsequence matching [7,8], classification [10], clustering [11], time series modeling [12], and association rule mining [9].

It is clear that most research in time series data mining does not address privacy issues, let alone time series privacy issues. While current privacy preserving techniques can be applied to preserve snap-shot privacy in time series data, they are inadequate for protecting time series privacy.

3. Time series privacy issues

We identify privacy issues for time series data in addition to traditional privacy issues in data mining. Time series data from a data source can be regarded as a time-domain signal. All the characteristics of a time-domain signal can be potentially regarded as private information by the data provider. Below, we list common characteristics in time series data that a data provider may need to keep confidential.

– Amplitude: Amplitude indicates the strength of a signal. This is the same as in traditional privacy research.
– Average: The average signal strength over time. For example, for a series of sales data, average amplitude indicates the average sales.
– Peak and trough: Peak and trough indicate extreme situations. The information is usually considered confidential as it may disclose extreme changes in underlying causes such as difficulties in money flow.
– Trend: By observing trends of time series data, an adversary may predict future changes of time series data. Thus trend information should be protected from competitors as well.
– Periodicity: Periodical changes in time series data indicate existence of periodically changing factors. For sales data of a store, the factor can be periodical changes in marketing strategies such as promotions which are usually regarded as confidential information for stores. Unlike the previous characteristics which are in time domain, periodicity is in frequency domain.

There are other characteristics which may be regarded as confidential by some data providers. However, as an initial study on time series privacy, we focus on the common characteristics listed above. Since the data flow separation attack aims to recover the original signal, the attack may be effective to disclose these common characteristics.

4. Threat model

In this paper we assume that data providers care about the sensitive information contained in their time series data. To protect their privacy, data providers will only supply their data to trusted research companies. Research companies will aggregate time series data provided by different data providers according to different criteria. An example of aggregating sales data provided by auto manufacturers is shown in Fig. 1. In Fig. 1, there is only one aggregation layer. In practice there can be many layers of aggregation because some research companies may aggregate data provided by other research companies or aggregate data provided by both original data providers and research companies.

We assume research companies will publish aggregated data for profit or for public usage. The research companies will disclose criteria used in aggregation. But research companies will not disclose information of data sources, specifically identities of data providers to protect their privacy.

We assume adversaries to have capabilities summarized as follows:

– Adversaries can obtain aggregated data from research companies for a small amount of fee or for free.
– Adversaries can not obtain data generated from original data sources because of lack of trust with original data sources. This assumption excludes the possibility of an original data provider being a privacy attacker. We do not study the case of compromised data provider in this paper. But obviously the data flow separation attack will be more effective if an adversary, being a provider of original data, can know part of original data aggregated by research companies.
– Adversaries can obtain data aggregated according to different criteria.
– Research companies have various data providers as their data sources and they do not want to disclose the composition of data sources. It is similar to an investment company does not want to disclose the composition of stocks in possess.

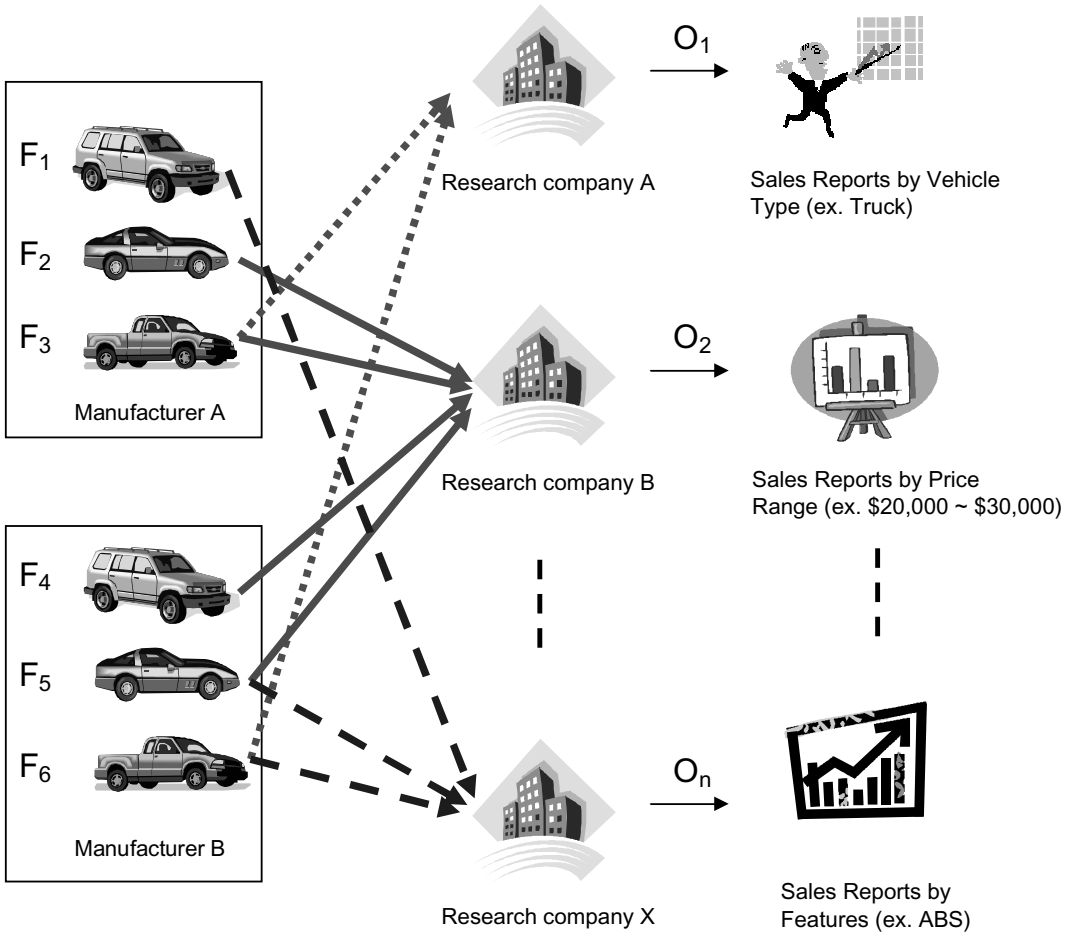


Fig. 1. An Example for Data Flow Model.

The threat model M can be represented as $M =< F, G, O >$, where F is a set of original data sources, G is a set of aggregation operations, and O is a set of observations available to adversaries. Though observations are obtained by applying aggregation operations to data sources, $O = G(F)$, aggregation operations and data sources are unknown to adversaries.

The model assumed in our paper is realistic. Many research companies compile weekly or monthly sales of large items, such as cars, TVs, computers, etc, from retailers or manufacturers. Each research company has its own sources and publishes its reports with aggregated totals. Since these reports are available with a small fee, someone can collect all these reports and try to separate data to recover original data. The manufacturers do not want to share data with parties other than trusted research companies, but would like to see the aggregated data to understand industry.

5. Data flow separation attack

In this section, we will first define the problem in the context of blind source separation and then describe how to apply the data flow separation attack in practice.

5.1. Blind source separation

Blind source separation is a methodology in statistical signal processing to recover unobserved "source" signals from a set of observed mixtures of the signals. The separation is called "blind" to emphasize that the source signals are not observed and that the mixture is a black box to the observer. While no knowledge is available about the mixture, in many cases it can be safely assumed that source signals are independent. In its simplest form [27], the blind source separation model assumes n independent signals $F_1(t), \cdots, F_n(t)$ and n observations of mixture $O_1(t), \cdots, O_n(t)$ where $O_i(t) = \sum_{j=1}^{n} a_{ij} F_j(t)$, where a_{ij} is the mix coefficient. The goal of blind source separation is to reconstruct the source signals $F_j(t)$ using only the observed data $O_i(t)$, with the assumption of independence among the signals $F_j(t)$. A very nice introduction to the statistical principles behind blind source separation is given in [27]. The common methods employed in blind source separation are minimization of mutual information [28,29], maximization of nongaussianity [30,31] and maximization of likelihood [32,33].

5.2. Data flow separation as a blind source separation problem

In this paper, we define an *individual data flow* as a series of time-stamped data generated by an original data source. *An aggregate data flow* is defined as the aggregate of individual data flows. Aggregate data flows are generated by research companies. If not specified, the phrase *data flow* in the remaining of this paper means the individual data flow for brevity.

For an attacker who is interested in sensitive information contained in individual data flow, it would be very helpful to separate the individual data flows based on the aggregate data flows. Because the separation of the data flows can recover the pattern of data flows, they can be use for further attacks such as frequency matching attack described in Section 5.3.

In this paper, we are interested in patterns carried in the time series data. For example, in Fig. 1, the attacker can get a time series $O_1 = [o_1^1, o_2^1, \cdots, o_n^1]$ of the aggregate data flow from Research Company A. We call n as the sample size in this paper. The attacker's objective is to recover the time series $F_i = [f_1^i, f_2^i, \cdots, f_n^i]$ for each individual data flow. The time series F_3 is contained in both aggregate flows O_1 and O_2, i.e., $O_1 = F_3 + F_6$, $O_2 = F_2 + F_3 + F_4 + F_5$.

In general, with l research companies and m individual data flows, we can rewrite the problem in vector-matrix notation,

$$\begin{pmatrix} O_1 \\ O_2 \\ \vdots \\ O_l \end{pmatrix} = \mathbf{A}_{l \times m} \begin{pmatrix} F_1 \\ F_2 \\ \vdots \\ F_m \end{pmatrix} \tag{1}$$

where $\mathbf{A_{l \times m}}$ is called the mixing matrix in blind source separation problems.

Data flow separation can be achieved using blind source separation techniques. The individual data flows are independent from each other since the individual data flows are from different sources. Given the observations $O_1, O_2, \cdots, O_l$, blind source separation techniques can be used to estimate the independent individual flows $F_1, F_2, \cdots, F_m$ by maximizing the independence between estimated individual data flows.

The issues about the blind source separation method are summarized as follows.

– Basic blind source separation algorithms require the number of observations to be greater than or equal to the number of independent sources. For data flow separation, it means $l \geqslant m$. Furthermore, we assume $m = l$ in this paper since it is fairly straightforward to extend our idea to cases where $l > m$.
– The l observations may have redundancy. In other words, the row vectors of the mixing matrix may be linearly dependent. The cost of the redundancy will be that some independent data flows are not separated.
– The data flow estimations by blind source separation algorithms are usually lifted, scaled versions of the actual data flows. Sometimes, the estimated data flow may be of different sign than the actual data flow. However, the attacker can still find characteristics of the actual data flow from the estimated data flow. Also, heuristic approaches can be used to fine tune the estimation, which is an interesting topic for further research.

5.3. Frequency matching attack

After the data flows have been separated, a number of data flows, each with a given time series, have been determined to be included in the aggregate.

We choose frequency spectrum matching to do further attack. Frequency spectrum can be generated by applying the Discrete Fourier Transform on time series data as below

$$X(k) = \sum_{j=0}^{N-1} f'_j e^{-\frac{2\pi i k j}{N}} \tag{2}$$

where f'_j denotes the jth data point in the time series data and N denotes the length of the time series and then calculating the magnitude of the transformed data. We match frequency spectrum by correlation.

The rationale for the use of frequency matching is two-fold: First, the dynamics of many data flows, such as sales, stock price, and weather, are characterized by their periodicities. By matching the frequency spectrum of a known data flow with the frequency spectrum of estimated data flows obtained by blind source separation techniques, we can identify corresponding flows with high accuracy. Second, frequency matching can easily remove the ambiguities introduced by the lifting and scaling in the estimated time series by removing the zero-frequency component. In summary, frequency spectrum analysis has excellent prerequisites to be highly effective.

Frequency matching can be applied to match data flows separated from different attacks. After collecting a set of aggregate data flows according to different criteria, the attacker can select arbitrary subsets of aggregate data flows as groups and apply data flow separation techniques to the groups to recover individual data flows. If a data flow separated from a group matches with a data flow separated from another group, then these two data flows should be generated from the same source. Moreover, the source generating these two data flows should satisfy at least one aggregation criteria in each group. If the attacker can match a data flow with data flows separated from several groups, the attacker can largely reduce the anonymity or possibly determine the identity of the source generating the data flow since the source should satisfy at least one criteria in each of these groups of aggregate flows. To better utilize the data, the attacker can try all the possible combinations to group available aggregate data flows and then match the data flows separated from these groups. Of course, when the number of aggregate flows in a group is too small, the data flow separation technique can not separate all data flows because the number of observations is smaller than the number of independent components. We will study the limit of this grouping and matching method in our future research.

6. Evaluation

In this section, we will evaluate the performance of data flow separation. We use the blind source separation algorithm proposed in [34] to separate the data flows. The accuracy of separation will be measured using correlation with actual flows. In our experiments, real stock market data [35] is used.

6.1. Performance metrics

In the following, we will adopt two metrics to evaluate the accuracy of data flow separation. Both metrics are based on a comparison of the separated data flows with the actual data flows.

As first performance metric, we use *mean square error* (*MSE*), a widely used performance criterion in blind source separation research. Let $F_A = [f_1^A, f_2^A, \cdots, f_n^A]$ represent the time series of the actual data flow and $F_B = [f_1^B, f_2^B, \cdots, f_n^B]$ represent the time series estimated by the blind source separation algorithm. In order to match the time series F_A with F_B, we first need to scale and lift F_B so that they have the same mean and variance.

$$\begin{aligned} F'_B = & \frac{std(F_A)}{std(F_B)} \cdot (F_B - mean(F_B) \cdot [1, 1, \cdots, 1]) \\ & + mean(F_A) \cdot [1, 1, \cdots, 1], \end{aligned} \tag{3}$$

where $std(F)$ and $mean(F)$ denote the standard deviation and the average of time series F, respectively. The *mean square error* is defined as follows:

$$\varepsilon_{A,B} = \frac{\|F_A - F'_B\|^2}{n} \quad . \tag{4}$$

Since the times series F_B can also be a flipped version of F_A, we also need to match F_A with $-F_B$.

As the second metric, we use *correlation* between the separated flow F_B and the corresponding actual flow F_A defined as follow:

$$R_{F_A,F_B} = \frac{\sum_i (f_i^A - mean(F_A))(f_i^B - mean(F_B))}{std(F_A)std(F_B)} \tag{5}$$

6.2. A small example

In this experiment, four time series of stock price selected from [35] are mixed into four aggregates. Figures 2(a) and 2(b) show the actual data flows and separated data flows from the aggregates.

We can observe for data flows 1, 2, and 3, the separated data flows are flipped, scaled and lifted versions of the corresponding actual data flows. We can also observe the resemblance between separated flow and the corresponding actual flow for data flow 4.

Figure 3 shows the performance of data flow separation in terms of the metrics introduced in Section 6.1. According to the correlation metric, the separated data flows are highly correlated to the actual data flows as shown in Fig. 3(a). In Fig. 3(b) both the separated data flow and its flipped time series are compared against the actual flows and the mean square error for each data flow shown in the figure is the smaller one. According to the mean square error metric 4 and 4, the reconstructed data flows are off by around 10% in comparison with the actual data flows. Both metrics indicate that the data flow separation is successful. In the following we will only use correlation to evaluate performance because the lifting and scaling in the mean square error metrics may introduce error.

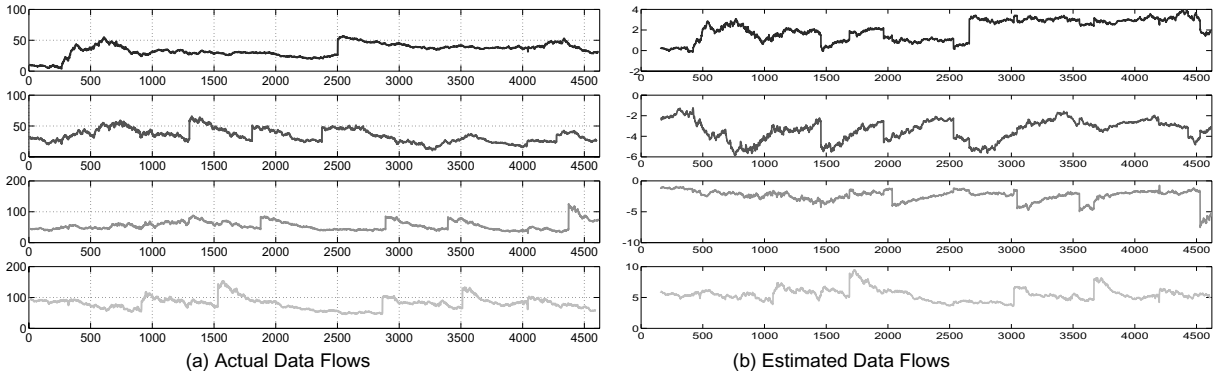


Fig. 2. Example of Data Flow Separation.

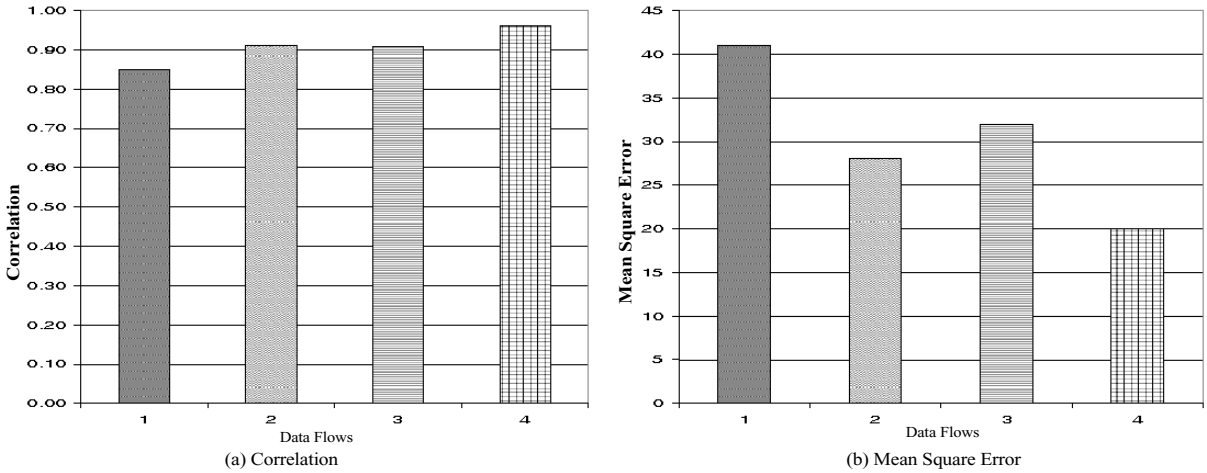


Fig. 3. Performance of Data Flow Separation on a Small Example.

6.3. Mixing degree

In this set of experiments, we would like to study the effect of mixing degree on the performance of data flow separation. We define mixing degree as follows:

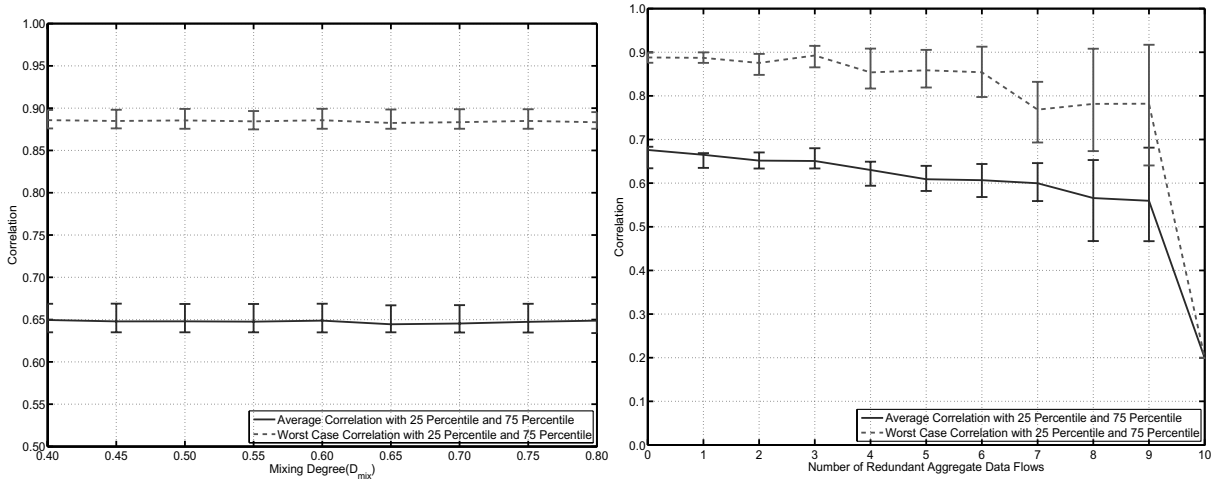
$$\mathrm{D_{mix}} = \frac{\text{average number of individual data flows mixed in aggregates}}{\text{number of individual data flows}} \tag{6}$$

It is equivalent as

$$D_{mix} = \frac{\text{number of non-zero entries in } A_{l\times m}}{l \times m} \quad . \tag{7}$$

Ten time series selected from stock data [35] are mixed randomly in this experiment to create ten aggregates. A total of 10000 randomly-generated *full-rank*[1] binary mixing matrices were used in this experiment.

[1]Experiments on rank deficient mixing matrices are described in Section 6.4.



(a)Effect of Mixing Degree (b) Effect of Redundant Observations

Fig. 4. Effect of Mixing Degree and Redundant Observations (Lower bar: 25 percentile, Upper bar: 75 percentile).

Figure 4(a) shows the effect of mixing degree on the performance of data flow separation. We plot statistics of both average correlation and worst case correlation. In this paper average correlation is defined as mean of correlation between separated data flows and actual data flows for each trial. We use the term worst case to refer to the most accurately separated data flow in each trial. It corresponds to the worst privacy compromising in each trial.

From Fig. 4(a), we can observe that data flow separation is effective since the separated flows are highly correlated to actual flows especially for the worst case. We can also observe that the performance of data flow separation is not sensitive to mixing degree for full-rank mixing matrices. This experiment indicates that a countermeasure to the data flow separation attack by simply increasing the mix degree is not effective.

6.4. Redundant aggregate data flows

In this set of experiments, we focus on the cases with redundant aggregate data flows. In our setting, redundant aggregate data flows mean that some aggregate data flows are linear combinations of other aggregate data flows. Redundant aggregate data flows will reduce the number of effective aggregate data flows. Redundant aggregate data flows are caused by rank deficient mixing matrices.

To study the effect of redundant observations, we randomly generated 1000 mixing matrices for each possible rank. Ten data flows randomly selected from the stock data are mixed using the randomly-generated mixing metrics of different ranks.

Figure 4(b) shows the performance of the data flow separation algorithm with redundant observations. We can observe that the performance of data flow separation decreases as the number of redundant observations increases. The performance degrades because the number of knowns decreases. When the number of aggregate data flows is larger than the number of individual data flows, the data flow separation problem becomes an over-complete base problem in blind source separation literature. In general an over-complete base problem is harder to solve.

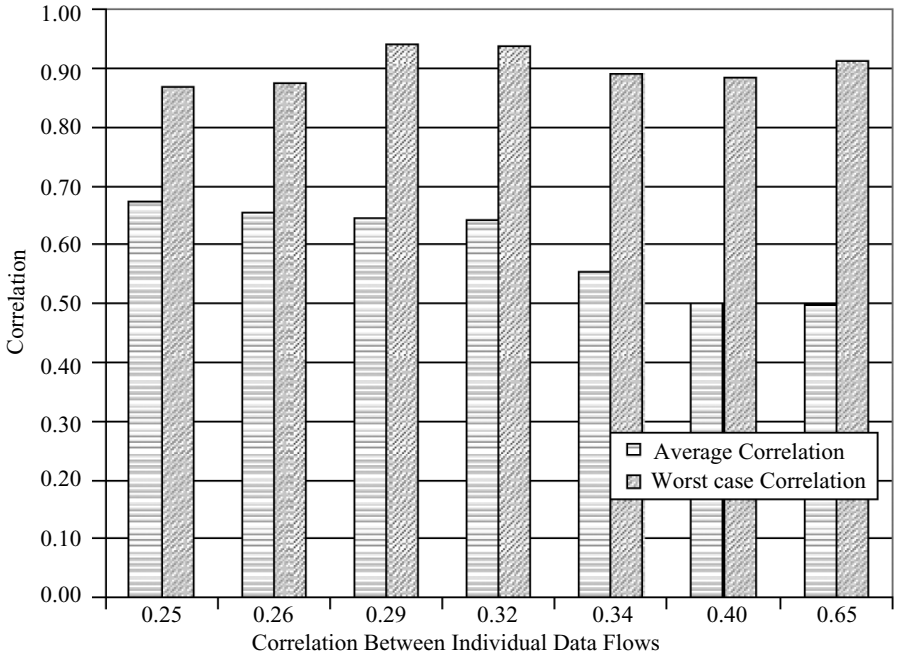


Fig. 5. Effect of Dependence between Individual Data Flows.

6.5. Dependence between individual data flows

In this set of experiments, we study the effect of dependence between individual data flows on data flow separation performance. We did this series of experiments because of the fact that most blind source separation algorithms assume relative independence between actual signals.

Groups of ten data flows are randomly picked from the stock data [35]. These groups of data flows have different average correlation among data flows in the same group. In this set of experiments, dependence between individual data flows is defined as average correlation between any pair of data flows in the group. The time series in each group are mixed randomly and we apply the data flow separation technique on the generated aggregates.

Figure 5 shows that the performance of the data flow separation technique decreases when the dependence among individual data flows increases. This happens because blind source separation algorithms used in data flow separation assume independence between underlying components. Even for the blind source separation algorithm [34] which takes advantage of both independence and timing structure of the underlying signals, the dependence among individual data flows can still degrade the performance of data flow separation attack. We can also observe that the worst case correlation is not sensitive to the dependence between individual data flows.

6.6. Frequency matching

In this subsection, we show the performance of the frequency matching attack proposed in Section 5.3. In this experiment, two groups of ten data flows are formed by selecting data flows from the stock dataset. Three data flows in both groups are the same. These two groups of data flows are mixed randomly to form two groups of aggregate data flows. Data flow separation is performed on the two groups of aggregate

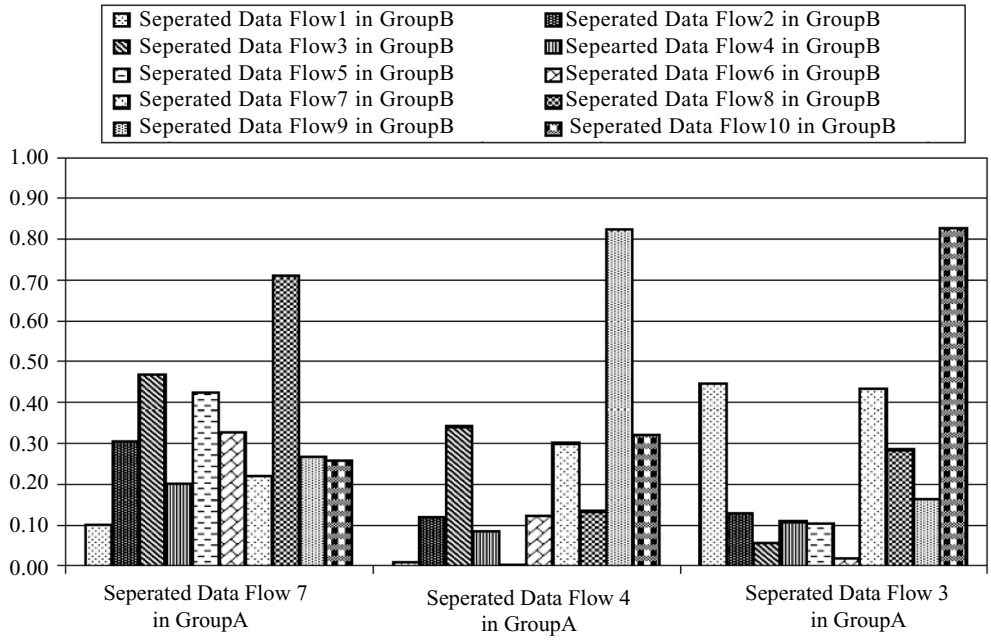


Fig. 6. Performance of Frequency Matching.

data flows. We identify common flows in both groups by matching frequency spectrums of separated data flows in two different groups.

Figure 6 shows the correlation between three identified separated data flows in one group and the ten separated data flows in the other group. As shown in Fig. 6, we can easily find out the data flows common to both groups from correlation of frequency spectrums: Separated data flow 3, 4, and 7 in Group A correspond to separated data flow 10, 9, and 8 in Group B respectively.

7. Discussion

From the experiments in Section 6, it is apparent that aggregation methods are not sufficient to effectively counter data flow separation attacks. Additional measures are needed.

One naive countermeasure can be adding independent noise into aggregate data flows to counter the data flow separation attack. This approach may not work because the noise and the original data flow are regarded as independent components and thus they can be separated by the blind source separation algorithm.

According to our experiments, the following countermeasures will be effective against data flow separation attacks:

- Increase the dependence among data flows by adding dependent noise to the data flows. Further research is needed to investigate how to optimally add noise so that privacy can be preserved and the performance of time series data mining will not be significantly affected.
- Limit the number of aggregate data flows that can be obtained by adversaries so that the number of observations is much less than the number of independent components. This countermeasure requires cooperation among research companies and it is hard to be enforced.

– Data sources should know from research companies about how the supplied data is going to be aggregated and restrict the usage of supplied data.

Also, research in blind source separation shows most blind source separation algorithms fail when the signals mixed are Gaussian distributed. Therefore, another countermeasure against data flow separation attacks is padding each aggregate data flow so that the distribution of the aggregated data is Gaussian. But different classes of blind source separation algorithms that make use of the time structure of the signals can still separate the data flows [36,37].

As mentioned in [38], aggregation is a major technique used to preserve privacy in data mining. Since data flow separation attacks can separate individual data flows from aggregates, the aggregation technique based privacy-preserving data mining systems are potentially vulnerable to data flow separation attacks.

8. Conclusion

In this paper, we introduced the concept of privacy in time series data mining. We presented a new attack against privacy in time series data mining, called the data flow separation attack, which can be used either alone or in conjunction with other attacks to significantly reduce the effectiveness of privacy-preserving techniques in time series data mining. The data flow separation attack is based on the blind source separation algorithms widely used to recover individual signals from mixtures of signals. Our experiments show that the attack is effective. With the aid of further attacks such as frequency matching attack, the data flow separation attack can be used to determine data sources of separate data flows.

We discussed countermeasures against the data flow separation attack. Our future work will focus on countermeasures to balance privacy-preserving and performance of data mining. We also plan to analytically model the effectiveness of the attack.

Acknowledgements

We thank Professor Keogh for the data sets used in our experiments and anonymous reviewers for feedbacks on the initial version of this paper.

References

[1] R. Agrawal and R. Srikant, Privacy-preserving data mining, *In SIGMOD Conference* (2000), 439–450.
[2] A.V. Evfimievski, R. Srikant, R. Agrawal and J. Gehrke, Privacy preserving mining of association rules, *In SIGKDD* (2002), 217–228.
[3] W. Du and Z. Zhan, Using randomized response techniques for privacy-preserving data mining, *in SIGKDD* (2003), 505–510.
[4] Y. Lindell and B. Pinkas, Privacy preserving data mining, *In CRYPTO* (2000), 36–54.
[5] J. Vaidya and C. Clifton, Privacy preserving association rule mining in vertically partitioned data, *In SIGKDD* (2002), 639–644.
[6] M. Kantarcioglu and C. Clifton, Privacy-preserving distributed mining of association rules on horizontally partitioned data, *IEEE Trans Knowl Data Eng* **16**(9) (2004).
[7] R. Agrawal, C. Faloutsos and A.N. Swami, Efficient similarity search in sequence databases, *In FODO* (1993), 69–84.
[8] C. Faloutsos, M. Ranganathan and Y. Manolopoulos, Fast subsequence matching in time-series databases, *In SIGMOD Conference* (1994), 419–429.
[9] G. Das, K.-I. Lin, H. Mannila, G. Renganathan and P. Smyth, Rule discovery from time series, *In SIGKDD* (1998), 16–22.

[10] P. Geurts, Pattern extraction for time series classification, *In PKDD* (2001), 115–127.
[11] E.J. Keogh and J. Lin, Clustering of time-series subsequences is meaningless: implications for previous and future research, *Knowl Inf Syst* **8**(2) (2005), 154–177.
[12] A.T. Ihler, J. Hutchins and P. Smyth, Adaptive event detection with time-varying poisson processes, *In SIGKDD* (2006), 207–216.
[13] F.Mörchen and A. Ultsch, Optimizing time series discretization for knowledge discovery, *in SIGKDD* (2005), 660–665.
[14] E.J. Keogh and M.J. Pazzani, An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback, *in SIGKDD* (1998), 239–243.
[15] R. Cole, D. Shasha and X. Zhao, Fast window correlations over uncooperative time series, *in SIGKDD* (2005), 743–749.
[16] T. Mielikäinen, E. Terzi and P. Tsaparas, Aggregating time partitions, *in SIGKDD* (2006), 347–356.
[17] C. Jutten and J. Herault, Blind separation of sources, part 1: an adaptive algorithm based on neuromimetic architecture, *Signal Process* **24**(1) (1991), 1–10.
[18] Z. Huang, W. Du and B. Chen, Deriving private information from randomized data, *in SIGMOD Conference* (2005), 37–48.
[19] Y. Zhu and L. Liu, Optimal randomization for privacy preserving data mining, *in SIGKDD* (2004), 761–766.
[20] W. Du and M.J. Atallah, Secure multi-party computation problems and their applications: a review and open problems, *in New Security Paradigms Workshop 2001*, Cloudcroft, New Mexico, USA, September 10–13, 2001, 13–22.
[21] J. Vaidya and C. Clifton, Privacy-preserving k-means clustering over vertically partitioned data, *in SIGKDD* (2003), 206–215.
[22] G. Jagannathan and R.N. Wright, Privacy-preserving distributed k-means clustering over arbitrarily partitioned data, *in SIGKDD* (2005), 593–599.
[23] R.N. Wright and Z. Yang, Privacy-preserving bayesian network structure computation on distributed heterogeneous data, *in SIGKDD* (2004), 713–718.
[24] A.J. Bagnall and G.J. Janacek, Clustering time series from arma models with clipped data, *in SIGKDD* (2004), 49–58.
[25] D. Berndt and J. Clifford, Using dynamic time warping to find patterns in time series, *in AAAI-94 Workshop on Knowledge Discovery in Databases*, 1994.
[26] E.J. Keogh and M.J. Pazzani, Scaling up dynamic time warping for datamining applications, *in SIGKDD* (2000), 285–289.
[27] J. Cardoso, Blind signal separation: statistical principles, **9**(10) (1998), 2009–2025, special. issue on blind identification and estimation. [Online]. Available: citeseer.ist.psu.edu/cardoso98blind. html.
[28] P. Comon, Independent component analysis, a new concept? *Signal Process* **36**(3) (1994), 287–314.
[29] Z. He, L. Yang, J. Liu, Z. Lu, C. He and Y. Shi, Blind source separation using clustering-based multivariate density estimation algorithm, *IEEE Trans. on Signal Processing* **48**(2) (2000), 575–579.
[30] A. Hyvärinen, Fast and robust fixed-point algorithms for independent component analysis, *IEEE Transactions on Neural Networks* **10**(3) (1999), 626–634, 1999. [Online]. Available: citeseer.ist.psu.edu/hyv99fast.html.
[31] A. Hyvärinen and E. Oja, A fast fixed-point algorithm for independent component analysis, *Neural Comput* **9**(7) (1997), 1483–1492.
[32] M. Gaeta and J.-L. Lacoume, Source separation without prior knowledge: the maximum likelihood solution, *in Proc. EUSIPCO'9* (1990), 621–624.
[33] D.-T. Pham, P. Garrat and C. Jutten, Separation of a mixture of independent sources through a maximum likelihood approach, *in Proc. EUSIPCO* (1992), 771–774.
[34] S.A. Cruces-Alvarez and A. Cichocki, Combining blind source extraction with joint approximate diagonalization: Thin algorithms for ICA, *in Proc. of the Fourth Symposium on Independent Component Analysis and Blind Signal Separation* Nara, Japan, apr 2003, pp. 463–468.
[35] E. Keogh, X. Xi, L. Wei and C.A. Ratanamahatana, The UCR time series classification/clustering homepage, http://www.cs.ucr.edu/?eamonn/time series data/, 2006.
[36] L. Tong, R.-W. Liu, V.C. Soon and Y.-F. Huang, Indeterminacy and identifiability of blind identifi-cation, *Circuits and Systems, IEEE Transactions on* **38**(5) (1991), 499–509.
[37] L. Molgedey and H.G. Schuster, Separation of a mixture of independent signals using time delayed correlations, *Physical Review Letters* **72**(23) (June 1994), 3634–3637.
[38] N. Zhang and W. Zhao, Privacy-preserving data mining systems, *Computer* **40**(4) (2007), 52–58.