Privately detecting correlations in distributed time series

Mehmet Sayal, Lisa Singh

HP Laboratories
HPL-2010-167

Keyword(s):

time-series, correlation, privacy, multi-dimensional

Abstract:

In this paper, we consider privacy preservation in the context of independently owned, distributed time series data. Specifically, we are interested in discovering correlations even though we cannot share the raw time series values. We propose developing a generic framework for identifying similarities or correlations of a particular behavior or statistic across participants. Our generic framework makes use of the additive combining property of certain statistics. It also allows for sharing of scaled bin values instead of raw data or statistical values to improve levels of privacy. We find that while there is a natural trade off between privacy and accuracy, we can maintain reasonable correlation accuracy for different levels of privacy.

External Posting Date: October 21, 2010 [Fulltext] Approved for External Publication
Internal Posting Date: October 21, 2010 [Fulltext]

Privately detecting correlations in distributed time series

Mehmet Sayal * Lisa Singh†

Abstract

In this paper, we consider privacy preservation in the context of independently owned, distributed time series data. Specifically, we are interested in discovering correlations even though we cannot share the raw time series values. We propose developing a generic framework for identifying similarities or correlations of a particular behavior or statistic across participants. Our generic framework makes use of the additive combining property of certain statistics. It also allows for sharing of scaled bin values instead of raw data or statistical values to improve levels of privacy. We find that while there is a natural trade off between privacy and accuracy, we can maintain reasonable correlation accuracy for different levels of privacy.

1 Introduction

Time series data is prevalent in many applications including, transaction data, stock trend data, and procurement data. In some domains, this information is sensitive and distributed across independent entities, e.g. procurement and medical. Researchers have investigated ways to share noisy versions of these data [6], as well as Fourier and wavelet transformed versions of time series and data streams for different data mining applications [14, 17]. Our problem focuses on using time series data from independent entities to discover higher level correlations without sharing the raw time series data. We must also ensure that the original data can not be determined from the shared information. At the same time, we need to preserve enough of the original data characteristics to identify correlations with other participants' time series data.

For example, given store purchase data, multiple stores may be interested in identifying correlations in sales across the industry in different regions around the country. Because each store maintains its data independently, the data must be merged to identify correlations across all independent companies. Sharing the raw time series data is a concern. Each company only wants the resulting correlation information provided that the company data remains private. The correlation information gives the participating companies insight into the market trend in different regions. Companies can then take this information and see how their data compares to the trend in the region.

In this paper, we propose developing a generic framework for identifying correlations of a particular behavior or statistic. We consider different privacy models and make use of the additive combining property of certain statistics to find correlations among participants. We explore three different privacy levels and demonstrate how the type and accuracy of the correlations vary based on the level of privacy guaranteed. We also show that scaled bin values of the raw data are sufficient for finding pairwise correlations and that they improve the level of privacy for many statistics.

Our contributions are as follows: 1) we present a generic framework for identifying behavioral similarities or correlations in time series data using a number of different statistics; 2) we present a low overhead method for aggregating data that obscures raw data values while maintaining basic scalar information that is important for correlation analysis; 3) we present a number of different privacy constraints and show how accuracy is affected when more constraints are added to the privacy problem; 4) we conduct an experimental analysis on both synthetic and real world data to better illustrate the trade off between accuracy and privacy.

The remainder of the paper is organized as follows. We begin with some background and problem formulation in section 2. We then describe our approach in section 3 and the privacy guarantees in 4. Our experiments and evaluation are presented in section 5. We then review the related literature in section 6 and present conclusion in section 7.

2 Problem formulation and notation

This section presents definitions, terminology, and notation. Because there are a number of different notations that have been introduced for distributed time series data, we begin by presenting the notation we will use throughout this paper.

2.1 Time series notation Each input time series contains items e_1, e_2, ..., e_N. The items represent an underlying signal E that can be viewed as a vector or

*Hewlett Packard Company.
†Georgetown University.

as a one dimensional function $E : [1 \dots N] \rightarrow \mathbb{R}$. Here, each e_j equals $E[j]$ and the time series is of length N.

The distributed data can be horizontally or vertically partitioned. This paper focuses on a *horizontal partitioning* of the data, where each participant has the same time series variable for a disjoint set of data.Suppose there are P participants each having a local time series vector E. We denote the i-th participant's time series as E^i. The combined or aggregate time series A is the average of all the participants time series, $A = \sum_{i=1}^{P} \frac{E^i}{P}$.

2.2 Correlations Correlations are useful for showing a relationship between two or more random variables or observed data values. Data that are correlated are not considered statistically independent. There are several correlation coefficients that measure the amount of correlation. While any of them can be used, we focus on the Pearson correlation coefficient. It measures the linear relationship between data. Given two time series, E^i and E^j, the Pearson correlation coefficient, ρ_{E^i,E^j}, is calculated by dividing the co-variance of two variables by their standard deviation: $\rho_{E^i,E^j} = \frac{cov(E^i,E^j)}{\sigma_{E^i}\sigma_{E^j}}$. In the case where there are more than two time series, we can use a covariance matrix (ρ) that is a PxP matrix whose i, j entry is the correlation coefficient ρ_{E^i,E^j}.

2.3 Representative Values We analyze the time series in equal size chunks or *windows*. A time series of length N contains ω windows with $n = N \div \omega$ elements in each window. The window size n is also the same for all participants and the time intervals are aligned across participant sites.[1] We denote the k-th window of participant i as E^i_k and the k-th window of the aggregate time series as A_k.

For each window, i, we compute a single representative value, r_i. The representative values themselves create a new time series, $R = \{r_1, \dots, r_\omega\}$ that, in some privacy schemes we propose, is shared among participants. This representative time series will be described more in the next section. Because there is only one representative value for each window, in order to compute correlations, we define a *correlation window* as the subset of windows considered for identifying a correlation. The length of the correlation window, l_{CW}, is defined as $l_{CW} = c\omega$, where c is a constant between 0 and 1 that represents the fraction of the total number of windows, ω in the time series. In the simplest case, $c = 1$ and the correlation window equals ω. In other words, one correlation coefficient will be computed for the entire time series. For this paper, c is pre-specified by the participants. For ease of exposition, we will focus on the case where $c = 1$.

[1] In this paper, we assume that the time series are re-sampled to a common rate based on arrival rate, etc. The discussion of that is beyond the scope of this paper.

3 Overview of Method

At a high level, the proposed method for privacy preserving correlation identification can be broken down into four major steps: preprocessing, local representative value calculation, distribution data schemes, and correlation identification. In the remainder of this section, we describe each of these steps in more detail. The privacy schemes and breaches will be the focus of the next section.

3.1 Preprocessing Steps Prior to beginning, one of the participants is designated as the master location or the *leader*. The leader begins by sending the participants a number of parameters across a secure channel: date range (d), window size (n), number of participants (P), correlation window length (l_{CW}), privacy scheme (PS) and any parameters associated with the selected privacy scheme. [2]

The date range identifies the time period of the analysis. This is used to ensure that the time series are aligned along the time dimension. The window size defines the chunk of data processed at once and also specifies the number of representative values for the time series. The number of participants is used to determine parameters for the different privacy schemes. The correlation window length identifies the period over which to calculate correlations. Finally, the privacy scheme identifies the form of the representative time series R and the procedure to use for communicating R.

3.2 Local Representative Time Series Calculation There are different values that can be used as representative values for a time series. Here, we begin by describing the different statistics or behaviors that can be calculated in our framework. We then describe two method for calculating the representative value for each window.

3.2.1 Possible Behaviors Using our framework, we want the ability to correlate a number of different statistics or behaviors. Users are able to correlate one or more statistics as long as each statistic can be combined through addition of the scalar values. In other words, if we sum all the participant values of a particular statistic, the result is meaningful. We refer to statistics

[2] The participants can determine these parameters apriori. In that case, the leader only activates the process.

that can be combined through addition as having an *additive combining property*.

Statistics with this additive combining property that we consider in this paper are the following:

- Mean: the average value of the window.
- Median: the median of the data values in the window.
- Min: the minimum value of the window.
- Max: the maximum value of the window.
- Range: the difference between the maximum and minimum value of the window.
- First: the first value of the window.
- Last: the last value of the window.
- Difference: the difference between the last and first value of the window.
- AbsoluteDistance: the sum of the absolute differences between all the adjacent points in a window. This represents the absolute distance traveled.
- DirectionChanges: the number of direction changes in the window.

3.2.2 Basic and Scaled Representative Time Series To generate representative values for windows, we bin values based on the statistic of interest. In previous work, a binning approach was used for privacy of bursts [15]. Here, we consider two forms of binning, basic and scaled binning.

Basic binning puts raw values within each window into bins or buckets to obscure the actual data value. For example, if the behavior of interest is the mean sales, then the mean value of each window is calculated: $r_i = \frac{\sum_{j=0}^{n-1} e_{n*i+j}}{n}$.

Scaled binning uses a binned distance from the mean of the time series R generated using basic binning to generate representative values. To calculate the scaled bin value, the participant calculates the representative values for the behavior of interest R, takes the mean of the representative values $\mu(R)$, and the standard deviation of the representative values, $\sigma(R)$. Then for each window, the representative value is replaced with a scaled bin value rs_i based on the rounded distance between r_i, $\mu(R)$, and $\sigma(R)$: $rs_i = round(\frac{|r_i - \mu(R)|}{d \times \sigma(R)})$. In this equation, d is a small constant. A scaled bin value of 0 means that the representative value r_i for window i is within some constant d standard deviations away from the mean, positive or negative. A bin value of 1 means that r_i is more than d standard deviations away from the mean, but less than $2d$ standard deviations away. Scaled bin values increase in this manner. Each scaled bin may represent a fraction of a standard deviation $d = 0.5$, one standard deviation, $d = 1$ or multiple standard deviations, $d = 2$. Clearly, the selected value of d impacts the precision of the results.

To illustrate these binning techniques, we present an example in Table 1. The first row shows the time points in the time series. The second row contains 8 elements or data values in the time series. Suppose the size of each window is 2 ($n = 2$) and our behavior statistic is *min*. The final two rows of the table show the basic bin representative values and the scaled bin representative values with $d = 0.5$, respectively. Intuitively, this example demonstrates two points. First, the basic bin value can be very different from the original values depending on the behavior captured. Second, it is more difficult to determine the original time series values if the scaled bin values are shared instead of basic bin representative values. We will investigate this further in the next section.

Table 1: Binning Example: $n = 2$, $\mu = 13.25$, $\sigma = 1.33$

	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8
e_i	12	11	22	10	15	15	17	18
r_i-min	11		10		15		17	
rs_i-min	1		2		1		2	

3.3 Distribution Data Scheme We consider a number of different ways to distribute or share representative time series values - broadcasting, trusted collector and self-eliminating noise. Each of these distribution schemes leads to different privacy concerns that will be discussed in the next section.

3.3.1 Broadcasting In the broadcast distribution scheme, the participants communicate by broadcasting their representative time series R to each other using standard encryption techniques. We consider both basic binning and scaled binning representative values. Because R is broadcast to all the participants, correlations can be found for any pair or group of participants.

3.3.2 Trusted Collector We consider three different trusted collector distribution schemes. In all three distribution schemes, the participants send their representative time series R to a collector. Once again, for this distribution scheme either binning technique can be

used to generate representative values. The difference in the schemes is what is returned by the trusted collector. In the first trusted collector distribution scheme, the trusted collector returns a correlation matrix. In the second trusted collector distribution scheme, the trusted collector sends a reordered correlation matrix. In both of these schemes, each participant is told by the collector which row contains the participant's information.

In the final distribution scheme, the trusted collector creates an average behavior time series, A and returns it to the participants. In this case, pairwise correlations can not be identified by the participants. We pause to mention that for this final trusted collector distribution scheme, the collector can be eliminated and a secure multiparty computation protocol [8] can be followed by the participants to compute A. Doing so eliminates the bottleneck associated with having a collector. However, as the number of participants increases, the communication overhead and delay increase. In this paper we will focus on the trusted collector distribution scheme instead of the secure multiparty computation.

3.3.3 Self Eliminating Noise In this final distribution scheme, we do not want to share the representative time series values with the collector, but we still want to compute aggregate correlations. Instead, we consider a simple approach that involves insertion of 'self-eliminating' random noise to generate an average behavior representative time series for the different participants.

This idea is used in [17]. We paraphrase the explanation here. *Self-eliminating* noise is a vector of values that cancels out after the elements in one dimension are summed together. The vector $[6, -1, -2, -3]$ is an example of a self-eliminating noise vector since (6 + -1 + -2 + -3)=0. Self-eliminating noise removes the need to either aggregate representative time series of different participants in a round robin fashion or using a trusted collector.

A *noise generator* (or generator for short) is a participant assigned to generate self-eliminating noise. Each generator j creates a random noise list $N^{i,j}$ for each participant i. There are G participants that serve as generators. The protocol requires $G \geq 2$. If there is only one generator, $G = 1$, then the generator could intercept any participant's communication and remove the added noise to get A.

In order for the noise to be self-eliminating during aggregation of the representative time series, each generator must ensure that the sum of values across all generated, participant noise lists sum to zero at any index on the list:

$$\forall_{\ j \leq G}(\forall_{\ 1 \geq k \leq log(N)} \ \sum_{i=1}^{P} N^{i,j}[k] = 0)$$

By enforcing this constraint, the noise will cancel out once the individual representative time series are aggregated. The noise prevents participants from potentially causing a privacy breach while also avoiding more costly aggregation using a round robin secure summation scheme.

$N^{i,j}$ is distributed to each participant as a preprocessing step. Once participant i receives the random number list from the generators, the participant adds the next random number from each list to the next representative value. To ensure that the data from participants with unusually large deviations is adequately hidden, a maximum and minimum data value threshold can be enforced. Participants with values outside of those thresholds would replace the actual data value with the corresponding threshold value. The idea of thresholding is similar to the idea of transform thresholding presented in [17].

After the noise has been added for all the elements in R_k^i, participant i can send the 'noisy' representative vector $R*_k^i$ to the collector. An example of this is illustrated in Table 2. In this example, there are 2 noise generators and six participants. Each participant receives a list of random numbers. Each participant will take the first number from the list and add it to his first representative value. Then he will add the second number to the second representative value and so on. Once the representative values are aggregated by the collector, the added noise cancels out and the actual aggregate representative time series is obtained by the leader.

Table 2: $\mathbf{R}^{i,j}$ for $P = 6$ and $G = 2$

P	Generator 1			Generator 2		
1	0	-7	2	-1	-8	4
2	6	4	8	-2	-1	-4
3	-3	7	3	-1	5	6
4	2	-4	-1	2	-3	3
5	-5	4	-6	7	-2	1
6	-2	-4	-6	5	9	2
sum	0	0	0	0	0	0

Because the collector is not one of the noise generators, the collector does not see the actual representative time series of the participants during aggregation. For additional privacy, each generator can use different

means and variances for data generated for each participant as long as it is self-eliminating across all participants.

3.4 Correlation Identification We are interested in finding two types of correlations - one that correlates the behavior of an individual's time series, E_i, to another individual's time series, E_j and one that correlates a behavior of an individual's time series, E^i, to a time series that represents the average behavior of all the time series , A. Notice that in this formulation, each participant has a single time series. We can easily generalize this so that each participant has multiple time series. For ease of exposition, we focus on a single time series for each participant for the majority of the paper and discuss the multiple time series case at the end of the next section.

4 Privacy Analysis

We discuss the amount of privacy preserved for the different distribution schemes. What is the likelihood of determining the original time series when different representative values are shared or from the correlation results themselves. We also consider the privacy implications of sharing multiple behaviors for the same time series data, e.g. computing both the mean and absoluteDistance.

4.1 Privacy breaches We consider two different types of privacy breaches, an absolute breach and a moderate breach.

DEFINITION 1. *An absolute breach occurs if for any participant P_i, the local time series E^i can be determined by an adversary. A moderate breach occurs if for any participant P_i, the mean and variance of the representative values in time series R^i can be determined by an adversary.*

An absolute breach means that the adversary can determine a participant's exact time series. Sometimes getting close is also a problem. Therefore, we choose to define the moderate breach as one where the adversary knows the mean and variance of the representative time series. With that information the adversary can approximate the time series well and in the worst case, this approximation of the original time series is as good or better than R. The adversaries are presumed to be naive non-participants or semi-honest participants, i.e. do not exhibit malicious behavior, but will cheat if sufficient information is available to them.

Since the original time series E^i is never shared among participants, we need to understand what the adversary can do if he/she determines the representative time series, R^i. What is the probability that an absolute breach will occur when the adversary knows R^i? Depending on the distribution scheme, determining R^i may be straightforward. We defer discussing the privacy associated with different distribution schemes until the next subsection. For now, we assume the worst case, that the adversary can accurately determine R^i.

To determine the amount of noise introduced by using a representative value, we measure the discrepancy between the original time series, E, and the perturbed time series E^*, generated using R, as the normalized Euclidean distance, $D(E, E^*) = \frac{\sqrt{E^2-(E^*)^2}}{N}$. The amount of error introduced depends on five factors, the binning strategy used, the statistical behavior being measured, the window length n, and the variance between the representative value r_i, and the actual values in window i. We claim that the worst privacy case scenario is when the basic binning strategy is used, the window length is small, the variance is low and the statistical behavior of interest can bound the window values. We intuitively explain this claim.

Privacy Statement 1: Since basic binning has representative values that are of the same general magnitude as the original time series values and the magnitude information is lost when scaled binning representative values are used, the likelihood of a breach is higher when the basic binning strategy is used.

Privacy Statement 2: When the variance is low, the representative values are a good approximation of the actual values in each window.

While this is the case, the adversary does not know that the variance is low based on the representative values themselves if only a single behavior is being shared. Therefore, even though using the representative value to approximate all the values in a window will lead to a low discrepancy, potentially an absolute breach, $D(E, E^*)$, the adversary cannot be certain this is the case. If the variance is high, then using the representative value to approximate all the values in a window will lead to a high discrepancy. In this case, the probability of an absolute breach occurring is similar to randomly guessing each value in the window.

Privacy Statement 3: While a single statistical measure can bound the values in the window, the measure does not provide insight into the order of the elements in the window or the actual values of all the elements.

The different statistical measures also lead to different levels of privacy. If the first or last value of a window is the representative value r_i for window i, one value of the window is known, but all the other values in the window are not bounded by the representative value. The median or mean give more insight into the

values of the elements and will tend to be a better approximation of each individual behavior in the window. The minimum or maximum values of the window can be used to bound the range of possible values in the window from a single direction. In all these cases, even though the statistical behavior can bound window values, it does not tell the adversary the value of all the elements in the window or the order of all the elements in the window.

Privacy Statement 4: Increasing the size of the window, n, generally, reduces the ability of the adversary in determine the original time series values, thereby improving the amount of privacy.

When the window length n is small, more representative values are being used to approximate the time series. As n increases, r_i represents a larger number of events. If the magnitude of the events is not constant, then the error between the actual data values and the representative data value will increase as n increases.

To better quantify these ideas, we define a generic probability for determining the values in a window i to be $P(w_i) = \sum_{j=1}^{n} P(e_j|r_i)$, where $P(e_j|r_i)$ is the probability that e_j can be determined given that r_i is known. When the behavior does not bound the window, then e_j and r_i are independent and $P(e_j|r_i) = P(e_j)$. If n is large, the behavior is unbounded, and the variance is high, $P(w_i)$ is no better than a random guess, $P(w_i) = n \times \frac{1}{x}$, where x represents the size of the domain of E.

Privacy Statement 5: Using scaled binning reduces the probability of determining actual data values to that of a random guess for adversaries.

We now consider the additional of privacy introduced using scaled binning. First, we notice that the range of the scaled bin values is between 0 and some small constant. This is the case irrespective of the range of the original data, positive or negative. Intuitively, we have flattened the data and for added privacy, can take the absolute value. The adversary will not be able to reconstruct the original time series with the scaled bin values because the scaled bins do not give enough insight about the magnitudes of the elements in the original time series. In the worst case, if the adversary determines the representative value, he/she will know the distance to the mean of R. However, he will not know the value of $\mu(R)$ or $\sigma(R)$. Also, he will be uncertain about the values that are above the mean versus those that are below the mean if they are bucketed together. Therefore, the adversary's prediction of E is no better than a random guess.

We now consider whether the result returned to participants can help determine E. Does the correlation matrix provides insight into the actual data values

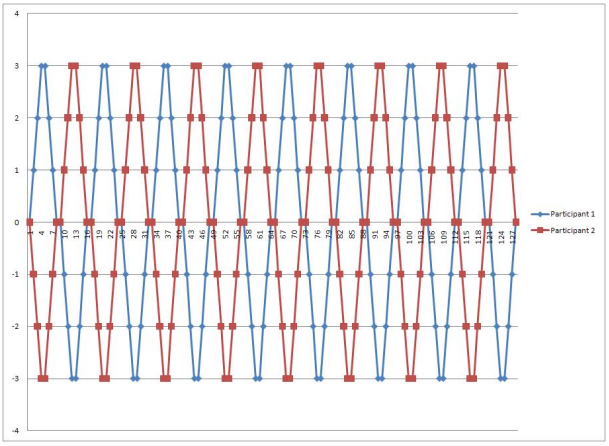


Figure 1: Participant time series having the same R time series

if that is what is shared among participants. Let's consider to two extreme cases, (1) two participants are extremely correlated and (2) the two participants are not correlated at all.

For case 1, these two participants can determine the shape or envelope of the other's time series. If R is known, then the correlation matrix does not provide any additional insight from R for an absolute breach to occur. While correlated data could imply similar time series values, Figure 1 shows an example where that is not the case. The figure shows two participant time series, one red and one blue, oscillate in opposite direction within the same time range The x axis represents time and the y axis represents the magnitude of the time series. The representative time series R_1 and R_2 for both these participants are the same and the correlation is 1. In this example, let us suppose the window size covers a single cycle. We show the values for all the different behaviors for a single window in Table 3 [3]. Theoretically, this may imply that if one of the participant's is an adversary and the two time series are completely correlated, the adversary may consider plugging in his/her own data distribution to approximate the other participant's time series. However, if the adversary does this, the final signal would approximate the other participant's time series accurately. Therefore, an adversary cannot gain insight based on having the correlation matrix.

For case 2, where the two participants are not correlated at all, if the adversary is a participant, the adversary assumes that his/her data distribution is different from his/her data distribution. However, many

[3]While the example is symmetric for ease of exposition, non-symmetric examples can also be generated

Table 3: Single window r for Figure 1 data

Statistic	r_i^1	r_i^2
Mean	0	0
Variance	3.73	3.73
Median	0	0
Min	-3	-3
Max	3	3
First	0	0
Last	0	0
Range	6	6
Difference	0	0
AbsoluteDistance	12	12
DirectionChanges	3	3

other distributions exist, making this close to a random guess.

Let us also consider the case when the correlation window is less than the size of the time series. Then for each correlation window, a correlation matrix is created. If the order of the individuals in each matrix is the same, then a participant P_k can see that his values are always correlated to a particular individual in the correlation matrices (even if he is uncertain about which particular participant it is). To remove this additional insight, we reordering successive matrices, thereby obscuring this information.

Another shared result is the aggregate time series, A. If the adversary knows that all the participants have the same distribution, then the privacy defaults to that of basic binning. Otherwise, the likelihood of determining individual time series is less than the cases described above.

The final shared result we consider is when multiple behaviors are shared.

Privacy Statement 6: For basic binning, an absolute breach is possible when all the representative time series are shared.

Proof Sketch: Suppose the adversary knows the mean, median, min, max, range, first, last, difference, absoluteDistance and directionChanges. Right away, the adversary knows the first, last and middle values of the bin. Then, using the absoluteDistance, min, max and directionChanges, it is possible to determine the other values in the window. A straightforward example assumes that the window has a single value v for each element. Then the mean = median = min = max = first = last = v, while the absoluteDistance = range = difference = directionChanges = 0. The adversary knows all the values and the order of the values, an absolute breach has occurred.

Privacy Statement 7: With scaled binning, the adversary cannot be certain about the original magnitude of the data. Therefore, even though the shape of the time series can be approximated, the actual values cannot be determined - a breach does not occur.

4.2 Privacy of distribution schemes In the previous subsection, we assumed that the adversary was able to determine the representative time series R_i for all the participants, P_i. In this subsection we consider how difficult it is to determine R using the different distribution schemes.

By definition, when broadcasting is used and basic binning is used, a moderate breach occurs. The representative times series are broadcasted to all the participants. If the adversary is one of the participants, he/she has all the representative time series. If the distribution scheme is broadcasting, but scaled binning is used for representative values, a moderate breach does not occur since, as described in the previous subsection, the domain and magnitude of events cannot be approximated better than a random guess. In fact, irrespective of distribution scheme, if scaled binning is used, the probability of determining the basic binning representative values does not increase.

In the other two distribution schemes, the representative values are not shared. The only information shared is the correlation matrix or the aggregate representative time series, A. As explained in the previous subsection, the correlation matrix and the aggregate time series do not improve the likelihood of determining the original participant time series. In the trusted collector scheme, the collector does have all the representative time series. In the worst case, this can lead to a moderate breach if the collector is an adversary. In the distribution scheme involving self eliminating noise, noisy representative time series values are shared, obscuring basic bin values. Therefore, a moderate breach does not occur for basic or scaled binning.

5 Experiments

In this section, we attempt to quantify the amount of utility and the accuracy of the calculated correlations under different privacy schemes. How quickly does the correlation accuracy decrease when we change the privacy related parameters? He we focus on correlation accuracy when varying window size, time series variance, behavior of interest, binning strategy, and bin scale size.

The data sets we consider include stock ticker, network packet, sensor, random walk, and synthetic time series, e.g. periodic, random, monotonic, etc. The majority of the non-synthetic data sets were obtained from the UCR time series data repository [11]. Information

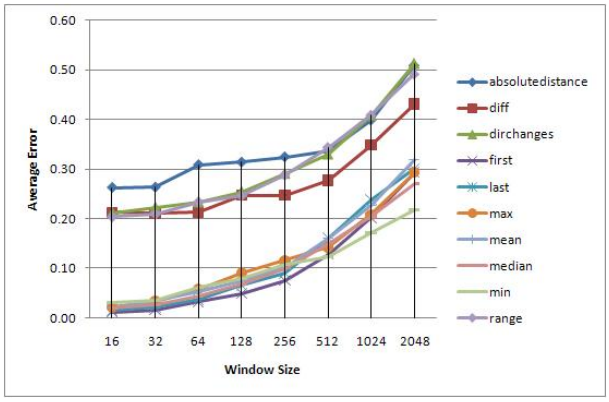
about the other data sets, including the synthetic ones can be found at (reference removed for double-blind). Again, while none of these data sets actually requires privacy, we consider these data sets because they serve as a representative set of time series data that has varying energy and noise characteristics.

5.1 Pairwise correlation detection accuracy for basic binning We now analyze correlation accuracy for pairwise correlations. A pairwise correlation is the Pearson correlation of time series data between two participants. Unless otherwise specified, the bin scale size, $d = 0.5$. Figure 2(a) compares the correlation error for different statistical behaviors and window sizes using basic binning. This is an average result across all the different data sets. As expected, as the window size increases, the error increases.

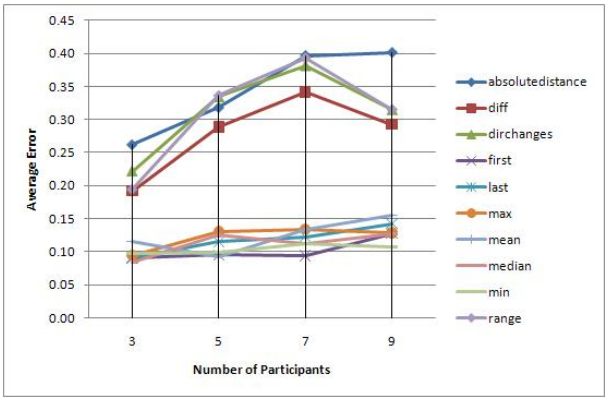
Figure 2(b) compares the correlation error for different statistical behaviors and different number of participants using basic binning. While the error does increase as the number of participants increases to a point, for some measure it decreases as the number of participants increase. We surmise that this results because of the variation in the participant data. Once the number of participants is greater than 5, the variation in the data between some of the participants decreases in our data sets.

Figure 2(c) shows the error for the different statistical behaviors for 3 participants, 5 data sets, and a window size of 16 elements. Due to space limitations, we cannot show the time series for each of these data sets. Instead, we highlight a few interesting findings. First, the ticker data has no noticeable error across any statistical behavior. The ticker data set is a relatively flat time series for most participants with little variance. Therefore, the representative values are very accurate approximations of the original signals. Second, in general, the error is relatively low. There are a few exceptions. For example, the absolute distance of the random walk data did particularly badly. We believe this results because of the high variability of random walk data. Notice, however, that while some behaviors did particularly poorly, others had negligible error.

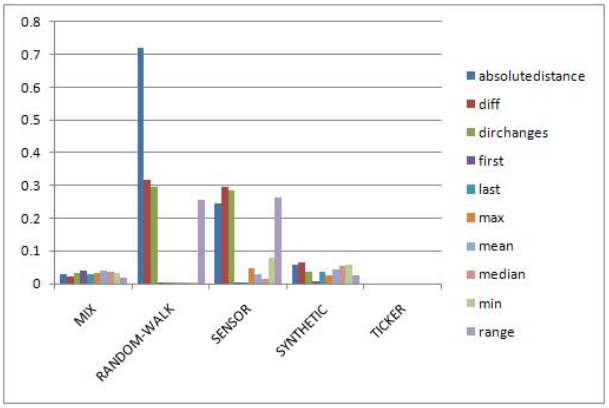
5.2 Pairwise correlation detection accuracy for scaled binning We now consider the scaled binning error rates for correlation. Figure 3(a) compares the correlation error for different statistical behaviors and window sizes using scaled binning. As expected, the error rate increases as the window size increases for the different statistical behaviors. Figure 3(b) compares the correlation error for different statistical behaviors and different number of participants using scaled binning.



(a) Correlation Accuracy Across Window Sizes



(b) Correlation Accuracy Across Number of Participants



(c) Correlation Accuracy Across Data Sets

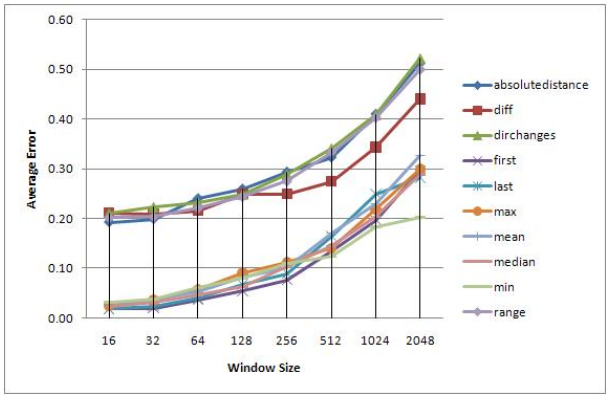
Figure 2: Pairwise Correlation Accuracy for Basic Binning Method

This graph behaves similar to the basic case. One should notice that the magnitude of the error is very similar, and in some cases lower. Finally, Figure 3(c) shows the error for the different statistical behaviors for 3 participants, 5 data sets, and a window size of 16 elements using scaled binning. While the general results are similar to the basic binning, the error rate for some behaviors of the random walk data are less than the basic binning case. In other words, there are times when the error rate is less. In fact, if we just take the average error across all the data sets, across all the window sizes, across all the number of participants, and across all the behaviors, the basic binning pairwise correlation error is 0.1899 and the error for the scaled binning pairwise correlation is 0.1872. The correlation accuracy using scaled binning is as good as basic binning with a higher privacy guarantee.

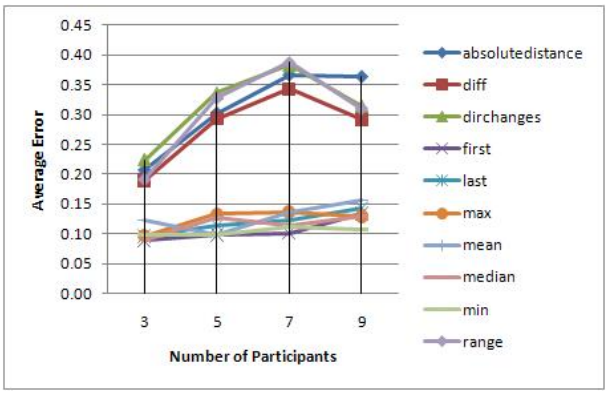
5.3 Aggregate correlation accuracy With the aggregated representative values, A, we find more of a difference between the basic binning and scaled binning correlation error. Figures 4(a) and 4(b) show the correlation error for the basic and scaled versions, respectively. In both cases, the error increases as the window size increases. However, for basic binning, there is more variation in the error depending on the behavior of interest, with some behaviors having much lower correlation error than others. For scaled binning, the slope of the error is smaller as the window size increases and error is more similar across the behaviors.

Figures 5(a) and 5(b) compare the aggregate correlation error for different statistical behaviors and different number of participants for both binning strategies. For basic binning, the shape of the error is similar to the pairwise case for many of the behaviors, peaking at five. For the scaled binning case, the error increases as the number of participants increases. However, the slope of the error does decrease as the number of participants increases. In general, the error is lower for basic binning than for scaled binning for these two charts.

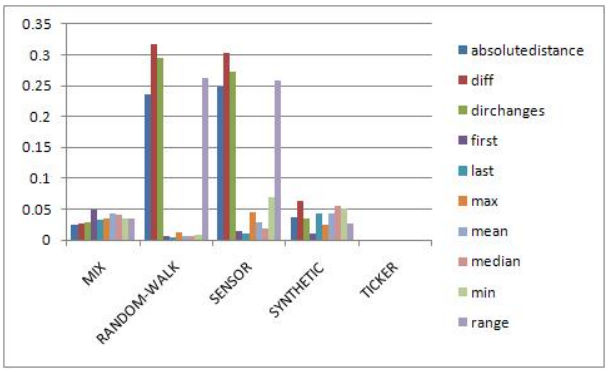
Finally, we do a comparison of the aggregate correlation for different data sets. Figures 6(a) and 6(b) show the correlation error for the basic and scaled versions, respectively. Here, we focus on the case with three participants and a window size of 16. When comparing the two figures, both do well on ticker data, both do poorly on the synthetic data, and the rest are mixed, depending on the behavior. Once again, we take the average error across all the data sets, across all the window sizes, across all the number of participants, and across all the behaviors. We find that the basic binning aggregate correlation error is 0.2467 and the error for the scaled binning aggregate correlation is 0.4596. Here, there is



(a) Correlation Accuracy Across Window Sizes



(b) Correlation Accuracy Across Number of Participants



(c) Correlation Accuracy Across Data Sets

Figure 3: Pairwise Correlation Accuracy for Scaled Binning Method

a clear difference between the correlation accuracy of basic binning and scaled binning.

5.4 Bin scale size In the previous experiments, the bin scale size d was constant ($d = 0.5$). Here we vary d to understand if the correlation accuracy changes as d increases for basic and scaled binning. Figure 7 plots the pairwise and aggregate correlation error as d increases for the basic and scaled binning methods. For the pairwise case, the error basically the same for both basic binning and scaled binning. Also as d increases, the error remains fairly constant. The error for the aggregate case was also fairly constant across both binning methods. However, in this case, the basic binning error was consistently less than scaled binning.

5.5 Discussion and Analysis When analyzing all these results, there are some interesting findings we want to highlight. First, for pairwise correlation, the correlation error for both the basic binning and scaled binning methods was essentially the same. This suggests that we should generally use scaled binning for pairwise correlations because the original signal cannot be reconstructed using the scaled bins and there is no penalty in terms of correlation accuracy. Also, it is sometimes useful to share many different behaviors. Because the accuracy is very similar across the different behaviors, when using scaled binning participants can share as many behaviors as they want without additional privacy concerns.

Another interesting result is that there is a difference in error between basic binning and scaled binning in the aggregate correlation results. The correlation error is significantly higher for the scaled binning method. In this case, there is a clear privacy/accuracy trade off.

Finally, as expected, as bin sizes increased, error rates increased for all the behaviors, particularly for data sets with a lot of variability. Therefore, while increasing the width of the window improves privacy, the correlation accuracy does decrease - again, a privacy/accuracy trade off.

6 Related Literature

A number of works related to privacy and time series data have been proposed [6, 14, 15, 17]. Li et al investigate privacy preservation in the context of streaming data [6]. They also investigate ways to maintaining the correlation and autocorrelation statistics of the data stream while hiding the raw data stream. They accomplish this by inserting random perturbations that mirror certain statistics of each window of data. Our approach is very different since we do not add random noise to the data.

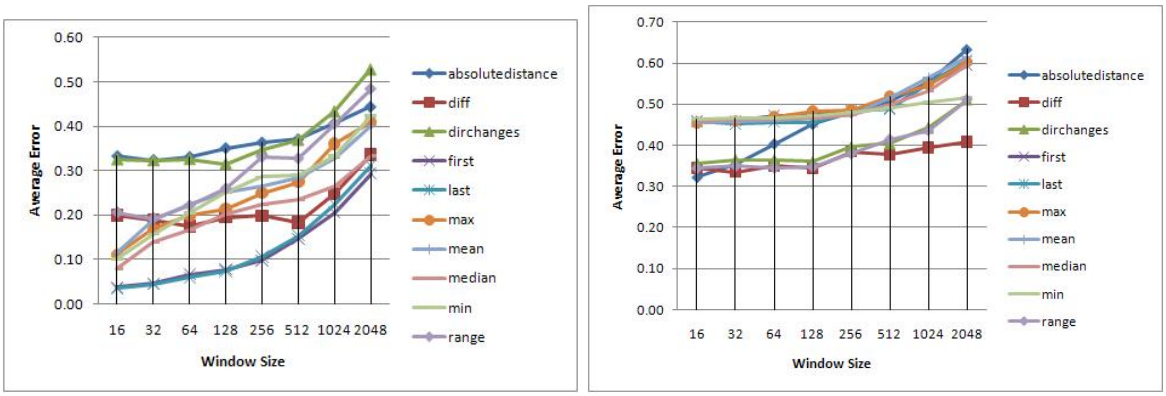
Papadimitrious et al investigate approaches for compressing time series data using Fourier transforms and wavelets transforms [14]. The focus of that paper is on reconstruction of the original time series, not correlations. Singh and Sayal also consider Fourier transforms and wavelet transforms for time series data [17]. They focus on finding bursts in the data using compressed representations of the original time series. Sayal and Singh propose a method for finding bursts using bins [15]. However, they do not consider correlations in that work.

Much tangential work in the area of privacy preserving data mining also exists. The work cited here is a representative sample of topics and is not meant to be an exhaustive list. Some work has investigated privacy preservation within the context of traditional data mining problems, e.g. classification [2] [20], association rule mining [5] [9], clustering [12] [19], and regression analysis [10, 7]. For a survey of current approaches and tools for privacy preservation in data mining, we refer you to [3] [21]. Some privacy preservation work exists on horizontally partitioned data sets [9] [13]. However, none of it is applied to the problem of correlation identification.

Finally, a great deal of work in the area of statistical databases focuses on providing statistics about data without compromising individual data values. There have been a number of data perturbation strategies used including swapping values [4], adding noise [18], and data partitioning [16]. While all these ideas are relevant to correlation identification, they differ because these works attempt to minimize the total error of the query result while blocking inference channels to sensitive data. In our case, we are only concerned with the error associated with correlations in the time series. Instead, we want to introduce as much bias or error as possible to improve the level of privacy without decreasing the accuracy of the correlation detection procedure. For a more detailed analysis of statistical databases, we refer you to [1].

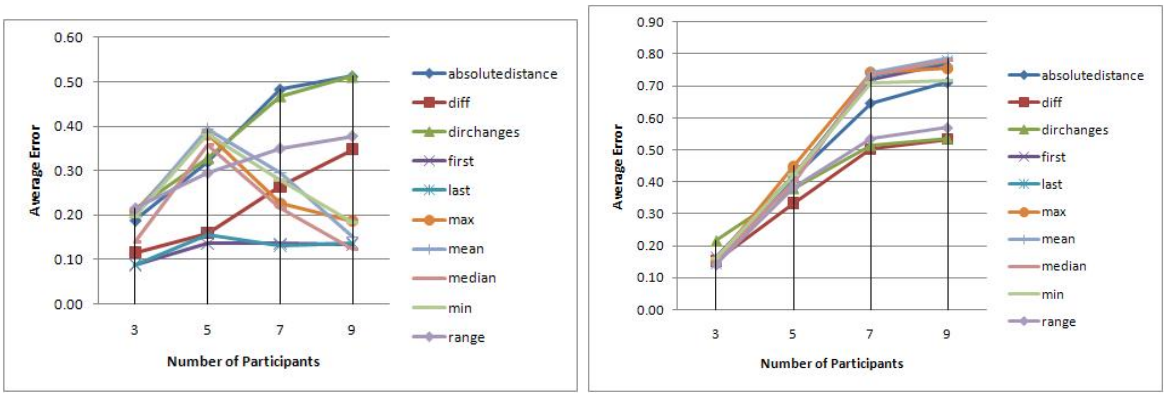
7 Conclusions

In this paper we describe a new method for identifying correlated statistics across independent participant sites. We show the trade off between the accuracy of the correlations and the privacy scheme used and propose a scaled binning method that maintains a high level of correlation accuracy and privacy. Our experiments on different time series data sets demonstrate the effectiveness of this technique. Future work includes investigating correlations for behavioral statistics that do not have the additive combining property and considering other interesting properties to measure.



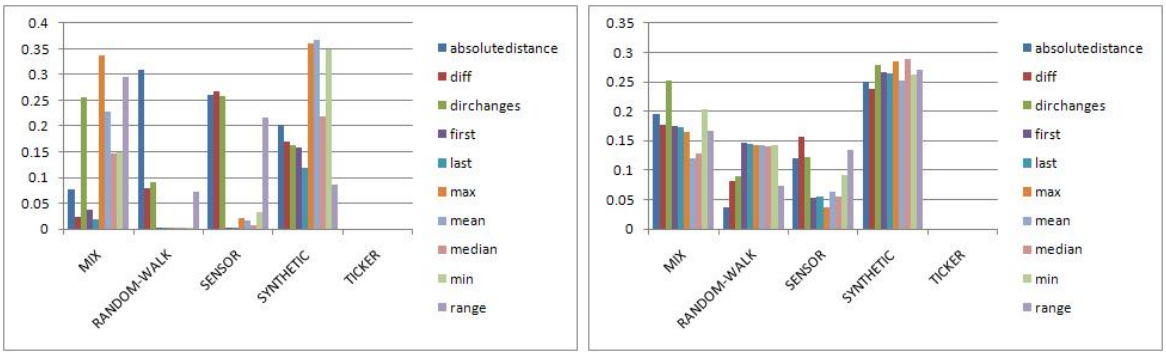
(a) Basic Binning (b) Scaled Binning

Figure 4: Aggregate Correlation Accuracy Across Window Sizes



(a) Basic Binning (b) Scaled Binning

Figure 5: Aggregate Correlation Accuracy Across Number of Participants



(a) Basic Binning (b) Scaled Binning

Figure 6: Aggregate Correlation Accuracy Across Data Sets

References

[1] N. R. Adam and J. C. Worthmann. Security-control methods for statistical databases: a comparative study. *ACM Computing Surveys*, 21(4):515–556, 1989.

[2] R. Agrawal and R. Srikant. Privacy-preserving data mining. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, pages 439–450. ACM Press, May 2000.

[3] C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, and M. Zhu. Tools for privacy preserving distributed data mining. *SIGKDD Exploration*, 4(2):1–7, 2003.

[4] D. E. Denning. *Cryptography and Data Security*. Addison-Wesley, 1982.

[5] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke. Privacy preserving mining of association rules. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, July 2002.

[6] S. P. Feifei Li, Jimeng Sun and et al. Hiding in the crowds: privacy preservation on evolving streams through correlation tracking. In *IEEE International Conference on Data Engineering*, 2007.

[7] S. E. Fienberg, Y. Nardi, and A. B. Slavković. Valid statistical analysis for logistic regression with multiple sources. *Protecting Persons While Protecting the People: Second Annual Workshop on Information Privacy and National Security, ISIPS 2008, New Brunswick, NJ, USA, May 12, 2008. Revised Selected Papers*, pages 82–94, 2009.

[8] O. Goldreich, A. Micali, and A. Wigderson. How to play any mental game. In *ACM Symposium on Theory of Computing*, pages 218–229, 1987.

[9] M. Kantarcioglu and C. Clifton. Privacy-preserving distributed mining of association rules on horizontally partitioned data, June 2002.

[10] A. Karr, X. Lin, J. Reiter, and A. Sanil. Secure regression on distributed databases. *Journal of Computational and Graphical Statistics*, 2004.

[11] E. Keogh. The ucr time series data mining archive. University of California, Riverside, Computer Science & Engineering Department. http://www.cs.ucr.edu/~eamonn/TSDMA/.

[12] X. Lin and C. Clifton. Privacy-preserving clustering with distributed em mixture modeling. *Knowledge and Information Systems*, 8(1):68–81, 2005.

[13] S. Merugu and J. Ghosh. Privacy-preserving distributed clustering using generative models. In *IEEE International Conference on Data Mining*, November 2003.

[14] S. Papadimitriou, F. Li, G. Kollios, and P. S. Yu. Time series compressibility and privacy. In *Proceedings of the International Conference on Very Large Data Bases*, pages 459–470. VLDB Endowment, 2007.

[15] M. Sayal and L. Singh. Detecting aggregate bursts from scaled bins within the context of privacy. In *IEEE ICDE Workshop on Privacy Data Management*, 2007.

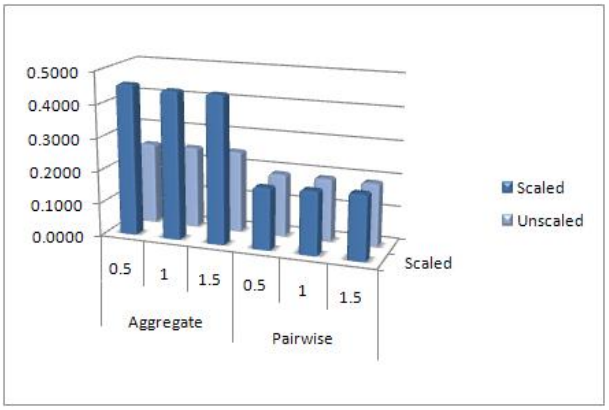


Figure 7: Comparison of Correlation Accuracy

[16] J. Schlörer. Information loss in statistical databases. *Computer Journal*, 26(3):218–223, 1983.

[17] L. Singh and M. Sayal. Privately detecting bursts in streaming, distributed time series data. *Data and Knowledge Engineering*, 68(6):509 – 530, 2009.

[18] J. F. Traub, Y. Yemini, and H. Wozniakowski. The statistical security of a statistical database. *ACM Trans. Database Syst.*, 9(4):672–679, 1984.

[19] J. Vaidya and C. Clifton. Privacy-preserving k-means clustering over vertically partitioned data. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 206–215, New York, NY, USA, 2003. ACM Press.

[20] J. Vaidya and C. Clifton. Privacy preserving naive bayes classifier for vertically partitioned data. In *SIAM International Conference on Data Mining*, 2004.

[21] V. S. Verykios, E. Bertino, I. N. Fovino, L. P. Provenza, Y. Saygin, and Y. Theodoridis. State-of-the-art in privacy preserving data mining. *SIGMOD Record*, 33(1), 2004.