

Compare Results

Old File:

effects_privacy_on_forecasting_MS (2).pdf

18 pages (437 KB)

8/28/2022 5:46:35 AM

versus

New File:

effects_privacy_on_forecasting (14).pdf

21 pages (354 KB)

9/14/2022 2:06:14 PM

Total Changes

794

Content

162 Replacements
186 Insertions
186 Deletions

Styling and Annotations

40 Styling
220 Annotations

Go to First Change (page 1)

The Effects of Privacy Protection on Forecast Accuracy

Cameron D. Bale, Matthew J. Schneider, Jinwook Lee

Drexel University Lebow College of Business

3220 Market St, Philadelphia, PA 19104

Abstract

Forecasts generated using protected time series change significantly from those using the original time series. While prior experiments have demonstrated severe degradations in forecast accuracy from a VAR model applied to differentially private time series, little is known about how privacy protection affects other forecasting models. We measure the effects of several data protection methods (top and bottom coding, additive noise, differential privacy, and cluster-based swapping) on both simple and complex forecasting models. We find that data protection degrades forecast accuracy the majority of the time regardless of forecast horizon. Surprisingly, when the time series are protected with differential privacy or additive noise, we find that exponential smoothing models have better accuracy than LGBM models for all forecast horizons. We investigate the reasons behind these results and offer guidance for practitioners in selecting a forecast model for privacy protected time series data.

Keywords: Etc., Etc., Etc.

1. Introduction

Forecasting is popular in a variety of fields, such as consumer analytics, renewable energy and power industries, and census tracking, all of which may benefit from the use of commercially or personally sensitive data. Examples include using social media information (Boone et al., 2019) and collaboratively shared power generation data (Gonçalves et al., 2021) to improve forecast accuracy. The privacy concerns associated with sensitive data have been demonstrated across many domains. Data ranging from consumer locations (De Montjoye et al., 2013) to smart meter usage (Véliz & Grunewald, 2018) can be used to identify individuals and/or infer sensitive information about them. Furthermore, a large number of privacy laws such as the General Data Protection Regulation (GDPR) require organizations to protect their sensitive data to avoid fines.

Various protection approaches are available depending on whether time series are stored in a single data set or spread across multiple data owners/data sets. In the multiple data owners scenario, techniques such as secure multi-party computation or federated learning enable privacy-preserving collaborative forecasting to ensure accurate forecasts while protecting sensitive data (Gonçalves et al., 2021; Sommer et al., 2021; Goncalves et al., 2021). On the other hand, we focus on scenarios in which a single data owner uses privacy methods to protect a time series data set. These privacy methods alter the sensitive data to produce protected time series which limit

¹For examples in the United States, see [this map](#).

the ability of a bad actor to identify data subjects or learn sensitive information about them. One example is the Census' use of random noise to perturb the individual and business level data that goes into calculating Quarterly Workforce Indicator data (Abowd et al., 2012). Privacy methods are attractive to organizations since when applied correctly, the data produced by these privacy methods can be exempt from privacy laws². The concern for forecasters is that privacy methods can drastically alter time series, leading to privacy adjusted forecasts. Empirical evidence of the effects of privacy methods on forecasts and the reasons behind changes in forecast accuracy would help forecasters adapt to using protected data.

While it has been demonstrated that differential privacy degrades forecast accuracy for VAR models and recurrent neural networks (RNNs) (Gonçalves et al., 2021; Imtiaz et al., 2020), there is no work which compares how multiple forecasting models perform on protected data. This comparison is needed because different forms of data protection produce different data points which will ultimately have different forecasts than what would be produced based on the original data. This paper provides an empirical analysis of forecasting with protected data. First, we investigate the drivers of changes in forecast accuracy for protected data. Specifically, we examine time series characteristics that give insight into why forecast accuracy changes. We develop a new matrix-based privacy method which swaps the values of time series with similar characteristics to balance the trade-off between privacy and forecast accuracy. We provide empirical results of forecast accuracy for protected data, and examine model-specific behavior to understand why certain models perform better than others. Motivated by findings from the judgmental forecasting literature, we investigate characteristics of privacy adjusted forecasts with improved forecast accuracy and assess the parallels between privacy and judgmentally adjusted forecasts.

The rest of the paper proceeds as follows. Section 2 reviews the relevant literature and...

2. Literature Review

2.1. Extant Studies on Data Privacy and Forecasting

Some forecasters have studied data privacy and forecasting in the context of collaborative forecasting. Gonçalves et al. (2020) explored a data market where data owners are compensated for sharing their data, and purchase forecasts based on the data from other parties. While data owners have a monetary incentive to share their data, they may be discouraged from doing so due to privacy concerns over sharing data with a central party. In such a situation, our work would help answer how forecast accuracy would be affected if the data owners applied data protection methods prior to sharing their data in the market. In the absence of a data market, other privacy-preserving solutions for collaborative forecasting include secure multi-party computation, decomposition-based methods, and data transformation techniques, all of which are succinctly described by Gonçalves et al. (2021).

Our interest is in privacy methods which generate protected data sets. The first methods we consider, known as additive or multiplicative noise and differential privacy, are based on incorporating random noise into the data. Given a confidential time series \mathbf{A} , a differentially private time series can be created using a randomized mechanism $\mathcal{M}(\mathbf{A}) = \mathbf{A} + \mathbf{N}$, which adds Laplace random noise \mathbf{N} with scale parameter $\Delta f_1 / \epsilon$. The sensitivity Δf_1 is determined as the maximum absolute difference between two time series \mathbf{A} and \mathbf{A}' , which differ in at most one

²See the description of *anonymous information* given in Recital 26 of the GDPR.

observation, where $\Delta f_1 = \max\{\|\mathbf{A} - \mathbf{A}'\|_1\}$. The mechanism \mathcal{M} satisfies ϵ -differential privacy by guaranteeing that, for every output \mathbf{t} of \mathcal{M} and every pair of series \mathbf{A} and \mathbf{A}' ,

$$\Pr(\mathcal{M}(\mathbf{A}) = \mathbf{t}) \leq \exp(\epsilon) \Pr(\mathcal{M}(\mathbf{A}') = \mathbf{t}). \quad (1)$$

Using this definition, Gonçalves et al. (2021) show that differential privacy reduces the forecast accuracy of VAR models even under very high values of the privacy parameter ϵ (weak privacy protection). Others have also studied the application of differential privacy to time series (Imtiaz et al., 2020; Fan & Xiong, 2012). Additive and multiplicative noise infuse random noise in the data but without the theoretical privacy guarantees of differential privacy. While Abowd et al. (2012) study the use of multiplicative noise, they do not offer forecast accuracy results. Through simulated data integrity attacks, however, we know that multiplicative noise reduces forecast accuracy Luo et al. (2018).

One interesting result from Imtiaz et al. (2020) is that differentially private data did not always produce worse forecast accuracy when forecasting individuals' health data using a recurrent neural network. Adding random noise to time series mirrors a technique used to prevent overfitting when forecasting with neural networks (Hewamalage et al., 2022, 2021). We explore whether data protection with random noise can achieve this same regularization at meaningful levels of privacy, and improve accuracy for recurrent neural networks.

Another privacy method is *generalization*, where data records are generalized to create equivalence classes of identical records. This privacy method is particularly popular for tabular data. The principle of k -anonymity (Sweeney, 2002) is used to describe when every record (or time series) is identical to at least $k - 1$ other records on a pre-determined set of attributes (or time periods). Nin & Torra (2009) evaluate the change in forecast accuracy for simple exponential smoothing, double exponential smoothing, linear regression, multiple linear regression, and polynomial regression applied to k -anonymized data. The authors find an overall reduction in forecast accuracy even for $k = 2$, but do not provide the accuracy of each model individually.

There are also privacy methods which are commonly used in practice but have not been studied in the forecasting literature. Top- and bottom-coding are used to replace the top (bottom) p percent of observations with the $1 - p$ (p) quantile. These techniques are useful for protecting data with sensitive values in the tails of distributions, such as income levels or smart meter data. Crimi & Eddy (2014) study the effect of top coding the Census' Public Use Microdata Samples on analyses of interest. They find that the sample correlation between two variables is shrunk towards zero when one or both of the variables are top coded. This may be relevant to multivariate forecasting model accuracy, which relies on the correlations between time series, and may be negatively affected when series are top- or bottom-coded. On the other hand, top- and bottom-coding could have an effect similar to adjusting for outliers, which can improve forecast accuracy when the outliers are close to the forecast origin (Chen & Liu, 1993).

Overall, while recent attention has been paid to privacy preserving collaborative forecasting, our interest is in forecasting using a single protected dataset. There has been no work which compares multiple forecasting models' accuracies when forecasting for a single protected dataset, or a comparison of models' accuracies under various privacy methods. The works which have shown that data protection degrades forecast accuracy have also not given detailed explanations as to why model performance is worse on protected data. Finally, there exist no privacy methods which are specifically designed with forecasters in mind.

2.2. Privacy Adjusted Forecasts

Judgmental adjustments to forecasts can improve accuracy by accounting for information that was not incorporated into a forecasting model (Fildes et al., 2009). Incorporating the intuition and experience of the adjuster, knowledge of special events, or insider or confidential information can add information with high diagnosticity that is useful for forecasting. However, adding information with low diagnosticity can degrade forecast accuracy Fildes et al. (2019). Adjusting forecasts for the sake of gaining control of the forecasting process, incorporating practitioner expectations, and compensating for judgmental biases can be detrimental to forecast accuracy ((Petropoulos et al., 2022), section 3.7.3). Despite varying motivations for adjusting forecasts, judgmental adjustments have been found to improve the accuracy of monthly demand forecasts from statistical models by an average of 10% (Davydenko & Fildes, 2013). The accuracy improvements are greater for low volatility time series which are easier to forecast (Fildes et al., 2009).

The characteristics of adjustments have an effect on forecast accuracy as well. Both positive and negative adjustments can improve accuracy, but positive adjustments tend to give only a marginal improvement (Davydenko & Fildes, 2013). Forecast bias can be reduced by negative adjustments, whereas positive adjustments maintain bias or exacerbate it (Fildes et al., 2009). The size or magnitude of judgmental forecast adjustments is positively associated with the size of accuracy improvements, which can occur when larger adjustments are made by adjusters who are confident in reliable information (Fildes et al., 2009).

The reasons for applying privacy methods are varied as well. Privacy legislation places strict limitations on data transfers and processing³ which can hurt business performance. Goldfarb & Tucker (2011) found that privacy regulation in the European Union led to a 65% average reduction in banner ad effectiveness at influencing purchase intent. Legal limitations can be circumvented when data is properly protected², but this comes at the cost of reducing the utility of the data. Data from regulated domains, such as healthcare (Centers for Medicare & Medicaid Services, 1996) and finance⁴ must also be protected. Other reasons for implementing privacy methods include reducing consumers' privacy concerns (Martin et al., 2017) or attempting to gain a competitive advantage through privacy-conscious brand positioning (Goldfarb & Tucker, 2013). Several of the largest tech companies in the world, including IBM (Holohan et al., 2019), Google⁵, Meta (Yousefpour et al., 2021), and Microsoft⁶, implement privacy methods and provide open-source code to enable others to do the same. Notably, Apple has positioned themselves as a privacy-focused company⁷.

Regardless of the motivation for data protection, privacy adjusted forecasts arise from changes to the data, which are made without regard to the effects on forecast accuracy. Privacy methods based on random noise add information with low diagnosticity, and are likely to reduce forecast accuracy. While the direction of adjustment is purposefully chosen in judgmental forecasting, the direction of privacy adjustments will occur indirectly via data protection. Under data protection, the adjustment size will again be determined by the forecasting models' responses to data protection. These responses are likely related to the strength of data protection, where stronger data protection results in larger changes to the data.

³For examples, see articles 6, 45, and 46 of the GDPR.

⁴See the Gramm-Leach-Bliley act.

⁵See several python libraries including PipelineDP and PyDP.

⁶See the OpenDP project.

⁷See descriptions of Apple's privacy features.

Certain time series characteristics, such as the strength of trend and spectral entropy, are related to forecast accuracy. These characteristics are useful for predicting the performance of forecasting models and predicting optimal forecast combinations (Li et al., 2022; Talagala et al., 2022). Due to their relationship with forecast accuracy, we analyze changes in these characteristics resulting from data protection to give insights into why forecasts change.

Our contributions are two-fold. First, we analyze privacy adjusted forecasts for multiple forecasting models and privacy methods, giving detailed explanations as to why model performance changes on protected data. To explain the improvement and/or degradation of forecast accuracy from data protection, we analyze privacy adjusted forecasts from two perspectives: (1) How data protection changes time series characteristics which translate into changes in forecast accuracy, and (2) How changes to forecasts are related to changes in forecast accuracy. To address (1), we use the approach outlined by Kang et al. (2017) to extract measures of forecastability such as spectral entropy and first order autocorrelation parameters which show fundamental changes to the data from data protection, and help explain why certain models perform better than others on protected data. To address (2), we measure whether forecasts are positively or negatively adjusted under data protection as well as the magnitude of each adjustment. Similar to Fildes et al. (2009) and Khosrowabadi et al. (2022), we classify privacy adjusted forecasts as either improving or degrading forecast accuracy. We use the random forest approach of Khosrowabadi et al. (2022) to identify the time series characteristics and adjustment features that are predictive of whether privacy adjusted forecasts improve or degrade accuracy.

For our second contribution, we propose a novel privacy method designed with forecasters in mind. Motivated by the random forest results, we implement a matrix-based privacy method which swaps the values of time series with similar characteristics to help maintain forecast accuracy. Results show that our method... We describe our proposed method next.

3. The k -nearest Time Series (nTS) Swapping Method

Let $\mathbb{X} = \{x_1, \dots, x_J\}$ be a given set of time series data (n -vectors). Using the standard privacy methods shown later in Table 1, a data provider releases protected data $P_{j,t}$ for each time series x_j based on the confidential values $A_{j,t}$ up until time t . The main issue with these methods is that they choose protected values based on predefined rules and not changes in forecast accuracy. However, the goal of the data provider should be to change $A_{j,t}$ to $P_{j,t}$ with minimal reductions in forecast accuracy while increasing privacy to an acceptable threshold.

We solve the data protection problem for the data provider using a matrix based k -nTS (k -nearest time series) swapping method, where the data provider releases a set of protected time series $\mathbb{X}' = \{x'_1, \dots, x'_J\}$, where $x'_j = (P_{j,1}, \dots, P_{j,t})^T$ is based on \mathbb{X} , the confidential values of all series up through time t . The k -nTS method matches time series that have similar characteristics within a rolling window of their past values. Then, it uses randomization to replace each $A_{j,t}$ with a confidential value from another time series with similar characteristics to balance the trade-off between forecast accuracy and privacy.

Depending on the quantity of available data, k -nTS can use rolling windows of data which adjust for dynamic changes in the relationships between time series. For example, if we choose a rolling window size, say n , then $x_j = (A_{j,t-n+1}, A_{j,t-n+2}, \dots, A_{j,t-1}, A_{j,t})^T$ where $x_j \in \mathbb{R}^n$. Protection in subsequent time periods from $t+1$ to T rolls x_j forward from $x_j = (A_{j,t-n+2}, A_{j,t-n+3}, \dots, A_{j,t}, A_{j,t+1})^T$ to $(A_{j,t-n+1}, A_{j,t-n+2}, \dots, A_{j,t-1}, A_{j,t})^T$, respectively. We label the time series characteristics for the current window as $f_{j,t}$, which we refer to as the feature vector for time series j in time period t based on the n values in $x_j = (A_{j,t-n+1}, A_{j,t-n+2}, \dots, A_{j,t-1}, A_{j,t})^T$. For simplicity, we omit the

t subscript for the feature vectors, and write f_j . See Li et al. (2022) for another example of the computation of time series features based on a rolling, fixed-length window.

For each time series $x_j \in \mathbb{R}^n$, the data provider computes the feature vector $f_j \in \mathbb{R}^m$. This vector can contain any single-valued feature calculated based on the values x_j , such as the strength of the trend and seasonality, the spectral entropy, or the mean of the values in the current window. This produces a set $\mathbb{C} = \{f_1, \dots, f_J\}$ of m -vectors containing the characteristics of each of the J time series windows. For each of the feature vectors $f_j \in \mathbb{R}^m$, the data provider computes a set of squared distances of the elements of the set \mathbb{C} . We define $\text{dist}(f_j, f_i) = d_{j,i}$ as the distance between f_j and f_i , i.e., the feature vectors corresponding to two distinct time series from a given set \mathbb{X} . Without loss of generality, we use the Euclidean norm, or ℓ^2 -norm, as a distance metric⁸. Since our case is multivariate and partially ordered, we can get a totally ordered set based on the Euclidean distance.

Let us define $x_j^{(k)}$ as the k th nearest neighbor of x_j , with corresponding feature vector $f_j^{(k)}$. Then, for a time series x_j , we have $\{d_{j,(1)}, d_{j,(2)}, \dots, d_{j,(J-1)}\}$ such that $d_{j,(k)} \leq d_{j,(l)}$ for any integers $k < l$ where $d_{j,(k)} = \|f_j - f_j^{(k)}\|$. Note that $x_j^{(i)} \in \mathbb{X} \setminus \{x_j\}$ and the superscript (i) means the i^{th} order statistic of the related Euclidean distances of all $f_j^{(i)} \in \mathbb{C} \setminus \{f_j\}$ from f_j . Thus, for a given time series vector x_j , its k -nTS (k -nearest time series) can be represented as the set $K_j = \{x_j^{(1)}, \dots, x_j^{(k)}\}$ based on $\|f_j - f_j^{(1)}\| \leq \dots \leq \|f_j - f_j^{(k)}\|$ or an ordered set $\{d_{j,(1)}, d_{j,(2)}, \dots, d_{j,(k)}\}$.

For more efficient computation of such ordering, we introduce a symmetric distance matrix D containing the squared distances between time series feature vectors. The squared distance between f_i and f_j is given by $d_{i,j} = \|f_i - f_j\|^2$, and $d_{i,j}$ is the (i, j) th entry of D (also note that $\text{rank}(D) \leq m + 2$). Suppose that we are given a data matrix $X = [x_1, x_2, \dots, x_J]$, $x_j \in \mathbb{R}^n$ (i.e., $X \in \mathbb{R}^{n \times J}$). We can write X , i.e., a confidential data matrix, as the following:

$$X = [x_1, x_2, \dots, x_J] = \begin{pmatrix} A_{1,t-n+1} & A_{1,t-n+2} & A_{1,t-n+3} & \cdots & A_{1,t-1} & A_{1,t} \\ A_{2,t-n+1} & A_{2,t-n+2} & A_{2,t-n+3} & \cdots & A_{2,t-1} & A_{2,t} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ A_{J,t-n+1} & A_{J,t-n+2} & A_{J,t-n+3} & \cdots & A_{J,t-1} & A_{J,t} \end{pmatrix}^T \quad (2)$$

where $x_j = (A_{j,t-n+1}, A_{j,t-n+2}, \dots, A_{j,t-1}, A_{j,t})^T$ where $x_j \in \mathbb{R}^n$ and $x_j \in \mathbb{X}$. We calculate the desired features based on each x_j and construct a feature matrix C (where $C \in \mathbb{R}^{m \times J}$) as follows,

$$C = [f_1, f_2, \dots, f_J] = \begin{pmatrix} f_{1,1} & f_{1,2} & f_{1,3} & \cdots & f_{1,m-1} & f_{1,m} \\ f_{2,1} & f_{2,2} & f_{2,3} & \cdots & f_{2,m-1} & f_{2,m} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ f_{J,1} & f_{J,2} & f_{J,3} & \cdots & f_{J,m-1} & f_{J,m} \end{pmatrix}^T. \quad (3)$$

We calculate the matrix D using the fact that $\|f_i - f_j\|^2 = (f_i - f_j)^T (f_i - f_j) = f_i^T f_i - f_i^T f_j - f_j^T f_i + f_j^T f_j$, which can be written up as the following:

$$D = \mathbf{1} \text{diag}(C^T C)^T - 2C^T C + \text{diag}(C^T C) \mathbf{1}^T, \quad (4)$$

⁸ All norms on \mathbb{R}^n are equivalent to the Euclidean norm.

Algorithm 1 The k -nTS Swapping method

Require: [Initialization]

(i) [Time Series Matrix $X \in \mathbb{R}^{n \times J}$] $X = [x_1, x_2, \dots, x_J]$, $x_j \in \mathbb{R}^n$ for $j = 1, \dots, J$ as in (2).

(ii) [Feature Matrix $C \in \mathbb{R}^{m \times J}$] $C = [f_1, f_2, \dots, f_J]$, $f_j \in \mathbb{R}^m$ for $j = 1, \dots, J$ as in (3).

(iii) [Distance Matrix D] $D = \mathbf{1} \text{diag}(C^T C)^T - 2C^T C + \text{diag}(C^T C) \mathbf{1}^T$ as in (4).

for $j = 1, 2, \dots, J$ **do**

[Finding a set K_j for x_j] Let d_j denote the j th column of D . Sort d_j from the smallest to largest components and find the k th smallest component, followed by K_j as in (5).

[Random swapping] $x_j \leftarrow x_j^{(i)}$ (last components only) for some $i \in \{1, \dots, k\}$ as in (6).

end for

where the symbol $\mathbf{1}$ denotes a column vector of J ones⁹. It is easy to see that the column vector $\text{diag}(C^T C) = (\|f_1\|^2, \dots, \|f_J\|^2)^T$. Let d_j denote the j th column of D . Then we can write the $J \times J$ distance matrix $D = [d_1, \dots, d_J]$, where $d_j \in \mathbb{R}^J$.

In the general case where $k \ll J$, for each time series x_j we sort d_j , the j th column of D from the smallest to largest components and find the k th smallest component so that we have

$$K_j = \{x_j^{(1)}, \dots, x_j^{(k)}\}. \quad (5)$$

That is, the data provider then selects a value of k from 1 to a maximum of $J - 1$ and selects the k -nearest time series to x_j based on the m features. In case of $k = J - 1$, random swapping is simply done by rearranging the components in the last row of matrix X . Let the i th most similar time series to x_j be $x_j^{(i)} = (A_{j,t-n+1}^{(i)}, A_{j,t-n+2}^{(i)}, \dots, A_{j,t}^{(i)})^T$ where n is the length of the rolling window of past data. Then the swap of the last component of x_j with the last component of one of its k -nearest time series, $x_j^{(i)} = 1, \dots, k$ can simply be written as the following random swapping:

$$k\text{-nTS Swapping : } P_{j,t} = \begin{cases} A_{j,t}^{(1)} & \text{with probability } \frac{1}{k} \\ \vdots \\ A_{j,t}^{(k)} & \text{with probability } \frac{1}{k}, \end{cases} \quad (6)$$

which is equivalent to the following: the last component of x_j is randomly replaced by the last component of $x_j^{(i)} \in K_j$ with probability $\frac{1}{k}$ for $i = 1, \dots, k$.

By Algorithm 1, we can obtain X' : a matrix of protected time series data at time point t for all J time series for a given rolling window size n . The k -nearest time series data protection method can be written up as the following protected data matrix

$$X' = [x'_1, x'_2, \dots, x'_J] = \begin{pmatrix} A_{1,t-n+1} & A_{1,t-n+2} & A_{1,t-n+3} & \cdots & A_{1,t-1} & P_{1,t} \\ A_{2,t-n+1} & A_{2,t-n+2} & A_{2,t-n+3} & \cdots & A_{2,t-1} & P_{2,t} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ A_{J,t-n+1} & A_{J,t-n+2} & A_{J,t-n+3} & \cdots & A_{J,t-1} & P_{J,t} \end{pmatrix}^T. \quad (7)$$

⁹Note that we could also define a distance matrix based on the actual time series values x_j , where D would become a function of X rather than C .

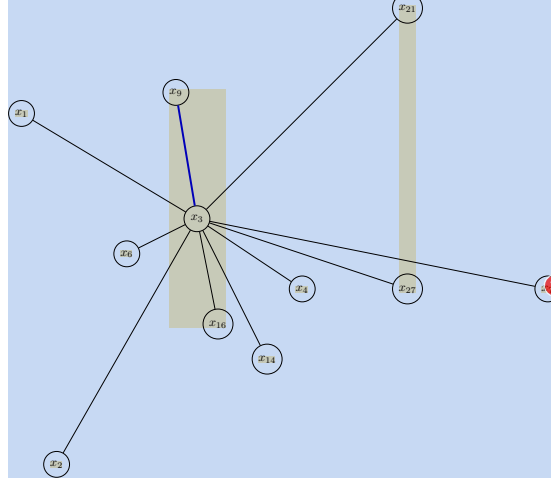


Figure 1: Random edge selection for k -nTS Swapping

As an example, consider a given time series x_3 where its last component is replaced by the last component of x_9 which was randomly selected among x_3 's 10-nearest time series. Using our notation, we can write

$$K_3 = \{x_1, x_2, x_4, x_6, x_7, x_9, x_{14}, x_{16}, x_{21}, x_{27}\}.$$

We can represent each time series x_j , $j = 1, \dots, J$ as a vector, and then put them in a graph $G = (V, E)$, which consists of a set V of vertices (or nodes) and a set E of undirected edges. In our case, we can use weighted edges to represent the Euclidean distance between associated time series feature vectors: $w_{i,j} = \text{dist}(f_i, f_j)$. If we put all the nodes on the graph and assign weight on every edge (every pair of nodes), e.g., $w_{i,j} = \text{dist}(f_i, f_j)$ for all $i \neq j$, then we will have a complete graph. The k -nearest time series swapping method can be considered as a random edge selection problem of the graph. Figure 1 depicts the case of $k = 10$ for the k -nearest time series of x_3 , where the last component of x_3 is swapped with that of x_9 .

4. Empirical Application

4.1. Standard Privacy Methods

4.2. Data

Complicated forecasting models are known to forecast more accurately than simple models using the unprotected version of the M3 competition monthly micro data (Koning et al., 2005), and models that explicitly capture trend and seasonality performed the best in the overall M3 competition (Makridakis & Hibon, 2000). We are interested in whether these results hold when forecasting using protected versions of the data. For our analyses, we use the monthly micro dataset from the M3 competition, which includes 474 strictly positive time series with values ranging from 120 to 18,100 (Makridakis & Hibon, 2000). Of the 474 series, 18 consist of 67 time periods, 259 consist of 68 time periods, and 197 consist of 125 time periods.


Table 1: Standard Data Protection Methods at Time t

Data Protection Method	Description	Formulation
None	Release confidential observation	$P_{j,t} = A_{j,t}$
Bottom-Coding	Bottom p percent of observations are replaced with the p quantile	$P_{j,t} = \begin{cases} B & \text{if } A_{j,t} \leq B \\ A_{j,t} & \text{if } A_{j,t} > B \end{cases}$ where $B = \inf\{x_j \in \mathbb{R} \mid F(x_j) \geq p\}$
Top-Coding	Top p percent of observations are replaced with the $1 - p$ quantile	$P_{j,t} = \begin{cases} T & \text{if } A_{j,t} \geq T \\ A_{j,t} & \text{if } A_{j,t} < T \end{cases}$ where $T = \sup\{x_j \in \mathbb{R} \mid F(x_j) \leq p\}$
Additive Noise	Add a normal random number with mean zero and standard deviation σ	$P_{j,t} = A_{j,t} + r, \text{ where}$ $r \sim N(0, \sigma)$ $\sigma = s * E[(x_j - E[x_j])^2]$
Differential Privacy	Add a laplace random number with mean zero and scale parameter $\Delta f_1 / \epsilon$	$P_{j,t} = A_{j,t} + r, \text{ where}$ $r \sim Lap(0, \Delta f_1 / \epsilon),$ $\Delta f_1 = \max\{\ f(\mathbf{A}_j) - f(\mathbf{A}'_j)\ _1\},$ and ϵ is given

4.3 Forecasting Models

In this section, we introduce the forecasting models which we apply to the original and protected data. The models are separated into ‘simple’ models which are trained to forecast one series at a time, and ‘complex’ models which are trained to generate forecasts for multiple series. For all models, we perform minimal data pre-processing, and allow the models to capture the important components of the series. Our goal is to assess the effects of privacy protection on the accuracy of popular forecasting models which are readily available to implement in R and/or Python, and have served as benchmarks or winners in recent forecasting competitions. Please see the appendix for full implementation details, including hyperparameter optimization for the complex models.

4.4 Simple Models



 We consider four simple models. The first three are common variants of exponential smoothing: (1) single exponential smoothing; (2) double exponential smoothing with an additive trend; and (3) triple exponential smoothing with additive trend and seasonality components. Our final simple model is auto-arima which automatically selects the best seasonal ARIMA model parameterization. All of the simple models were implemented in Python using the sktime forecasting module (Löning et al., 2022).

A SES model can be written as follows:

$$\begin{aligned}\hat{y}_{t+h|t} &= \ell_t \\ \ell_t &= \alpha y_t + (1 - \alpha)\ell_{t-1},\end{aligned}\tag{8}$$

where ℓ_t is the level of the series at time t and α is the smoothing parameter for the level. The SES model is suitable for series that do not exhibit trend or seasonality.

4.5 Complex Models

  We also consider three complex models. The first is a variant of the VAR model known as VARX-L (Nicholson et al., 2017), which we fit using the BigVAR package in R. Fitting large VAR models is already computationally expensive since the number of estimated coefficients is $K + pK^2$, where K is the number of series and p is the number of lags (Hyndman & Athanassopoulos, 2018). The problem is compounded in our case since we have fewer time periods than series. We use the VARX-L model which implements the Lasso penalty to induce sparsity in the coefficients and enable model estimation, and fit one VAR model to each subset of series with the same length, resulting in three VARX-L models being estimated per data set.

The second complex model we consider is a light gradient boosted machine (LGBM) (Ke et al., 2017), which was the top performing model in the M5 competition (Makridakis et al., 2022). This is a global forecasting model trained on all time series in our data set. We use the LGBM implementation provided in the Darts python module (Herzen et al., 2022).

Our final complex model is an RNN with long short-term memory cells (LSTM). This model was a component in the winning solution to the M4 competition (Smyl, 2020; Makridakis et al., 2018). Similar to the LGBM, the RNN is a global forecasting model, and we used the implementation provided in Darts (Herzen et al., 2022).

5. Results

We apply the standard privacy methods outlined in Table 1 for the parameter values shown in Table 2. In section 5.1, we use the random forest approach of Khosrowabadi et al. (2022) to identify the changes in time series features that are most predictive of whether privacy adjusted forecasts have improved or degraded accuracy. Section 5.2 describes the use of these features in the feature vectors f_j in our k -nTS swapping method, which we apply using the k values shown in Table 2. We then examine specifically how each privacy method affects these time series features, and how these relate to the behavior of each model on protected data in Section 5.3. These results help explain the forecast accuracy and forecast error distribution of each forecasting model under each privacy method, which we highlight in Sections 5.4 and 5.5. Finally, in Section 5.6 we study privacy adjusted forecasts through the lens of judgmental forecasting, and examine how adjustment characteristics such as magnitude and direction relate to the accuracy of privacy adjusted forecasts, and compare our results with judgmentally adjusted forecasts.

Privacy Method	Parameter	Values
Top- and Bottom-Coding	p	0.1, 0.2, 0.4
Additive Noise	s	0.5, 1, 1.5, 2
Differential Privacy	ϵ	0.1, 1.0, 4.6, 10.0, 20.0
k -nTS Swapping	k	5, 10, 15

Table 2: Parameter values for privacy methods.

5.1. Predicting Accuracy of Privacy Adjusted Forecasts

There are thousands of time series features which could predict the accuracy of privacy adjusted forecasts (Fulcher & Jones, 2014). Rather than extracting as many features as possible, we focus on a set of interpretable features used by Bandara et al. (2020) for clustering and forecasting similar time series, which were chosen to capture the majority of time series dynamics. We expand this feature set slightly to include additional interpretable features offered in the `tsfeatures` package in R (Hyndman et al., 2022), giving us twenty-two features to consider which are shown in the appendix.

For each protected data set (one for each parameter value for the standard privacy methods in Table 1) and the original data set, we calculate the feature matrix $C \in \mathbb{R}^{m \times J}$ where $m = 22$ and $J = 474$. Similarly, for the privacy adjusted forecasts based on each protected dataset, we compute a binary variable $v \in \{0, 1\}$ that is equal to one when the accuracy of the privacy adjusted forecast is improved relative to the original forecast. We then calculate the differences in time series features, i.e., $C_{diff} = C_{orig} - C_{protected}$ between the feature matrix for the original series and each feature matrix for the protected series. We use a random forest to identify the most important features for predicting v using the changes in time series characteristics that result from data protection under the standard privacy methods. However, random forests struggle to properly identify feature importance in the presence of highly collinear predictors, so we use an intermediate feature selection step. We rank the features' on their usefulness for predicting v using the minimum Redundancy - Maximum Relevance algorithm (Ding & Peng, 2005), and use the ten most relevant features in our random forest. Based on the random forest results, the features that are most important for predicting v are used in our k -nTS swapping method.

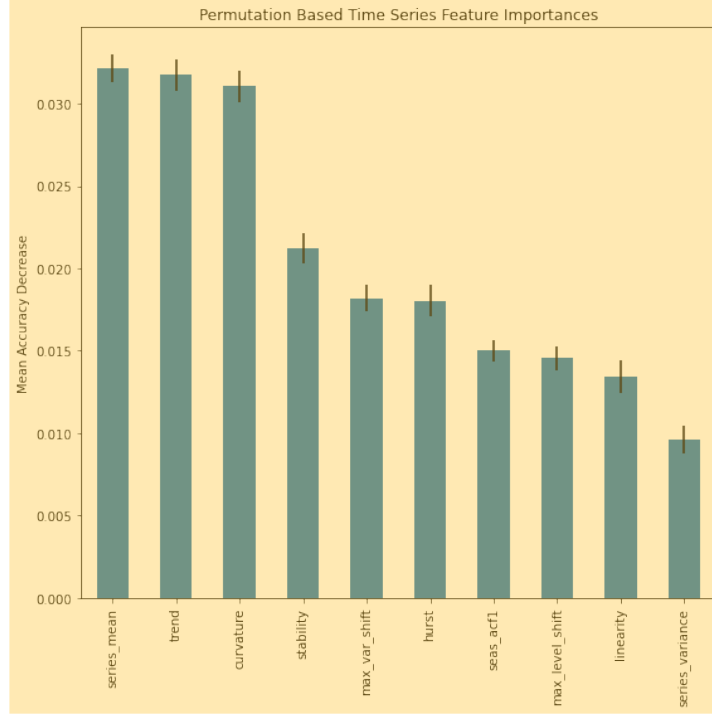


Figure 2: Feature importance from a random forest used to predict whether a privacy adjusted forecast had improved accuracy.

Figure 2 shows feature importance based on the mean accuracy decrease from permuting each of the ten features prior to predicting v . Differences in the mean of a time series are most important for predicting the accuracy of privacy adjusted forecasts, closely followed by differences in the strength of trend and the coefficient from an orthogonal quadratic regression measuring the curvature of the series. The remaining features we include in k -nTS swapping are Stability, the variance of the means of non-overlapping windows, *max_var_shift*, which is the largest shift in variance between two overlapping windows, and the Hurst coefficient, which measures the long term memory or persistence of a time series, and falls into the interval $[0.5, 1]$.

Following Khosrowabadi et al. (2022), we examine the marginal distributions of these predictors as they relate to the proportion of privacy adjusted forecasts with improved accuracy. For every feature, privacy adjusted forecasts are most likely to improve accuracy when the privacy method results in little to no change in the feature. This is our primary motivation for including these features in k -nTS: if we can swap values between series with similar features, we can limit the changes in these features that are most harmful to forecast accuracy. Privacy protection tends to bias the series downward, resulting in decreases in the series means and decreasing the accuracy of privacy adjusted forecasts. The differences in the strength of trend exhibit a similar relationship with the accuracy of privacy adjusted forecasts. We observe a small proportion of protected series with stronger trends, although these tend to have worse accuracy as well. Privacy protection tends to have little effect on the magnitude of the curvature coefficient (approximately

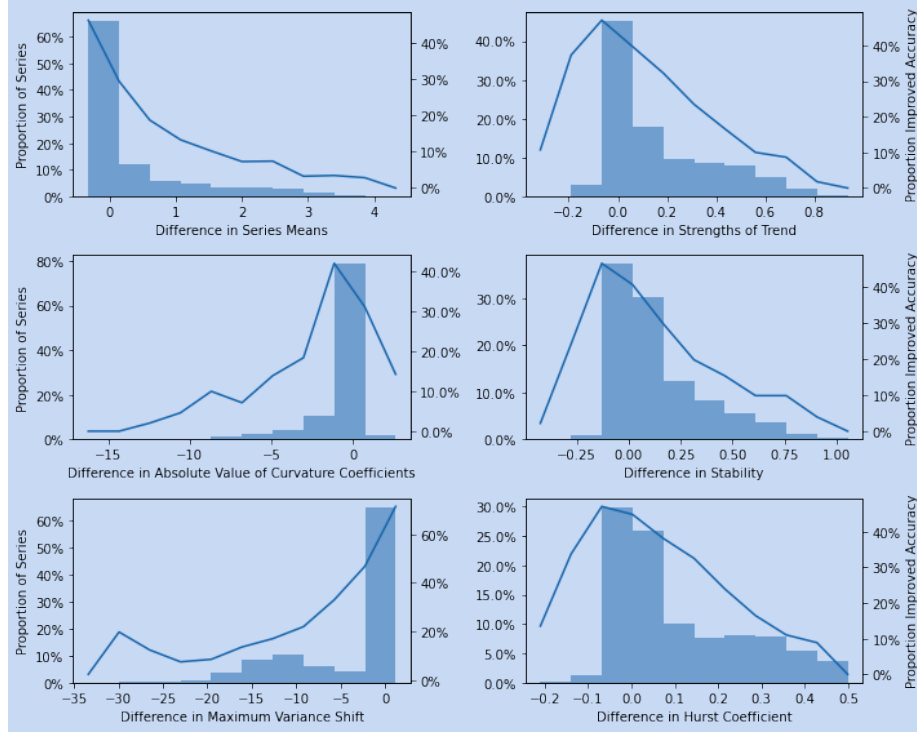


Figure 3: Marginal distributions (histograms) of the differences in the six most important time series features. The blue lines overlaying the histograms show the proportion of series within each bin for which the privacy adjusted forecasts had improved accuracy.

80% of protected series have little to no change in the curvature coefficient), but when it does, the coefficient tends to be larger in magnitude for protected series. In contrast to the curvature coefficients, we do observe a change in the stability of series under privacy protection. Standard privacy methods tend to alter the data such that differences in means in non-overlapping windows (which result from characteristics such as strong trends) are reduced in protected data, leading to lower values of stability (less variance in window means) in protected data. Similar to the curvature coefficients, the maximum variance shift in just over 60% of cases changes little between the original and protected series. Of all the features, this one provides the bin with the highest proportion of series with improved forecast accuracy - well over 60% for the largest bin, compared to just over 40% for the largest bins of the other features. For some series, there are large increases in the maximum variance shift, which are related to worse accuracy. Finally, most protected series have smaller Hurst coefficients than their original counterparts, indicating that privacy protection reduces the persistence of a series, sometimes quite significantly, which is associated with worse accuracy of privacy adjusted forecasts.

5.2. k - n TS Swapping Application

We implement k -nts for $k \in \{5, 10, 15\}$ for each subset of series within our data that are the same length. We include the six features from Figure 3 in the feature vectors f_j . We use a rolling

window of length $n = 25$, where the time series values for $t = 1, \dots, 25$ are swapped based on the features in the first window. The features are recalculated and swapping is performed on a rolling-window basis thereafter.

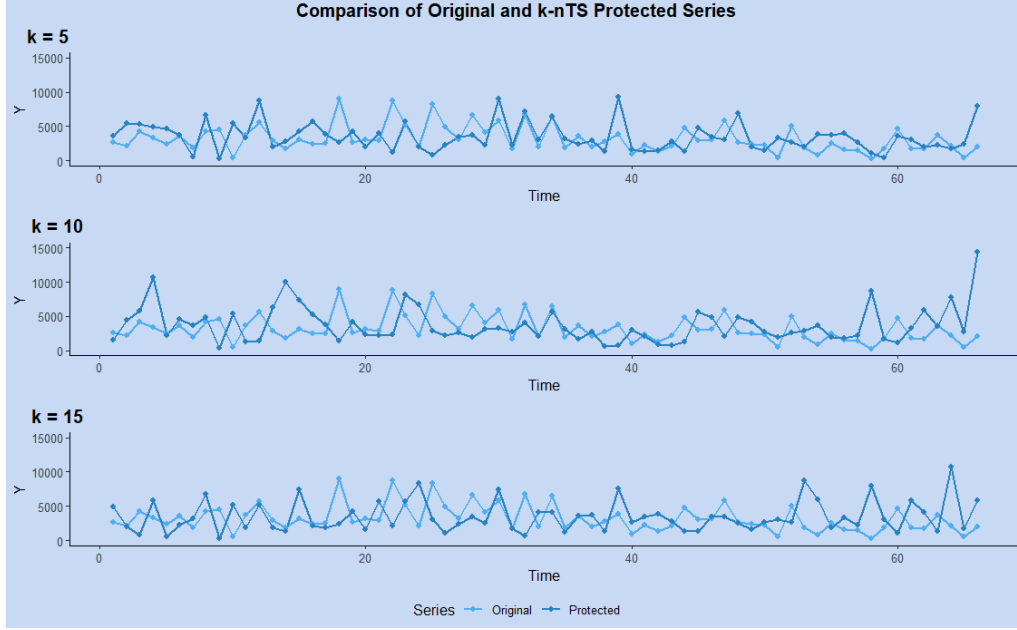


Figure 4: Original and k -nTS protected versions of two time series for $k = 15$.

Figure 4 compares an original series to its k -nTS protected counterparts.

5.3. Changes in Time Series Features and Model Behavior

We now examine the changes in time series features on a per-privacy method basis. For brevity, we include results for top-coding, k -nTS, and differential privacy in this section, and results for bottom-coding and additive noise in the appendix.

Figure 5 shows the distributions of time series features for the original and top-coded datasets, which are relatively stable as the strength of top-coding increases. These small changes in time series features result in small decreases in forecast accuracy, as we show in Section 5.4.

Next, we examine the distributions of features for the differentially private data sets, shown in Figure 6. The changes in features are drastic as the strength of privacy protection increases (ϵ decreases). For meaningful levels of $\epsilon \in \{0.1, 1\}$, we see the destruction of time series features that are vital for accurate forecasting. The Hurst coefficient distributions show the removal of most of the persistence of the series, making them have a tendency to revert to a long-term mean which is biased downward, leading to extremely poor forecast accuracy.

We examine the distributions of features for the k -nTS protected data sets in Figure 7. These distributions tell a story somewhere between top-coding and differential privacy. While the distribution of means is stable under k -nTS, we do see reductions in the distributions of the strength of

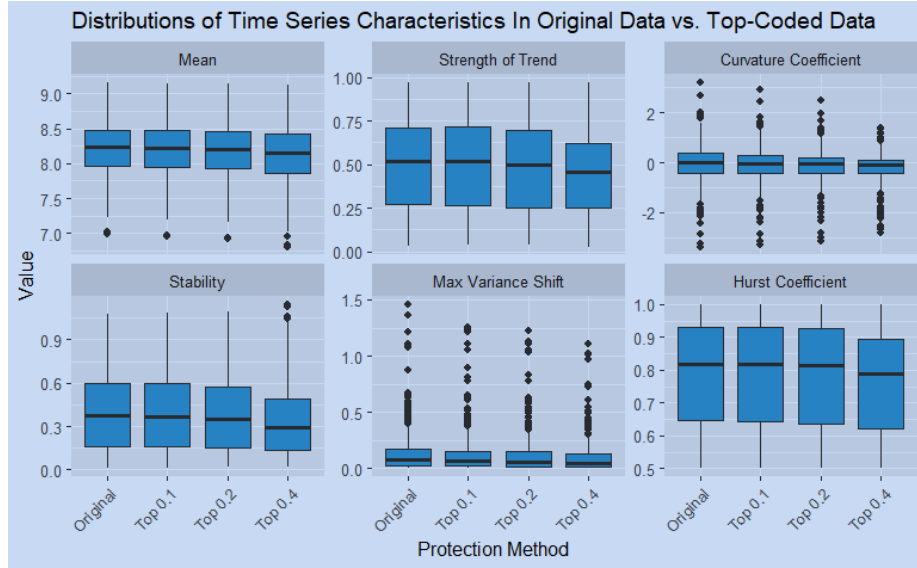


Figure 5: Distributions of time series characteristics in the original and top-coded data sets.

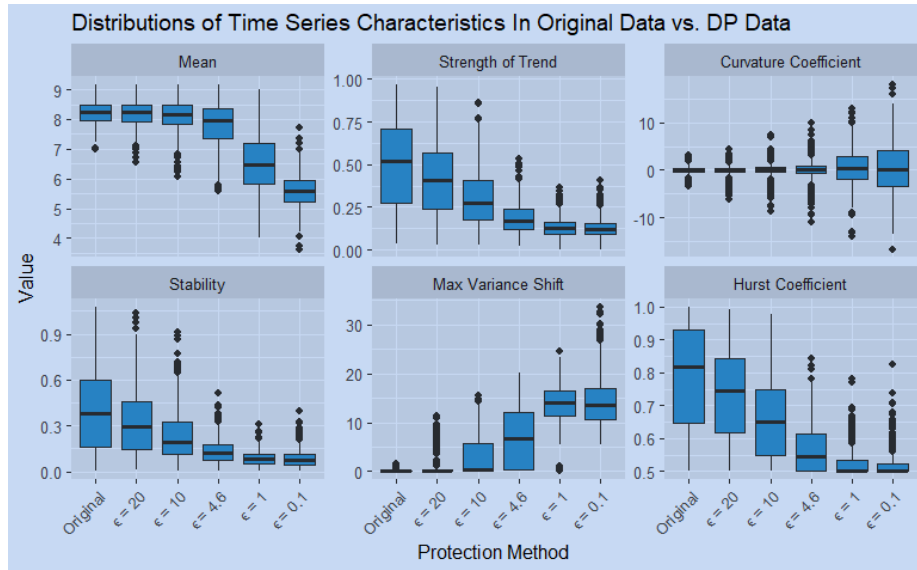


Figure 6: Distributions of time series characteristics in the original and differentially private data sets.

trend, stability, and the Hurst coefficient. These changes are consistent with the forecast accuracy shown in 5.4.

Finally, Figure 8 shows the distributions of the α parameter for simple exponential smooth-



Figure 7: Changes in time series characteristics under k -nTS for $k \in \{5, 10, 15\}$.

ing for the data sets protected using differential privacy, top-coding, and k -nts. Larger values of α apply less smoothing and weight the most recent observations the heaviest when forecasting. Small values of α correspond to high levels of smoothing and give more weight to older observations. Under differential privacy, we see that SES significantly reduces the magnitude of α as ϵ increases. For $\epsilon \in \{0.1, 1\}$, most of the α values are so small as to lead the model to simply forecast the long-run average of the series, which we know to be biased downward based on Figure 6. Since top-coding did not produce large changes in time series characteristics, we do not observe large changes in α for top-coded data, while α tends to be smaller for k -nTS protected data, they tend to lie between the values for top-coding and differential privacy with the exception of $\epsilon = 20$.

5.4. Accuracy of Point Forecasts

We first examine the mean accuracy of one-step ahead point forecasts across all models and privacy parameters for each privacy method. Table 3 displays the average MAE across all models and protected datasets for each privacy method. More granular results are presented in tables in the appendix.

Privacy Method					
Original	Top Coding	Bottom Coding	Differential Privacy	Additive Noise	k-nTS
681.24	696.97	749.41	3172.11	1274.51	959.92

Table 3: Average MAE across all forecasting models and protected datasets for each privacy method.

We also present the average ranks of each model based on (1) average MAE across all protected datasets for each privacy method, and (2) variance of the forecast errors across all protected

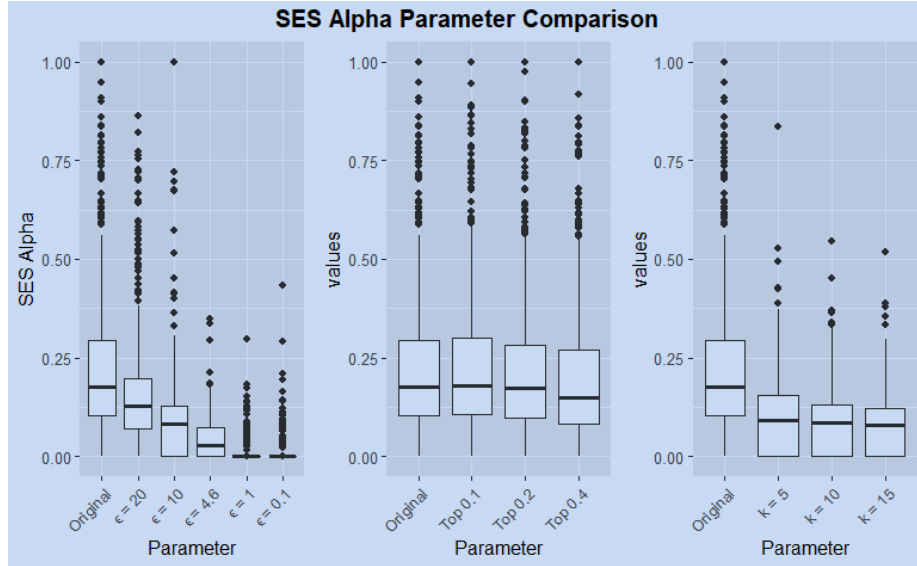


Figure 8: Distributions of α across all time series for each differential privacy, top-coded, and k -nTS protected data set.

datasets for each privacy method.

Mean MAE		Error Variance	
Model Ranks	Average Rank	Model Ranks	Average Rank
DES	2.6	DES	2.2
RNN	2.8	Auto-ARIMA	3.2
Auto-ARIMA	3.2	RNN	3.2
SES	3.2	SES	3.6
TES	3.8	TES	3.8
LGBM	5.6	LGBM	5.2
VAR	6.8	VAR	6.8

5.5. Variance of Forecast Errors

To compare forecast error distributions, we consider the variance of the forecast errors as a measure of forecast quality. Figure 9 shows the error distribution for each model for each privacy method across protection parameters. Models with lower forecast error variance will be more desirable since forecasters can have higher confidence that they have not generated forecasts for some series which are significantly worse than others. For the forecasts for each model for each level of top coding and the original data, we calculate the variance of the forecast errors (MAE values) and normalize the variances by the lowest variance on the original data, which was for the auto-ARIMA forecasts. Figure 10 shows these normalized variances, where the models in each quadrant are arranged in ascending order. Auto-ARIMA and TES continue to offer the best performance, as these models have the lowest error variance and lowest mean absolute errors. Notably, the VAR model performs particularly poorly, producing some of the

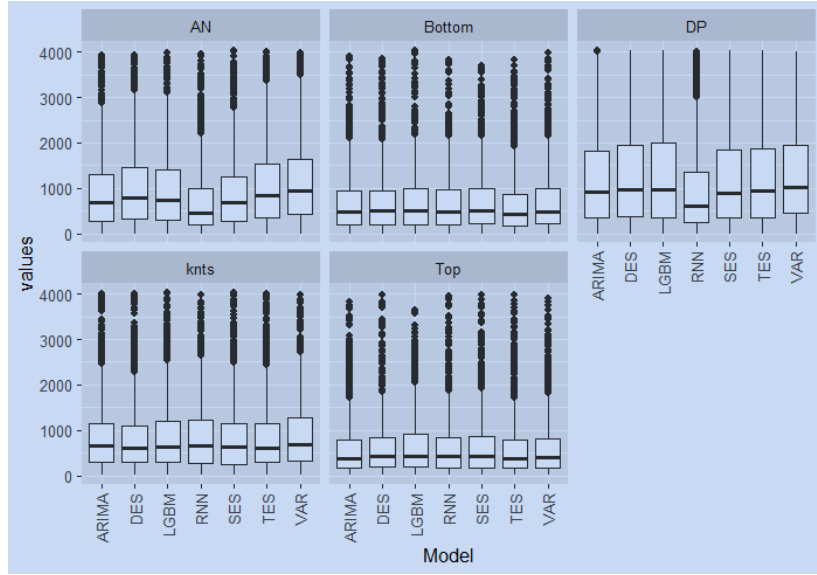


Figure 9: Changes in time series characteristics under k -nTS for $k \in \{5, 10, 15\}$.

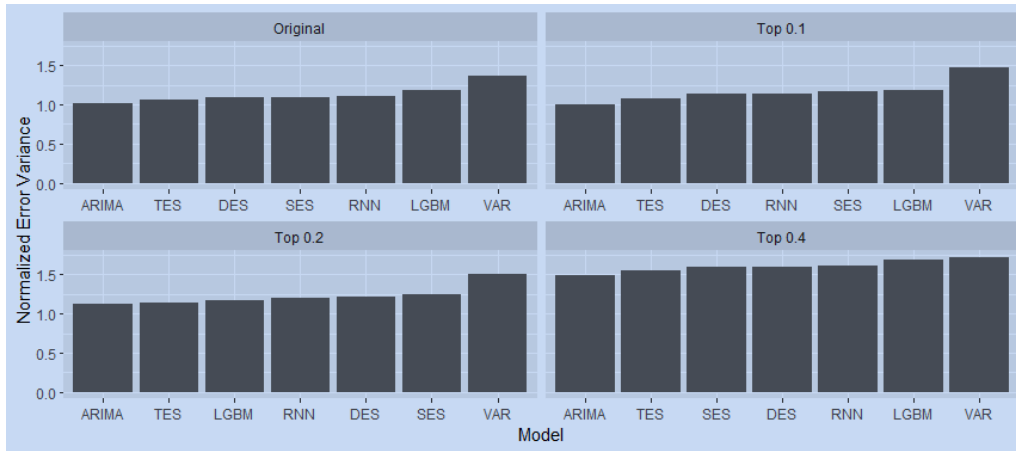


Figure 10: Model rankings for each top-coded dataset based on normalized forecast error variance.

highest mean forecast errors, and the highest forecast error variance. Another important finding is that the increase in error variance for all models is at least 50% compared to the forecast error variance for auto-ARIMA on the original data. So, while the increase in mean forecast error appears palatable, there will be much more variance in the quality of the forecasts from strongly protected data.

5.6. Comparing Privacy and Judgmentally Adjusted Forecasts

6. Conclusion

References

- Abowd, J. M., Gittings, R. K., McKinney, K. L., Stephens, B., Vilhuber, L., & Woodcock, S. D. (2012). Dynamically consistent noise infusion and partially synthetic data as confidentiality protection measures for related time series. *US Census Bureau Center for Economic Studies Paper No. CES-WP-12-13*, .
- Bandara, K., Bergmeir, C., & Smyl, S. (2020). Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach. *Expert systems with applications*, 140, 112896.
- Boone, T., Ganeshan, R., Jain, A., & Sanders, N. R. (2019). Forecasting sales in the supply chain: Consumer analytics in the big data era. *International Journal of Forecasting*, 35, 170–180.
- Centers for Medicare & Medicaid Services (1996). The Health Insurance Portability and Accountability Act of 1996 (HIPAA). Online at <http://www.cms.hhs.gov/hipaa/>.
- Chen, C., & Liu, L.-M. (1993). Forecasting time series with outliers. *Journal of forecasting*, 12, 13–35.
- Crimi, N., & Eddy, W. (2014). Top-coding and public use microdata samples from the us census bureau. *Journal of Privacy and Confidentiality*, 6.
- Davydenko, A., & Fildes, R. (2013). Measuring forecasting accuracy: The case of judgmental adjustments to sku-level demand forecasts. *International Journal of Forecasting*, 29, 510–522.
- De Montjoye, Y.-A., Hidalgo, C. A., Verleysen, M., & Blondel, V. D. (2013). Unique in the crowd: The privacy bounds of human mobility. *Scientific reports*, 3, 1–5.
- Ding, C., & Peng, H. (2005). Minimum redundancy feature selection from microarray gene expression data. *Journal of bioinformatics and computational biology*, 3, 185–205.
- Fan, L., & Xiong, L. (2012). Real-time aggregate monitoring with differential privacy. In *Proceedings of the 21st ACM international conference on Information and knowledge management* (pp. 2169–2173).
- Fildes, R., Goodwin, P., Lawrence, M., & Nikolopoulos, K. (2009). Effective forecasting and judgmental adjustments: an empirical evaluation and strategies for improvement in supply-chain planning. *International journal of forecasting*, 25, 3–23.
- Fildes, R., Goodwin, P., & Önköl, D. (2019). Use and misuse of information in supply chain forecasting of promotion effects. *International Journal of Forecasting*, 35, 144–156.
- Fulcher, B. D., & Jones, N. S. (2014). Highly comparative feature-based time-series classification. *IEEE Transactions on Knowledge and Data Engineering*, 26, 3026–3037.
- Goldfarb, A., & Tucker, C. (2013). Why managing consumer privacy can be an opportunity. *MIT Sloan Management Review*, 54, 10.
- Goldfarb, A., & Tucker, C. E. (2011). Privacy regulation and online advertising. *Management science*, 57, 57–71.
- Gonçalves, C., Bessa, R. J., & Pinson, P. (2021). A critical overview of privacy-preserving approaches for collaborative forecasting. *International journal of Forecasting*, 37, 322–342.
- Goncalves, C., Bessa, R. J., & Pinson, P. (2021). Privacy-preserving distributed learning for renewable energy forecasting. *IEEE Transactions on Sustainable Energy*, 12, 1777–1787.
- Goncalves, C., Pinson, P., & Bessa, R. J. (2020). Towards data markets in renewable energy forecasting. *IEEE Transactions on Sustainable Energy*, 12, 533–542.
- Herzen, J., Lässig, F., Piazzetta, S. G., Neuer, T., Tafti, L., Raille, G., Pottelbergh, T. V., Pasieka, M., Skrodzki, A., Huguenin, N., Dumonal, M., Kościsz, J., Bader, D., Gusset, F., Benheddi, M., Williamson, C., Kosinski, M., Petrik, M., & Grosch, G. (2022). Darts: User-friendly modern machine learning for time series. *Journal of Machine Learning Research*, 23, 1–6. URL: <http://jmlr.org/papers/v23/21-1177.html>.
- Hewamalage, H., Bergmeir, C., & Bandara, K. (2021). Recurrent neural networks for time series forecasting: Current status and future directions. *International Journal of Forecasting*, 37, 388–427.
- Hewamalage, H., Bergmeir, C., & Bandara, K. (2022). Global models for time series forecasting: A simulation study. *Pattern Recognition*, 124, 108441.
- Holohan, N., Braghin, S., Mac Aonghusa, P., & Levacher, K. (2019). Diffprivlib: the IBM differential privacy library. *ArXiv e-prints*, 1907.02444 [cs.CR].
- Hyndman, R., Kang, Y., Montero-Manso, P., Talagala, T., Wang, E., Yang, Y., & O'Hara-Wild, M. (2022). *tsfeatures: Time Series Feature Extraction*. URL: <https://pkg.robjhyndman.com/tsfeatures/> r package version 1.0.2.9000.
- Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: principles and practice*. OTexts.
- Imtiaz, S., Horchidan, S.-F., Abbas, Z., Arsalan, M., Chaudhry, H. N., & Vlassov, V. (2020). Privacy preserving time-series forecasting of user health data streams. In *2020 IEEE International Conference on Big Data (Big Data)* (pp. 3428–3437). IEEE.

- Kang, Y., Hyndman, R. J., & Smith-Miles, K. (2017). Visualising forecasting algorithm performance using time series instance spaces. *International Journal of Forecasting*, 33, 345–358.
- Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30.
- Khosrowabadi, N., Hoberg, K., & Imdahl, C. (2022). Evaluating human behaviour in response to ai recommendations for judgemental forecasting. *European Journal of Operational Research*, .
- Koning, A. J., Franses, P. H., Hibon, M., & Stekler, H. O. (2005). The m3 competition: Statistical tests of the results. *International Journal of Forecasting*, 21, 397–409.
- Li, L., Kang, Y., & Li, F. (2022). Bayesian forecast combination using time-varying features. *International Journal of Forecasting*, .
- Luo, J., Hong, T., & Fang, S.-C. (2018). Benchmarking robustness of load forecasting models under data integrity attacks. *International Journal of Forecasting*, 34, 89–104.
- Löning, M., Király, F., Bagnall, T., Middlehurst, M., Ganesh, S., Oastler, G., Lines, J., Walter, M., ViktorKaz, Mentel, L., chrisholder, RNKuhns, Tsaprounis, L., Owoseni, T., Rockenschaub, P., danbartl, jesellier, eenticott shell, Gilbert, C., Bulatova, G., Lovkush, Parker, M., Take, K., Schäfer, P., Khrapov, S., Meyer, S. M., AidenRushbrooke, oleskiewicz, Xu, Y.-X., & Ansari, A. (2022). alan-turing-institute/sktime: v0.13.2. URL: <https://doi.org/10.5281/zenodo.7017832>. doi:10.5281/zenodo.7017832.
- Makridakis, S., & Hibon, M. (2000). The m3-competition: results, conclusions and implications. *International journal of forecasting*, 16, 451–476.
- Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). The m4 competition: Results, findings, conclusion and way forward. *International Journal of Forecasting*, 34, 802–808.
- Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2022). M5 accuracy competition: Results, findings, and conclusions. *International Journal of Forecasting*, .
- Martin, K. D., Borah, A., & Palmatier, R. W. (2017). Data privacy: Effects on customer and firm performance. *Journal of Marketing*, 81, 36–58.
- Nicholson, W. B., Matteson, D. S., & Bien, J. (2017). Varx-l: Structured regularization for large vector autoregressions with exogenous variables. *International Journal of Forecasting*, 33, 627–651.
- Nin, J., & Torra, V. (2009). Towards the evaluation of time series protection methods. *Information Sciences*, 179, 1663–1677.
- Petropoulos, F., Apiletti, D., Assimakopoulos, V., Babai, M. Z., Barrow, D. K., Taieb, S. B., Bergmeir, C., Bessa, R. J., Bijak, J., Boylan, J. E. et al. (2022). Forecasting: theory and practice. *International Journal of Forecasting*, .
- Smyl, S. (2020). A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *International Journal of Forecasting*, 36, 75–85.
- Sommer, B., Pinson, P., Messner, J. W., & Obst, D. (2021). Online distributed learning in wind power forecasting. *International Journal of Forecasting*, 37, 205–223.
- Sweeney, L. (2002). k-anonymity: A model for protecting privacy. *International journal of uncertainty, fuzziness and knowledge-based systems*, 10, 557–570.
- Talagala, T. S., Li, F., & Kang, Y. (2022). Fformpp: Feature-based forecast model performance prediction. *International Journal of Forecasting*, 38, 920–943.
- Véliz, C., & Grunewald, P. (2018). Protecting data privacy is key to a smart energy future. *Nature Energy*, 3, 702–704.
- Yousefpour, A., Shilov, I., Sablayrolles, A., Testuggine, D., Prasad, K., Malek, M., Nguyen, J., Ghosh, S., Bharadwaj, A., Zhao, J., Cormode, G., & Mironov, I. (2021). Opacus: User-friendly differential privacy library in PyTorch. *arXiv preprint arXiv:2109.12298*, .

7. Appendix

We first examine the point forecast accuracy and error distributions for forecasts based on top-coded data. In Table 4, we see the MAE for each model on the original and top-coded datasets. TES (which explicitly models trend and seasonality of each series) outperforms the rest of the models on the original data, and is closely followed by auto-ARIMA. These models outperform the more complex LGBM and RNN models, although this is to be expected given that we did not apply advanced pre-processing techniques which can substantially improve the accuracy of machine learning models.¹⁰

¹⁰See the appendix for results using LGBM and RNN with advanced pre-processing techniques.

Model	Original	Top (10%)	Top (20%)	Top (40%)
SES	686.71	678.27	694.71	748.84
DES	680.54	671.48	682.41	736.33
TES	637.90*	627.09*	642.72*	716.24**
Auto-ARIMA	647.34**	631.61**	653.35**	714.68*
VAR	729.57	711.89	707.07	734.03
LGBM	700.89	694.93	705.39	772.65
RNN	671.28	671.05	694.50	749.51

Table 4: Original MAE vs. MAE under Top-Coding. * = best accuracy ** = second best accuracy

In terms of the effect of top-coding on forecast accuracy, we see some remarkable results. When forecasting with 10% top-coded data, the MAE for each model improves (although the improvement for RNN is marginal). Typically, data users face a trade-off between the privacy and utility of data, where increasing privacy reduces data utility. In this case, some privacy is obtained for free while actually improving forecast accuracy. Stronger levels of top-coding tend to reduce forecast accuracy. Overall, TES and auto-ARIMA performed the best on this data under all levels of top-coding.