~~The Effects of Privacy Protection on Forecast Accuracy~~Preserving Time Series Features for Forecasting with Protected Data

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 data from smart devices (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)[1] require organizations to protect their sensitive data to avoid fines, and place strict limitations on data transfers and processing[2].

These legal limitations can be circumvented when data is properly anonymized[3]. For example, during the COVID-19 pandemic, mobile phone position data was anonymized through aggregation to origin-destination matrices, which were used to study population mobility patterns in the EU (Santamaria et al. 2020). This data was subsequently used in nowcasting GDP during the COVID-19 pandemic (Barbaglia 2020). Data that is not anonymous, on the other hand, is subject to purpose limitation, and cannot be freely re-used[4]. In addition to legal compliance, other reasons for protecting data include reducing consumers' privacy concerns (Martin et al., 2017) or attempting to gain a competitive advantage through privacy-conscious brand positioning (Goldfarb & Tucker, n.d.). Several of the largest tech companies in the world, including IBM, Google[5], Meta, and Microsoft[6] implement various approaches to privacy protection and provide open-source code to enable others to do the same. ~~Notably,~~ Apple has also positioned ~~themselves~~ itself as a privacy-focused company[7].

Various data protection approaches are available depending on whether time series are stored in a single data set (centralized) or spread across multiple data owners/data sets (decentralized). In the decentralized scenario, multi-party computation or federated learning enable privacy-preserving collaborative forecasting to ensure accurate forecasts while protecting sensitive data (Gonçalves et al., 2021; Goncalves, Bessa, et al., 2021; Sommer et al., 2021). We focus on the centralized scenario 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 the ability of a bad actor to identify data subjects and 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). The concern for forecasters is

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

[2] See articles 6, 45, and 46 of the GDPR.

[3] See the description of anonymous information given in Recital 26 of the GDPR.

[4] See article 5(b) of the GDPR.

[5] See several python libraries including PipelineDP and PyDP.

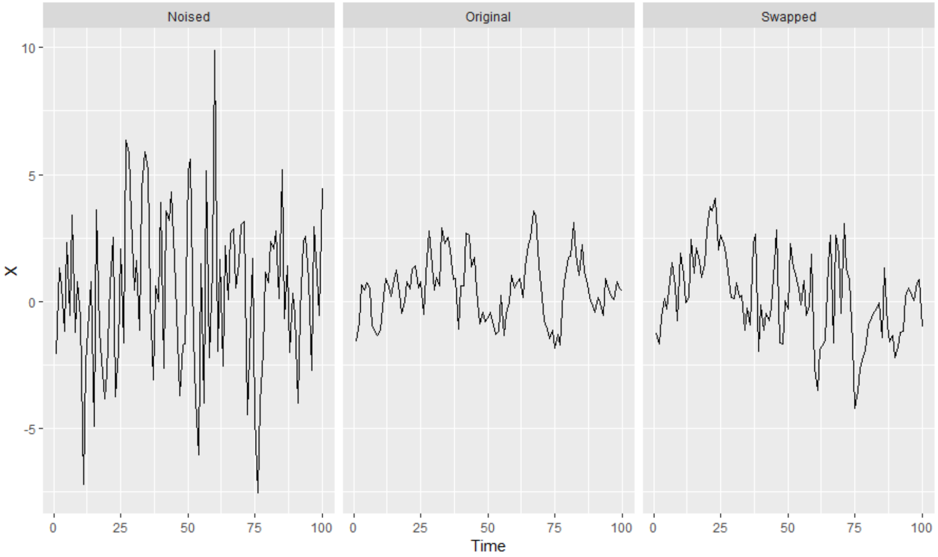
[6] See the OpenDP project.

[7] See descriptions of Apple's privacy features.

that privacy methods can drastically alter time series features which can significantly affect forecast accuracy.

Consider the example shown in Figure 1. The series shown in the middle plot is a simulated AR(1) process with autoregressive parameter $\phi_1 = 0.8$. The series on the left is the original series with random noise added to each period that is proportional to the standard deviation of the original series. Estimating an ARIMA(1, 0, 0) model on the original series yields an estimate of $\widehat{\phi_1} = 0.73$, while the noised series yields an estimate of $\widehat{\phi_1} = 0.16$. The series on the right was created by swapping the original series values with values from two other simulated AR(1) processes, both with $\phi_1 = 0.8$. The swapped series better preserves the autocorrelation of the original series, with an estimate of $\widehat{\phi_1} = 0.56$. Visually, the swapped version of the original series is more plausible than the noised version.

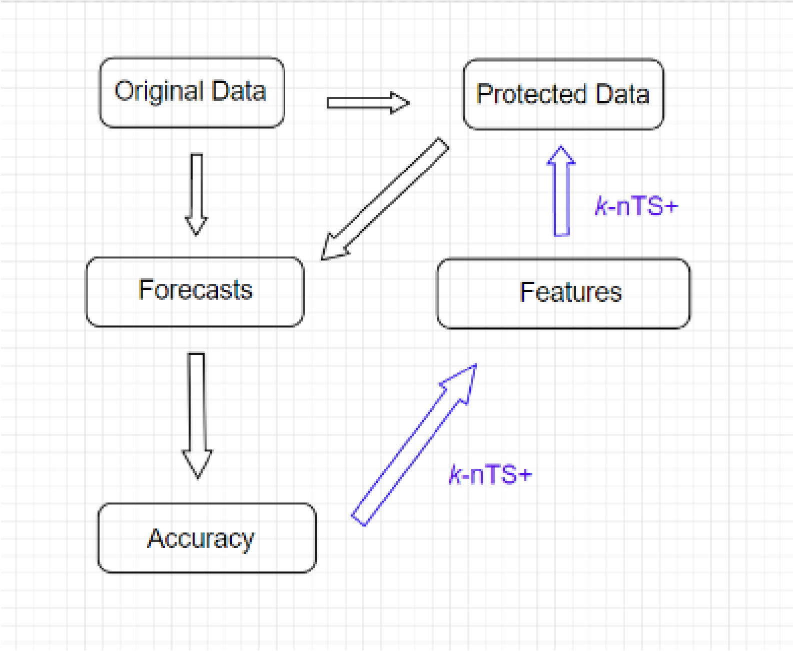
Figure 1: comparison of protected AR(1) processes to the original AR(1) process.



This paper examines how data protection changes time series features and how this affects forecasting model performance. As shown in Figure 2, we first create protected time series using baseline privacy methods and generate forecasts for both the original and protected series using several forecasting models. We compare the accuracy of these models for each protected data set relative to the original data and investigate how changes in time series features drive changes in forecast accuracy. ~~First, we investigate the drivers of changes in forecast accuracy for protected data. Specifically, we examine the effects of privacy protection on time series features, which provide a link to explain why privacy methods produce changes in forecast accuracy.~~ To limit reductions in forecast accuracy that can arise from data protection, we develop a new matrix-based privacy method called *k*-nts+ which swaps the values of time series with similar features (e.g., variance, strength of trend, etc.). This ~~to~~ balances the trade-off between privacy and forecast accuracy by preserving the features of time series even while the time series values are altered.~~.~~

The *k*-nTS+ method uses a feedback loop from the baseline privacy methods that learns which time series features are most predictive of forecast accuracy. These features are used as the basis for swapping to create new protected data sets which maintain the features of the original time series significantly better than the baseline methods. This ensures that time series features which are important for forecasting are preserved such that the time series are representative of the original data and enable models to produce usable forecasts. Organizations will be more likely to adopt data protection measures if they trust the measures to produce useful data while improving privacy. Our method produces significantly improved forecast accuracy at similar levels of privacy compared to the baseline methods.~~-~~ We illustrate the application of our method and provide empirical results of forecast accuracy and privacy for protected data~~a and examine model-specific behavior to understand why certain models perform better than others~~.

Figure 2: Framework to study the effects of privacy methods on forecast accuracy using time series features. Blue arrows indicate the ~~flow of results which~~ feedback loop which informs our proposed *k*-nTS ~~-~~protection method ~~based on swapping values of time series with similar features to limit the changes in features and forecast accuracy between the original and protected data~~.



The rest of the paper proceeds as follows. In Section 2, we review the relevant literature and outline the contributions of this paper. Section 3 describes the swapping mechanism of the baseline *k*-nTS method, and Section 4 outlines nine well known time series features which are used for swapping in our empirical application. In Section 5 we outline the feedback loop shown in Figure 2 for our *k*-nTS+ method. Section 6 contains an application of several forecasting models to protected data sets produced using *k*-nTS+ and other baseline privacy methods. We examine the forecast accuracy and the time series features produced under each privacy method and how this relates to forecasting model performance. We discuss our conclusions in Section 7, including recommendations for future research.~~s…~~

2. Lit Review

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.

Some forecasters have studied data privacy and forecasting in the context of collaborative forecasting. (Goncalves, Pinson, et al., 2021) 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. (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; Liyue Fan & Li Xiong, 2014). 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 is a regularization technique ~~mirrors a technique~~ used to prevent overfitting when forecasting with neural networks (Hewamalage et al., 2021, 2022). We explore whether data protection with random noise can achieve this same regularization at meaningful levels of privacy.

Another type of 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 methods 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 shru~~shrinks~~nk~~ 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).

Commented [B1]: Should we remove all mention of top/bottom coding? It doesn't show up anywhere in the forecasting literature. Or, do we keep top/bottom coding in the literature review but mention later that we don't consider it since censored models can account for it?

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, which our work remedies.

2.1. 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 judgmentally adjusting forecasts, these 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. 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 magnitude of judgmental 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).

Regardless of the motivation for data protection, privacy methods alter time series features which leads models to generate different forecasts than what would be produced based on the original data. The ‘adjustment’ is applied to the data rather than directly to the forecasts, but both privacy protection and judgmental adjustments result in adjusted forecasts. However, privacy protection may be applied without regard for forecast accuracy. Privacy methods based on random noise add information with low diagnosticity, and are likely to reduce forecast accuracy. While the direction and size of adjustment are purposefully chosen in judgmental forecasting, the direction and size of adjustments to forecasts from privacy protection will be determined by forecasting models’ response to changes in time series features. These responses are likely related to the strength of data protection since stronger data protection results in larger changes to the data.

2.2. Time Series Features and Forecasting

There are thousands of features which have been used for time series classification (Fulcher & Jones, 2014). A small~~er~~ set of interpretable features was used by (Bandara et al., 2018) for clustering and forecasting similar time series, which improved the accuracy of recurrent neural network models. Our focus is on features which are predictive of forecast accuracy, since privacy methods which alter these features will be most detrimental for forecasting.

The initial results from the M4 competition suggested that the randomness and linearity of time series were the most important determinants of forecast accuracy, and that seasonal time series (which are typically less noisy) are easier to forecast (Makridakis et al., 2018). In a follow-up study, (Spiliotis et al., 2020) used multiple linear regression to confirm the importance of randomness, linearity, and seasonal strength in predicting the MASE values of the ETS, ARIMA, Theta, and Naïve 2 (random walk applied to seasonally adjusted data) models from the M4 competition. On average, increasing the frequency, kurtosis, linearity, and seasonal strength of time series contributed to improved forecast accuracy. However, increasing skewness, self-similarity, and randomness affected accuracy negatively. While strength of seasonality improved the accuracy of all models, strength of trend had no statistically significant effect on accuracy for ETS, ARIMA, and Theta, while hurting accuracy for Naïve 2, which has no means of accounting for trend.

Outside of predicting forecast accuracy, time series characteristics can be used as a basis for making and combining forecasts. Features such as the strength of trend and seasonality have been used in exponential smoothing model selection (Qi et al., 2022). Forecasts based on this feature-based model selection had lower MASE, sMAPE, and MSIS than information-based selection methods for the majority of forecast horizons. Time series ~~characteristics~~ features have also been used to select optimal model and forecast combinations (Li et al., 2022; Talagala et al., 2022). ~~Model selection based on the representativeness of~~The REP metric ~~forecasts~~ (Petropoulos & Siemsen, 2022) selects models with trend and seasonality components when the respective signals of these components are strong, and has been shown to outperform information criteria-based and cross-validation based model selection.

2.3. Our Contributions

Our contributions are two-fold. First, we analyze privacy adjusted forecasts from multiple forecasting models and privacy methods. We extract time series features which are predictive of forecast accuracy and show how these features change under data protection. We use these changes to explain the performance of various forecasting models ~~and provide recommendations on forecasting with protected data.~~in our empirical application.

Second, we propose a novel privacy method designed with forecasters in mind. Existing privacy methods apply protection based on the values of time series and give no regard to the preservation of time series features. Motivated by~~Due to~~ the relationship between time series features and forecast accuracy (Spiliotis et al. 2020), and the usefulness of time series features for performing model selection and forecast combination (Li et al., 2022; Talagala et al., 2022), we create a privacy method that uses a feedback loop based on the relationship between time series features and forecast accuracy to forecast well under data protection. ~~existing privacy methods fail to provide utility for forecasting at meaningful levels of privacy. We address this issue using a~~This matrix-based method ~~which~~ swaps the values of time series with similar features to help maintain those features and forecast accuracy. Results show that our method provides significantly better accuracy at similar levels of privacy protection compared to competitor privacy methods. Further, using the *performance gap* from the REP metric (Petropoulos & Siemsen, 2022), we show that the protected time series produced under *k*-nTS+ are much more representative of the original series compared to the protected series from other methods. We define our proposed method in the next section.

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

See attached pdf.

4. Time Series Features for Swapping Protection

In this section, we describe the time series features which have been demonstrated to have a relationship with forecast accuracy. We let x_j' denote a univariate stationary time series with a finite mean and constant variance. The spectral density $f_x(\lambda)$ of x_j' is estimated as the scaled fourier transform of the autocovariance function $\gamma_x(k)$ of x_j'. The spectral density can be thought of as the probability density function of a random variable Λ on the unit circle (Goerg, n.d.), where for a non-zero integer k, when $\gamma_x(k) \neq 0$, the spectral density $f_x(\lambda)$ will have a peak at the corresponding frequency λ. The forecastability, or spectral entropy, of x_j' is measured using the Shannon entropy of $f_x(\lambda)$, given by

$$SpecEntropy = -\int_{-\pi}^{\pi} \hat{f}_x(\lambda) log\, \hat{f}_x(\lambda) d$$

where the maximum entropy is attained when $\Lambda \sim U(-\pi, \pi)$. In practice, estimates of $F_1 \in [0,1]$, where high F_1 values represent a low signal-to-noise ratio, indicating that x_j' is difficult to forecast (Kang et al., 2017).

Next, we consider the self-similarity feature quantified using the Hurst parameter (Wang et al., 2006), which measures the long-range dependence of a time series. This feature had the largest magnitude effect on forecast accuracy in the study of the M4 data performed by (Spiliotis et al. 2020). We use the definition of self-similarity of a time series described by (Willinger et al., n.d.). Suppose that x_j' is the increment process of x_j, i.e., $x_{j,t}' = x_{j,t+1} - x_{j,t}$

An aggregated sequence, denoted $x_j'^{(m)}$, is created by averaging x_j' over non-overlapping blocks of size m, where

$$x_{j,k}'^{(m)} = 1/m \sum_{i=(k-1)m+1}^{km} x_{j,i}', \;\; k = 1,2,\ldots$$

and k indexes the block. If x_j is a self-similar time series, then

$$x_j' = m^{1-H} x_j'^{(m)}$$

for all integers m. We focus on the definition of second-order self-similarity, where x_j' is exactly second-order self-similar if $m^{1-H} x_j'^{(m)}$ has the same variance and autocorrelation as x_j' for all values of m, or is asymptotically second-order self-similar if this holds as $m \rightarrow \infty$ (Rose, n.d.). The parameter H is the Hurst exponent, which is estimated using the differencing term d from a fractional ARIMA model, i.e., FARIMA(0, d, 0) (Wang et al., 2006) (Hyndman et al., 2022), where

$$Hurst \; = H = d + 0.5.$$

Estimates of H fall in the interval (0, 1), where $H \; = \; 0.5$ corresponds to a random walk (Sobolev, 2017), $H \; < \; 0.5$ corresponds to anti-persistent or mean-reverting series, and $H \; > \; 0.5$ corresponds to persistent time series that are more likely to maintain their current trend. (Rose, n.d.) notes that a self-similar process has a spectral density that follows a power law near $\lambda = \; 0$, where $f_x(\lambda) \sim a\lambda^{1-2H}$ as $\lambda \rightarrow \; 0$ with $0.5 \; < \; H \; < \; 1$. When $H \; \approx 1$, the spectral density increases rapidly as $\lambda \rightarrow \; 0$ and will tend to have low spectral entropy, whereas when $H \; \approx \; 0.5$, the spectral density increases slowly as $\lambda \rightarrow 0$ and will tend to have high spectral entropy. For a random walk with $H \; = \; 0.5$, i.e., the spectral density is finite at the origin (Rose, n.d.).

We consider the remaining features from (Spiliotis et al., 2020) which had the largest effects on forecast accuracy. Since none of the privacy methods we consider will change the time series' frequency, we omit this feature from consideration, noting that higher frequencies are associated with improved forecast accuracy. We include skewness and kurtosis which measure the shape of the distribution of time series' values.

Skewness, which we denote F_3, measures the lack of symmetry in the distribution of the values of x_j (Wang et al., 2006), where positive (negative) values are associated with a right- (left-) skewed data distribution:

$$Skewness = \frac{1}{n\sigma^3} \sum_{t=1}^{n} \left(x_j - \overline{x}_J\right)^3$$

We use a measure of Kurtosis relative to the standard normal distribution (Wang et al., 2006). Positive kurtosis corresponds to distributions that tend to have a distinct peak near the mean with heavy tails, whereas negative kurtosis corresponds to distributions that are relatively flat near the mean,

$$Kurtosis = \frac{1}{n\sigma^4}\sum_{t=1}^{n}\left(x_j - \bar{x}_j\right)^4 - 3,$$

where 3 is the kurtosis of the standard normal distribution.

Next, we perform STL decomposition (Cleveland et al. 1990) to obtain the trend, seasonal, and remainder components of x_j. We use the approach of (Hyndman et al. 2019) which is designed to handle multiple seasonalities to obtain

$$x_j = f_j + s_{1,j} + \cdots + s_{M,j} + e_t,$$

where f_j, $s_{i,j}$, and e_j are the trend, ith Seasonal, and remainder components, respectively.

We extract the first order autocorrelation coefficient of the detrended and deseasonalized series, referred to as 'linearity' by (Spiliotis et al. 2018).

$$E\_acf = \frac{\sum_{t=2}^{T}\left(e_{j,t} - \bar{e}\right)\left(e_{j,t-1} - \bar{e}\right)}{\sum_{t=1}^{T}\left(e_{j,t} - \bar{e}\right)^2}$$

This feature gives a measure of the forecastability of a time series after the trend and seasonality have been accounted for.

Continuing with the decomposed series, we compute the strength of trend F_6 and strength of the ith seasonal component $F_{7,i}$ as follows,

$$Trend = 1 - \frac{Var(e_j)}{Var(f_j + e_j)},$$

$$Seasonality_i = 1 - \frac{Var(e_j)}{Var(s_{i,j} + e_j)}.$$

In practice, the values of $Trend$ and $Seasonality_i$ are bounded to [0,1] (Hyndman 2022).

Our final two features are included to maintain data utility throughout the k-nTS swapping process. The idea is to swap values between series that not only have similar characteristics, but whose values have similar magnitudes. Toward this end, we include the mean F_8 and the variance F_9 of x_j,

$$SeriesMean = \frac{1}{T}\sum_{t=1}^{T} x_{j,t},$$

$$SeriesVariance = \frac{1}{T-1}\sum_{t=1}^{T}\left(x_{j,t} - \bar{x}_j\right)^2.$$

Figure 3 compares two monthly time series on the nine time series features discussed in this section, the values of which are shown in Table 1. The good forecastability of the series on the left is indicated by the low spectral entropy and high Hurst coefficient values. The series on the right, however, is essentially a random walk as indicated by the value of the Hurst coefficient, and a spectral entropy of one indicates a very low signal to noise ratio. Another notable difference is in the strength of the trend of each series – most of the variance of the series on the left is due to a strong trend, which is forecastable, whereas the variance in the series on the right appears to be due to the randomness of the series. The series on the left has low *Kurtosis*, i.e., light tails relative to the standard normal distribution, whereas the opposite is true for the series on the right.

Fig 3: Comparison of a time series with desirable features (easy to forecast) and a time series with undesirable features (difficult to forecast).

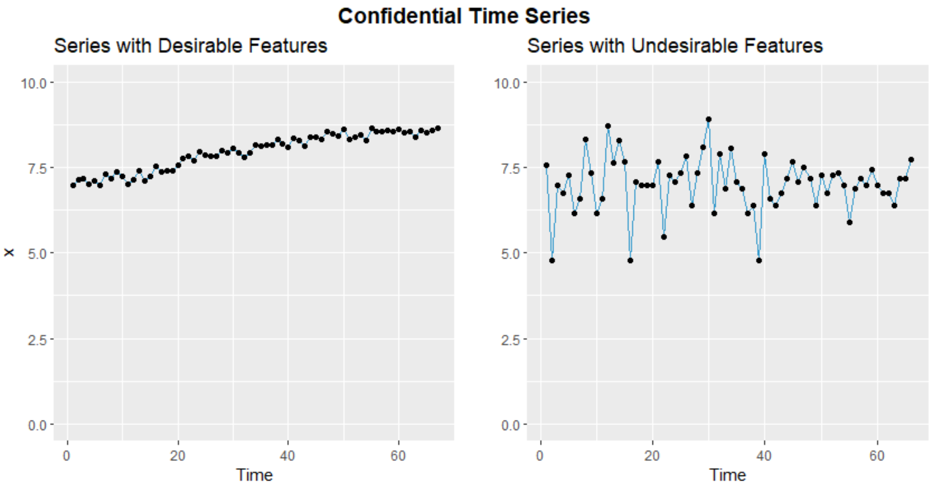


Table 1~~2~~: feature value comparison between a series with desirable features (easy to forecast) and a series with undesirable features (difficult to forecast)

Feature	Desirable Features (left Fig. 1)	Undesirable Features (right Fig. 1)

SpecEntropy	0.07	1.00
Hurst	1.00	0.50
Skewness	-0.42	-0.57
Kurtosis	-1.24	1.16
E_acf	-0.09	-0.19
Trend	0.97	0.12
Seasonality	0.16	0.23
SeriesMean	7.96	7.01
SeriesVariance	0.29	0.65

5. The *k*-nearest Time Series + (nTS+) Swapping Method

The *k*-nTS+ privacy method builds on *k*-nTS by including a feature selection process based on the ~~changes in~~ time series features and forecast accuracy under baseline privacy methods. The idea is to perform swapping which limits the changes in features which are most predictive of ~~changes in~~ forecast accuracy.

The feature selection process starts with the data controller generating forecasts for period $T - 1$ for both the original data and the data protected using baseline privacy methods such as differential privacy and additive noise. The data controller measures the ~~*difference* in~~ forecast errors a~~, denoted e_j^d,~~ and time series feature values for ~~, denoted f_j^d for feature f, between~~ each original and protected time series. Our reasoning is that a feature should be included in the *k*-nTS swapping process (and preserved in the protected data) ~~if changes in that feature are predictive of changes in~~if it is predictive of the forecast error across the original and baseline protected data sets.

An initial filtering of the features is performed using the RReliefF algorithm (Robnik-Sikonja & Kononenko 2003) which assigns a weight to each feature that indicates the ability of that feature to discriminate between nearest neighbor time series with different forecast errors compared to nearest neighbor time series with similar forecast errors. This is an intuitive choice for a feature importance measure since our privacy method swaps values between nearest neighbor time series. A similar filtering was performed by Li et al. (2022) for their forecast combination methodology.

One downside of Relief-based algorithms is that they do not remove redundant features. These algorithms can be applied recursively, removing the feature(s) with the lowest weights in each iteration, but there is not a straight-forward method for choosing how many features to ultimately include (Urbanowicz et al. 2018). Including all important features from RReliefF in *k*-nTS would significantly increase the dimensionality and reduce the efficiency of the swapping process. To address this problem, we apply a recursive feature elimination (RFE) algorithm based on a random forest which predicts forecast accuracy using subsets of the most important features from the RReliefF algorithm. Prior work has shown that random forest based RFE is efficient when applied to sets of highly correlated features, i.e., it selects a small set of features with good prediction performance (Gregorutti et al. 2017).

In each iteration of the RFE algorithm, we store the out-of-bag (oob) mean-squared error (MSE) and permutation-based feature importance values from a random forest used to predict the accuracy of the forecasts for each original and protected time series using the time series features. The least important feature is removed, and the model is then retrained for the next iteration.

These steps continue recursively until only one feature remains. The full RFE algorithm is repeated N_{rfe} times.

We compute the average MSE of the oob predictions for each feature subset size across the N_{rfe} repetitions of RFE. Similarly, for each repetition, we rank the features based on the inverse of the order in which they were eliminated and average these ranks. Let p_{min} denote the subset size with the minimum average MSE across repetitions and let MSE_{min} denote that minimum average MSE. Our final subset size p^* is chosen as follows,

$$p^* = \min \mathrm{p}, \qquad \mathrm{p} = 1, \ldots, \mathrm{P}$$

$$\leq$$

$$s.t. \left(RMSE_p - RMSE_{min}\right)/\mathrm{RMSE_{min}} \leq \tau$$

Where τ is the maximum percentage difference between the minimum MSE and the MSE of our chosen subset size. The chosen subset size is the smallest number of features which offers an average predictive accuracy with similar performance to the subset size with the best average accuracy. We include the p^* features with the highest average ranks in the *k*-nTS+ swapping method.

Algorithm: RFE for *k*-nTS+ Using Random Forest

```
for i, $i = 1, \ldots N_{rfe}$:
    -  ≠Train random forest ≠
    -  Calculate MSE of oob predictions
    -  Calculate permutation-based feature importance
    for subset size $p$,  $p = P - 1, \ldots, 1$:
            o  Keep the $p$ most important features
            o  Retrain random forest using only the $p$ most important features
            o  Calculate MSE of oob predictions
            o  Calculate permutation-based feature importance
    end
end
```

Calculate the average MSE for each subset size across all iterations.

Calculate the rank of each feature as the average of the elimination orders from each iteration.

Calculate the desired number of features p^* as the minimum number of features with an average prediction error within τ% of the minimum average prediction error.

~~The feature selection process starts with the data controller generating forecasts for period $T - 1$ for both the original data and the data protected using baseline privacy methods such as differential privacy and additive noise. The data controller measures the *difference* in forecast errors, denoted e_j^d, and time series feature values, denoted f_j^d for feature f, between each original and protected~~

Formatted: Font: Cambria, Not Italic

Formatted: Font: Cambria

Formatted: Font: Cambria, Not Italic

Formatted: Font: Cambria

Formatted: Centered

Formatted: Font: Cambria, Not Italic

Formatted: Font: Cambria, Not Italic

Formatted: Underline

Formatted: Font: Cambria, Not Italic

Formatted: Font: Cambria, Not Italic

Formatted: Font: Cambria, Not Italic

Formatted

Formatted: Font:

~~time series. Our reasoning is that a feature should be included in the *k*-nTS swapping process if changes in that feature are predictive of changes in forecast error.~~

~~We perform a two-stage feature selection process. An initial filtering of the features is performed by the RReliefF algorithm (Robnik-Sikonja & Kononenko 2003) which assigns a weight to each feature that indicates the ability of changes in that feature to predict changes in forecast accuracy. Next, we use random forest to generate permutation-based feature importance scores for the features with the largest RReliefF weights. Ultimately, we include the subset of features that have the largest importance values based on the random forest.~~

The *k*-nTS+ algorithm can be used collaboratively between the data controller and the forecaster. If, for example, the forecaster specifies their preferred forecasting model~~l(s)~~, the data controller can apply the model~~(s)~~ to the original and protected data up through time period *T – 1*, assess which changes in features are most predictive of changes in accuracy for the specified model~~(s)~~, and release data to the forecaster using *k*-nTS+ based on these features up through time period *T*.

6. Empirical Application

6.1. Data

Recent work by (Spiliotis et al., 2020) showed that the M3 competition data are representative of the real world on the basis of time series ~~characteristics~~features. Further, since the M3 competition data is publicly available, it will enable researchers to easily compare future privacy methods applied to the same data.~~ C~~ It is also known that complex 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~~We test 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. Of the 474 series, 18 consist of 67 time periods, 259 consist of 68 time periods, and 197 consist of 125 time periods.

6.2. Forecasting Models

The forecasting models under study 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. 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. Additional model descriptions are given in Section 5.4, where we explore the performance of each model mathematically based on the time series features. Please see the appendix for full implementation details.

Table 2~~3~~: forecasting models under study. Includes relevant information for the variant of model and whether it is a local or global forecasting model.~~ We consider the VAR somewhere in between a local and a global model – it must be trained on subsets of the M3 data due to its computational complexity.~~

	Model Name	Variant	Global (Yes/No)
Simple Models	SES	-	N
	DES	Additive trend	N
	TES	Additive trend/seasonality	N
	Auto-ARIMA	seasonal	N
Complex Models	VAR	-	N-
	LGBM	-	Y
	RNN	LSTM	Y

6.3. Privacy Methods

We apply each of the privacy methods shown in Table 3 below to the original M3 monthly micro data for each of the displayed parameter values.

Table 3: privacy methods, and their parameter values, which we apply to the m3 monthly micro data. Values are arranged in order of strength of privacy protection.

Privacy Method	***Parameter***	***Values***
Additive Noise	s	0.25, 0.50, 1.0, 1.5, 2.0
Differential Privacy	ϵ	20.0, 10.0, 4.6, 1.0, 0.1
k-nTS	k	3, 5, 7, 10, 15
k-nTS+	k	3, 5, 7, 10, 15

6.3.1. Differential Privacy

Given an original time series $\boldsymbol{A}$, a differentially private time series can be created using a randomized mechanism $M(\mathbf{A}) = \mathbf{A} + \mathbf{N}$ which adds Laplace random noise $\boldsymbol{N}$ with scale parameter $\Delta f_1/\epsilon$. The sensitivity Δf_1 is determined as the maximum absolute difference between two time series $\boldsymbol{A}$ and $\boldsymbol{A'}$, which differ in at most one observation, where $\Delta f_1 = max\left|\left|A - A'\right|\right|_1$. The mechanism M satisfies ϵ-differential privacy by guaranteeing that, for every output $\boldsymbol{t}$ of M and every pair of series $\boldsymbol{A}$ and $\boldsymbol{A'}$,

Commented [B2]: Could say we do not consider top/bottom coding because it can be accounted for (censored model)

$$Pr(M(\boldsymbol{A}) = \boldsymbol{t}) \leq exp(\epsilon)Pr(M(\boldsymbol{A'}) = \boldsymbol{t}).$$

6.3.2. Additive Noise

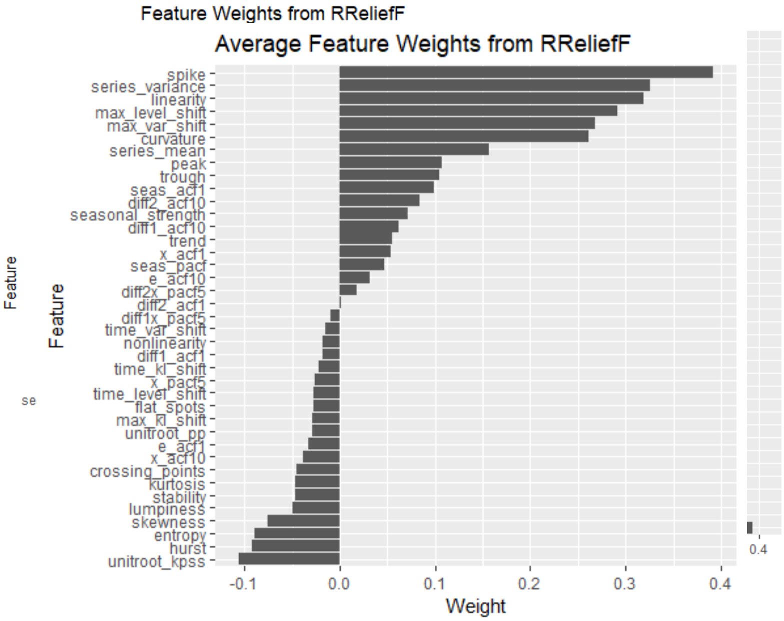
Additive noise protection is achieved by adding a normal random number with mean zero and standard deviation σ to each value in a time series x_j. Protected values can be written $P_{j,t} = A_{j,t} + r$, where $r \sim N(0, \sigma^2)$ and $\sigma = s * \sqrt{\left(E\left[\left(x_j - E\left[x_j\right]\right)^2\right]\right)}$. The protection parameter s denotes the number of standard deviations of x_j that define the standard deviation of the sampling distribution of r.

6.3.3. k-nTS and k-nTS+

For *k*-nTS, the distance between time series is calculated using the nine features described in Section 4.3. To perform feature selection for *k*-nTS+, we create protected versions of our selected data using additive noise and differential privacy for all of the parameter values shown in Table 3 (i.e., 10 protected data sets and 1 original data set). We generate forecasts for each of the 11 data sets for time period $T - 1$ using each of the forecasting models shown in Table 3 and compute the absolute error of each forecast for each series. So that the variation in forecast accuracy is due to changes in time series features and not the forecasting model used, we apply the *k*-nTS+ selection method for each forecasting model separately and select the features that are most important across all models.

~~We compute the differences in the absolute error between the original and protected forecasts for each series and model and use these differences as the target variable in the RReliefF algorithm.~~ W~~Next, w~~e compute 39 time series features using the tsfeatures package in R, including the nine features described in Section 3, and use custom functions for the series mean, variance, skewness, and kurtosis. Using RReliefF, ~~the differences in~~ the~~se~~ features ~~for each series~~ are used to predict the ~~differences in~~ absolute forecast errors for each model for each series ~~between~~ across the original and protected data sets. Figure 4 shows the average RreliefF weights for each of the 39 features across forecasting models.~~.~~

Figure 4: Average RReliefF weights across the results for each forecasting model.~~**Feature weights from RReliefF algorithm.**~~

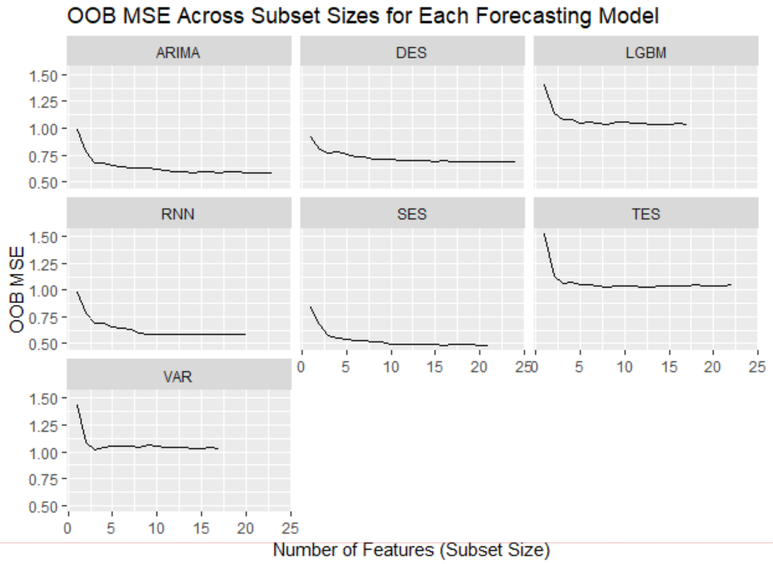


~~Several~~ Many features are poor predictors of ~~changes in~~ forecast error and are assigned negative weights. ~~Many of the features have small positive weights which are less than 0.10.~~ We select the features for each forecasting model with positive weights ~~greater than 0.10~~ for ~~inclusion in the random forest~~inclusion in the RFE algorithm.

We apply the RFE algorithm once for each forecasting model for $N_{rfe} = 50$ Iterations. The average oob MSE across feature subset sizes and models is shown in Figure 5. For all of the forecasting models, most of the reduction in oob MSE is accomplished using five or fewer features.

Fig. 5: Average OOB MSE across feature subset sizes when predicting the MAE of each forecasting model.

Formatted: Font: Bold

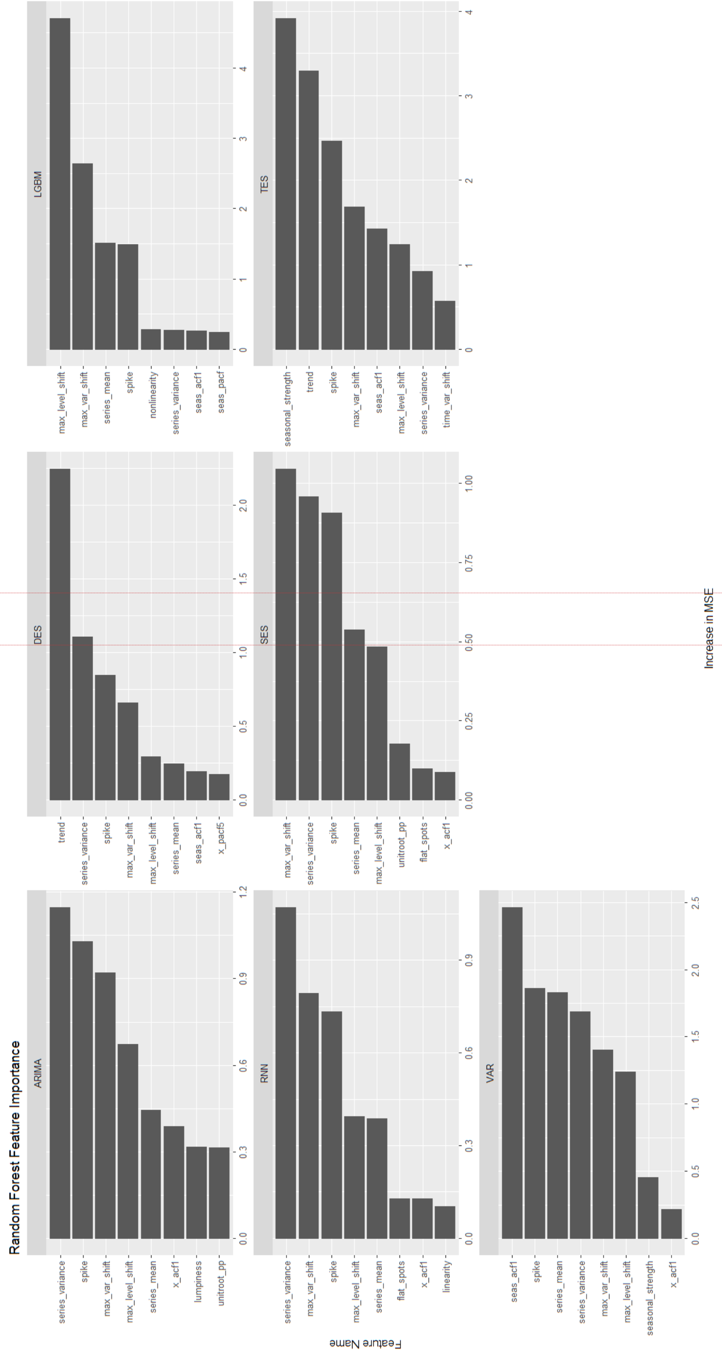


Formatted: Font: Cambria, Not Italic

Figure 6 shows the permutation-based importance values for the eight most highly ranked features for each forecasting model. Some features, such as *spike, max_var_shift, max_level_shift, series_mean* and *series_variance* are highly ranked across most or all forecasting models. Other features appear to be highly important only for specific models. Examples include *trend*, which is highly important for DES and TES, *seasonal_strength*, which is highly important for TES, and *unitroot_pp*, which is important for Auto-ARIMA and SES.~~T~~

Formatted: Font: (Default) +Body (Calibri)

Fig 6: Permutation importance for the eight most important features (based on RFE elimination rank) for each model.



Formatted: Font: Bold

Formatted: Font: Cambria, Not Italic

Formatted: Font: Not Bold

For our empirical application, we select $\tau = 0.05$. The average p^* across forecasting models is six, so we select the six features with the highest average rank across the RFE iterations for all forecasting models. These features are shown in Table 4.

~~We include all features with RReliefF weights greater than 0.10 in the random forest selection procedure, and those results are shown in Figure 5. The random forest flags several features as poor predictors of changes in forecast error, assigned them negative weights. We include the features with positive random forest importance values in *k*-nTS+. These features and their descriptions are shown in Table 4.~~

~~FIG 5: Random forest feature importance results.~~

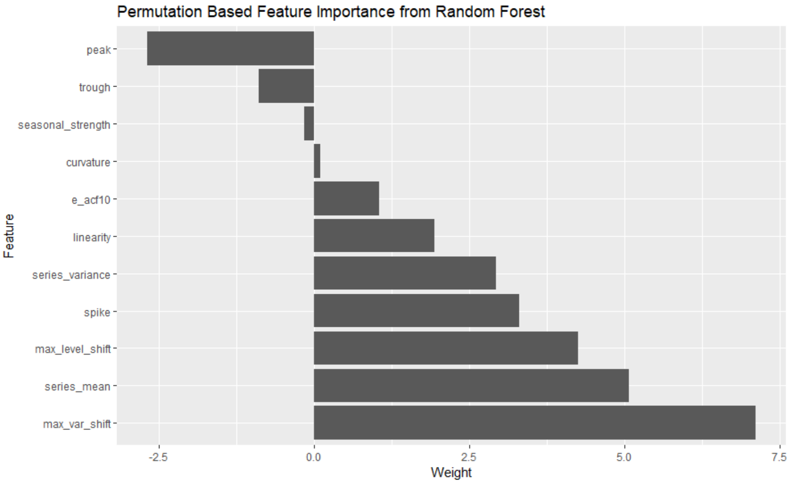


Table 4: Names and descriptions of features selected for *k*-nTS+.

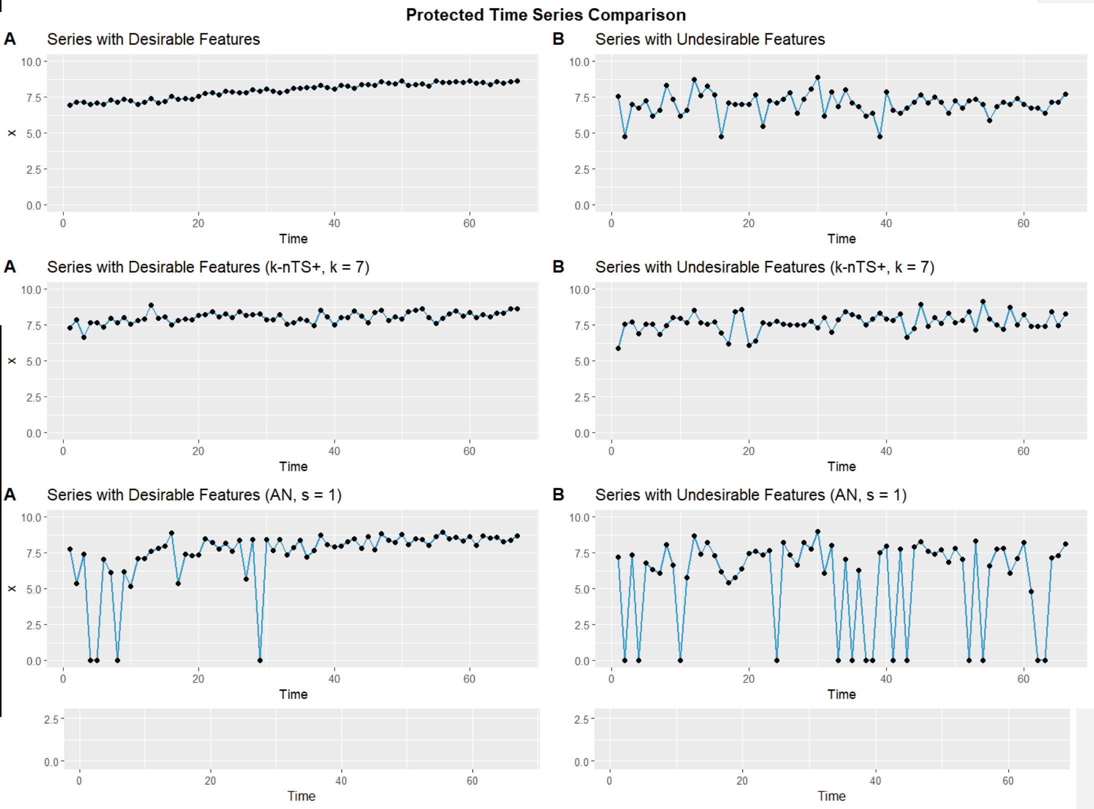
Feature Name	Description
trend~~series_mean~~	~~Mean of the time series.~~Strength of trend.
~~series_variance~~ spike	~~Variance of the time series.~~Variance of the leave-one-out variances of the remainder component of the decomposed series.
max_var_shift~~linearity~~	~~First non-intercept coefficient of an orthogonal quadratic regression.~~The largest variance shift between two consecutive sliding windows.
~~curvature~~ series_variance	~~Second non-intercept coefficient of an orthogonal quadratic regression.~~Variance of the series.

~~max_var_shift~~ max_level_shift	The largest mean shift between two consecutive sliding windows.~~Largest shift in the variance between two consecutive sliding windows.~~
~~max_level_shift~~ series_mean	~~Largest shift between the means of two consecutive sliding windows.~~Mean of the series.

~~The features selected for *k*-nTS+ can be divided into three groups. The first group, consisting of the mean, variance, linearity, and curvature, defines the overall shape of a time series (in this case the overall shape of the series within the sliding window). The second group measures shifts in the location (mean) and shape (variance) within a time series window. The third group captures important characteristics of the remainder component.~~

Figure 7~~X~~ compares some protected versions of the series with desirable and undesirable features from Section 4.

Figure 7~~X~~: Comparison of original, AN (s = 1), and *k*-nTS+ (*k* = 7) versions of the series with desirable and undesirable features from Section 4.~~.~~



6.4. *Forecast Accuracy and Privacy Assessment*

The participants of the M3 competition were not supposed to identify the original time series. Therefore, w~~W~~e assess the ability of each privacy method to protect against *identification disclosure*, which occurs when a third party correctly predicts the identity of a protected time series. ~~We assume a protected~~Each protected dataset consists of ~~each~~ the protected series along with a pseudo identifier, i.e., $X = \left[(PID_1, x_1), (PID_2, x_2), \ldots \left(PID_J, x_J\right)\right]^T$. This pseudo-identifier has no relation with the true identity of a time series. The pseudo identifier in our application is the ~~For example,~~ `Series` column from the original M3 data which contains a PID for each time series, e.g., `N1402`. ~~a data set that requires protection might consist of the daily sales quantities of various grocery retailers. In a protected version of this data set, a pseudo-identifier could consist of a randomly generated number for each retailer which links observations across time for a given retailer but does not reveal the retailer's identity.~~ Identification disclosure would occur if a competition participant (or any other third party) ~~correctly predicts~~were to correctly predict the identity of one or more of the time series in the M3 data set based on the protected time series and some original time series values which the third party possesses. For example, identification disclosure would occur if a third party were to correctly make the statement, "Series N1402 comes from [retailer] store number 5249." ~~of a retailer, based on the protected daily sales quantities, and some original daily sales quantities which the third party possesses.~~ For simplicity, we assume the third party does not know which privacy method was applied to the data and knows that the time series of interest is contained in the protected data set.[8]

For brevity, we have included the math behind our definition of identification disclosure in the appendix. The metric we use to measure the risk of identification disclosure, $\bar{P}$, gives the average proportion of the J time series which are correctly identified across S simulated privacy attacks:

$$\bar{P} = \frac{1}{J * S} \sum_{s=1}^{S} \sum_{i=1}^{J} \left[\widehat{M_i^s} = j^*\right]$$

[8] We note that there are other privacy leaks such as attribute disclosure (Li et al. 2007) or membership inference (Shokri et al. 2017). Identification disclosure is the most applicable for our data, and we consider this a steppingstone to additional privacy leaks – e.g., identifying a time series within a protected data set enables a third party to learn unknown information with greater certainty.

where $\widehat{M_i^s}$ is the third party's prediction of the identity of the ith protected time series, and identification disclosure occurs when the predicted identity is equal to the true identity j^*.

While we seek to protect the time series identities, we also want the protected versions of the time series to be realistic. Forecasters prefer forecasts that are representative, i.e., typical of the data used to produce the forecast (Petropoulos & Siemsen, 2022). Likewise, organizations (and forecasters) will prefer using protected data that is representative of the original data, which our method provides through the preservation of time series features. We use the *performance gap* portion of the REP metric of Petropoulos & Siemsen (2022) to measure the distance between the protected and original time series values,

$$\text{Performance gap} = \left|\left|\boldsymbol{x}_t, \boldsymbol{x}'_t\right|\right|_1.$$

Table 5 contains the average MAE of one-step ahead point forecasts across all models, ~~and~~ the identification disclosure metric $\bar{P}$, and the average performance gap across all series ~~ ~~for several privacy parameters for each privacy method. For the simulated privacy attacks, we extract $S = 20$ random samples of external data containing ten values from each time series. ~~There is~~The results show a negative ~~clear~~ relationship between forecast accuracy and the strength of privacy protection. While strong differential privacy provides the lowest risk of identification ~~and attribute disclosure~~, it more than triples the average forecast error relative to the original data. Essentially unusable forecasts are produced under differential privacy and additive noise unless privacy is quite weak ($\epsilon \ \geq$~~$=$~~ 10, or $s \ \geq$~~$=$~~ 1). For example, under differential privacy with $\epsilon = 10$, nearly 50% of series are identified correctly on average, while MAE has increased by just over 31~~0~~%. Protection against identification disclosure is better under additive noise with $s = 1$: about 22% of series are correctly identified on average. But, this comes at further cost to forecast accuracy, which is reduced by nearly 45%. Standard k-nTS with k = 3 offers a better trade-off – protection against identification disclosure is quite good, since only 2% of series are correctly identified on average, while accuracy is reduced by about 40%. So, for a similar reduction in accuracy to additive noise with $s = 1$, k-nTS gives better privacy. k-nTS+ offers better protection against identification disclosure than additive noise ($s = 1$) and differential privacy ($\epsilon = 10$) with a reduction in accuracy of only 14~~3~~%.~~%.~~

The results also show that k-nts+ produces protected time series with the smallest performance gaps by a large margin. This is important because organizations need to trust the protection measures before implementing them – since k-nts+ produces plausible time series, this protection is more likely to be used. Further, k-nTS+ ($k = 3$) produces forecasts which are usable, i.e., accuracy is only reduced by approximately 14%.

~~δ~~

Formatted: Not Superscript/ Subscript

Formatted: Font: (Default) Cambria Math

Formatted: Font: +Body (Calibri)

		Privacy Method											
	Original	**Additive Noise**			**Differential Privacy**			***k*-nTS**			***k*-nTS+**		
Parameter	-	**1.0**	**1.5**	**2.0**	**10**	**4.6**	**1.0**	**3**	**7**	**15**	**3**	**7**	**15**
AvgPropIdent	-	22.51~~29~~	10.359~~9.99~~	5.84~~57~~	49.038~~8.66~~	13.552~~2.44~~	1.85~~95~~	2.052~~2.03~~	2.122~~2.00~~	1.57~~96~~	3.264~~4.5~~	3.50~~8~~	2.733~~3.28~~
Average MAE	685.71	993.95	1343.29	1821.38	899.38	1400.95	3310.34	956.89	987.04	1066.16	781.027~~78.27~~	822.267~~797.91~~	839.784~~8.11~~
Average Performance Gap	-	142,095	304,482	489,840	73,803	305,396	1,930,653	112	120	127	77	82	90

Formatted: Font: Not Italic

Commented [B3]: Make sure epsilon value for DP is correct - review write-up on it. Make sure it is still differentially private in the presence of external information.

Formatted Table

* The average across models for additive noise and differential privacy excludes the VAR model error for AN (s = 1) and DP ($\epsilon = 0.1$) as the errors in these cases were over 1000% larger than the error of any other model. This was due to extremely large noise values being added to the very end of some time series, causing the VAR forecasts to explode.~~ The reasons for this will be explored in section -.~~

In **Table 5~~4~~,** the models are rank-ordered based on their MAE on the original data. We also display the rank of each model based on forecast error variance. TES and ARIMA, which explicitly model the seasonality of the series, had the best accuracy which is consistent with the findings of the original M3 competition (Makridakis & Hibon, 2000). These models also had the lowest error variance~~.~~…. ~~However, their performance suffers on the protected data, as the SES, DES, and RNN models outperform them on both accuracy and error variance. Overall, DES and SES are the simplest models and have the best accuracy and lowest error variance across protected data sets. LGBM is ranked equal to Auto-ARIMA on average rank, while VAR consistently performs the worst.~~

Table 5~~4~~: the rank of each model in terms of MAE and forecast error variance on the original data vs. the ~~average rank across protected datasets~~knts+ (*k*=3) data. The rightmost column contains the average of the protected MAE and error variance ranks.

Commented [B4]: For k-nts+ only

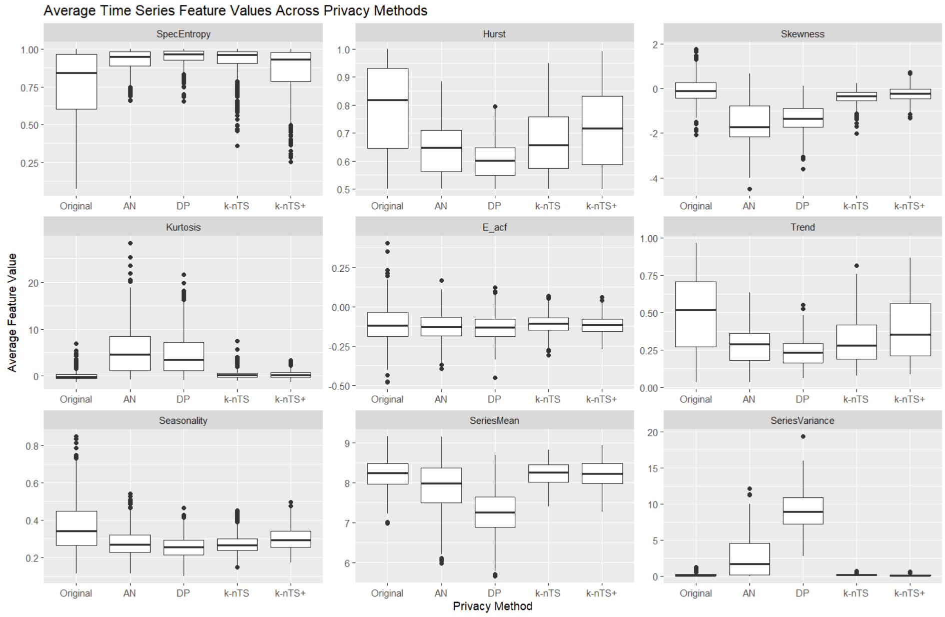
	MAE Ranks		Forecast Error Variance Ranks		Average Rank
Model	Original	Protected	Original	Protected	Protected
TES	1	1~~4.5~~	2	4~~3.5~~	2.5~~4~~
ARIMA	2	4~~.5~~	1	1~~4.25~~	2.5~~4.38~~
RNN	3	5~~3~~	5	5~~3.25~~	5~~3.13~~
DES	4	2~~.5~~	3	2	2~~2.25~~
SES	5	3~~2.5~~	4	3~~.25~~	3~~2.88~~
LGBM	6	6~~4~~	7	6~~4.75~~	6~~4.38~~
VAR	7	7~~.0~~	6	7	7

6.5. *Changes in Time Series Features*

~~6.5.~~

In **Figure** 8~~5~~, we calculate the average feature value for each series across the protected datasets for each privacy method. We plot these distributions next to the distribution of each feature from the original data.

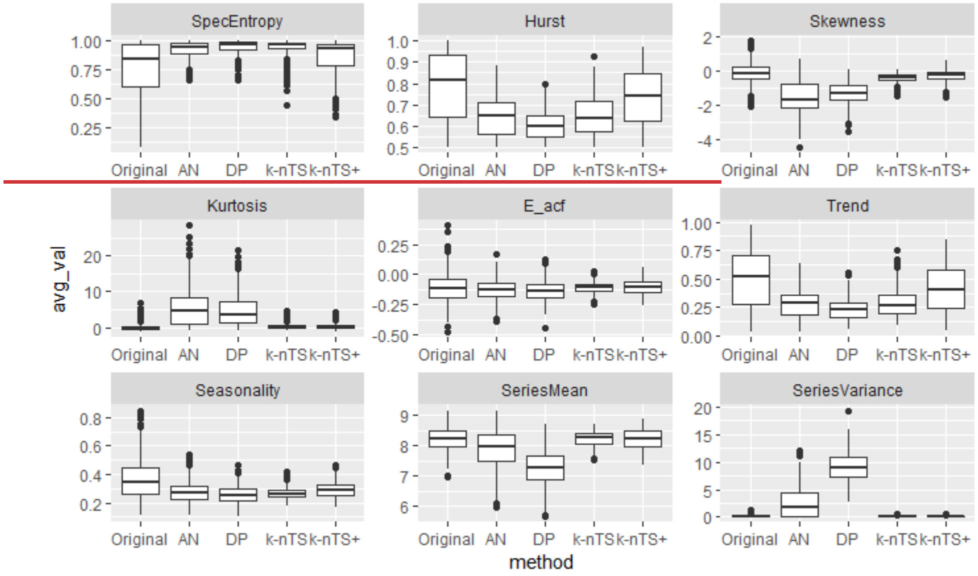
Fig 8: distributions of average time series features across protected data sets for each privacy method.



~~**Figure 5: distributions of the original feature values for each series and the average feature values for each series across protected datasets for each privacy method.**~~

Formatted: Indent: Left: 0.55", No bullets or numbering

Formatted: Font: Bold



- While all privacy methods tend to increase spectral entropy, the distribution under *k*-nTS+ is much closer to the original than the other methods.
- The most notable differences between the privacy methods show up in skewness, kurtosis, series mean, and series variance.
  - Random noise-based protection produces a negative skew in log-transformed time series, creates heavy tails, biases the mean downward, and blows up the variance.
- Standard K-nTS increases the randomness and reduces the long-term dependence of series, including reducing the strength of trend and seasonality. Swapping by directly matching on these features does not preserve these features well.
- Distributional features, namely *SeriesVariance*, *SeriesMean*, *Skewness*, and *Kurtosis*, are much less affected under standard k-nTS than random noise protection.
- The features selected in *k*-nTS+ are the "building blocks" for maintaining the other features such as Hurst coefficient and the strength of trend.
- None of the protection methods appear to do particularly well at preserving the strength of seasonality, although features related to seasonality (strength of seasonality, peak, and trough) were determined not to be useful for predicting changes in forecast error from data protection (for this particular data set).
- Overall, k-nTS makes the series more random, but preserves some of their distributional patterns. K-nTS+ does a better job of preserving the spectral entropy, hurst, and strength of trend, leading to improved forecast accuracy.
- Random noise protection produces large changes both in the distributions of time series increases their randomness, and poor forecast accuracy.

~~6.6. Detailed Model Performance Explanations~~

Formatted: Font: (Default) Cambria

Formatted: List Paragraph, Bulleted + Level: 1 + Aligned at: 0.5" + Indent at: 0.75"

Formatted: List Paragraph

Formatted: Indent: Left: 0.5", No bullets or

~~The simplest model we examine is SES, where the forecast for time period $t + h$ is a weighted average of the time series values up through time period t,~~

Formatted: List Paragraph

~~$$\hat{x}_{t+h|t} = l_t$$~~

Formatted: List Paragraph, Left

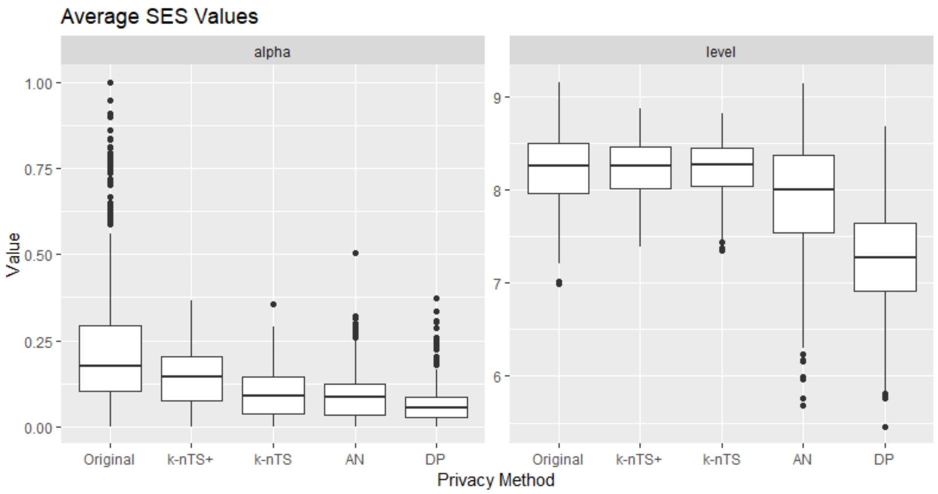
~~$l_t = \alpha x_t + (1 - \alpha) l_{t-1}$.~~

Formatted: List Paragraph

~~Changes in the distribution of x_t will directly affect the value of the level l_t. For example, positively (negatively) skewing the distribution of x_t will shift the level in the positive (negative) direction, directly affecting the accuracy of the forecasts.~~

~~In terms of model fitting, SES tends to select smaller values of α when applied to data that is less forecastable. Figure 5 shows that on average, the value of α is lower for series with higher spectral entropies and lower Hurst coefficients. The good news is that this means SES is not treating the random noise in the series as a signal, and instead tends to smooth it out.~~

~~**Fig 5:** Average SES α value and level l_t across series and protected data sets for each privacy method.~~



~~DES builds on SES by accounting for the trend of a series,~~

~~$\hat{x}_{t+h|t} = l_t + hb_t$~~
~~$l_t = \alpha x_t + (1-\alpha)(l_{t-1} + b_{t-1})$~~
~~$b_t = \beta(l_t - l_{t-1}) + (1-\beta)b_{t-1}.$~~

~~TES builds on DES by incorporating seasonal components,~~

~~$\hat{x}_{t+h|t} = l_t + hb_t + s_{t+h-m(k+1)}$~~
~~$l_t = \alpha(x_t - s_{t-m}) + (1-\alpha)(l_{t-1} + b_{t-1})$~~
~~$b_t = \beta(l_t - l_{t-1}) + (1-\beta)b_{t-1}$~~
~~$s_t = \gamma(y_t - l_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m}.$~~

~~The ARIMA$(p, d, q)(P, D, Q)_m$ model is a function of *p* autoregressive parameters ϕ_p, *P* seasonal autoregressive parameters Φ_P, *q* moving average parameters θ_q, and *Q* seasonal moving average parameters Θ_Q,~~

~~$x_t = c + \phi_1 x_{t-1} + \cdots + \phi_p x_{t-p} +$~~
~~$\theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q} +$~~
~~$\Phi_1 x_{t-m} + \cdots + \Phi_P x_{t-mP} +$~~
~~$\Theta_1 \epsilon_{t-m} + \Theta_Q \epsilon_{t-mQ} + \epsilon_t.$~~

~~The VAR model forecasts *K* time series where the forecast for series *k* is a function of *p* lagged values of series *k* and each of the other *k – 1* series,~~

~~$x_{k,t} = c_k + \phi_{k1,1} x_{1,t-1} + \phi_{k2,1} x_{2,t-1} + \cdots + \phi_{kk,1} x_{k,t-1} +$~~
~~$\phi_{k1,2} x_{1,t-2} + \cdots + \phi_{kk,p} x_{k,t-p} + \epsilon_{k,t}$~~

-

Formatted: List Paragraph, Left

Formatted: List Paragraph

Formatted: List Paragraph, Left

Formatted: List Paragraph

Formatted: List Paragraph, Left

Formatted: List Paragraph

Formatted: List Paragraph, Bulleted + Level: 1 + Aligned at: 0.5" + Indent at: 0.75"

7. Conclusions

This paper has examined forecasting using protected data. The privacy-utility tradeoff is manifest in the reduction in forecast accuracy that occurs when improving the privacy of time series. A substantial portion of the privacy literature is focused on theoretical privacy guarantees, i.e., differential privacy. Our findings agree with past research (Goncalves et al. 2021) and show that differential privacy (and additive noise) lead forecasting models to generate unusable forecasts at meaningful levels of privacy. This undesirable privacy-utility tradeoff under differential privacy has been demonstrated in other contexts as well. A recent paper by Blanco-Justicia et al. (2022) found that much of the work on differential privacy and deep learning utilized relaxed versions of differential privacy with values of ϵ that theoretically do not provide meaningful levels of privacy protection. Their experiments found that model regularization (e.g., L2-regulatization) provided comparable privacy protection with better accuracy and lower model learning cost than differential privacy.

Rather than adding random noise to time series, our proposed *k*-nTS+ privacy method uses time series features as a basis for swapping the values between time series. We demonstrated the effectiveness of our protection approach using data from a ~~well known~~well-known forecasting competition where the identities of the time series needed to be kept confidential. The proposed method kept average forecast accuracy within 10-15% of the original data and ~~Compared to other protection methods, k-nts+~~ preserved~~s~~ the ranking of the best and worst forecasting models, which was the focus of the competition. Research in other domains found that a 10-15% loss in usability is often a best-case scenario under privacy protection (Schneider et al. 2018). Further, the proposed privacy method provides comparable levels of privacy to differential privacy at meaningful levels of ϵ while enabling models to produce usable forecasts.
~~(i.e., which model performs the best?)~~

Formatted: Font: Not Bold

Formatted: Font: Not Bold

Formatted: Font: Not Bold

Formatted: Font: Not Bold

Formatted: Font: Not Bold

Our goal is to increase the ease with which organizations can protect their data. The proposed *k*-nTS+ method reduces the frictions of implementing privacy protection since organizations need only select an appropriate value for *k* and apply the method to their data. Decentralization based methods, such as those in (Gonçalves et al., 2021; Goncalves, Bessa, et al., 2021; Sommer et al., 2021) are effective but require a more complicated decentralized framework. Our proposed method ~~We have produced a privacy method and keeps forecast accuracy within 10-15% by using features to inform the protection. This~~ increases the likelihood that organizations will protect their data ~~– reduces the frictions with implementing privacy protection. Enables~~and enables data sharing.

Formatted: Normal, No bullets or numbering

Formatted: Font: (Default) Cambria

While we showed that *k*-nTS+ preserves data utility for forecasting, future work should examine the utility of protected time series for other use cases, such as classification. Future work should also assess forecasting with protected data using multiple forecast horizons. A theoretical examination of forecasting model performance on protected data would help us further understand which models will perform well under different conditions. This would help improve model selection procedures when forecasting with protected data since we are seek a model that forecasts well for unseen data, not necessarily the protected data. Future work should also assess whether forecast combinations, which tend to improve accuracy on unprotected data (Makridakis et al. 2018), are also beneficial when forecasting with protected data.

ϵ

- ~~**Usability**~~
- ~~**Preservation of privacy**~~
- ~~**Trustworthiness**~~
- ~~**Privacy literature has focused on theoretical privacy guarantees which results in unusable forecasts**~~
- ~~**We have produced a privacy method and keeps forecast accuracy within 10-15% by using features to inform the protection. This increases the likelihood that organizations will protect their data – reduces the frictions with implementing privacy protection. Enables data sharing.**~~
- ~~**We demonstrated the effectiveness of our protection approach using data from a well known forecasting competition where the identities of the time series**~~

Formatted: Normal, No bullets or numbering

~~needed to be kept confidential. Compared to other protection methods, k-nts+ preserves the ranking of forecasting models which was the focus of the competition (i.e., which model performs the best?)~~

- ~~Research in other domains found that 10-15% loss in usability is often a best case scenario for protection (Matt's MS paper). (There's no free lunch).~~
- ~~Future research~~
    - ~~Preservation of time series features – does it preserve utility for other use cases?~~
    - ~~Multiple forecast horizons~~
    - ~~Theoretical treatment?~~
    - ~~Forecast combination of methods after protection~~

References

Abowd, J. M., Gittings, K., McKinney, K. L., Stephens, B. E., Vilhuber, L., & Woodcock, S. (2012). *Dynamically consistent noise infusion and partially synthetic data as confidentiality protection measures for related time-series*. 41.

Bandara, K., Bergmeir, C., & Smyl, S. (2018). Forecasting Across Time Series Databases using Recurrent Neural Networks on Groups of Similar Series: A Clustering Approach. ArXiv:1710.03222 [Cs, Econ, Stat]. http://arxiv.org/abs/1710.03222

Blanco-Justicia, A., Sanchez, D., Domingo-Ferrer, J., & Muralidhar, K. (2022). A Critical Review on the Use (and Misuse) of Differential Privacy in Machine Learning. arXiv preprint arXiv:2206.04621.

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*(1), 170–180. https://doi.org/10.1016/j.ijforecast.2018.09.003

Chen, C., & Liu, L.-M. (1993). Forecasting time series with outliers. *Journal of Forecasting*, *12*(1), 13–35. https://doi.org/10.1002/for.3980120103

Formatted: Font: Not Italic

Formatted: Font: (Default) Cambria, 11 pt, Font color: Auto, Pattern: Clear

Crimi, N., & Eddy, W. (2014). Top-Coding and Public Use Microdata Samples from the U.S. Census Bureau. *Journal of Privacy and Confidentiality*, *6*(2). https://doi.org/10.29012/jpc.v6i2.639

Davydenko, A., & Fildes, R. (2013). Measuring forecasting accuracy: The case of judgmental adjustments to SKU-level demand forecasts. *International Journal of Forecasting*, *29*(3), 510–522. https://doi.org/10.1016/j.ijforecast.2012.09.002

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), 1376. https://doi.org/10.1038/srep01376

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*(1), 3–23. https://doi.org/10.1016/j.ijforecast.2008.11.010

Fildes, R., Goodwin, P., & Önkal, D. (2019). Use and misuse of information in supply chain forecasting of promotion effects. *International Journal of Forecasting*, *35*(1), 144–156. https://doi.org/10.1016/j.ijforecast.2017.12.006

Fulcher, B. D., & Jones, N. S. (2014). Highly Comparative Feature-Based Time-Series Classification. *IEEE Transactions on Knowledge and Data Engineering*, *26*(12), 3026–3037. https://doi.org/10.1109/TKDE.2014.2316504

Goerg, G. M. (n.d.). *Forecastable Component Analysis*. 9.

Goldfarb, A., & Tucker, C. (n.d.). *Why Managing Consumer Privacy Can Be an Opportunity*. 6.

Goldfarb, A., & Tucker, C. E. (2011). Privacy Regulation and Online Advertising. *Management Science*, *57*(1), 57–71. https://doi.org/10.1287/mnsc.1100.1246

Gonçalves, C., Bessa, R. J., & Pinson, P. (2021). A critical overview of privacy-preserving approaches for collaborative forecasting. *International Journal of Forecasting*, *37*(1), 322–342. https://doi.org/10.1016/j.ijforecast.2020.06.003

Goncalves, C., Bessa, R. J., & Pinson, P. (2021). Privacy-Preserving Distributed Learning for Renewable Energy Forecasting. *IEEE Transactions on Sustainable Energy*, *12*(3), 1777–1787. https://doi.org/10.1109/TSTE.2021.3065117

Goncalves, C., Pinson, P., & Bessa, R. J. (2021). Towards Data Markets in Renewable Energy Forecasting. IEEE Transactions on Sustainable Energy, 12(1), 533–542. https://doi.org/10.1109/TSTE.2020.3009615

Gregorutti, B., Michel, B., & Saint-Pierre, P. (2017). Correlation and variable importance in random forests. *Statistics and Computing*, 27(3), 659-678.

Hewamalage, H., Bergmeir, C., & Bandara, K. (2021). Recurrent Neural Networks for Time Series Forecasting: Current status and future directions. International Journal of Forecasting, 37(1), 388–427. https://doi.org/10.1016/j.ijforecast.2020.06.008

Hewamalage, H., Bergmeir, C., & Bandara, K. (2022). Global models for time series forecasting: A Simulation study. *Pattern Recognition*, *124*, 108441. https://doi.org/10.1016/j.patcog.2021.108441

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. *2020 IEEE International Conference on Big Data (Big Data)*, 3428–3437. https://doi.org/10.1109/BigData50022.2020.9378186

Kang, Y., Hyndman, R. J., & Smith-Miles, K. (2017). Visualising forecasting algorithm performance using time series instance spaces. *International Journal of Forecasting*, *33*(2), 345–358. https://doi.org/10.1016/j.ijforecast.2016.09.004

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (n.d.). *LightGBM: A Highly Efficient Gradient Boosting Decision Tree*. 9.

Formatted: Font: Not Italic

Formatted: Font: Not Italic

Formatted: Font: (Default) Cambria, 11 pt, Font color: Auto, Pattern: Clear

Formatted: Font: Not Italic

Formatted: Font: Not Italic

Khosrowabadi, N., Hoberg, K., & Imdahl, C. (2022). Evaluating human behaviour in response to AI recommendations for judgemental forecasting. *European Journal of Operational Research*, *303*(3), 1151–1167. https://doi.org/10.1016/j.ejor.2022.03.017

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(3), 397–409. https://doi.org/10.1016/j.ijforecast.2004.10.003

Li, N., Li, T., & Venkatasubramanian, S. (2006, April). t-closeness: Privacy beyond k-anonymity and l-diversity. In 2007 IEEE 23rd international conference on data engineering (pp. 106-115). IEEE.

Li, L., Kang, Y., & Li, F. (2022). Bayesian forecast combination using time-varying features. International Journal of Forecasting, S0169207022000930. https://doi.org/10.1016/j.ijforecast.2022.06.002

Liyue Fan & Li Xiong. (2014). An Adaptive Approach to Real-Time Aggregate Monitoring With Differential Privacy. *IEEE Transactions on Knowledge and Data Engineering*, *26*(9), 2094–2106. https://doi.org/10.1109/TKDE.2013.96

Luo, J., Hong, T., & Fang, S.-C. (2018). Benchmarking robustness of load forecasting models under data integrity attacks. *International Journal of Forecasting*, *34*(1), 89–104. https://doi.org/10.1016/j.ijforecast.2017.08.004

Makridakis, S., & Hibon, M. (2000). The M3-Competition: Results, conclusions and implications. *International Journal of Forecasting*, *16*(4), 451–476. https://doi.org/10.1016/S0169-2070(00)00057-1

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). The M4 Competition: Results, findings, conclusion and way forward. *International Journal of Forecasting*, *34*(4), 802–808. https://doi.org/10.1016/j.ijforecast.2018.06.001

Formatted: Font: Not Italic

Formatted: Font: Not Italic

Formatted: Font: (Default) Cambria, 11 pt, Font color: Auto, Pattern: Clear

Formatted: Font: Not Italic

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2022). M5 accuracy competition: Results, findings, and conclusions. *International Journal of Forecasting*, S0169207021001874. https://doi.org/10.1016/j.ijforecast.2021.11.013

Martin, K. D., Borah, A., & Palmatier, R. W. (2017). Data Privacy: Effects on Customer and Firm Performance. *Journal of Marketing*, *81*(1), 36–58. https://doi.org/10.1509/jm.15.0497

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*(3), 627–651. https://doi.org/10.1016/j.ijforecast.2017.01.003

Nin, J., & Torra, V. (2009). Towards the evaluation of time series protection methods. *Information Sciences*, *179*(11), 1663–1677. https://doi.org/10.1016/j.ins.2009.01.024

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., Browell, J., Carnevale, C., Castle, J. L., Cirillo, P., Clements, M. P., Cordeiro, C., Oliveira, F. L. C., Baets, S. D., Dokumentov, A., … Ziel, F. (2022). *Forecasting: Theory and practice*. 167.

Petropoulos, F., & Siemsen, E. (2022). Forecast Selection and Representativeness. *Management Science*, mnsc.2022.4485. https://doi.org/10.1287/mnsc.2022.4485

Qi, L., Li, X., Wang, Q., & Jia, S. (2022). fETSmcs: Feature-based ETS model component selection. *International Journal of Forecasting*, S0169207022000954. https://doi.org/10.1016/j.ijforecast.2022.06.004

Schneider, M. J., Jagpal, S., Gupta, S., Li, S., & Yu, Y. (2018). A flexible method for protecting marketing data: An application to point-of-sale data. Marketing Science, 37(1), 153-171.

Smyl, S. (2020). A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *International Journal of Forecasting*, *36*(1), 75–85. https://doi.org/10.1016/j.ijforecast.2019.03.017

Formatted: Bibliography

Formatted: Font: Cambria

Formatted: Font: (Default) Cambria, 11 pt, Font color: Auto, Pattern: Clear

Formatted: Font: Cambria

Formatted: Font: +Body (Calibri)

Sobolev, D. (2017). The effect of price volatility on judgmental forecasts: The correlated response model. *International Journal of Forecasting*, *33*(3), 605–617. https://doi.org/10.1016/j.ijforecast.2017.01.009

Sommer, B., Pinson, P., Messner, J. W., & Obst, D. (2021). Online distributed learning in wind power forecasting. *International Journal of Forecasting*, *37*(1), 205–223. https://doi.org/10.1016/j.ijforecast.2020.04.004

Spiliotis, E., Kouloumos, A., Assimakopoulos, V., & Makridakis, S. (2020). Are forecasting competitions data representative of the reality? *International Journal of Forecasting*, *36*(1), 37–53. https://doi.org/10.1016/j.ijforecast.2018.12.007

Sweeney, L. (2002). k-ANONYMITY: A MODEL FOR PROTECTING PRIVACY. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *10*(05), 557–570. https://doi.org/10.1142/S0218488502001648

Talagala, T. S., Li, F., & Kang, Y. (2022). FFORMPP: Feature-based forecast model performance prediction. *International Journal of Forecasting*, *38*(3), 920–943. https://doi.org/10.1016/j.ijforecast.2021.07.002

Véliz, C., & Grunewald, P. (2018). Protecting data privacy is key to a smart energy future. *Nature Energy*, *3*(9), 702–704. https://doi.org/10.1038/s41560-018-0203-3

Wang, X., Smith, K., & Hyndman, R. (2006). Characteristic-Based Clustering for Time Series Data. *Data Mining and Knowledge Discovery*, *13*(3), 335–364. https://doi.org/10.1007/s10618-005-0039-x

Willinger, W., Paxson, V., Taqqu, M. S., & Willinger, W. (n.d.). *Self-Similarity and Heavy Tails: Structural Modeling of Network Tra c*. 26.

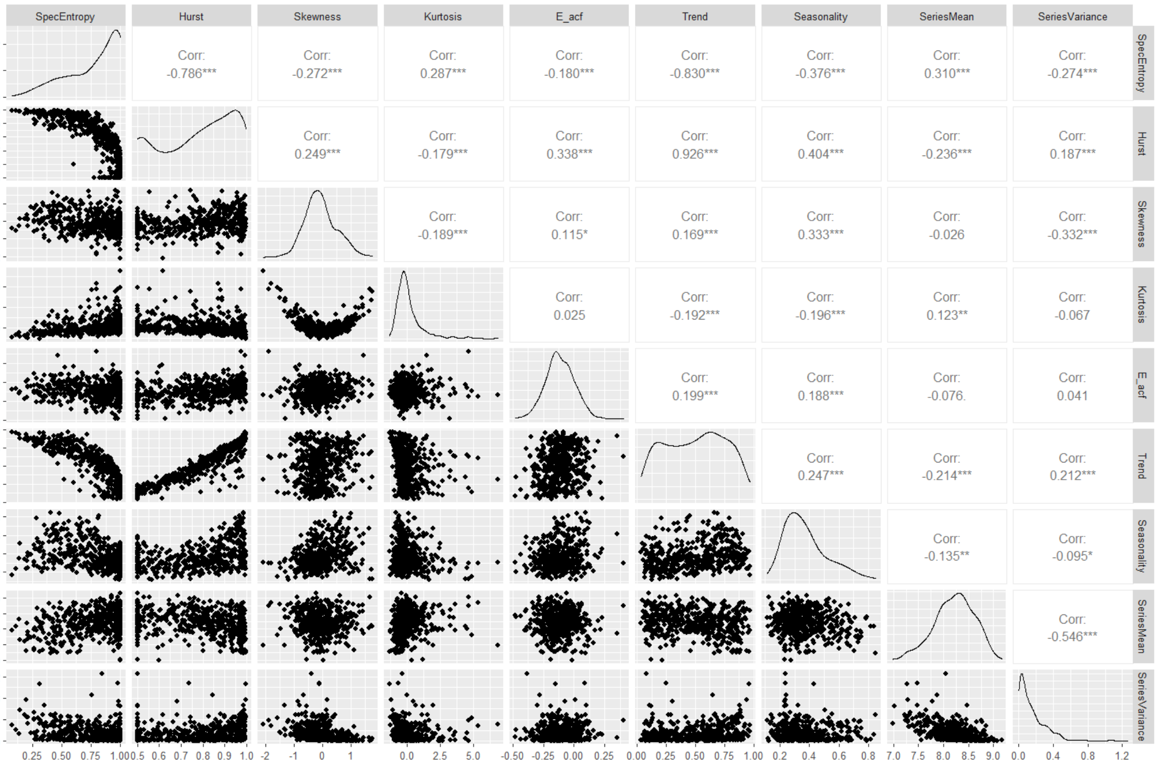
8. Appendix

The relationships between the time series features in the original data are shown in Figure 4.

- Spectral entropy distribution is skewed left, approximately 63% of series have a spectral entropy of at least 0.75 (a lot of series are already difficult to forecast).

- Spectral entropy is negatively correlated with features that improve forecastability such as Hurst exponent, strength of trend and seasonality, and remainder first autocorrelation coefficient (e-acf).
- Hurst is strongly correlated with strength of trend and seasonality
- Series with a larger mean tend to have higher variance

FIG 4 (Scatterplot matrix showing scatterplots between each feature pair, kernel density of each feature, and correlations between each feature pair for the original data.)



In the upper triangle of **FIG 4**, "***" denotes p-value < 0.001, "**" denotes p-value < 0.01, "*" denotes p-value < 0.05, "." denotes p-value < 0.10, "" is shown otherwise.

The Kullback-Leibler divergence between two probability densities $p(f)$ and $q(f)$ is defined as

$$\int_{\mathbb{R}^d} p(f) \log p(f)/q(f) \, d\text{f}$$

where $p(f)$ is the probability density of the original feature, and the probability density of the feature from the protected data is denoted $q(f)$. Following the approach of (Spiliotis et al., 2020), we approximate $p(f)$ and $q(f)$ using normalized kernel densities, and estimate the KL-divergence between $p(f)$ and $q(f)$ as

$$KL(p,q) = \sum_f \hat{p}(f) log \frac{\hat{p}(f)}{\hat{q}(f)}.$$

Letting $H(p)$ denote the entropy of $p(f)$, the percentage difference between $p(f)$ and $q(f)$ is approximately

$$PD(p,q) = \frac{KL(p,q)}{H(p)} * 100.$$

Mathematical Details of Identification and Attribute Disclosure

To perform identification disclosure, we assume a third party possesses some original data pertaining to a unit of interest in the protected dataset. For the above example, this would be some sequence of original daily sales quantities for a known retailer. Denote this original data $\boldsymbol{c_i} = (ID_i, c_i)$, which contains a direct identifier ID_i (e.g., the identity of retailer i) and original data $c_i = (A_{i,t'}, \dots, A_{i,t'+E})$ which contains a sequence of values which are components of the original time series x_j.

We let M_i denote the random variable (from the perspective of the third party) that indicates the corresponding PID_j for ID_i, i.e., $M_i = j$ when the values in $\boldsymbol{c}_i$ are components of the original version of protected series j. Since the true value $M_i = j^*$ is unknown, the third party predicts the value of M_i to be the series j with the highest match probability, conditional on the known values, as follows

$$\widehat{M}_i = argmax_j P(M_i = j|c_i), \quad (1)$$

where identification disclosure occurs when $\widehat{M_i} = j^*$. The probability $P(M_i = j|c_i)$ is calculated as follows. Let $\tilde{x}_j = (P_{j,t'}, \dots, P_{j,t'+E}), j = 1, \dots, J$ denote the protected values of each time series j that occur in the same time periods as c_i. The third party computes the similarity between c_i and the protected values $\widetilde{x_J}, j = 1, \dots, J$ using the Euclidean distance,

$$s(c_i, \widetilde{x_J}) = \frac{1}{\left|\left|c_i - \widetilde{x_J}\right|\right|_2}, j = 1, \dots, J.$$

Using these similarities the third party builds a probability mass function for M_i over all protected series in X' as

$$P(M\_i = j \mid c_i) = \frac{s(c_i, \widetilde{x_j})}{\sum_{j=1}^{N} s(c_i, \widetilde{x_j})},$$

and predicts $\widehat{M_i}$ as in (1).

To estimate the risk of identification disclosure, we perform simulations in which E sequential values are sampled from each original time series x_j, and we measure the average proportion of series which are identified. The sampled values are denoted $C = [\boldsymbol{c}_1, \ldots, \boldsymbol{c}_J]^T$. Each of the vectors $\boldsymbol{c}_i$ corresponds to one of the J original time series and we compute r_j conditional on the sampled c_i from series j. We repeat this simulation S times to obtain $\boldsymbol{C} = \{C_1, \ldots, C_S\}$, and compute the average proportion of correctly identified time series across all external data samples and original time series,

Commented [B5]: Membership inference paper

$$\bar{P} = \frac{1}{J * S} \sum_{s=1}^{S} \sum_{i=1}^{J} \left[\widehat{M_i^s} = j^*\right]$$

where [.] are Iverson brackets.

These simulations assume that the third party in possession of C predicts the match for each vector $\boldsymbol{c}_i$ independently of the predicted matches for other vectors. The risk estimate from a given simulation is equivalent to the identification risk when J independent third parties are each in possession of one of the vectors $\boldsymbol{c}_i$ and each attempts identification risk as described above. Overall, multiple vectors may be matched to the same protected time series.

8.1. Privacy Assessment – Attribute Disclosure

To perform attribute disclosure, we assume the third party predicts additional original values for each protected time series based on the known values C and the predicted match $\widehat{M_i}, i = 1, \ldots, J$ for each vector $\boldsymbol{c}_i$, as simulated above. For our purposes, we assume the third party is interested in the value $A_{j,t'+E+1}$ from each time series that immediately follows the known values $c_i = (A_{i,t'}, \ldots, A_{i,t'+E})$. The third party regresses the $J \times E$ known values in C on the corresponding protected values from each matched series,

$$A_{i,t'} \sim \beta_0 + \beta_1 P_{\widehat{M_i},t'} + \epsilon \qquad (1)$$

and predicts the unknown value of each time series in time period $t' + E + 1$ based on the protected value from that period,

$$A_{\widehat{i,t'+E+1}} = \beta_0 + \beta_1 P_{\widehat{M_i},t'+E+1}\,. \qquad (2)$$

To estimate the risk of attribute disclosure in a protected dataset, i.e., the risk of the third party correctly predicting the value of each time series in time period $t' + E + 1$, we perform the regression and prediction steps (1) and (2) for each $C_s \in \boldsymbol{C}$, and measure the mean absolute error between the predicted and actual original values.

$$\bar{A} = \frac{1}{S * J} \sum_{s=1}^{S} \sum_{i=1}^{J} \left| \widehat{A^{s}_{i,t+E+1}} - A_{i,t+E+1} \right|$$