Can We Protect Time Series Data While Maintaining Accurate Forecasts?

Abstract

In the context of increasing data breaches and privacy concerns, we focus on the often-neglected goal of maintaining the usefulness of protected data by exploring the intersection of data privacy and time series forecasting. Using both simulated and real-world time series data sets, we test various privacy methods, including a proposed swapping-based method (k-nTS+) designed to maintain time series features, a differentially private method, and an approach based on sharing model weights trained on unprotected data. Based on forecasts from seven different forecasting models, none of the privacy methods based on swapping or random noise addition can consistently maintain forecast accuracy at an acceptable level of privacy. On the other hand, sharing model weights trained on unprotected data enables accurate forecasting, but accurate forecasts themselves can uncover the identities of the unprotected time series. We show that transforming time series into rates increases the similarity of time series features, values, and forecasts, and enables k-nTS+ to produce protected rate time series with a reduction in forecast accuracy of just 3.6% on average. Overall, this paper shows that except under certain conditions, creating protected time series with acceptable privacy is incompatible with obtaining accurate forecasts.

Keywords: data privacy; time series features; machine learning; forecast accuracy; identification disclosure risk

1. Introduction

Despite the past few years of worldwide data protection legislation, the number of reported data breaches has more than doubled. Fundamental drivers of data breaches include the increased availability of data, parties in data sharing collaborations, and threats from adversaries. Of these threats, it has been estimated that 74% of all data breaches involved human error or deliberate misuse, and 19% of all data breaches were internal involving full-time employees, independent contractors, interns, or other staff[1]. The overarching message is that although privacy can be assured legally and contractually, the data still gets out.

To support good privacy practices, data owners may protect time series data by altering the unprotected data values. This approach assumes the data eventually gets out and protects against the worst-case scenario of a hacker or internal employee breaching the data. Typically, data protection has two goals. The first goal is to limit the ability of an adversary to identify data subjects (identification disclosure, *i.e.*, revealing the true identity of a time series). The second goal is to limit the ability of an adversary to learn sensitive information about data subjects (attribute disclosure, *e.g.*, learning sensitive values after an identification is made). In tandem,

[1]See the summary of the 2023 Verizon Data Breach Investigations Report.

these goals limit an adversary's increased knowledge about specific data subjects and have been a primary focus of data privacy researchers (Reiter, 2005; Dwork & Naor, 2010; Hu, 2019).

However, an often-overlooked goal is maintaining the usefulness of protected data, which has been treated as an afterthought by many researchers (Blanco-Justicia et al., 2022). For publishing standards in academic journals, the importance of the theoretical underpinnings of differential privacy dominates the importance of practical problems with real applications (Drechsler, 2023). The forecasting literature also found that the trade-off between forecast accuracy of vector autoregressive (VAR) models and differential privacy levels was poor. Gonçalves et al. (2021) showed that even protected data with a weak differential privacy level of $\epsilon = 20$ (it is recommended that ϵ lie in the range [0.01, ln 3] (Dwork, 2011; Blanco-Justicia et al., 2022) reduced forecast accuracy to unusable levels for some data owners. Thus, the focus of this paper is to more broadly assess whether forecast accuracy can be maintained under privacy protection.

To achieve good forecast accuracy and comply with privacy practices, Goncalves et al. (2020) propose a data market where sensitive data is shared only with a market operator, who utilizes the data to sell forecasts to interested buyers. Assuming a trustworthy market operator, this approach provides privacy since sellers only have access to their own time series and buyers only have access to forecasts from the market operator. In practice, however, data owners may be hesitant to share their data over privacy concerns with the market operator. Only sharing forecasts deprives forecasters of useful time series data and has other limitations including the assumption of the forecasting model (*e.g.*, an exponential smoothing or VAR model) and the inability of a forecaster to calculate time series features (*e.g.*, the spectral entropy or the strength of trend). Other researchers share more information such as forecasting model parameters (Goncalves et al., 2021), which has the advantage of sharing non-personal data which obviates the need to implement data security measures or anonymize personal data according to the General Data Protection Regulation (GDPR)[2], but this assumes the data generating process. Our goal is to test whether data owners can bypass these limitations using privacy solutions that alter the unprotected time series data directly to produce protected time series that can be shared with forecasters. To assess this goal, we propose a new privacy method k-nearest time series+ (k-nTS+) swapping which generates protected data that preserves time series features that are important for forecast accuracy. This usefulness-based method gives us the best chance of maintaining the forecast accuracy of protected time series.

Consider two time series in Figure 1, one with the best of features (series A) and one with the worst of features (series B). Series A has a strong positive trend, low spectral entropy indicating good forecastability, and small shifts in variance. Series B has a negligible trend, high spectral entropy indicating poor forecastability, and large shifts in variance. Any forecaster interested in good forecast accuracy would prefer series A over series B. Furthermore, the only features of series B of interest to a forecaster may be the mean and variance.

[2]See this map for legislation examples in the United States. Also see Articles 6, 45, and 46 of the GDPR.

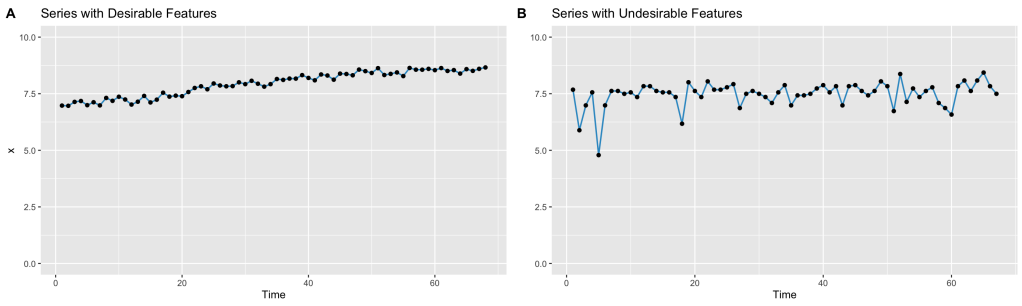

Figure 1: A Tale of Two Time Series: Series A (desirable) and Series B (undesirable).

Now, suppose both time series include personal data requiring protection such as monthly household energy consumption, retail transactions, or movements from smart devices (Boone et al., 2019). How can a data owner alter the time series values in series A so that an adversary cannot easily re-identify series A and a forecaster can still make good forecasts? The method we propose in this paper involves swapping the past values of series A with the values similar time series on the feature space. One idea may be to swap the values of series A randomly with time series that have a similar spectral entropy and strength of trend. However, this method may prove ineffective for maintaining forecast accuracy because similarity based on a low spectral entropy and high strength of trend could select time series with strong seasonality and negative trends. Hence, we propose a machine learning based feature selection method to be included in *k*-nTS+ which initially selects features using RReliefF (Robnik-Šikonja & Kononenko, 2003) and consolidates the remaining correlated features with a random forest-based recursive feature elimination (RFE) algorithm (Gregorutti et al., 2017).

We test our proposed *k*-nTS+ privacy method and other privacy methods based on random noise addition (including differential privacy) on both simulated time series with controlled features and the M3 and M4 forecasting competition data sets, which have features that are representative of real world data (Spiliotis et al., 2020). We also assess whether model parameters, trained on the unprotected data, can be shared in place of protected data. We measure both the identification and forecast disclosure risks of the protected time series, as well as the forecast accuracy of a selection of univariate and multivariate forecasting models applied to the protected data.

Our contributions to the literature are three-fold. First and most importantly, we show that achieving accurate forecasts based on protected data is not always possible. Furthermore, sharing model parameters trained on the unprotected data can enable accurate forecasts that compromise privacy. We perform a thorough evaluation of the performance of seven forecasting models (ranging from exponential smoothing to a recurrent neural network (RNN)) applied to time series that are protected using four privacy methods. Using time series data sets with features that are representative of real world data, we show that time series (on their original scale) can rarely be protected with good forecast accuracy, and that data owners and forecasters will need to adapt to using scaled rate data if protected time series are desired, regardless of the forecasting model used. At an acceptable level of privacy, forecast accuracy cannot consistently be maintained unless time series are transformed into rates which normalizes time series values and increases

cross-series feature similarity. Importantly, this normalization ensures that forecasts themselves are highly similar and cannot be used to identify the protected series. This enables k-nTS+ to produce protected rate time series with a reduction in forecast accuracy of just 3.6% on average, and a reduction in accuracy of 19% when the rate forecasts are transformed back to the original time series scale.

Second, under certain conditions such as select data sets and/or normalized time series values, we enable data owners to share a protected version of a time series data set with good forecast accuracy and reasonable privacy (European Parliament and Council of European Union, 2016). We measure privacy based on two disclosure risks: the probability of reidentifying protected time series using past protected values (Nin & Torra, 2009) and forecasts. While low probabilities of these risks do not guarantee the theoretical standard of differential privacy, our measures are easily interpreted and enable data owners to perform a reasonable privacy assessment as per the GDPR (Bale et al., 2023). Furthermore, we find that data that is differentially private in name (with privacy parameter $\epsilon \geq 10$) provides weak protection against reidentification. In comparison, our proposed k-nTS+ method consistently outperforms benchmark methods by a significant margin on the trade-off between privacy and forecast accuracy, and enables highly accurate forecasts using the protected rate data. The proposed method is flexible since it gives the user the option to customize the protected data for specific forecast horizons, accuracy metrics, or forecasting models. To the best of our knowledge, this paper is the first to produce protected time series data with both acceptable privacy and usable forecast accuracy.

Third, we use a machine-learning based feature selection process that incorporates the usefulness of data (forecast accuracy) into the protection process. The results show that features which are statistically significant predictors of forecast accuracy (Makridakis et al., 2018; Spiliotis et al., 2020) are not necessarily the most useful for swapping time series values. Specifically, we find that *Max Variance Shift*, *Variance*, *Max Level Shift*, *Spike*, *Mean*, and *Kurtosis* are the most important features to consider for swapping values from similar time series. We analyze how these features change vis-à-vis accuracy and show how various privacy methods differentially affect time series features. Overall, using this feature selection process improves the performance of our proposed swapping method anywhere from 4%-30% depending on the data set relative to swapping without the machine learning process. This advantage follows a theme in recent research (Schneider et al., 2018) which maintained marketing metrics within 10-15% of the original by including a marketing loss function in their privacy method. Furthermore, the feature selection method is not limited to data privacy applications, and can be used to select an efficient set of features for predicting any categorical or continuous single valued metric.

The rest of the paper is as follows. In Section 2, we review the relevant literature. Section 3 describes the k-nTS swapping method and proposes the k-nTS+ privacy method. Section 4 assesses the performance of k-nTS+ compared to baseline privacy methods on sets of simulated time series with controlled features and describes the rate transformation used to normalize the time series values and improve the performance of the privacy methods. We build on the results of Section 4 by testing the privacy methods on the M3 and M4 forecasting competition data sets. Section 5 describes the results from the M3 data set while the results from the M4 data set are contained in the Appendix. Section 6 concludes.

2. Literature Review

2.1. Time Series Features

Time series features are commonly used for classification (Fulcher & Jones, 2014), clustering (Bandara et al., 2020), forecast accuracy prediction (Makridakis et al., 2018; Spiliotis et al., 2020), and model selection and forecast combination (Montero-Manso et al., 2020; Qi et al., 2023; Talagala et al., 2022; Li et al., 2023a,b; Kang et al., 2022). Table 1 illustrates a subset of features used in the literature along with values of these features from series A and series B in Figure 1.

Feature	Description	Value Range	Series A	Series B
Spectral Entropy	Signal to noise ratio of a time series.	$[0, 1]$	0.07	1.00
Hurst	Long range dependence (self-similarity) of a time series.	$[0, 1]$	0.99	0.50
Skewness	Symmetry of the distribution of time series values.	$(-\infty, \infty)$	-0.41	-0.57
Kurtosis	Weight of the tails of the distribution of time series values.	$(-\infty, \infty)$	-1.24	1.16
Error ACF	First autocorrelation coefficient of the error component of a decomposed series.	$[-1, 1]$	-0.09	-0.19
Trend	Strength of the trend.	$[0, 1]$	0.97	0.12
Seasonality	Strength of the seasonality.	$[0, 1]$	0.16	0.23
Mean	Mean of a time series.	$[0, \infty)$	7.96	7.01
Variance	Variance of a time series.	$[0, \infty)$	0.29	0.65

Table 1: Time series features descriptions, ranges, and values from A Tale of Two Time Series: Series A (desirable) and Series B (undesirable).

In examining which features improve forecast accuracy, Bandara et al. (2020) clustered similar time series based on eighteen interpretable features, including *Mean*, *Variance*, and *Seasonality* to improve the accuracy of RNNs between 2 and 11%. Furthermore, the initial results from the M4 competition suggested that the randomness (measured using *Spectral Entropy*) and linearity (measured using *Error ACF*) of time series were the most important determinants of forecast accuracy and that seasonal time series (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 mean absolute scaled error (MASE) values of the error, trend, and seasonality (ETS), autoregressive integrated moving average (ARIMA), Theta, and Naïve 2 (random walk applied to seasonally adjusted data) models from the M4 competition. Specifically, they found that increasing the value of *Frequency*, *Kurtosis*, *Error ACF*, and *Seasonality* of time series improved forecast accuracy, but increasing *Skewness*, *Hurst*, and *Spectral Entropy* degraded forecast accuracy.

The predictive power of time series features has been leveraged for other tasks such as forecasting model selection. Qi et al. (2023) found that forecasts using the strength of trend and seasonality for exponential smoothing model selection had lower errors across multiple forecast accuracy metrics than information-based selection methods for the majority of forecast horizons.

Petropoulos & Siemsen (2023) created a representativeness metric that selects models with trend and seasonality components when the respective signals of these components are strong. For most data frequencies, their approach lowered MASE on the M, M3, and M4 competition data and selected the best forecasting model approximately 3% more often than the other selection methods.

Time series features can also improve forecast combination accuracy. Montero-Manso et al. (2020) used forty-two time series features on historical data with FFORMA (Feature-based FORecast Model Averaging) to tune a weighted combination of forecasts. They applied their method to the M4 competition data and found that their point estimates and prediction intervals outperformed all but one method in the M4 competition. Talagala et al. (2022) applied a meta-learning algorithm based on Bayesian multivariate surface regression to 37 features, including *Spectral Entropy* and *Hurst*, to predict the model combination that would yield the minimum forecast error for the M4 competition data. This approach achieved forecast accuracy on par with the top M4 competition methods with less computational cost. Li et al. (2023a) proposed a feature-based Bayesian forecast combination framework with time-varying weights. In experiments using the M3 competition data, this method reduced the MASE by approximately 1.1% relative to the next-best forecast combination method. Interestingly, using the diversity of forecasts as a single feature achieves similar or better forecast accuracy compared to using dozens of features from historical data (Kang et al., 2022; Li et al., 2023b).

2.2. Privacy Methods

Time series data are either stored in a single data set (centralized) or spread across multiple data owners and/or data sets (decentralized). In the decentralized scenario (*e.g.*, in the renewable energy sector), multi-party computation and federated learning enable privacy-preserving collaborative forecasting to ensure accurate forecasts while protecting sensitive data (Gonçalves et al., 2021; Goncalves et al., 2021; Sommer et al., 2021). For example, data owners can sell time series data to a market operator who then sells forecasts of the time series data to multiple buyers. This approach has the advantage of creating a market of economic incentives for data sharing while limiting data transfer and protecting privacy. However, there are still privacy risks including potential data breaches with the transfer and storage of the time series data to the market operator. Goncalves et al. (2020) modeled a data market where data owners are compensated for sharing their time series data and purchase only forecasts based on the data from other parties. However, the original time series were still shared with a central party which discourages data owners from sharing time series due to the possibility of a data breach. Other privacy-preserving solutions for collaborative forecasting include secure multi-party computation, decomposition-based methods, and data transformation techniques (Gonçalves et al., 2021).

In a centralized scenario, the data owner anonymizes the time series data by directly altering the values within the data set to increase the privacy level of the protected data set. Gonçalves et al. (2021) showed that differential privacy reduces the forecast accuracy of VAR models 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, 2013). Luo et al. (2018) simulated data integrity attacks and found that multiplicative noise reduces forecast accuracy by over 21% when only half the data points are altered. Their results likely understate the reduction in forecast accuracy from privacy methods because only half the data points were altered.

Other privacy methods include generalization where the structure of the original data set is changed. Data records can be aggregated or combined to make every record (or time series)

identical to at least $k - 1$ other records (or time series). For example, daily time series data can be aggregated to weekly time series data (temporal aggregation), or each time series can be averaged with its most similar time series ($k = 2$ anonymity). Using $k = 2$ anonymity (weak privacy), Nin & Torra (2009) evaluated the change in forecast accuracy for simple exponential smoothing, double exponential smoothing, linear regression, multiple linear regression, and polynomial regression. They found an overall reduction in forecast accuracy but did not provide the accuracy of each model individually. Also, top- and bottom-coding can be used to replace the tails of distributions with a threshold value, such as $150,000 for income or 10 kilowatts-hours for household energy usage. Top- and bottom-coding limit attribute disclosure risk (*i.e.*, preventing knowledge of specific values within a time series), but may not be effective at limiting identification disclosure risk (*i.e.*, preventing the identification of an entire time series). Top- and bottom-coding could have an effect similar to adjusting for outliers which improves forecast accuracy when the outliers are close to the forecast origin (Chen & Liu, 1993). Overall, a key question emerges from the literature: *Is it possible to share protected time series data with good forecast accuracy?*

Sharing protected time series data with good forecast accuracy would enable organizations to use privacy-enhancing technologies to improve consumer experiences and perceptions. For example, automating the data protection process can limit human intrusions on personal data (Goldfarb & Tucker, 2013). Prioritizing privacy by implementing privacy methods could create a competitive advantage through increased consumer loyalty, trust, and positive performance (Martin & Murphy, 2017). These effects could mitigate damaging outcomes, such as poor stock returns and consumers falsifying information, that arise when consumers feel that their data is vulnerable. Furthermore, replacing sensitive data with protected data could dampen the negative effects of a data breach if the data were to be compromised (Martin et al., 2017). There are also potential benefits from complying with privacy law by using protected time series, such as avoiding the need for consumer consent to re-use data, removing data retention limits, and enabling cross-border data transfers (European Parliament and Council of European Union, 2016; Arbuckle & El Emam, 2020). Using protected data could also help organizations avoid fines such as the $1.3 billion fined against Meta for transferring user data from the EU to the US[3].

2.3. Adjusted Forecasts

Privacy methods adjust forecasts by altering the original time series data. Similar to judgmental adjustments, this presents the forecaster with multiple forecasts to choose from. We reference the long history on judgmental forecasting (Petropoulos et al. (2022) see sections 2.11.2 and 3.7.3) investigating how the direction and magnitude of adjustments, and the volatility of forecasts affect forecast accuracy.

There are two critical differences between privacy adjustments and judgmental adjustments. First, judgmental adjustments alter a forecast after it is output from a forecasting model. The underlying time series and their features are not changed. For the direction of the adjustment, Davydenko & Fildes (2013) found that both positive and negative adjustments can improve accuracy, but positive adjustments tend to give only a marginal improvement. Khosrowabadi et al. (2022) similarly found that beneficial positive adjustments tended to be small, and beneficial negative adjustments tended to be large. Fildes et al. (2009) showed that negative adjustments reduce forecast bias, whereas positive adjustments maintain bias or exacerbate it. The magnitude

[3]E.U. slaps Meta with record $1.3 billion fine for data privacy violations.

of judgmental adjustments is also positively associated with the size of accuracy improvements when adjustments are based on reliable information. For volatility, accuracy improvements are greater for time series that have forecast errors with low volatility, presumably because adjusters struggle to assess the effect of future events accurately when a time series is more volatile (Fildes et al., 2009).

Second, the motivation for judgmental adjustments is different. Motivations include gaining control of the forecasting process, incorporating practitioner expectations, and compensating for judgmental biases (Petropoulos et al. (2022) sec. 3.7.3). The goal is to incorporate the intuition and experience of the adjuster, knowledge of special events, or insider or confidential information to improve forecast accuracy (Fildes et al., 2019). Despite varying motivations, judgmental adjustments have been shown to improve forecast accuracy by 5-10% on average (Davydenko & Fildes, 2013; Khosrowabadi et al., 2022). For privacy adjustments, the goal is to improve privacy by blurring the data. The assumption is that forecast accuracy will not improve – instead, utility (forecast accuracy) will tradeoff with privacy (Duncan & Stokes, 2004).

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

We solve the data protection problem for the data owner using a matrix-based k-nTS (k-nearest time series) swapping method, where the data owner releases a set of protected time series $\mathbb{X}' = \{x'_1, \ldots, x'_J\}$ where $x'_j = \left(P_{j,1}, \ldots, P_{j,t}\right)^T$ is based on $\mathbb{X} = \{x_1, \ldots, x_J\}$, the original values of all series through time t. To create a protected series x'_j, the k-nTS swapping method finds the k most similar time series to x_j where similarity is based on time series features. For each period t, it randomly chooses one of the k similar series to x_j and replaces $A_{j,t}$ with the original value at time t from the randomly chosen series.

Depending on the quantity of available data, k-nTS swapping can use rolling windows of data that adjust for dynamic changes in time series features. For example, if we choose a rolling window of size m, then $x_j = \left(A_{j,t-m+1}, A_{j,t-m+2}, \ldots, A_{j,t-1}, A_{j,t}\right)^T$ where $x_j \in \mathbb{R}^m$. Protection in subsequent periods from $t+1$ to T rolls x_j forward by one time period. We label the time series features 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 m values in x_j. For simplicity, we omit the t subscript for the feature vectors and write f_j.

For each time series $x_j \in \mathbb{R}^m$, the data owner computes the feature vector $f_j \in \mathbb{R}^N$. This vector can contain any single-valued feature calculated from the values in x_j, such as the strength of the trend and seasonality, the spectral entropy, or the mean of the current window. Let $\mathbb{C} = \{f_1, \ldots, f_J\}$ be the set of N-vectors containing the features from each of the J time series windows. For each f_j, the data owner computes a set of squared distances of the elements of $\mathbb{C}$. We define $dist\left(f_j, f_i\right) = d_{j,i}$ as the distance between f_j and f_i, *i.e.*, the feature vectors corresponding to two distinct time series from $\mathbb{X}$. Without loss of generality, we use the Euclidean norm, or ℓ_2-norm, as a distance metric[4]. Since our case is multivariate and partially ordered, we can get a totally ordered set based on the Euclidean distance.

We define $x_j^{(k)}$ as the kth nearest neighbor of x_j, with the corresponding feature vector $f_j^{(k)}$. Then, for a time series x_j, we have $\{d_{j,(1)}, d_{j,(2)}, \ldots, d_{j,(J-1)}\}$ 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 ith order

[4]All norms on $\mathbb{R}^n$ are equivalent to the Euclidean norm.

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-nearest time series can be represented as the set $K_j = \{x_j^{(1)}, \ldots, x_j^{(k)}\}$ based on an ordered set $\{d_{j,(1)}, d_{j,(2)}, \ldots, d_{j,(k)}\}$.

For more efficient computation, 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 $d_{i,j}$, that is the (i, j)th entry of D (also note that $rank\,(D) \leq N + 2$). Suppose we have an original data matrix $X = [x_1, x_2, \ldots, x_J]$, where $x_j \in \mathbb{R}^m$ (*i.e.*, $X \in \mathbb{R}^{m \times J}$). We calculate the desired features based on each x_j and construct a feature matrix C (where $C \in \mathbb{R}^{N \times J}$) as follows:

$$C = [f_1, f_2, \ldots, f_J] = \begin{pmatrix} f_{1,1} & f_{1,2} & \cdots & f_{1,N} \\ f_{2,1} & f_{2,2} & \cdots & f_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ f_{J,1} & f_{J,2} & \cdots & f_{J,N} \end{pmatrix}^T, \tag{1}$$

where each of the rows of C (*i.e.*, the J values of the nth feature) is normalized to have zero mean and unit variance. 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 as the following:

$$D = \mathbf{1} diag\left(C^T C\right)^T - 2C^T C + diag\left(C^T C\right)\mathbf{1}^T, \tag{2}$$

where $\mathbf{1}$ denotes a column vector of J ones, and $diag(.)$ denotes the diagonal of a matrix.[5]. It is easy to see that the column vector $diag\left(C^T C\right) = \left(\|f_1\|^2, \ldots, \|f_J\|^2\right)^T$. We can also compute a weighted distance matrix as follows,

$$D = \mathbf{1} diag\left(C^T WC\right)^T - 2C^T WC + diag\left(C^T WC\right)\mathbf{1}^T, \tag{3}$$

where W is a $N \times N$ diagonal matrix, $diag(W) = (w_1, \ldots, w_N)^T$, and w_n is the weight for the nth feature. This enables a data owner to prioritize certain features in the swapping process by choosing weights that indicate the relative importance of each feature. For example, suppose the features are ordered $n = 1, \ldots, N$ in ascending order of importance. Then the weights in $diag(W)$ can be chosen such that $w_1 < w_2 < ... < w_N$. If the data owner does not provide feature weights then each weight $w_n = 1$ and equation (3) becomes equivalent to equation (2).

Let d_j denote the jth column of D. Then we can write the $J \times J$ distance matrix $D = [d_1, \ldots, d_J]$, 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 and take the k smallest components so that we have

$$K_j = \{x_j^{(1)}, \ldots, x_j^{(k)}\}. \tag{4}$$

That is, the data owner 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 N features. Let the ith most similar time series to x_j be $x_j^{(i)}$. Swapping the last component of x_j with the last component of one of its k-nearest time series $x_j^{(i)}, i = 1, \ldots, k$, is:

$$P_{j,t} = A_{j,t}^{(i)} \text{ with probability } 1/k \text{ for } i = 1, \ldots, k. \tag{5}$$

[5]Note 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.

By Algorithm 1, we can obtain X', a matrix of protected time series data through time t for all J time series. All of the first m values of each time series are swapped based on the k-nearest time series to the first rolling window. For each successive time period, the window is rolled forward, the k-nearest neighbors are re-calculated, and swapping is performed on a rolling basis. The output of the k-nTS swapping method can be written as the following protected data matrix,

$$X' = [x'_1, x'_2, \ldots, x'_J] = \begin{pmatrix} P_{11} & P_{12} & \ldots & P_{1T} \\ P_{21} & P_{22} & \ldots & P_{2T} \\ \vdots & \vdots & \ddots & \vdots \\ P_{J1} & P_{J2} & \ldots & P_{JT} \end{pmatrix}^T . \tag{6}$$

We also present the time complexity of the k-nTS swapping algorithm (Algorithm 1). The matrix multiplication $C^T WC$ takes $O(JmN)$ and computation of the distance matrix D takes $O(J^2mN)$. Then, for k-nTS swapping, we use each column d_j of the distance matrix D, a $J \times J$ matrix. For each $j = 1, \ldots, J$, k-nTS swapping takes $O(J + klogJ)$ since k is typically much smaller than J. Therefore, the final time complexity is $O(J^2mN) + O(J + k\log J)$.

Due to the protected values $P_{j,t}$ being chosen at random from the set of neighbors K_j, it is possible that a swapped value could be an outlier compared to the rest of the protected data points, which could negatively affect forecast accuracy. We mitigate this possiblity by replacing any outlying swapped data points after isolating the seasonality and trend. This outlier replacement strategy is conveniently implemented in the R *forecast* package (Hyndman & Khandakar, 2008) and is performed as follows. Suppose a given protected time series x'_j is decomposed into seasonal, trend, and remainder components as follows

$$x'_j = s_j + g_j + r_j, \tag{7}$$

and the 25th and 75th percentiles of the values of the remainder component r_j are denoted $p_{0.25}$ and $p_{0.75}$, respectively. Points are identified as outliers if they are less than $p_{0.25} - 3 \times IQR$ or greater than $p_{0.75} + 3 \times IQR$, where $IQR = p_{0.75} - p_{0.25}$, and are replaced using linear interpolation based on the neighboring non-outlying points.

Algorithm 1 The k-nTS Swapping Method

Initialization:

$X = [x_1, \ldots x_J]$: the $m \times J$ matrix of original time series.

$C = [f_1, \ldots, f_J]$: the $N \times J$ matrix of time series features.

W: the $N \times N$ diagonal matrix of feature weights.

k: the number of nearest neighbor time series to consider for swapping.

Compute the feature distance matrix $D = \mathbf{1} diag\left(C^T WC\right)^T - 2C^T WC + diag\left(C^T WC\right)\mathbf{1}^T$.

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

  Find the set K_j for x_j by sorting the jth column of D from smallest to largest and finding the kth smallest component.

  Replace the last component of x_j with the last component of $x_j^{(i)}$ for a randomly chosen $i \in \{1, \ldots, k\}$.

end for

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

The k-nTS+ swapping method adds a feature selection process to k-nTS swapping which selects features that are good predictors of forecast accuracy. The data owner can provide a large selection of time series features to the method. However, using all of the features for the basis of swapping would significantly increase the dimensionality and reduce the efficiency of the swapping process. To address this problem, a random forest-based recursive feature elimination (RFE) algorithm is applied to features initially selected by RReliefF (Robnik-Šikonja & Kononenko, 2003). Prior work has shown that random forest-based RFE is efficient when applied to sets of highly correlated features (Gregorutti et al., 2017), ensuring that the method selects a small set of features that predict forecast accuracy well.

The feature selection methodology is outlined in Algorithm 2 and proceeds as follows. Let f_j^κ denote one of the $\mathcal{K}$ nearest neighbor feature vectors to f_j, where $\mathcal{K}$ is the number of nearest neighbors considered by RReliefF, and let ϵ_j^κ and ϵ_j denote the forecast errors for the corresponding time series. Let π_ϵ and π_n denote the events that series x_j and x_j^κ have different forecast errors and different values for the nth feature, respectively, conditional on being nearest neighbors. The RReliefF weight for the nth feature approximates the difference in conditional probabilities,

$$\omega_n = p\left(\pi_n|\pi_\epsilon\right) - p\left(\pi_n\middle|\pi_\epsilon^c\right). \tag{8}$$

The RReliefF weights approximate the difference between the probability that the nth feature discriminates between series with different forecast errors, and the probability that the nth feature discriminates between series with the same forecast error. In the first stage of the feature selection method we eliminate all features that are poor predictors of differences in forecast error with $\omega_n \leq 0$.

The second stage of the feature selection method utilizes random forest-based RFE. First, we train a random forest model to predict the forecast errors from a given forecasting model using a set of time series features and compute the mean-absolute error (MAE) on the out-of-bag (OOB) predictions using that set of features. Next, we compute the permutation-based feature importance, *i.e.*, the change in OOB MAE from permuting the values of a given feature. The least important feature is dropped from the model and the model training and feature elimination process is performed recursively until one feature remains. Each feature is then assigned a rank based on the reverse of the elimination order. This process is performed N^{rfe} times. Different permutations of the data may result in different importance measures across RFE iterations. Thus, we compute s^*, the number of features that resulted in the lowest average OOB MAE across all RFE iterations, and select the s^* features with the best (lowest) average rank across all RFE iterations.

Algorithm 2 Two-Stage Feature Selection

Initialization:
  $\mathcal{F} = \{\mathcal{F}_1, \ldots, \mathcal{F}_N\}$: set of N time series feature functions.
  $C^T = [f_1, \ldots, f_J]^T$: the $J \times N$ matrix of time series feature values.
  $\epsilon = (\epsilon_1, \ldots, \epsilon_J)^T$: the $J \times 1$ vector of forecast errors.
  $\mathcal{K}$: the number of nearest neighbor time series to consider for RReliefF.
  N^{rfe}: the number of RFE iterations.

Stage 1: RReliefF Weight Selection
Compute RReliefF weights ω_n, $n = 1, \ldots, N$ from predicting ϵ using C^T.
Compute first stage feature selection $\mathcal{F}' = \{\mathcal{F}_n \in \mathcal{F} : \omega_n > 0\}$, where $|\mathcal{F}'| = S$.
Compute $J \times S$ feature matrix C' with columns corresponding to features in $\mathcal{F}'$.

Stage 2: Recursive Feature Elimination
for $i = 1, \ldots N^{rfe}$ **do**:
  Set $s = S - 1$.
  while $s > 0$ **do**
    Train a random forest to predict ϵ using C'.
    Compute $e_{i,s+1}$, the MAE of the OOB predictions using $s + 1$ features.
    Compute permutation-based importance $I_{n'}$ of each feature $\mathcal{F}_{n'} \in \mathcal{F}'$.
    Drop the least important feature from C' and $\mathcal{F}'$ such that s features remain.
    Assign rank $r_{i,n'} = s + 1$ to the dropped feature for iteration i.
    Set $s = s - 1$.
  end while
  Assign rank $r_{i,n'} = 1$ to most important feature.
  Add all dropped features with $\omega_n > 0$ back to $\mathcal{F}'$ and C'.
end for
Compute $\bar{e}_s = \frac{1}{N^{rfe}} \sum_{i=1}^{N^{rfe}} e_{i,s}$, $s = 1, \ldots, S$, the average MAE when using s features.
Identify the subset size s^* corresponding to $\bar{e}_s^{min}$, the minimum average OOB MAE.
Compute $\bar{r}_{n'} = \sum_{i=1}^{N^{rfe}} r_{i,n'}$, $n' = 1, \ldots, N'$, the average rank of each feature $\mathcal{F}_{n'}$.
Select $\mathcal{F}^*$, the s^* features with the best (lowest) average ranks $\bar{r}_{n'}$.

Figure 3.1 illustrates the full k-nTS+ protection process, which can be used collaboratively between a data owner and the forecaster. The forecaster specifies their preferred forecasting model $\mathcal{M}$. The data owner generates protected versions of the original data set up through time period $T - h$ using baseline privacy methods with varying levels of privacy strength. The data owner computes the forecast errors from using the model $\mathcal{M}$ to forecast time periods $T - h + 1$ through T for each version of the data. These errors are the target variable in the feature selection process, which is performed on a feature matrix C containing the combined feature values from the original and baseline protected data sets. We note that the feature values should be normalized *after* they are combined in C. The selected features F^* are then used as the basis for swapping to create the protected data set X' from through original data set up through time period T. The data owner can perform the swapping process over successive time periods conditional on the selected features to release protected data for times $T + 1$ and beyond, or may choose to perform the feature selection process again for any time period. Algorithm 3.1 specifies the full k-nTS+ swapping method.

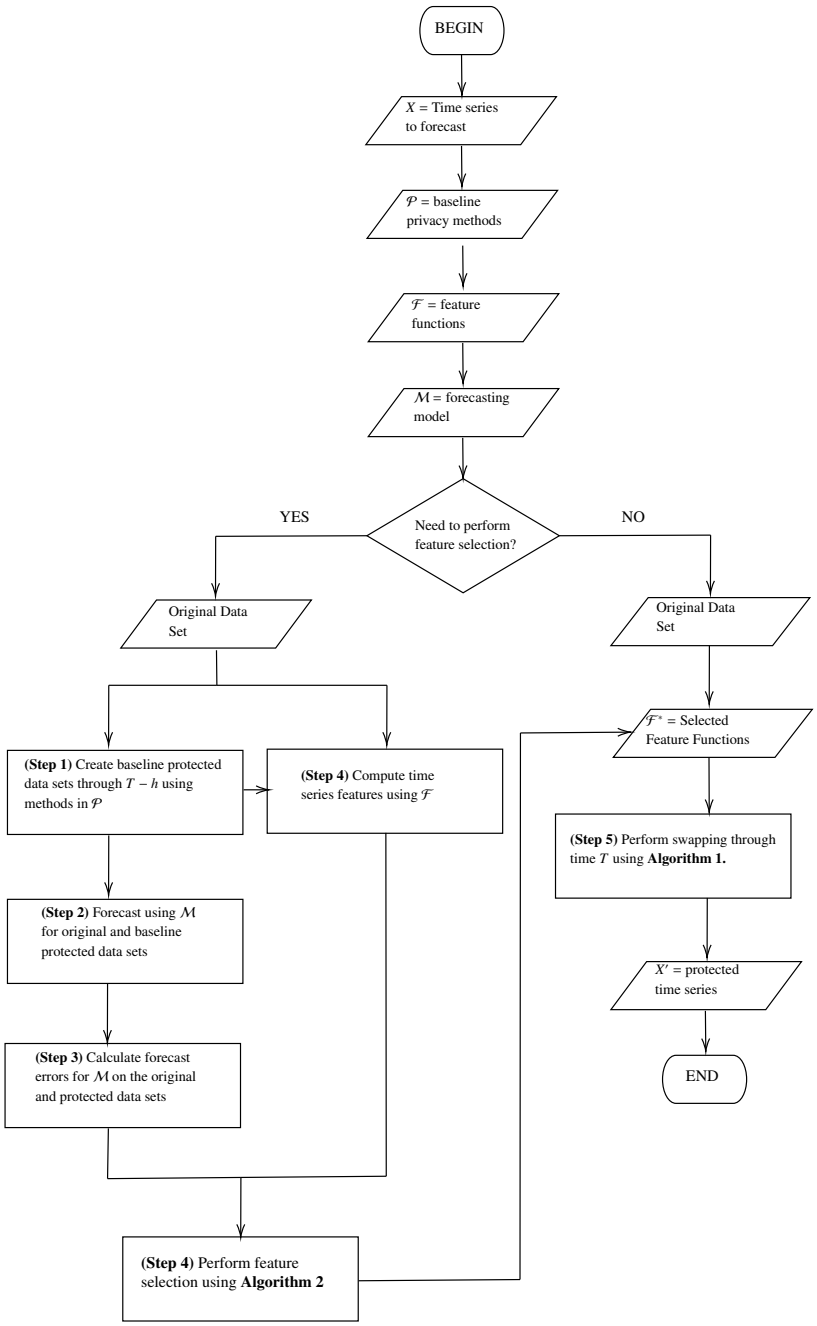

Figure 2: Flowchart of the k-nTS+ data protection process.

Algorithm 3 The k-nTS+ Swapping Method

Initialization:

$X = [x_1, \ldots x_J]$: the $T \times J$ matrix of original time series.

$\mathcal{P} = \{\mathcal{P}_1, \ldots, \mathcal{P}_B\}$: the B baseline privacy methods.

$\mathcal{F} = \{\mathcal{F}_1, \ldots, \mathcal{F}_N\}$: set of N functions for calculating time series features.

$\mathcal{K}$: the number of nearest neighbor time series to consider for RReliefF.

N^{rfe}: the number of recursive feature elimination iterations.

k: the number of nearest neighbor time series to consider for swapping.

$\mathcal{M}$: the desired forecasting model.

h: forecast horizon.

Step 1: Create Baseline Protected Data Sets

Store values from time(s) $T - h$ through T in X as test set $X^{test} = X[T - h : T]$.

Create baseline protected data set $\mathcal{P}_b(X[1 : T - h]) = X^b$ for each $\mathcal{P}_b \in \mathcal{P}$.

Step 2: Generate Baseline Forecasts

Generate forecasts using $\mathcal{M}$ for time(s) $T - h + 1$ through T based on the unprotected data $X[1 : T - h]$ and each protected data set $\mathcal{P}_b, b = 1, \ldots, B$.

Step 3: Measure Forecast Accuracy

Compute average forecast error at the series level $\epsilon = (\epsilon_1, \ldots, \epsilon_J)^T$ for the unprotected data forecasts and $\epsilon^b = \left(\epsilon_1^b, \ldots, \epsilon_J^b\right)^T$, $b = 1, \ldots, B$ for the protected data forecasts.

Step 4: Compute and Select Time Series Features

Compute time series features matrices C from $X[1 : T - h]$ and $C^b, b = 1, \ldots, B$, from $X^b, b = 1, \ldots, B$.

Create $N \times J(B + 1)$ feature matrix $C = \left[C^1, \ldots, C^B, C\right]$ and $J(B + 1) \times 1$ forecast error vector $\mathcal{E} = \left(\epsilon^b, \ldots, \epsilon^B, \epsilon\right)^T$ by concatenating the columns of the feature matrices and the error vectors, respectively, from the baseline protected and original data sets.

Compute $\mathcal{F}^*$ using **Algorithm 2** with inputs C, $\mathcal{E}$, $\mathcal{F}$, $\mathcal{K}$, and N^{rfe}.

Step 5: Create Protected Data Set Using k-nTS Swapping

Compute C^*, the $s^* \times J$ matrix of selected time series feature values.

Compute permutation-based importance I_{n^*} of each feature $\mathcal{F}_{n^*} \in \mathcal{F}^*$.

Compute feature weight matrix where $diag(W^*) = (w_1, \ldots, w_{s^*})^T$ and $w_{n^*} = I_{n^*} / \sum_{p=1}^{s^*} I_p$.

Use inputs X, C^*, W^*, k and selected features $\mathcal{F}^*$ to perform swapping through time T using **Algorithm 1**.

4. Simulation Application: A Tale of Two Time Series

4.1. A Tale of Two (Sets) of Time Series

We simulate two sets of ten time series each with "desirable" and "undesirable" features, respectively. The series are simulated using the GRATIS (GenerRAting TIme Series) R package (Kang et al., 2020) which uses genetic optimization to tune a mixture auto-regressive (MAR) model to generate synthetic time series with user-provided feature values. The MAR model is composed of three Gaussian autoregressive integrated moving average $ARIMA(p, d, 0)$ processes, where $p \in \{0, 1, 2, 3\}$ denotes the number of autoregressive parameters, and $d \in \{0, 1, 2\}$ indicates the degree of first differencing. Both p and d are sampled randomly for each process, and the autoregressive parameters are sampled from the stationary parameter space (Zhao & Hyndman, 2023).

We sample a target spectral entropy value s^* from a uniform distribution over the range $[0.3, 0.4]$. We then sample a target value for the first autocorrelation coefficient ϕ_1^* from a uniform distribution over the range $[0.7, 0.9]$. The target for the second autocorrelation coefficient ϕ_2^* is sampled from a uniform distribution over the range $[\phi_1^* - 0.2, \phi_1^* - 0.1]$. The target feature values for the series with undesirable features are sampled in a similar fashion, where the distributions for s^* and ϕ_1^* are defined over the ranges $[0.6, 0.7]$ and $[0.2, 0.4]$, respectively. Note that we are not requiring the MAR model to simulate an AR(2) process. Rather, we are simulating time series where the first two coefficients of the estimated autocorrelation function (ACF) are approximately equal to the target values. Our goal is to create sets of time series with relatively high signal-to-noise ratios and strong autocorrelation (desirable features) compared to relatively low signal-to-noise ratios and weak autocorrelation (undesirable features). Each of the simulated time series is re-scaled using a mean and standard deviation sampled from uniform distributions over the ranges $[2000, 12000]$ and $[50, 500]$, respectively. If necessary, we shift entire time series by a constant to ensure that the minimum value for all of the time series is at least one. Figure 3 plots the simulated time series.

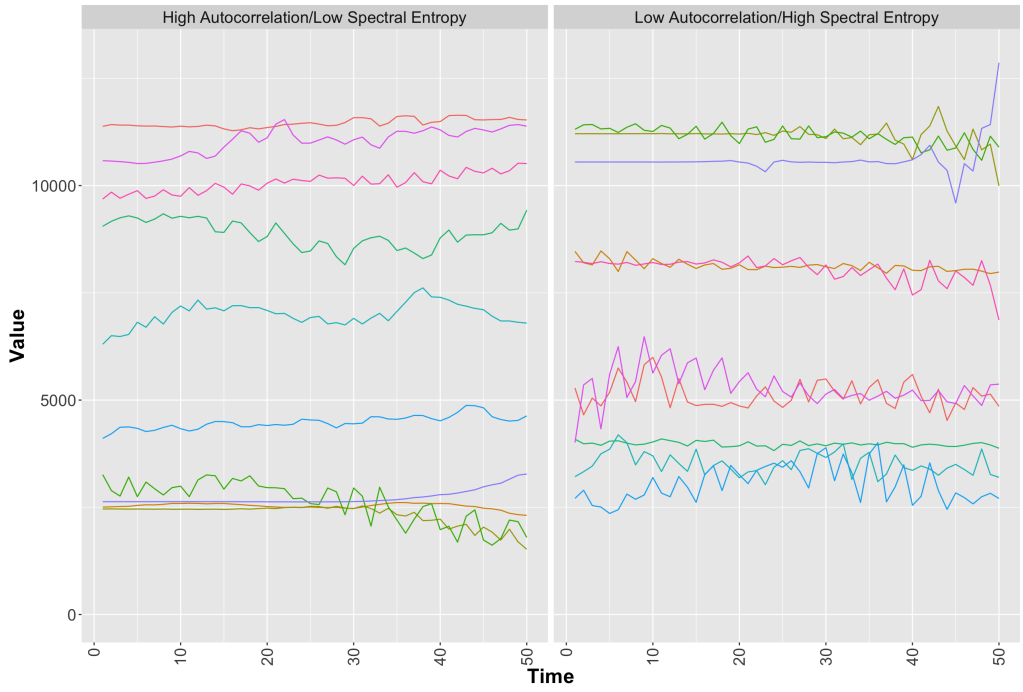

Figure 3: A Tale of Two (Sets) of Time Series. The sets of ten series with desirable and undesirable features are shown in the left and right plots, respectively.

The rest of the example proceeds as follows. Subsection 4.2 describes the time series features used for the k-nTS+ swapping method, subsection 4.3 describes the privacy methods and defines our measures of privacy risk, and subsections 4.4 and 4.4 present the privacy and forecast accuracy results.

4.2. Time Series Features for Forecast Accuracy

While the time series were simulated with targeted values of spectral entropy and the first two autocorrelation coefficients, these features are correlated with many other features including those in Table 1 and will only be selected in the *k*-nTS+ swapping process if changes in these features after applying the baseline privacy method(s) are predictive of forecast accuracy. For *k*-nTS+, we include all features which could help improve accuracy in the swapping process and which are easily computable using the *tsfeatures* package in R (Hyndman et al., 2023). Examples include *Spike*, *Max Variance Shift*, and *Max Level Shift*. We allow our proposed feature selection method to select the efficient set of features which will balance the trade-off between forecast accuracy and data privacy well. We refer the reader to Yang & Hyndman (2023) for a detailed explanation of time series features and provide further mathematical detail on several time series features in the Appendix.

4.3. Privacy Methods

4.3.1. Differential Privacy

We follow the interpretation and implementation of differential privacy from Gonçalves et al. (2021). A mechanism M satisfies ϵ-differential privacy by guaranteeing that, for every output x' of M and every pair of series x_j and x_i which differ on at most one observation,

$$\Pr\Big(M(x_j) = x'\Big) \leq \exp(\epsilon) \Pr\Big(M(x_i) = x'\Big). \tag{9}$$

A differentially private time series can be created using a randomized mechanism $M(x_j) = x_j + \eta$ that adds a vector of random noise values, each of which is drawn from a Laplace distribution with scale parameter $\Delta f_1/\epsilon$, to an original time series x_j. The sensitivity Δf_1 is determined as the maximum absolute difference between two time series x_j and x_i, which differ in at most one observation, where $\Delta f_1 = \max \|x_i - x_j\|_1$. We assess thirty-nine values of ϵ ranging from 20 to 0.01, which ranges from least private to most private.

4.3.2. Additive Noise

Additive noise adds a normally distributed random number with mean zero and standard deviation σ to each value in a time series x. Protected values can be written $P_t = A_t + r$, where $r \sim N\left(0, \sigma^2\right)$ and $\sigma = s\sigma_{x_j}$. The protection parameter s denotes the number of standard deviations of x_j and we assess thirty-nine values of s between 0.08 and 30.00, which ranges from least private to most private.

4.3.3. k-nTS+

The k-nearest series for k-nTS+ swapping are determined using features computed by the data owner from the original data set. To initialize the swapping process, the features and neighbors are calculated for the first window of length n from each original series. Swapping is performed for each of the n values using this same set of neighbors. The window is then rolled forward by one period, the features and neighbors are re-calculated, and a new value is swapped into the last period of the window. This process continues until all the values from time 1 to T in each of the original series have been replaced. For the purposes of this example, we select $n = 25$ and use the features described in Section 4.2.

To perform feature selection for k-nTS+, we first create protected versions of the original data using differential privacy from time period 1 to $T - 1$ for each of the 39 ϵ values described in Section 4.3.1. We generate forecasts for each of these protected data sets for time T and compute the absolute error of each forecast for each series. These errors, along with the time series features, are used as input into the feature selection methodology described in Section 3.1. This feature selection method is performed separately for each forecasting model in order to detect the variation in forecast accuracy due to changes in time series features (and not the forecasting model), but our final protected data sets are created using the features which have the highest average importance across forecasting models.

For each k-nTS+ protected data set, we use the features with the highest average rank across the RFE iterations for all forecasting models. Next, we use these features to swap all values for time periods 1 to T for all series. The data owner shares this protected data with forecasters who forecast time period $T + 1$. We test k-nTS+ for $k = 2, 3, 4, 5, 6, 7, 8, 9$.

4.3.4. Sharing Model Weights

Instead of sharing protected data for forecasting, an alternative is for the data owner to share pre-trained models with the forecaster, as seen in collaborative scenarios (Goncalves et al., 2021). We compare the approaches to creating protected data by altering the data directly to a scenario where the data owner shares VAR model parameters, from a model trained on the unprotected data, with the forecaster. For example, suppose the data owner trains a VAR model with $l = 1$ lags on $J = 5$ differenced unprotected time series over time periods $t = 2, \ldots, T$. Using the VAR model and the first $l \times J$ differenced time series values, the data owner simulates $V = 5000$ potential future paths for each of the J series for the next $l+1$ time periods, and takes the average of the potential values in each period for each time series. These values, along with the model intercepts, lag coefficients, and the covariance matrix of the model residuals, are shared with the forecaster. To generate forecasts, the forecaster continues the simulation process described above through time T, and generates forecasts using the model weights and $l \times J$ most recent values of the simulated series. Overall, this approach enables the forecaster to obtain a data set (of simulated time series) and model weights trained on the unprotected data without sharing confidential data. We evaluate this approach on the simulated time series using two VAR models of lag order one trained on five time series each.

4.3.5. Identification and Forecast Disclosure Risk

We assess the ability of each privacy method to protect against identification disclosure, which occurs when an adversary correctly predicts the identity of a protected time series. While preventing (or limiting the success of) an identification disclosure is not a theoretical privacy guarantee, it is an interpretable and reasonable definition of privacy that can enable data controllers to achieve a targeted level of privacy protection while maintaining useful data[6] (Bale et al., 2023). Each protected data set consists of the protected series along with a pseudo identifier (PID), *i.e.*, $X = [(PID_1, x_1), (PID_2, x_2), \ldots (PID_J, x_J)]^T$. Identification disclosure occurs if an adversary (or forecaster) correctly predicts the identity of one or more of the protected time series based on the protected time series and some outside information the adversary possesses. For example, identification disclosure occurs when an adversary correctly states, "Series X comes from the monthly sales of the Roseville, Minnesota Target store." An acceptable level of identification disclosure risk depends on the context in which the protected data will be released and the preferences of the data controller. One example is the European Medicines Agency (EMA), which has set a maximum acceptable identification probability of 9% (European Medicines Agency, 2017).

We assess an identification disclosure attack where an adversary uses unprotected time series values to identify the protected time series. We perform $S = 20$ simulations of this privacy attack in which we sample ten sequential values from each original time series and treat these as external information available to the adversary. The adversary predicts the identity of each protected series based on which original values are closest to the protected values from the same time periods.

The metric we use is identification disclosure risk for time series (Nin & Torra, 2009), $\bar{P}$, or the average proportion of the J time series which are correctly identified across the S simulated privacy attacks,

$$\bar{P} = \frac{1}{J \times S} \sum_{s=1}^{S} \sum_{i=1}^{J} I\left(\hat{M}_i^s = j^*\right), \tag{10}$$

[6]See the Article 29 Working Party Guidelines 05/2014.

where $\hat{M}_i^s$ is the adversary's prediction of the identity of the ith protected time series. We use $I(*)$ to denote the indicator function which is equal to one when identification disclosure occurs, i.e., when the predicted identity is equal to the true identity j^*. We refer the reader to the Appendix for added mathematical details.

In addition to identification disclosure risk, we measure the proportion of time series which can be correctly identified based on their forecasts, which we term *forecast disclosure risk*. The attack is similar to the one previously described, except instead of using past unprotected values to identify protected series, the adversary compares values from the future to the forecasts from protected data. Let $\hat{X} = [(PID_1, \hat{A}_{1,T+1}), (PID_2, \hat{A}_{2,T+1}), \ldots, (PID_J, \hat{A}_{J,T+1})]$ denote the set of forecasts and pseudo identifiers for the protected time series, where $\hat{A}_{j,T+1}$ is the forecast for the jth series for time $T + 1$. Suppose the adversary observes the actual unprotected values for time $T + 1$ as external information and compares these values to the forecasts. The adversary predicts the identity of the ith protected time series based on which forecast is most similar the jth unprotected value. Forecast disclosure occurs when an identification is made correctly and we measure forecast disclosure using $\bar{P}^f$,

$$\bar{P} = \frac{1}{J} \sum_{i=1}^{J} I\left(\hat{M}_i^f = j^*\right), \tag{11}$$

where $\hat{M}_i^f$ is the adversary's prediction of the identity of the ith time series based on the forecasts. Although identification and forecast disclosure are only two of many possible attacks, successfully reidentifying a large portion of time series (*e.g.*, $> 9\%$) indicates that a time series data set is not sufficiently protected.

The attacks described in this section do not capture all of the ways identification disclosure could occur. For example, the values of features such as the mean and variance could uniquely identify time series, and it may be better for privacy if the values of these features are not *exactly* preserved through the k-nTS+ swapping process. Overall, the metric in (10) provides a reasonable lower bound on the identification disclosure risk associated with protected time series.

4.4. Results

Multiple protected data sets can be generated from the same unprotected data set due to the randomness of the privacy protection from additive noise (AN), differential privacy (DP), and k-nTS+. In practice, a data owner would choose one protected data set to share with forecasters to reduce knowledge of the distribution of time series values. For the purposes of this simulation, we assess the average forecast accuracy and identification disclosure risk of the simulated data by generating twenty-five protected data sets for each of the privacy parameters and averaging the accuracy and privacy results. For each protected data set we perform the privacy simulation discussed in Section 4.3.5 and generate forecasts using simple exponential smoothing (SES) and double exponential smoothing (DES) forecasting models. We also generate VAR-simulated time series and compare the forecast accuracy and privacy under this approach to the k-nTS+ protected data.

Figure 4 shows the average $\bar{P}$ (top row) and the average MAE (bottom row) for each set of time series across the twenty-five simulations for each privacy method and parameter. AN did not produce protected data with acceptably low identification disclosure risk (over 24% of series can be correctly identified on average in both data sets) even for the largest value of s which reduced average forecast accuracy by over 1000% for both data sets. For $\epsilon = 0.01$, differential privacy produced protected data sets with an acceptable average identification disclosure risk of

approximately 9-10% (which is on-par with an adversary randomly guessing the time series' identities), but this comes at a reduction in forecast accuracy of over 2000%. k-nTS+ produces protected data sets with an average $\bar{P}$ of 8.4% and 5.9% for the desirable and undesirable data sets for $k = 4$ and $k = 7$, respectively. The adversary is better off randomly guessing the time series' identities in this case, and the forecast accuracy of the k-nTS+ protected data sets is reduced by approximately 790% and 1200% for the data with desirable and undesirable features, respectively. This is a significant improvement over the forecast accuracy of the differentially private data sets, but the forecasts from these protected data sets would still be unusable. Overall, none of the privacy methods achieved an acceptable trade-off between privacy and forecast accuracy.

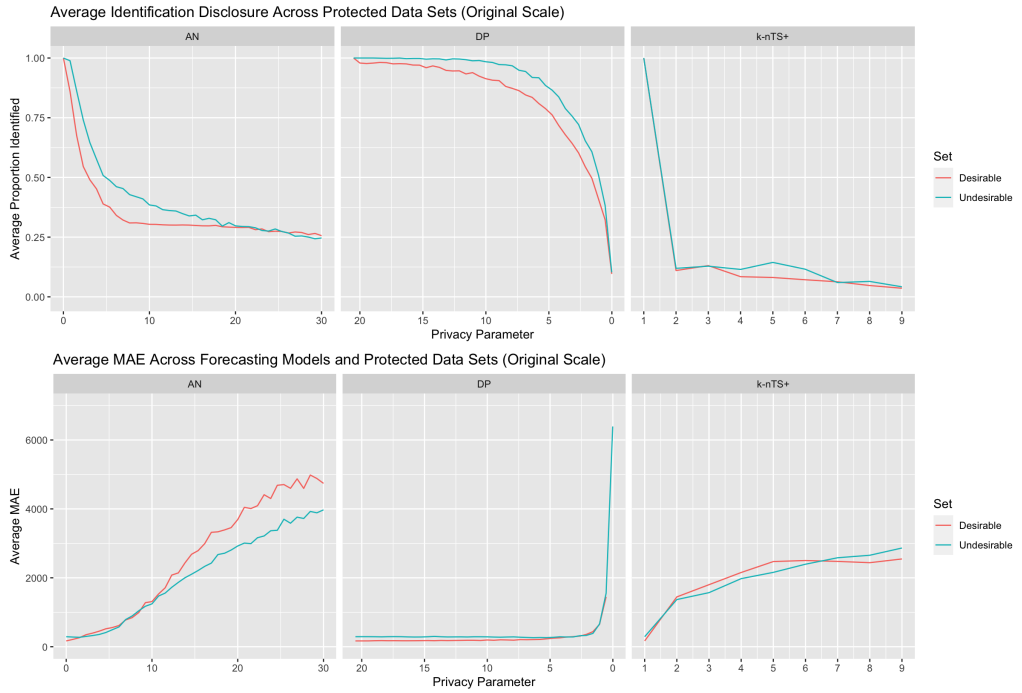

Figure 4: Average identification disclosure risk and MAE for all privacy parameters for additive noise, differential privacy, and k-nTS+ applied to sets of ten time series with desirable or undesirable features.

Table 2 compares the forecast accuracy of VAR models trained on k-nTS+ protected data sets to the approach based on the VAR-simulated series. Notably, the forecast accuracy using the undesirable VAR-simulated series did not degrade compared to the unprotected data. The desirable VAR-simulated series began diverging near the forecast origin which resulted in poor forecast accuracy albeit improved over the accuracy under k-nTS+. These accuracy improvement come at the cost of privacy; over 60% of the VAR-simulated series can be correctly identified on average, and over 20% can be identified based only on one forecast. k-nTS+ provides much stronger privacy protection with identification and forecast disclosure risk estimates falling at or below the 9% threshold.

Method	Data Set	Parameter	$\bar{P}$	$\bar{P}^f$	% Change MAE (VAR)
k-nTS+	Desirable	$k = 4$	8.44%	9.2%	4141.39%
	Undesirable	$k = 7$	5.98%	6.40%	420.13%
VAR Simulated	Desirable	-	61.50%	20%	3782.08%
	Undesirable	-	64.00%	60%	-0.01%

Table 2: A Tale of Two (Sets of 10) Time Series: Comparison of privacy and VAR model forecast accuracy (percent change in MAE) under k-nTS+ and the VAR weight sharing approach for the protected versions of ten simulated series with desirable and undesirable features.

Our results show that achieving good forecast accuracy at an acceptable level of privacy is not possible on small sets of time series with unique means and variances. When time series have very different values (and features) they are easily identified which is consistent with privacy results on non-time series data - that a small set of quasi-identifying attributes is often sufficient to identify large portions of data sets (Sweeney, 2002). Furthermore, DP, AN, k-nTS+, and the VAR-simulated approach are not able to produce protected versions of these series with acceptable privacy protection *and* forecast accuracy. Either the time series values (and features) are changed drastically to achieve acceptable privacy protection with unacceptable forecast accuracy (*e.g.*, under k-nTS+), or values and features are preserved with little to no privacy (*e.g.*, the VAR-simulated series). Given the poor accuracy results, this leads us to ask the question, *Under what conditions is it possible to achieve an acceptable trade-off between forecast accuracy and privacy?*

4.5. Releasing Protected Data with Good Forecast Accuracy

To create an environment where protected data can be released with good forecast accuracy, we propose performing the data protection and forecasting tasks after normalizing the original time series values. To illustrate, we transform the simulated time series into rates, which are preferred to original values for privacy reasons since they are bounded (Abowd et al., 2013) which relates to the global sensitivity of differential privacy (Dwork, 2011). We convert each simulated time series into a series of rates using the following formula

$$r_t = (A_t - A_{t-1}) / \bar{A}_t, \tag{12}$$

where $\bar{A}_t = (A_t + A_{t-1})/2$. We set $r_1 = 0$ to begin the conversion. Assuming the original time series values are non-negative, the resulting rates will be bounded to $[-2, 2]$. If sequential zeros occur in the original data, *i.e.*, if $A_t = A_{t-1} = 0$, the rate in (12) will be undefined and will have to be manually set to zero.

Transforming the original series into rates will make the series less identifiable by normalizing the values across series with different means and other features. This also reduces the identifiability of functions of the time series values, such as a mean forecast. For example, a rate $r_t = 0.1$ could correspond to a time series with $A_t \approx 1$ and $A_{t-1} \approx 0.9$, or $A_t \approx 100$ and $A_{t-1} \approx 90$. In addition, the ability of our proposed k-nTS+ method to swap time series values with minimal detriment to forecast accuracy will improve since the swappable values for any given time period will be on a similar scale to the replaced value. Figure 13 in the Appendix shows that the rate transformation automatically increases the cross-series similarity of some of the time series features such as *Mean*, *Variance*, *Linearity*, and *Curvature*.

We repeat the data protection and forecasting simulation described above on the transformed series. Table 3 summarizes the results for each privacy method that directly alters the unprotected

time series values. We display the weakest privacy parameter for each privacy method and data set that produced protected data with approximately 9% identification disclosure risk or less. The forecast accuracy results are averaged across the SES and DES forecasting models. We note that the forecasts from the VAR model(s) were relatively unstable for the additive noise protected data so we omit these results to avoid skewing the values. In Table 2 we compare the VAR forecast accuracy and privacy results for the k-nTS+ protected data sets and the VAR-simulated approach directly.

Method	**Data Set**	**Parameter**	$\bar{P}$	$\bar{P}^f$	**% Change MAE**
Additive Noise	Desirable	$s = 30.00$	25.62%	16.8%	2712.62%
	Desirable (rate)	$s = 10.77$	10.00%	11.0%	57.36%
	Undesirable	$s = 30.00$	24.64%	21.8%	1265.12%
	Undesirable (rate)	$s = 6.15$	10.00%	11.4%	87.12%
Differential Privacy	Desirable	$\epsilon = 0.01$	9.60%	10.4%	4647.70%
	Desirable (rate)	$\epsilon = 0.54$	10.00%	12.2%	81.05%
	Undesirable	$\epsilon = 0.01$	10.28%	10.2%	2096.71%
	Undesirable (rate)	$\epsilon = 0.01$	10.10%	9.4%	14226.57%
k-nTS+	Desirable	$k = 4$	8.44%	9.6%	1178.70%
	Desirable (rate)	$k = 2$	8.28%	6.60%	-0.29%
	Undesirable	$k = 7$	5.98%	6.00%	787.42%
	Undesirable (rate)	$k = 2$	3.02%	7.00%	4.36%

Table 3: A Tale of Two (Sets of 10) Time Series: Comparison of privacy ($\bar{P}$) and forecast accuracy (percent change in MAE) for the protected versions of ten simulated series with desirable and undesirable features in original and rate form.

All of the privacy methods that directly alter the time series values offer an improved trade-off between privacy risk and forecast accuracy on the rate-transformed series. k-nTS+ offers the best trade-off with identification and forecast disclosure risks that are below the random guessing threshold of 10%, and a change in forecast accuracy that ranges from no change on the series with desirable features, to 4% less accurate on the series with undesirable features. For the parameters considered, the disclosure risks converge to random guessing under differential privacy on both the original and rate data, and under additive noise on the rate data. In other words, there is a threshold of random noise under both AN and DP that results in the protected time series and forecasts becoming indistinguishable from each other under our identification disclosure attack. Under k-nTS+, real time series values are swapped across time series, such that the adversary's guesses of the time series identities are misled to be less accurate than random guessing.

Method	Data Set	Parameter	$\bar{P}$	$\bar{P}^f$	% Change MAE
k-nTS+	Desirable	$k = 4$	8.44%	9.20%	4141.39%
	Desirable (rate)	$k = 2$	5.11%	3.60%	-4.51%
	Undesirable	$k = 7$	5.98%	6.40%	420.13%
	Undesirable (rate)	$k = 2$	2.34%	7.60%	-19.26%
VAR Simulated	Desirable	-	61.50%	20.00%	3782.08%
	Desirable (rate)	-	10.00%	20.00%	-11.26%
	Undesirable	-	64.00%	60.00%	-0.01%
	Undesirable (rate)	-	10.10%	10.00%	1.70%

Table 4: A Tale of Two (Sets of 10) Time Series: Comparison of privacy ($\bar{P}$) and VAR model forecast accuracy (percent change in MAE) under *k*-nTS+ and the VAR-simulated approach for the protected versions of ten simulated series with desirable and undesirable features in original and rate form.

In Table 4 we see similar improvements in the privacy and VAR model forecast accuracy for *k*-nTS+ and the VAR-simulated approach applied to the rate data. Notably, the identification disclosure of the VAR-simulated series reduces to random guessing while privacy accuracy is maintained (undesirable series) or even improves (desirable series). However, the forecast disclosure risk of the VAR-simulated approach remains high for the desirable series. In comparison, the forecasts from the *k*-nTS+ protected data offered accuracy improvements from 4%-19% while keeping average forecast disclosure well under random guessing.

4.6. Increasing the Number of Time Series

In Tables 5 and 6, we include the results from applying the privacy methods and VAR-simulated approach to sets of one hundred time series generated in the same manner described in Section 4.1. This allows us to see the effect of the number of time series in a data set on the choice of privacy parameter that meets the identification disclosure risk threshold and the resulting forecast accuracy. Again, we display the weakest privacy parameter for each privacy method and data set that produced protected data with 9% identification disclosure risk or less.

Method	Data Set	Parameter	$\bar{P}$	$\bar{P}^f$	% Change MAE
Additive Noise	Desirable	$s = 8.46$	8.96%	6.32%	595.51%
	Desirable (rate)	$s = 1.54$	6.18%	2.48%	2.11%
	Undesirable	$s = 8.46$	8.75%	4.88%	199.59%
	Undesirable (rate)	$s = 1.54$	6.65%	1.16%	1.88%
Differential Privacy	Desirable	$\epsilon = 0.54$	7.67%	0.92%	842.89%
	Desirable (rate)	$\epsilon = 4.22$	7.14%	2.28%	2.96%
	Undesirable	$\epsilon = 0.54$	6.82%	4.46%	428.99%
	Undesirable (rate)	$\epsilon = 4.22$	8.50%	1.42%	2.09%
k-nTS+	Desirable	$k = 2$	4.59%	7.20%	465.87%
	Desirable (rate)	$k = 2$	5.11%	1.20%	8.55%
	Undesirable	$k = 2$	1.84%	3.28%	189.96%
	Undesirable (rate)	$k = 2$	2.34%	1.38%	1.32%

Table 5: A Tale of Two (Sets of 100) Time Series: Comparison of privacy ($\bar{P}$) and forecast accuracy (percent change in MAE) for the protected versions of one hundred simulated series with desirable and undesirable features in original and rate form.

In Table 5, we find that increasing the number of time series in a given data set to one hundred (assuming the time series come from the same data generating process) allows the privacy methods to produce protected data with significantly better forecast accuracy with similar or improved protection against re-identification. The only exception occurred under k-nTS+ on the desirable rate series, where forecast accuracy was reduced by 8.6%. However, these forecasts would still be usable and the effect of transforming the series into rates is consistent with the results on the sets of ten series - the privacy-utility trade-off improved upon that achieved for the non-transformed series such that all methods were able to offer usable forecasts and protection against re-identification that met the 9% threshold. We note that due to the increase in the number of time series being protected, the identification risk threshold of 9% is much higher than the reidentification rate under random guessing (1%). This means that less noise needs to be added under additive noise and differential privacy to achieve the required level of privacy protection. For example, the necessary s value dropped from 30 on the set of ten desirable series to 8.46 on the set of one hundred desirable series. Since less noise is added, the percentage increase in MAE is lower (595% on the set of one hundred series vs. 2713% on the set of ten series). For k-nTS+, the increase in the number of time series means there is a greater chance that each time series has k-nearest neighbors with similar features to swap with, which resulted in improved forecast accuracy in all cases except for the desirable rate series.

Method	**Data Set**	**Parameter**	$\bar{P}$	$\bar{P}^f$	**% Change MAE**
k-nTS+	Desirable	$k = 2$	4.59%	9.20%	0.00%
	Desirable (rate)	$k = 2$	5.11%	3.60%	-16.16%
	Undesirable	$k = 2$	1.84%	6.40%	0.00%
	Undesirable (rate)	$k = 2$	2.34%	7.60%	-3.76%
VAR Simulated	Desirable	-	0.85%	-	-
	Desirable (rate)	-	1.70%	4.00%	> 1000%
	Undesirable	-	1.15%	1.00%	-0.00%
	Undesirable (rate)	-	2.05%	3.00%	-1.90%

Table 6: A Tale of Two (Sets of 100) Time Series: Comparison of privacy ($\bar{P}$) and VAR model forecast accuracy (percent change in MAE) under k-nTS+ and the VAR-simulated approach for the protected versions of one hundred simulated series with desirable and undesirable features in original and rate form.

Table 6 compares the privacy and VAR forecast accuracy for the k-nTS+ protected data and the VAR-simulated approach applied to one hundred time series. In this case, the k-nTS+ method produces no reduction in forecast accuracy for the VAR model on the desirable series, while the VAR-simulated approach produces many diverging series with forecasts equal to infinity. On the undesirable series, both methods offer comparable performance with little change in forecast accuracy on both the original and rate series and protection against reidentification that is slightly stronger for the VAR-simulated approach. Overall, increasing the number of time series greatly reduced the identification risk for the VAR-simulated approach. We note that if the VAR model was properly specified to achieve good forecast accuracy on the desirable series, the sets of ten and one hundred desirable time series may be more identifiable.

5. Empirical Application

For comparison, the results in Sections 4.4 and 4.5 are derived from synthetic time series created with controlled features which may not be representative of real-world data sets. In this section, we assess the balance between privacy and accuracy on time series more representative of real world data.

5.1. Real World Data

Recent work by Spiliotis et al. (2020) shows that the M3 and M4 competition data sets contain time series with features representative of real world data, which makes these data sets suitable for illustrating our feature-based k-nTS+ swapping method. Related to privacy, the organizers of the early M competitions did not disclose the true identity of the time series used in their competitions (Makridakis & Hibon, 2000). This provides a natural connection to privacy because we can compute the identification disclosure risk of each protected time series. Motivated by the results in the illustrative application in Section 4, we analyze the forecast accuracy and privacy when protecting the original M3 and M4 data sets and after converting the M3 and M4 data sets into rates. We focus on the M3 results in this section and include M4 results in Section 8.2.

As discussed in Section 4.3.3, the data owner protects every single time series value from time period 1 to T. The protected time series are given to forecasters to produce one-step ahead forecasts for time $T + 1$. The data owner then measures forecast accuracy using the protected and unprotected data against the actual values from $T + 1$. We assume the forecaster may be an adversary attempting to identify an unprotected time series by using the protected time series. For calculating the identification disclosure risk measure discussed in Section 4.3.5, we take the most conservative approach and assume that the adversary knows the length of each time series which makes identification easier. Thus, we separate the privacy analysis and protection into the groups of time series with equal lengths within each of the M competition data sets. We note that in order to test our proposed method with the desired values of k (see Section 5.3 below), we exclude any series that do not have the same length as at least fifteen other series. This left us with 2363 of the 3003 M3 competition series, where approximately 1.3% of these protected series would be correctly identified from random guessing.

The rest of the empirical application is outlined as follows. Subsection 5.2 defines the time series features used for the k-nTS+ swapping method, subsection 5.3 describes the privacy methods, subsection 5.4 presents the privacy and forecast accuracy results, subsection 5.7 analyzes how the time series features change after privacy protection, and subsection 5.9 discusses the computational cost of the proposed k-nTS+ privacy method. Subsection 8.4 also analyzes whether the volatility of the original time series and magnitude or direction of privacy adjustments maintained forecast accuracy.

5.2. Time Series Features

We include the set of time series features mentioned in Section 4.3.3 for consideration in k-nTS+ and let our machine learning-based feature selection method determine the set of efficient features for swapping. To demonstrate the uplift in forecast accuracy from using our machine learning-based feature selection method, we also perform k-nTS swapping based on manually selected features that were shown to be predictive of forecast accuracy based on the literature review in Section 2. We select *Mean*, *Variance*, *Spectral Entropy*, *Hurst*, *Skewness*, *Kurtosis*, *Error ACF*, *Trend*, and *Seasonality* since these features were used to improve the forecast accuracy of RNNs (Bandara et al., 2020) and/or were statistically significant predictors of forecast

accuracy on the M4 competition data (Spiliotis et al., 2020). We omit *Stability* and *Non-linearity* since these features were not significant predictors of forecast accuracy after controlling for the previously mentioned features (Spiliotis et al., 2020), and *Frequency* because none of the privacy methods we consider change the frequency of the original data.

5.3. Privacy Methods

We assess differential privacy for $\epsilon = 20, 10, 4.6, 1, 0.1$, additive noise for $s = 0.25, 0.5, 1.0, 1.5, 2.0$, and k-nTS and k-nTS+ for $k = 3, 5, 7, 10, 15$. To perform feature selection for k-nTS+, the data owner creates protected versions of the data using both AN and DP from time period 1 to $T - 1$ for the above parameter values and performs feature selection and data protection as described in Section 4.3.3. We continue to use $n = 25$ as the window length for the monthly data, and $n = 12$ for other frequencies, as $n \geq 12$ is required for the computation of some of the time series features such as the sum of the first ten coefficients of the ACF of the twice-differenced time series (*Second Difference ACF10*).

5.4. Results

For all privacy methods, we generate one-step ahead forecasts for time $T+1$ using forecasting models in R and Python listed in Table 7. Similar to the M Competitions, all reported forecast accuracy and standard deviation results are derived from comparing the forecasts for $T + 1$ to the actual data from $T + 1$. For the rate versions of the series, we examine the accuracy of the rate forecasts compared to the actual rate from time $T + 1$, and the rate forecast transformed back to the original time series scale compared to the actual original value from $T + 1$. Reported privacy results are derived from calculating the identification disclosure risk using the protected data from time period 1 to T. The light gradient-boosting machine (LGBM), RNN, and VAR models are trained separately on the subsets of time series with equal lengths. We perform minimal data pre-processing and use the standard settings in the off-the-shelf packages. Full implementation details can be found in the appendix.

Model	**Variant**
SES	-
DES	Additive trend
TES	Additive trend/seasonality
Auto-ARIMA	Seasonal
VAR	-
LGBM	-
RNN	-

Table 7: Univariate and multivariate forecasting models.

Table 8 displays the average MAE of one-step ahead point forecasts and the disclosure risk metrics $\bar{P}$ and $\bar{P}^f$ across all models and M3 series. The percentages in parentheses are the increase in average MAE relative to the average MAE from the unprotected series. For each privacy method, we display the two privacy parameters that had identification disclosure risk measures closest to the 9% threshold.

Privacy Method	Parameter	$\bar{P}$	$\bar{P}^f$	Accuracy (% Change MAE)
Unprotected	-	98.41%	18.36%	424.66
Additive Noise	$s = 1.50$	11.21%	5.95%	1361.37 (220.58%)
	$s = 2.00$	8.00%	4.86%	1689.36 (297.81%)
Differential Privacy	$\epsilon = 4.6$	15.68%	6.85%	1309.83 (208.44%)
	$\epsilon = 1.00$	3.70%	2.56%	3401.81 (701.07%)
k-nTS	$k = 3$	4.12%	2.57%	1119.30 (163.58%)
	$k = 5$	3.72%	2.46%	1187.88 (179.73%)
k-nTS+	$k = 3$	4.93%	3.23%	973.86 (129.33%)
	$k = 5$	4.54%	2.96%	1015.20 (139.06%)
k-nTS+ (k = 3)	$M = 1.0$	14.39%	5.47%	741.26 (74.55%)
	$M = 1.5$	8.42%	4.43%	815.70 (92.08%)

Table 8: Identification disclosure risk and forecast accuracy for the unprotected and protected original M3 data sets.

The results show an inverse relationship between forecast accuracy and the strength of privacy protection. However, on average, none of the methods offer good protection against identification disclosure while maintaining an acceptable level of forecast accuracy, which is consistent with the results from the simulation in Section 4.4. While not shown in this table, even small amounts of random noise that offer little privacy protection (additive noise with $s = 0.25$ and differential privacy with $\epsilon = 20$) degrade forecast accuracy by approximately 30%. On the other hand, k-nTS and k-nTS+ offer acceptable protection against reidentification at any value of k but still produce significant reductions in forecast accuracy of over 100%. For a comparable level of privacy (about 3-8% identification disclosure) the k-nTS+ ($k = 3$) protected series had approximately 166% better forecast accuracy than additive noise ($s = 2.00$) and 570% better than differential privacy ($\epsilon = 1.00$). Further, k-nTS+ had approximately 30% better forecast accuracy than standard k-nTS, which shows the uplift from the proposed feature selection methodology[7].

We attempt to improve the forecast accuracy of the k-nTS+ ($k = 3$) protected data by implementing an additional post-processing step. Let $B_j \geq \left|A_{j,t} - P_{j,t}\right|$ denote a non-negative constant that bounds the absolute difference between the original and protected time series values in time t for the jth series. In this case, $P_{j,t}$ is the value that was swapped into time period t during the k-nTS+ protection process. In related work, Lee & Schneider (2023) derive robust bounds on the absolute change in additive exponential smoothing-based forecasts based on protected data.

We define $B_j = M\sigma_{x_j}$ where M is a non-negative constant and σ_{x_j} denotes the standard deviation of time series x_j. Let $P^B_{j,t}$ denote the protected value for series j in time t after bounding the absolute difference between the unprotected and protected values in series j, which is defined as follows

[7]The results for the unprotected data exclude the application of the VAR model to the yearly micro subset of the M3 data due to the model producing explosive forecasts that skewed the overall average results. Similarly, the averages for additive noise with $s \in \{1.5, 2\}$ and for differential privacy with $\epsilon \in \{0.1, 1.0, 4.6\}$ exclude results from the VAR model applied to various subsets of the M3 data that produced MAE values over 100,000. Even with the large amount of random noise infused from these privacy methods, the VAR model did not decrease the magnitude of coefficients to smooth out the noise, resulting in extremely poor forecast accuracy. This is especially problematic if, for example, an extreme outlier occurs in a series at time T which causes the forecast at time T + 1 to explode and skew the overall average forecast error. This problem did not occur for the other forecasting models, which did a better job smoothing out the random noise.

$$P^B_{j,t} = \begin{cases} P_{j,t}, & if \ \ |A_{j,t} - P_{j,t}| \leq B \\ A_{j,t} - B_j, & if \ \ |A_{j,t} - P_{j,t}| > B \ \ and \ \ A_{j,t} > P_{j,t} \\ A_{j,t} + B_j, & if \ \ |A_{j,t} - P_{j,t}| > B \ \ and \ \ A_{j,t} < P_{j,t} \end{cases}. \quad (13)$$

The parameter M can be used to control the trade-off between privacy and forecast accuracy. Smaller (larger) values of M will restrict the protected values $P^B_{j,t}$ to a tighter (looser) interval around the unprotected values $A_{j,t}$, which will increase (reduce) forecast accuracy and increase (reduce) identification disclosure risk. This post-processing is useful in cases where $k - nTS+$ produces protected data with identification disclosure risk that is below the 9% threshold, *i.e.*, there is headroom to increase forecast accuracy at the expense of identification disclosure risk while remaining below the required threshold. This post-processing step is also computationally efficient because for a given value of k, the bounding post-processing step can improve forecast accuracy without having to repeat the swapping process for a lower value of k.

We apply the bounding step to the k-nTS+ ($k = 3$) protected data set and the results are shown in the last two rows of Table 8. We tested values of $M = 0.5, 1.0, 1.5$. Using $M = 1.5$ improved forecast accuracy by about 37% relative to the k-nTS+ ($k = 3$) protected data and increase identification disclosure risk to 8.42%, but the forecasts are still unusable in this scenario. While using $M = 1.0$ further improves forecast accuracy, it pushes the identification disclosure risk to over 14%. While the bounding improves forecast accuracy at acceptable levels of identification risk, it does not change our overall conclusion that none of the methods offer good protection against identification disclosure while maintaining an acceptable level of forecast accuracy.

Privacy Method	**Parameter**	$\bar{P}$	$\bar{P}^f$	**Accuracy (% Change MAE)**	**Accuracy (% Change MAE) Original Scale**
Unprotected	-	98.40%	2.67%	0.0137	424.66
Additive Noise	$s = 1.00$	21.47%	2.02%	0.0157 (14.98%)	1564.98 (268.53%)
	$s = 1.50$	7.22%	1.87%	0.0169 (23.92%)	771.74 (81.73%)
Differential Privacy	$\epsilon = 4.6$	10.16%	2.01%	0.0175 (28.05%)	812.72 (91.38%)
	$\epsilon = 1.00$	1.32%	1.65%	0.0402 (193.98%)	-
k-nTS	$k = 3$	4.90%	1.50%	0.0147 (7.33%)	531.19 (25.09%)
	$k = 5$	4.14%	1.53%	0.0147 (7.16%)	533.35 (25.59%)
k-nTS+	$k = 3$	8.20%	1.61%	0.0142 (3.63%)	505.94 (19.14%)
	$k = 5$	6.50%	1.54%	0.0144 (5.12%)	531.98 (25.27%)
k-nTS+ (k = 3)	$M = 1.0$	30.83%	1.64%	0.0143 (4.75%)	527.58 (24.24%)
	$M = 1.5$	15.81%	1.66%	0.0144 (5.44%)	509.17 (19.90%)

Table 9: Identification disclosure risk and forecast accuracy for the unprotected and protected M3 rate data sets.

Table 9 displays the same information as Table 8 but for the M3 rate data. Further, since the data owner possesses the original time series data, the forecaster can share rate forecasts with the data owner who can transform the rate forecasts back to the scale of the original time series. Let $\hat{r}_{T+1}$ denote the forecasted rate for time $T + 1$. A forecast for time $T + 1$ that matches the scale of the original data can be obtained using $\hat{r}_{T+1}$ and the original data point A_T from time T (which is known by the data owner),

$$\hat{A}_{T+1} = A_T \frac{1 + 0.5\hat{r}_{T+1}}{1 - 0.5\hat{r}_{T+1}}. \quad (14)$$

Column five of Table 9 contains the average MAE for these transformed rate forecasts. Under the k-nTS and k-nTS+ privacy methods, these transformed rate forecasts are usable, with only

19-26% less accuracy than the forecasts from the original data. These forecasts are significantly more accurate than any of the forecasts from the protected original data shown in column four of Table 8. We do not include the results for transformed rate forecasts based on differential privacy ($\epsilon = 1.00$) because many of the forecasts approached infinity and resulted in poor forecast accuracy.

The protected rate datasets consistently exhibit smaller forecast error increases across all privacy methods and parameters, with the rate transformation consistently reducing identification disclosure risk. For k-nTS and k-nTS+, identification disclosure risk is capped at 8%, with forecast accuracy reductions of approximately 7% or less which is a better trade-off than under AN and DP. The post-processing bounding method proved ineffective for k-nTS+ ($k = 3$) protected rate data as identification disclosure risk was pushed to unacceptable levels with no improvement in forecast accuracy. The feature selection method in k-nTS+ results in a 4-5% accuracy improvement over k-nTS, albeit with a slight increase in identification disclosure risk.

Additional accuracy results for k-nTS+ ($k = 3, M = 1.5$) protected original data and k-nTS+ ($k = 3$) protected rate data, broken down by forecasting model and M3 data subset, are provided in Tables 13 and 14 in the Appendix. Table 13 shows that the original 'Monthly Micro' data had an acceptable increase in forecast error of 14.79%, but all other subsets had increases in forecast error of 69% or more. Notably, the rate forecasts and transformed rate forecasts had acceptable changes in accuracy on most M3 subsets, with the exception of 'Monthly Finance' and 'Monthly Micro'. TES had the best accuracy on the original data, but was one of the least accurate for the rate data. The LGBM and RNN models performed well on both unprotected and protected rate data, while Auto-ARIMA excelled in most cases except for k-nTS+ ($k = 3, M = 1.5$) protected original data. While not displayed in the table, the forecasts based on unprotected rate data for LGBM and RNN models show notable accuracy improvements of 15% and 12% over forecasting the unprotected original series, aligning with established practices in time series normalization for global forecasting models (Hewamalage et al., 2021, 2022).

5.5. Can We Share Model Parameters Instead of Protected Data?

We apply the VAR-simulation approach discussed in Section 4.3.4 and attempt to identify the simulated series using the privacy analysis in Section 4.3.5. Rather than sharing simulated values with the forecaster, another option is for the data owner to share protected lagged values, which we assess using the best performing k-nTS+ protected original and rate data sets and the additive noise ($s = 0.25$) data sets. We compare the accuracy and privacy of these VAR-based approaches to our proposed approach of forecasting using VAR models trained on k-nTS+ protected original and rate data sets.

Table 10 reveals that while the VAR-simulation approach yields the best forecast accuracy (22% reduction over unprotected data) with acceptable identification disclosure risk, it barely reduces the forecast disclosure risk relative to the unprotected data. In other words, the time series are identifiable based on forecasts from VAR model parameters and its lagged data. This confirms the results from the simulation study in Section 4; on the original time series scale, good forecasts are not private.

Privacy Method	Shared Data	Shared Model	$\bar{P}$	$\bar{P}^f$	Accuracy (% Change MAE)
Unprotected	Full Data	No	98.4%	18.36%	418.01
k-nTS+ (k = 3, M = 1.5)	Full data	No	8.00%	4.43%	848.16 (102.91%)
AN (s = 0.25)	Lags only	Yes (VAR)	*70.89%	8.59%	882.64 (111.16%)
k-nTS+ (k = 3, M = 1.5)	Lags only	Yes (VAR)	*8.00%	3.30%	2910.43 (596.27%)
VAR-simulation	Lags only	Yes (VAR)	9.70%	17.99%	508.70 (21.70%)

Table 10: Disclosure risks and VAR model accuracy for the unprotected, k-nTS+ ($k = 3, M = 1.5$), and VAR-based approaches applied to the original M3 data. The * values indicate that the disclosure risk value from the privacy analysis on the full data, *i.e.*, the average $\bar{P}$ if an adversary collect ten lagged values over time.

The results for the M3 rate series in Table 11 show that the rate transformation enables accurate forecasts *because the forecasts are highly similar across all series*. Even on the unprotected data, just 2.67% of forecasts can correctly identify a time series. The disclosure risks for both the full k-nTS+ protected data sets and the VAR-simulation approach are acceptably low, while the best accuracy is achieved on the full k-nTS+ protected data sets for both the rate and transformed rate forecasts. Overall, the VAR-based approaches that utilize protected lags performed poorly. Even with weak privacy using additive noise-protected lags ($s = 0.25$), forecast accuracy is 8% worse on the original data and 45% worse on the rate data compared to k-nTS+.

Privacy Method	Shared Data	Shared Model	$\bar{P}$	$\bar{P}^f$	Accuracy (% Change MAE)	Accuracy (% Change MAE) Original Scale
Unprotected	Full data	No	98.42%	2.67%	0.0173	418.01
k-nTS+ (k = 3)	Full data	No	8.20%	1.61%	0.0162 (-6.28%)	578.50 (38.40%)
AN (s = 0.25)	Lags only	Yes (VAR)	*91.03%	1.52%	0.0242 (39.57%)	1367.35 (227.11%)
k-nTS+ (k = 3)	Lags only	Yes (VAR)	*8.20%	1.69%	0.0226 (30.43%)	718.05 (71.78%)
VAR-simulation	Lags only	Yes (VAR)	2.54%	1.78%	0.0172 (-0.67%)	880.92 (110.74%)

Table 11: Disclosure risks and VAR model accuracy for the unprotected, k-nTS+ ($k = 3$), and VAR-based approaches applied to the M3 rate data. The * values indicate that the disclosure risk value from the privacy analysis on the full data, *i.e.*, the average $\bar{P}$ if an adversary collect ten lagged values over time.

5.6. Summary of Privacy and Accuracy Results

When applied to the original time series, our k-nTS+ method achieves the highest forecast accuracy among privacy methods and outperforms the differential privacy approach by 570% at an acceptable level of privacy. However, none of the tested privacy methods could simultaneously protect the original time series and maintain usable forecast accuracy at acceptable privacy risk levels. Methods relying on random noise compromise forecast accuracy unless very little noise is added, providing minimal privacy protection. Similarly, the approach based on simulating time series from a VAR model produced relatively accurate forecasts, but with high disclosure risk. In contrast, k-nTS+ preserves key time series features but still experiences accuracy degradation due to increased randomness and the unique features and values present in the original time series. To address this, we recommend first transforming time series into rates which normalizes values to $[-2, 2]$ and enhances cross-series feature, value, and forecast similarity. The privacy-forecast accuracy trade-off improves with rate data, with k-nTS+ leading with a mere 3.6% reduction in forecast accuracy compared to 23-28% under random noise for an acceptable level of privacy. On the VAR model specifically, sharing a full k-nTS+ protected rate data set enabled more accurate forecasts than an approach based on simulating time series. Data owners could also reverse the rate transformation on the k-nTS+ based forecasts to achieve forecasts on the original scale which were only 19% less accurate on average than using unprotected data, compared to an 81-91% reduction in accuracy with additive noise and differential privacy. Overall, the results on

the M4 data in Section 8.2 are consistent with the ones described in this section. k-nTS+ ($k = 5$) produces protected M4 rate data with identification and forecast disclosure risks of 4.58% and 0.46%, respectively, an increase in accuracy of 15.10% on the rate data, and a reduction in accuracy for the inverse rate forecasts of only 14.62%.

5.7. Analysis of Time Series Features

5.7.1. Importance of Time Series Features

The RReliefF weights approximate the difference between the probability that the nth feature discriminates between series with different forecast errors, and the probability that the nth feature discriminates between series with the same forecast error. Features with $\omega_n > 0$ have a higher probability of varying across series with different forecast errors than varying across series with similar forecast errors. If we swap using features with $\omega_n > 0$, we will maintain the values of these features throughout the swapping process and better maintain forecast accuracy.

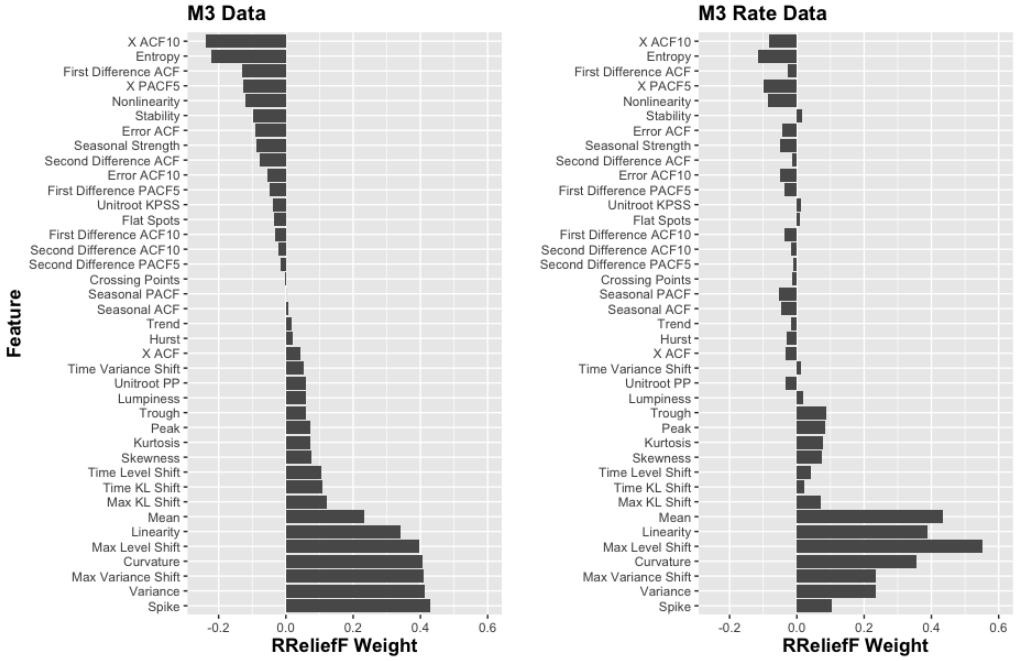

Figure 5: RReliefF weights ω_n averaged across all subsets of M3 data and forecasting models.

Figure 5 shows the RReliefF weights for each of the 39 features averaged across all subsets of the M3 data and all forecasting models. We order the features based on the weights from the original M3 data. RReliefF was used to predict the absolute forecast errors for each model and series across the unprotected and protected data sets. Surprisingly, *Entropy* and *Seasonal Strength* had negative weights for both the original and rate data sets which implied they were not useful to maintain forecast accuracy when swapping. On the other hand, *Max Level Shift*, *Mean*, *Linearity*, *Curvature*, *Max Variance Shift*, *Variance*, and *Spike* had large positive weights and

were important for maintaining forecast accuracy in both data sets. Overall, while the numeric values of the weights vary across the original vs. the rate data sets, the weights of most features retain the same sign. Most exceptions occur for features that had weights with small magnitudes that flipped to the opposite sign, *e.g.*, *Hurst* and *Flat Spots*.

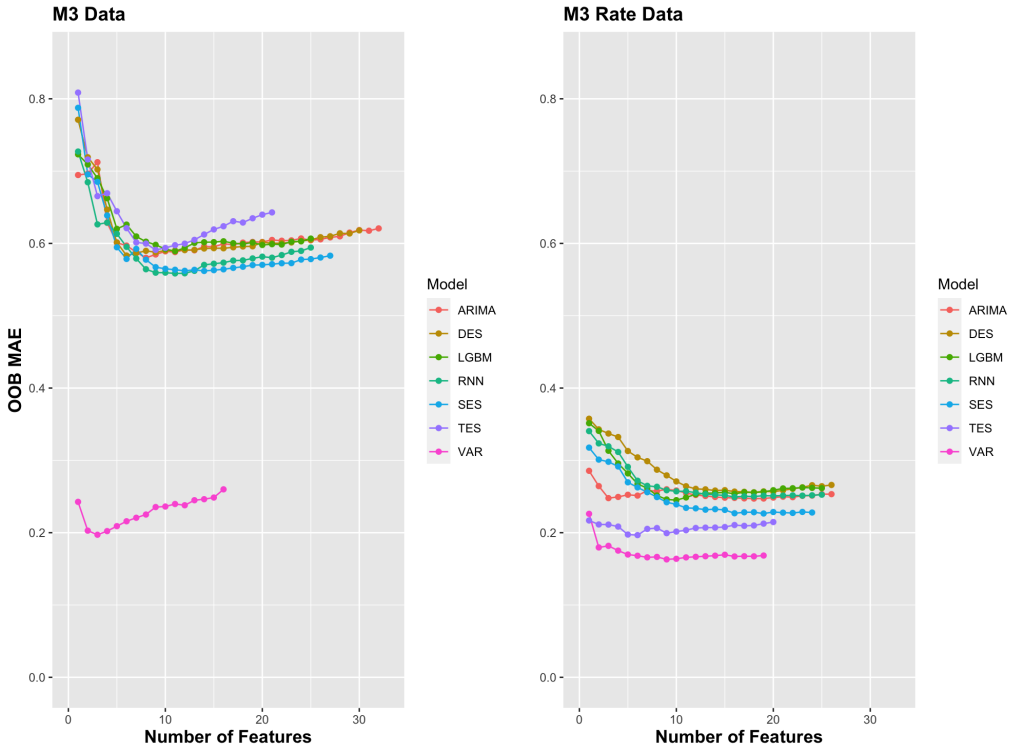

Figure 6: Average OOB MAE across feature subset sizes when predicting the MAE of each forecasting model applied to a subset of the M3 Monthly Micro data.

Figure 6 presents the MAE of the random forest predictions of the MAE of all forecasting models when applied to a subset of the M3 monthly micro data. Note that the number of features considered for each model (and subset of data) varies based on the RReliefF weights from the first stage of the feature selection method. If a feature was identified as being a poor predictor of forecast error (as indicated by $\omega_n < 0$, then it was eliminated prior to the RFE feature selection stage. For each of the forecasting models over $N^{rfe} = 25$ iterations, most of the reduction in OOB MSE occurred using roughly ten or fewer features.

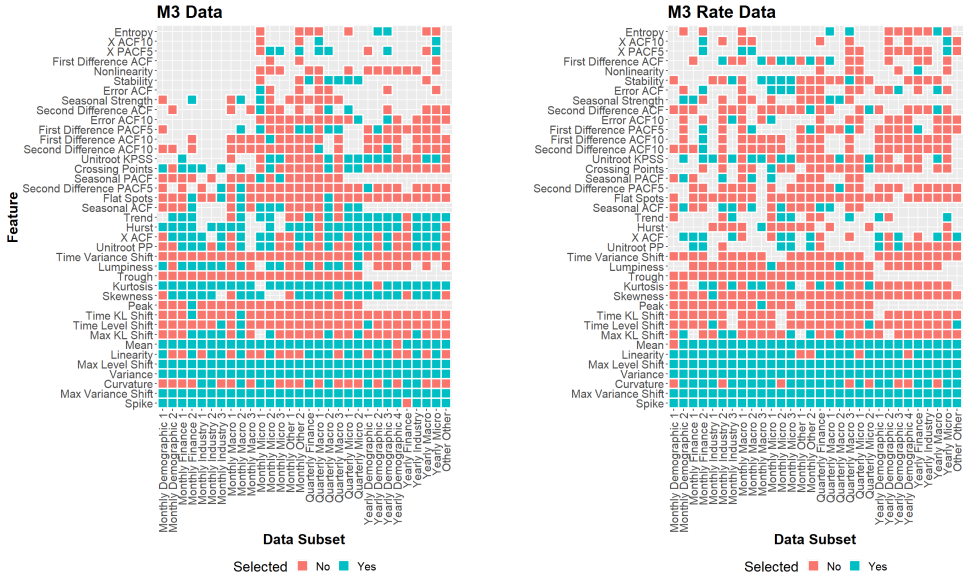

Figure 7: Results of RFE, the second stage of the proposed feature selection methodology. Blue (red) coordinates indicate that a given feature was (was not) selected to be used in k-nTS+ for the corresponding data set. Gray coordinates indicate that either a feature could not be computed for a given data set (*e.g.*, seasonal features for yearly data) or a feature had an importance weight $\omega_n \leq 0$ and was eliminated based on the first stage of the feature selection methodology.

Figure 7 summarizes the results of the RFE feature selection step broken out by data subset and feature. We list features in the order of the magnitude of the importance weights from the original M3 data shown in Figure 5. The uncolored coordinates indicate either that a feature had RReliefF weight $\omega_n \leq 0$ and was eliminated in the first feature selection step, or it could not be computed for a given data set, such as the seasonal features for the yearly data. The benefit of the proposed feature selection methodology becomes apparent when considering the small proportion of blue coordinates in each column, indicating that only a few features are needed to obtain the best accuracy when predicting the error of forecasting models. The features that had the highest average importance weights, such as *Spike* and *Variance*, tended to be selected most frequently regardless of the data under consideration. Some features such as *Curvature* and *Linearity* had high importance weights but were not frequently selected by the RFE stage on the original data. However, these features were selected much more frequently for the rate data. Others, such as *Trend*, *Hurst*, and *X ACF*, were selected relatively frequently for the original quarterly and yearly data, but were eliminated much more frequently for the rate data.

5.7.2. Illustration of Time Series Features After Protection: A Tale of Two (Protected) Time Series

Figure 8 illustrates the advantages of the proposed k-nTS+ privacy method by displaying two time series from the original M3 monthly micro data (where the trade-off between privacy and forecast accuracy was acceptable) with desirable (A) and undesirable (B) features. Plots A.1 and B.1 show the unprotected series, plots A.2 and B.2 illustrate the results using k-nTS+ with $k = 3$

and $M = 1.5$, and plots A.3 and B.3 illustrate the results using additive noise with $s = 1.5$. For the k-nTS+ protected series, there is little visual change for the undesirable series. For additive noise, there are drastic changes to both series.

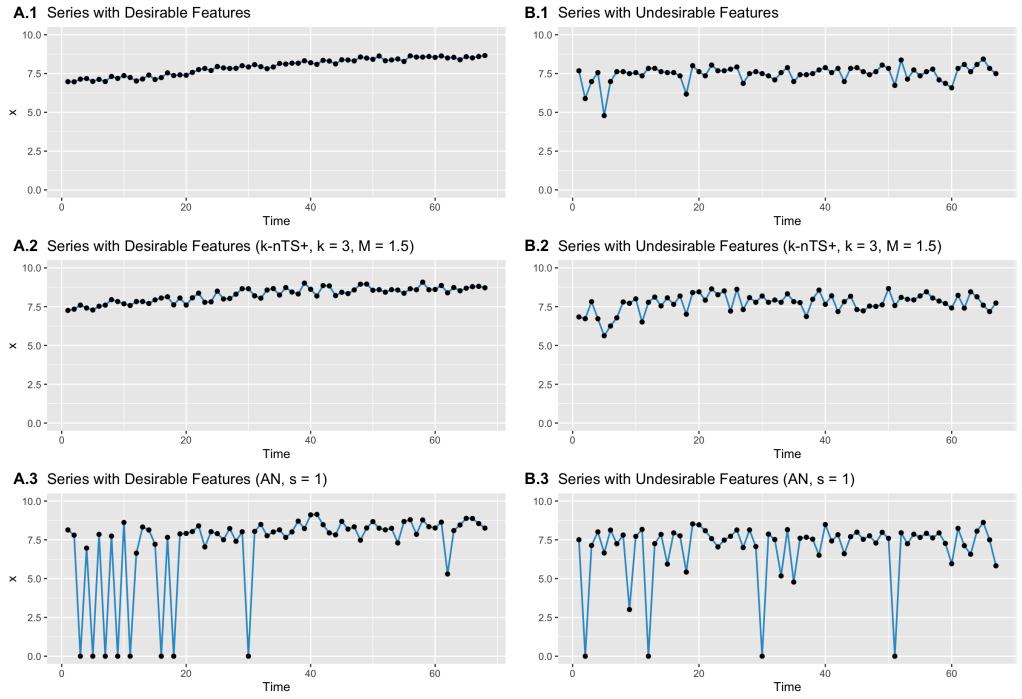

Figure 8: A Tale of Two (Protected) Time Series: Comparison of unprotected, k-nTS+ ($k = 3, M = 1.5$), and AN ($s = 1.5$) protected versions of original 'Monthly Micro' series with desirable (A) and undesirable (B) features.

Feature	**Desirable Time Series (left Fig. 8)**			**Undesirable Time Series (right Fig. 8)**		
	Unprotected	**k-nTS+ ($k = 3$, M = 1.5)**	**AN ($s = 1.5$)**	**Unprotected**	**k-nTS+ ($k = 3$, M = 1.5)**	**AN ($s = 1.5$)**
Spectral Entropy	0.07	0.68	0.92	1.00	0.97	1.00
Hurst	1.00	0.97	0.63	0.58	0.75	0.50
Skewness	-0.40	-0.40	-2.12	-2.12	-0.96	-2.57
Kurtosis	-1.28	-0.92	2.86	-7.02	1.12	6.00
Error ACF	-0.08	0.00	-0.54	-0.21	-0.20	-0.21
Trend	0.97	0.83	0.33	0.27	0.44	0.11
Seasonality	0.16	0.23	0.23	0.32	0.27	0.27
Mean	7.94	8.27	7.12	7.48	7.74	6.94
Variance	0.30	0.23	7.21	0.31	0.36	4.00
Spike	0.0000	0.0000	0.0111	0.0000	0.0000	0.0102
Max Variance Shift	0.05	0.09	11.10	0.60	0.38	8.75
Max Level Shift	0.57	0.49	2.67	0.43	1.02	1.56

Table 12: Feature values from A Tale of Two Time Series: M3 'Monthly Micro' Series A (desirable) and Series B (Undesirable).

Table 12 displays the values of the time series features before and after protection and shows

that the low spectral entropy and high Hurst coefficient values of the desirable time series indicate good forecastability. The undesirable series has a spectral entropy of one indicating a low signal-to-noise ratio. When comparing the two series, the variance of the desirable series is due to a forecastable trend, whereas the variance of the undesirable series is due to randomness. The desirable series also has low Kurtosis with a light tailed distribution compared to the undesirable series. One interesting finding is that the k-nTS+ ($k = 3, M = 1.5$) version of the desirable series has a lower variance than the unprotected series. However, the higher (long run) variance of the unprotected series is due to the strong trend. Figure 8 shows the short run month-to-month variance of the k-nTS+ protected series is higher than the unprotected series, as indicated by the values of *Max Variance Shift* in Table 12.

Finally, Figure 9 displays boxplots of the values of the time series features selected across all subsets of the M3 'Monthly Micro' data set, which included 474 time series. The feature values from the original M3 series are compared to the values from the series simulated from the VAR models and the protected series from the k-nTS+ ($k = 3, M = 1.5$), k-nTS ($k = 3$), AN ($s = 1.5$), and DP ($\epsilon = 4.6$) privacy methods. Random noise privacy methods increase the randomness and significantly change distributional characteristics of most of the features, leading to poor forecast accuracy. On the other hand, the k-nTS and k-nTS+ methods better preserve the feature distributions, with k-nTS+ providing the closest feature values for features such as *X ACF* and *Unitroot PP*. The series simulated from the VAR models in some cases exhibit feature values which are better for forecasting (*e.g.*, *X ACF*) but are unrepresentative of the features observed in the unprotected data. Compared to the features from the k-nTS+ protected series, the features from the VAR simulated series are less similar to the distributions from the unprotected series. Overall, the features of the k-nTS+ protected series are most similar to those of the unprotected series, which results in the best forecast accuracy of the privacy methods we considered.

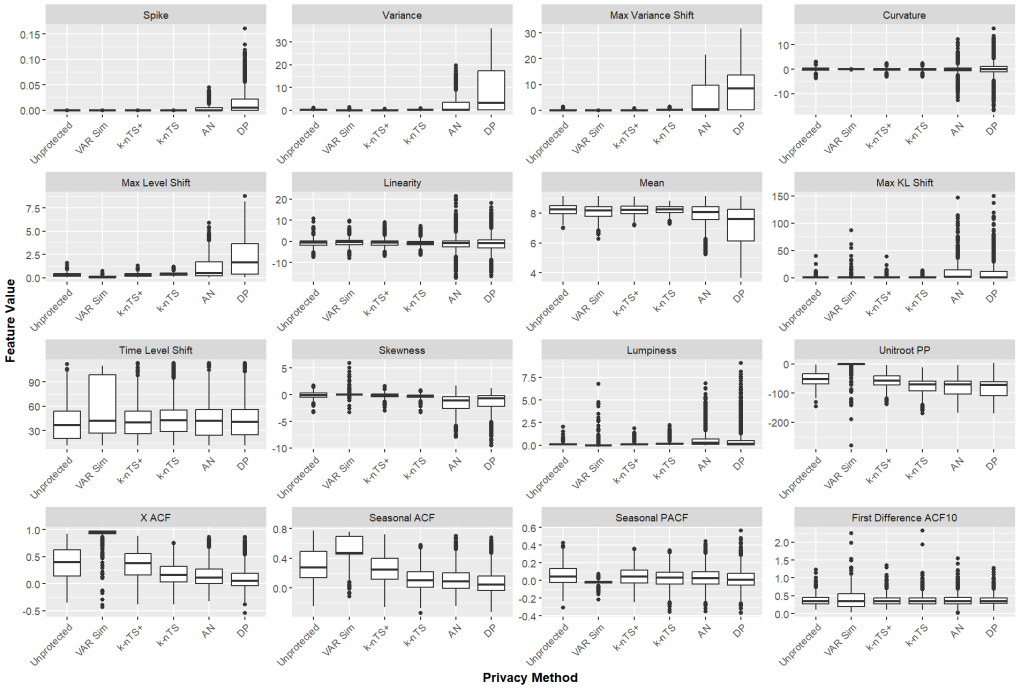

Figure 9: Distributions for each privacy method of the time series features selected for the 474 M3 'Monthly Micro' time series.

5.8. Preservation of M3 Time Series Features

We use principal components analysis (PCA) to compare the feature distributions from all unprotected and protected time series for the features that were selected in the majority of M3 data subsets (at least 28/30 cases), namely, *Max Variance Shift*, *Variance*, *Max Level Shift*, *Spike*, *Mean*, and *Kurtosis* (Kang et al., 2017; Spiliotis et al., 2020). Let $\mathbf{C}$ denote the matrix containing the combined features from all unprotected and protected series from the methods examined in Figure 9. The principal components are calculated as

$$\begin{bmatrix} \text{PC1} \\ \text{PC2} \end{bmatrix} = \begin{bmatrix} 0.50 & 0.48 & 0.30 & -0.40 & 0.49 & 0.15 \\ 0.05 & 0.19 & -0.30 & 0.23 & -0.15 & 0.90 \end{bmatrix} \mathbf{C}, \tag{15}$$

where PC1 and PC2 denote the first two principal components which are linear combinations of the six time series feature values. The smaller the distance between two points in the principal components space, the more similar the corresponding time series are in terms of the six features. The first two principal components explain 78% of the variation in the time series features. The first component increases with the features capturing the values (and changes in the values) of the variance and mean of the time series, and decreases with the spikiness of the time series. The second component is largely determined by *Kurtosis*, with large positive values of *Kurtosis* increasing the values of PC2 and vice versa.

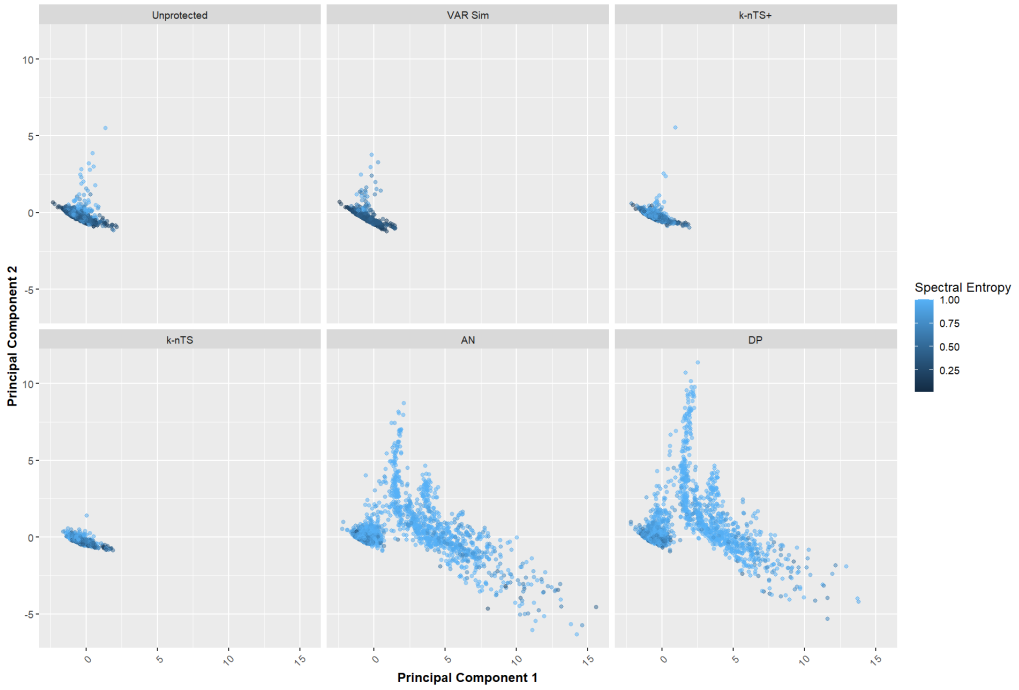

Figure 10: Principal components calculated from the most frequently selected features across all original M3 time series.

Figure 10 compares the distribution of principal component values for all of the unprotected original M3 time series and each privacy method. Visually, the features from the *k*-nTS+ protected data and the VAR-simulation series preserve the most realistic representation of the unprotected feature space. Many of the series protected using differential privacy and additive noise contain combinations of features that do not occur in the unprotected data, such as larger (smaller) values of PC1 (PC2) corresponding to the larger values of *Variance*, *Max Variance Shift*, and *Max Level Shift* shown in Figure 9. There are also many series that have similar values of PC1 but have a large range of PC2 values in the positive direction which corresponds to increases in *Kurtosis* which indicate a series with a peak near the mean and heavy tails.

Figure 10 also depicts the *Spectral Entropy* of the time series using the color of the points on the principal component axes. The poor forecasting performance of the series protected using additive noise and differential privacy is reflected by the light blue color of many points indicating high values of spectral entropy. On the other hand, the VAR simulated series tend to have *lower* spectral entropy than the unprotected series even though these series had reduced forecasting performance *i.e.*, the signal contained in the VAR simulated series is not as predictive of future unprotected values. Further, while the distribution of features that are most important for maintaining forecast accuracy is well maintained in the *k*-nTS+ protected series, forecast accuracy using the *k*-nTS+ protected data still decreased on average, and the average spectral entropy increased from 0.50 in the unprotected series to 0.70 in the *k*-nTS+ protected series. This is consistent with the results in Spiliotis et al. (2020) that showed using multiple linear regression that

an increase in spectral entropy (or randomness) of a time series is associated with an increase in forecast error when holding many other time series features constant.

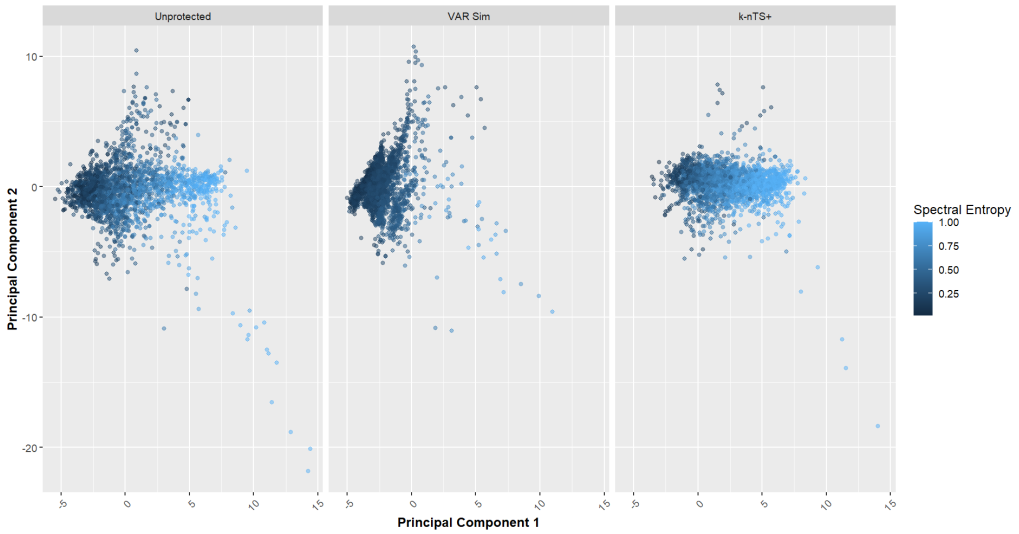

Figure 11: Principal components calculated from the most frequently selected features across all original M3 time series.

Figure 11 plots the first two principal components for the unprotected, VAR-simulation, and k-nTS+ ($k = 3$) series recalculated to include all features.[8] These components capture 40% of the overall variation. While the VAR-simulation series had a similar feature distribution for the six most commonly selected features under k-nTS+, k-nTS+ preserves a more accurate representation of the overall feature distribution.

5.9. Computational Cost

The analysis of the M3 data utilized a desktop PC with an Intel i9-12900k CPU and 32GB DDR5 6000MT/s CL36 RAM. Figure 12 (left) illustrates the total time (seconds) to generate the k-nTS+ protected M3 data subsets as a function of the total number of time periods swapped. The plot shows a roughly linear increase in computational cost, averaging 0.60 additional seconds for one additional swapped period (adjusted R^2 of 0.92). The RReliefF and RFE steps in the feature selection process increase computation time by 0.06 and 2.72 seconds (adjusted R^2 values of 0.91 and 0.92) on average for one additional time series, respectively. On average, k-nTS swapping sees an increase of 0.55 seconds for one additional time series, holding the number of time periods constant, and 1.32 seconds when the average time series length increases by one, holding the number of time series constant (adjusted R^2 value of 0.90). The most computationally intensive subset, 'Monthly Micro,' took just under 32000 seconds (about 533 minutes or 8.88 hours) for the full protection process.

[8]We exclude three features that capture the index of the maximum shifts in the level, variance, and Kulback-Leibler divergence since the VAR-simulation series have a shorter number of time periods overall.

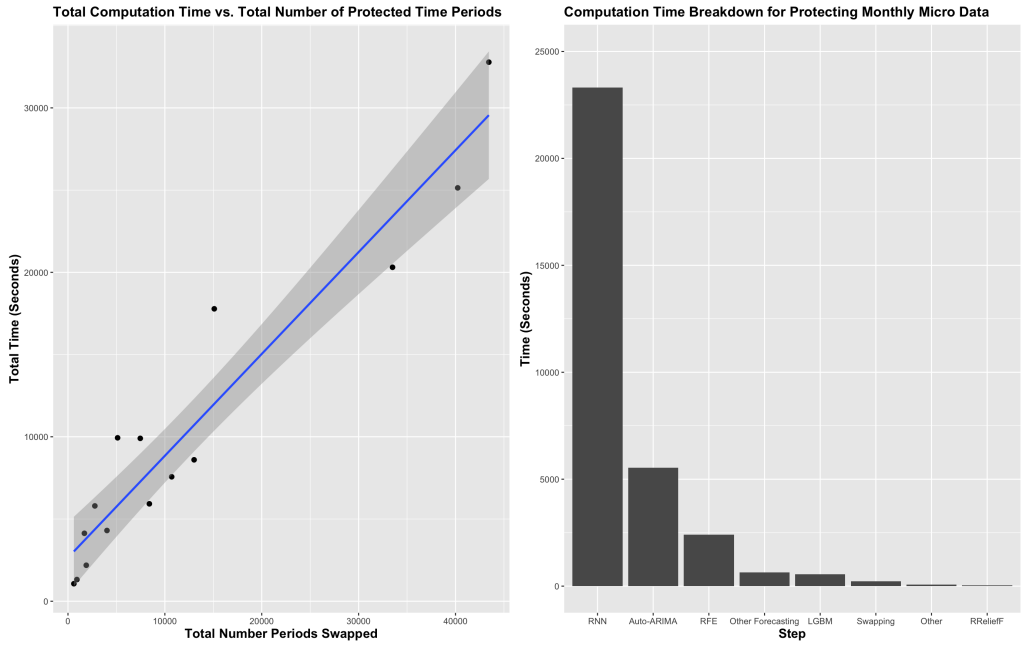

Figure 12: Total computation time (in seconds) to create the k-nTS+ protected data sets (left plot) and a breakdown of the computation time (in seconds) of steps in the protection process (right plot).

Figure 12 (right) illustrates the total computation time in seconds for generating k-nTS+ protected versions of the 'Monthly Micro' time series, segmented by protection process steps. Bars corresponding to forecasting models depict time spent generating forecasts for baseline protected datasets. The majority of computation time was dedicated to RNN forecasting, followed by Auto-ARIMA. Combined time for feature calculation (Other), feature selection (RReliefF and RFE), and swapping in protected values was less than the time needed for forecasting 474 'Monthly Micro' series with Auto-ARIMA models. Excluding RNN forecasting, the full 'Monthly Micro' protection process falls to 8395 seconds (about 140 minutes or 2.33 hours), making it the second most computationally intensive set. The 'Monthly Industry' series, excluding RNN time, takes 8866 seconds (about 147 minutes or 2.46 hours), despite containing fewer (289) but longer (136 periods on average) time series compared to the 'Monthly Micro' subset (474 time series, average length 86.7 periods). All of the results in this section are based on the thirty nine time series features calculated using the *tsfeatures* R package. Computation time will increase (decrease) with an increase (decrease) in the number of features input into the initial feature selection process.

6. Conclusion

We believe the answer to the question of '*Can we protect time series data while maintaining accurate forecasts?*' is: it depends. None of the methods we tested could consistently maintain a usable level of forecast accuracy at acceptable levels of privacy. We found that sharing model parameters or accurate forecasts themselves could re-identify time series, depending upon the

quantity and similarity of time series within a data set. Data sets with less time series and lower similarities were more difficult to achieve an acceptable trade-off between privacy and forecast accuracy. To achieve an acceptable trade-off, time series were transformed into rates, which increased the similarity of time series features and values to improve forecast accuracy and reduce privacy risk compared to the original series. Although it was not possible to protect every time series data set while maintaining accurate forecasts, we believe it is a far, far better thing that forecasters know.

Empirically, this paper examined the impact of data privacy methods on forecast accuracy, where a data owner shares a protected data set with forecasters. We proposed a new privacy method, k-nTS+ swapping, designed to maintain forecast accuracy by swapping the values between time series conditional on time series features. We tested existing additive noise and differential privacy methods, our proposed k-nTS+ method, and a VAR-based simulation method using simulated data and data from a well-known forecasting competition where the identities of the time series needed to be kept confidential.

To the best of our knowledge, this paper is the first to create a protected time series data set tailored to maintain forecast accuracy. We did this by carefully investigating the similarity between time series features rather than time series values. The goal was to find a balance where the few most important features for forecast accuracy are preserved such that forecasts remain usable and privacy is good. Our proposed feature selection method is designed to choose the features that achieve this balance when swapping time series values. We found that the most important features (based on selection frequency) for maintaining forecast accuracy while swapping were *Max Variance Shift*, *Variance*, *Max Level Shift*, *Spike*, *Mean*, and *Kurtosis*. However, forecasting is a unique use case, and there is an inherent tension between feature preservation, forecast accuracy, and privacy. To achieve an acceptable privacy-forecast accuracy trade-off, protected past values (and features) must be dissimilar enough from unprotected values (unprotected features) so that privacy is preserved, while producing forecasts that are as similar as possible to future values. However, we showed that good forecasts themselves are not private. Thus, an acceptable trade-off between privacy and forecast accuracy is unattainable unless time series are first normalized; increasing the similarity of time series features, values, and forecasts such that our proposed k-nTS+ method enables accurate forecasts that maintain privacy.

A substantial portion of the privacy literature is focused on theoretical privacy guarantees such as differential privacy. Our findings agree with past research (Gonçalves et al., 2021) and show that differential privacy (and additive noise) generates unusable forecasts at reasonable levels of privacy. This undesirable privacy-utility trade-off has also been demonstrated in contexts other than forecasting. For example, 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 large values of ϵ that theoretically do not provide meaningful levels of privacy protection. Their experiments found that model regularization (*e.g.*, L2-regularization) provided comparable privacy protection with better accuracy and lower model learning cost than differential privacy. In our application, we found that our k-nTS+ swapping method had better forecast accuracy at comparable levels of identification disclosure risk with differential privacy. We note that compared to differential privacy, our privacy metric could be considered ad hoc and only one possible attack; however, it was interpretable by a layperson and reasonable according to the GDPR.

A major limitation of our study was that we did not consider privacy metrics other than identification and forecast disclosure risk. Reconstruction attacks, where an adversary is able to recover some or all of the original data, are a valid concern especially when data is shared

between multiple parties (Gonçalves et al., 2021; Goncalves et al., 2021). Concern over reconstruction attacks would preclude the scenario where a data owner provides unprotected lagged time series values and a fitted time series model to a forecaster since the forecaster could use the lagged values to recover the unprotected time series. When sharing time series protected using k-nTS+, reconstruction attacks are unlikely to succeed since all past data is protected by releasing a single protected data point for each time period. Even if some of the released points are similar to the original points, the adversary does not necessarily know which points those are. Further, performing swapping using a rolling window allows the set of k nearest time series to change for each successive time period which will incorporate data from a set of time series potentially much larger than k and would make it difficult to reverse engineer the swapping process. While differentially private time series are immune to reconstruction attacks in theory, we found that forecasts from differentially private data at acceptable levels of $\epsilon \leq ln\ 3$ are unusable. Increasing ϵ improves accuracy but also results in significantly higher identification disclosure risk (about 40% and 44% for $\epsilon = 10$ for the original and rate data, respectively). Overall, the differentially private time series in our application could not provide an acceptable trade-off between privacy and forecast accuracy, as these series were either not useful for forecasting or not protective against identification disclosure.

Another potential privacy risk is attribute disclosure, where an adversary discerns sensitive data values for a data subject of interest. In the context of time series, this could occur if an adversary seeks to discern a sensitive time series feature, *e.g.*, the average monthly sales over a 6 month period for the Roseville, Minnesota Target store. Future work should look to rigorously define likely privacy attacks against time series data and propose approaches for managing the risks of such attacks while maintaining the usefulness of time series data.

For future work, when applying our proposed privacy method, we only considered the scenario where all time series are stored in a centralized location. Our method could be extended to decentralized scenarios where data is spread across multiple owners. In this case, each data owner could apply the method to their own time series before sharing/pooling data together. The time series from each data owner could only be swapped with values from the same data owner. In cases of limited data, data owners could consider swapping values with synthetic time series with desirable features (Kang et al., 2020). Our method could also be used by a trusted central party to protect data from multiple data owners before sharing the combined protected data with participating parties. This could produce interesting opportunities for forecasting with global models trained on a combination of a data owner's unprotected data and protected data from other parties. This protection approach may increase the willingness of owners to share data by alleviating privacy concerns with storing sensitive data in a central location (Goncalves et al., 2020).

Further research could examine forecasting strategies for protected data, such as whether combinations of privacy adjusted forecasts improve forecast accuracy or whether time series features can predict the best performing forecasting model(s) for a protected data set. Since many of the time series features were preserved and the entire protected data set was shared, forecasters could use this time series data for other applications such as clustering. Outside of data privacy applications, our k-nTS+ swapping method can also be applied to the tasks of outlier replacement, missing data replacement (Twumasi & Twumasi, 2022), and nowcasting (Barbaglia et al., 2023). It is important to note that the randomization from the swapping mechanism can be removed for tasks that do not require privacy. We used the proposed feature selection method to select features that were predictive of forecast accuracy, but both the RReliefF and RFE stages could be used to select an efficient set of time series features that are predictive of other categor-

ical or continuous target variables.

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

7. Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author(s) used ChatGPT in order to generate suggestions and paraphrase passages from the authors' original writing in Section 5.6. More specifically, the authors asked ChatGPT to write a more concise version of a three paragraph summary written in the author(s) own words. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

References

Abowd, J. M., Schneider, M. J., & Vilhuber, L. (2013). Differential privacy applications to bayesian and linear mixed model estimation. *Journal of Privacy and Confidentiality*, *5*.

Arbuckle, L., & El Emam, K. (2020). *Building an anonymization pipeline: creating safe data*. O'Reilly Media.

Bale, C. D., Fischer, J. L., Schneider, M. J., Weber, S., & Chang, S. (2023). Legally anonymizing location data under the gdpr. *ResearchGate Preprint*, . doi:10.13140/RG.2.2.17076.73609/1.

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.

Barbaglia, L., Frattarolo, L., Onorante, L., Pericoli, F. M., Ratto, M., & Pezzoli, L. T. (2023). Testing big data in a big crisis: Nowcasting under covid-19. *International Journal of Forecasting*, *39*, 1548–1563.

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. *ACM Computing Surveys*, *55*, 1–16.

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.

Chen, C., & Liu, L.-M. (1993). Forecasting time series with outliers. *Journal of forecasting*, *12*, 13–35.

Cleveland, R. B., Cleveland, W. S., McRae, J. E., & Terpenning, I. (1990). Stl: A seasonal-trend decomposition. *J. Off. Stat*, *6*, 3–73.

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.

Drechsler, J. (2023). Differential privacy for government agencies—are we there yet? *Journal of the American Statistical Association*, *118*, 761–773.

Duncan, G. T., & Stokes, S. L. (2004). Disclosure risk vs. data utility: The ru confidentiality map as applied to topcoding. *Chance*, *17*, 16–20.

Dwork, C. (2011). A firm foundation for private data analysis. *Communications of the ACM*, *54*, 86–95.

Dwork, C., & Naor, M. (2010). On the difficulties of disclosure prevention in statistical databases or the case for differential privacy. *Journal of Privacy and Confidentiality*, *2*.

European Medicines Agency (2017). External guidance on the implementation of the European Medicines Agency policy on the publication of clinical data for medicinal products for human use. URL: https://www.ema.europa.eu/en/documents/regulatory-procedural-guideline/external-guidance-implementation-european-medicines-agency-policy-publication-clinical-data-medicinal-products-h en.pdf.

European Parliament and Council of European Union (2016). Gdpr recital 26. URL: Eur-Lex-32016R0679-EN-EUR-Lex(europa.eu).

Fan, L., & Xiong, L. (2013). An adaptive approach to real-time aggregate monitoring with differential privacy. *IEEE Transactions on knowledge and data engineering*, *26*, 2094–2106.

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., & Önkal, 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.

Goerg, G. (2013). Forecastable component analysis. In *International conference on machine learning* (pp. 64–72). PMLR.

Goldfarb, A., & Tucker, C. (2013). Why managing consumer privacy can be an opportunity. *MIT Sloan Management Review*, *54*, 10.

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.

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

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.

Hu, J. (2019). Bayesian estimation of attribute and identification disclosure risks in synthetic data. *Transactions on Data Privacy*, *12*, 61–89.

Hyndman, R., Kang, Y., Montero-Manso, P., O'Hara-Wild, M., Talagala, T., Wang, E., & Yang, Y. (2023). tsfeatures: Time series feature extraction. URL: `https://github.com/robjhyndman/tsfeatures`.

Hyndman, R. J., & Athanasopoulos, G. (2021). *Forecasting: principles and practice*. 3rd Edition, OTexts: Melbourne, Australia. URL: `OTexts.com/fpp3`.

Hyndman, R. J., & Khandakar, Y. (2008). Automatic time series forecasting: the forecast package for R. *Journal of Statistical Software*, *26*, 1–22. doi:`10.18637/jss.v027.i03`.

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)*, (pp. 3428–3437).

Kang, Y., Cao, W., Petropoulos, F., & Li, F. (2022). Forecast with forecasts: Diversity matters. *European Journal of Operational Research*, *301*, 180–190.

Kang, Y., Hyndman, R. J., & Li, F. (2020). Gratis: Generating time series with diverse and controllable characteristics. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, *13*, 354–376.

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.

Khosrowabadi, N., Hoberg, K., & Imdahl, C. (2022). Evaluating human behaviour in response to ai recommendations for judgemental forecasting. *European Journal of Operational Research*, *303*, 1151–1167.

Lee, J., & Schneider, M. J. (2023). Geometric series representation for robust bounds of exponential smoothing difference between protected and confidential data. *Annals of Operations Research*, (pp. 1–11).

Li, L., Kang, Y., & Li, F. (2023a). Bayesian forecast combination using time-varying features. *International Journal of Forecasting*, *39*, 1287–1302.

Li, L., Kang, Y., Petropoulos, F., & Li, F. (2023b). Feature-based intermittent demand forecast combinations: accuracy and inventory implications. *International Journal of Production Research*, *61*, 7557–7572.

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.

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.

Martin, K. D., Borah, A., & Palmatier, R. W. (2017). Data privacy: Effects on customer and firm performance. *Journal of Marketing*, *81*, 36–58.

Martin, K. D., & Murphy, P. E. (2017). The role of data privacy in marketing. *Journal of the Academy of Marketing Science*, *45*, 135–155.

Montero-Manso, P., Athanasopoulos, G., Hyndman, R. J., & Talagala, T. S. (2020). Fforma: Feature-based forecast model averaging. *International Journal of Forecasting*, *36*, 86–92.

Nin, J., & Torra, V. (2009). Towards the evaluation of time series protection methods. *Information Sciences*, *179*, 1663–1677.

Nogueira, F. (2014). Bayesian Optimization: Open source constrained global optimization tool for Python. URL: `https://github.com/fmfn/BayesianOptimization`.

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*, *38*, 705–871.

Petropoulos, F., & Siemsen, E. (2023). Forecast selection and representativeness. *Management Science*, *69*, 2672–2690.

Qi, L., Li, X., Wang, Q., & Jia, S. (2023). fetsmcs: Feature-based ets model component selection. *International Journal of Forecasting*, *39*, 1303–1317.

Reiter, J. P. (2005). Estimating risks of identification disclosure in microdata. *Journal of the American Statistical Association*, *100*, 1103–1112.

Robnik-Šikonja, M., & Kononenko, I. (2003). Theoretical and empirical analysis of relieff and rrelieff. *Machine learning*, *53*, 23–69.

Rose, O. (1996). *Estimation of the Hurst parameter of long-range dependent time series* volume 137. University of Würzburg Institute of Computer Science Research Report Series: Report No. 137.

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*, 153–171.

Seabold, S., & Perktold, J. (2010). statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*.

Sobolev, D. (2017). The effect of price volatility on judgmental forecasts: The correlated response model. *International Journal of Forecasting*, *33*, 605–617.

Sommer, B., Pinson, P., Messner, J. W., & Obst, D. (2021). Online distributed learning in wind power forecasting. *International Journal of Forecasting*, *37*, 205–223.

Spiliotis, E., Kouloumos, A., Assimakopoulos, V., & Makridakis, S. (2020). Are forecasting competitions data representative of the reality? *International Journal of Forecasting*, *36*, 37–53.

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.

Twumasi, C., & Twumasi, J. (2022). Machine learning algorithms for forecasting and backcasting blood demand data with missing values and outliers: A study of tema general hospital of ghana. *International Journal of Forecasting*, *38*, 1258–1277.

Wang, X., Smith, K., & Hyndman, R. (2006). Characteristic-based clustering for time series data. *Data mining and knowledge Discovery*, *13*, 335–364.

Willinger, W., Paxson, V., & Taqqu, M. S. (1998). Self-similarity and heavy tails: Structural modeling of network traffic. *A practical guide to heavy tails: statistical techniques and applications*, *23*, 27–53.

Yang, Y., & Hyndman, R. (2023). Introduction to the tsfeatures package. URL: `https://cran.r-project.org/web/packages/tsfeatures/vignettes/tsfeatures.html`.

Zhao, B., & Hyndman, R. J. (2023). Introduction to gratis. URL: `https://cran.r-project.org/web/packages/gratis/vignettes/QuickStart.html`.

8. Appendix

8.1. M3 Accuracy Results by Model and Data Subset

	Original Forecasts		**Inverse Rate Forecasts**	**Rate Forecasts**	
Model	*Unprotected*	*k-nTS+ (k = 3) M = 1.5*	*k-nTS+ (k = 3)*	*Unprotected*	*k-nTS+ (k = 3)*
SES	433.99	799.18 (84.15%)	485.78 (11.93%)	0.0139	0.0137 (-1.31%)
DES	422.82	798.92 (88.95%)	498.54 (17.91%)	0.0142	0.0140 (-1.71%)
TES	377.43	712.73 (88.84%)	525.68 (39.28%)	0.0143	0.0147 (2.99%)
Auto-ARIMA	389.91	825.09 (111.61%)	470.12 (20.57%)	0.0117	0.0132 (13.25%)
VAR	418.01	848.16 (102.91%)	578.50 (38.40%)	0.0173	0.0162 (-6.28%)
LGBM	458.98	853.25 (85.90%)	476.49 (3.81%)	0.0111	0.0134 (20.27%)
RNN	458.91	848.11 (84.81%)	481.54 (4.93%)	0.0114	0.0136 (19.70%)

Table 13: MAE (% change in MAE) for each forecast model on the *k*-nTS+ protected M3 original and rate data.

	Original Forecasts		Inverse Rate Forecasts	Rate Forecasts	
Data	*Unprotected*	*k-nTS+ (k = 3) M = 1.5*	*k-nTS+ (k = 3)*	*Unprotected*	*k-nTS+ (k = 3)*
Monthly Demographic	112.53	444.88 (295.34%)	113.61 (0.96%)	0.0049	0.0051 (2.27%)
Monthly Finance	291.52	861.62 (195.56%)	482.21 (65.41%)	0.0073	0.0092 (26.57%)
Monthly Industry	517.46	875.63 (69.22%)	595.45 (15.07%)	0.0126	0.0134 (6.74%)
Monthly Macro	178.59	642.08 (259.53%)	206.46 (15.61%)	0.0036	0.0039 (8.50%)
Monthly Micro	687.53	789.24 (14.79%)	937.80 (36.40%)	0.0341	0.0363 (6.47%)
Monthly Other	400.28	704.90 (76.10%)	402.39 (0.53%)	0.0204	0.0122 (-40.44%)
Quarterly Finance	138.95	412.52 (196.89%)	117.95 (-15.11%)	0.0031	0.0027 (-11.19%)
Quarterly Macro	162.76	481.47 (195.82%)	193.12 (18.66%)	0.0034	0.0041 (21.54%)
Quarterly Micro	594.95	1044.80 (75.61%)	743.27 (24.93%)	0.0127	0.0142 (11.59%)
Yearly Demographic	351.97	1095.81 (211.34%)	333.12 (-5.35%)	0.0056	0.0056 (0.02%)
Yearly Finance	1678.25	3600.49 (114.54%)	982.78 (-41.44%)	0.0157	0.0139 (-11.90%)
Yearly Industry	429.82	1295.09 (201.31%)	397.04 (-7.63%)	0.0104	0.0102 (-2.19%)
Yearly Macro	184.97	358.28 (93.69%)	177.68 (-3.94%)	0.0029	0.0031 (8.09%)
Yearly Micro	949.64	1681.62 (77.08%)	864.38 (-8.98%)	0.0237	0.0198 (-16.35%)
Other Other	47.83	529.30 (1006.71%)	40.49 (-15.34%)	0.0013	0.0012 (-2.36%)

Table 14: MAE (% change in MAE) for each data subset from the k-nTS+ protected M3 original and rate data.

8.2. M4 Accuracy and Privacy Results

Tables 15 and 16 contain the disclosure risk and privacy results from applying the differential privacy and k-nTS+ methods to the M4 data and forecasting using SES, DES, VAR, and LGBM models. As discussed in Section 5, we separate the privacy analysis and protection into the groups of time series with equal lengths within each of the M4 competition data sets. To test our proposed method with the desired values of k we exclude any series that do not have the same length as at least fifteen other series. This left us with 88,605 of the 100,000 M4 competition series, where approximately 0.5% of these protected series would be correctly identified from random guessing. For computational efficiency, we excluded the additive noise method from the M4 analysis, meaning that the k-nTS+ feature selection method utilized only the differential privacy method as a baseline.

Privacy Method	**Parameter**	$\bar{P}$	$\bar{P}^f$	**Accuracy (% Change MAE)**
Unprotected	-	99.45%	10.23%	372.10
Differential Privacy	$\epsilon = 10$	25.06%	4.64%	6303.00 (1593.92%)
	$\epsilon = 4.6$	9.25%	2.97%	3641.92 (878.76%)
k-nTS+	$k = 3$	2.57%	1.18%	1500.46 (303.25%)
	$k = 5$	2.13%	1.11%	1438.92 (286.71%)
k-nTS+ (k = 3)	$M = 0.50$	21.95%	3.25%	680.36 (82.85%)
	$M = 1.00$	7.42%	2.23%	879.00 (136.23%)

Table 15: Disclosure risks and forecast accuracy for the unprotected and protected original M4 data sets.

Privacy Method	Parameter	$\bar{P}$	$\bar{P}^f$	Accuracy (% Change MAE)	Accuracy (% Change MAE) Original Scale
Unprotected	-	99.42%	0.78%	0.0125	372.10
Differential Privacy	$\epsilon = 10$	28.84%	0.64%	0.0116 (-6.86%)	521.89 (40.26%)
	$\epsilon = 4.6$	5.15%	0.59%	0.0139 (11.72%)	> 1000%
k-nTS+	$k = 3$	5.97%	0.49%	0.0106 (-14.71%)	431.84 (16.06%)
	$k = 5$	4.58%	0.46%	0.0106 (-15.10%)	426.50 (14.62%)
k-nTS+ (k = 3)	$M = 1.00$	21.71%	0.52%	0.0105 (-15.78%)	433.22 (16.43%)
	$M = 1.50$	11.40%	0.50%	0.0105 (-15.46%)	430.53 (15.70%)

Table 16: Disclosure risks and forecast accuracy for the unprotected and protected M4 rate data sets.

8.3. A Tale of Two Time Series: Simulated Example Feature Distributions

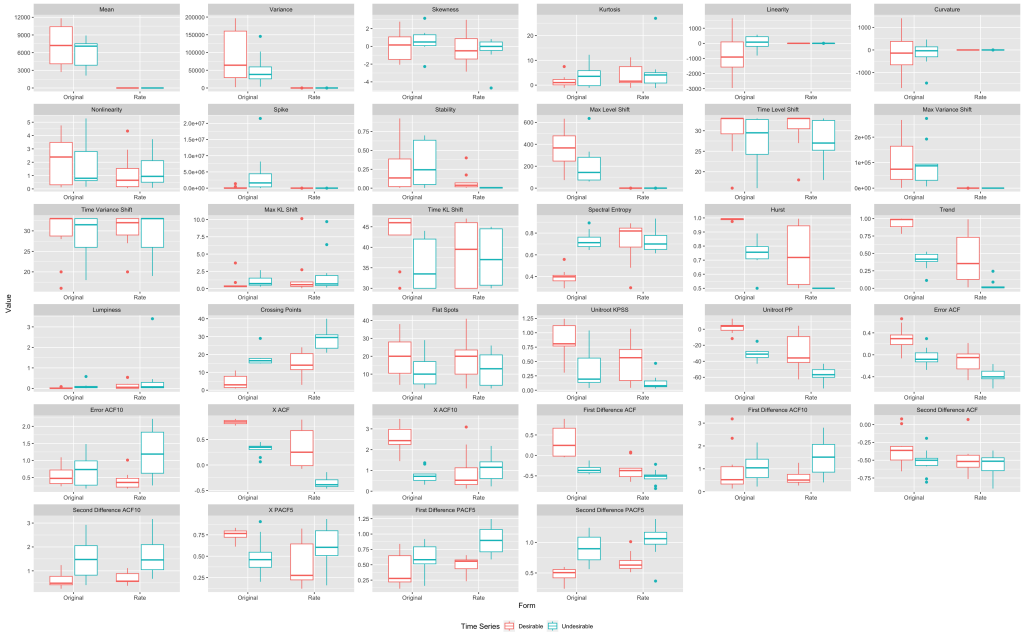

Figure 13: Feature distributions for original and rate versions of simulated series with desirable and undesirable features.

8.4. Comparing Privacy and Judgmentally Adjusted Forecasts

Similar to Fildes et al. (2009) and Khosrowabadi et al. (2022), we compare the percentage of forecast adjustments that improved accuracy across adjustment direction, magnitude, and the coefficient of variation of the unprotected time series. We compare these results for the adjusted forecasts using the bounded k-nTS+ ($k = 3, M = 1.5$) protected original data set, and the k-nTS+ ($k = 3$) protected rate data set, as these were the top performing privacy methods for these data sets.

Let $\hat{A}^{u}_{j,T+1}$ and $\hat{A}^{p}_{j,T+1}$ denote the forecasts for time $T+1$ based on the unprotected and protected versions of the jth time series. To compute adjustment magnitude, we normalize the absolute difference between the adjusted and original forecasts using the mean of the unprotected series,

$$Magnitude_j = \frac{\left|\hat{A}^{u}_{j,T+1} - \hat{A}^{p}_{j,T+1}\right|}{\bar{x}_j} \tag{16}$$

where the u and p superscripts denote a forecast based using the unprotected and protected series, respectively. Using the approach of Khosrowabadi et al. (2022), we bin the magnitudes into high ($>$ 0.75 quantile), low ($<$ 0.25 quantile) and medium ($\geq$ 0.25 quantile and $\leq$ 0.75 quantile) intervals.

We also compute the average relative absolute error (AvgRelAE, see Davydenko & Fildes (2013)) to compare the relative accuracy of the adjusted and original forecasts. The AvgRelAE of the adjusted forecasts is computed as

$$AvgRelAE \;=\; exp\left[\frac{1}{J}\sum_{j\,=\,1}^{J} log\,\frac{AE^{p}_{j}}{AE^{o}_{j}}\right], \tag{17}$$

where AE^{p}_{j} and AE^{o}_{j} are the absolute forecast error for the protected and unprotected versions of series j, respectively. An AvgRelAE less than one indicates an average improvement in accuracy and an AvgRelAE greater than one indicates an average reduction in accuracy[9]. This metric is also easily interpretable since the average percentage increase in forecast error can be calculated as $(AvgRelAE - 1) \times 100$. We remove the forecasts with the 5% smallest and 5% largest ratios (AE^{p}_{j}/AE^{o}_{j}) to prevent extreme outliers from affecting AvgRelAE (Davydenko & Fildes, 2013).

		Direction		
		Positive	*Negative*	**Total**
Magnitude	*Large*	4.44 (14.1%)	3.63 (14.0%)	4.05 (14.1%)
	Medium	3.06 (21.7%)	3.14 (17.6%)	3.10 (19.7%)
	Small	1.69 (32.9%)	1.57 (34.2%)	1.63 (33.5%)
	Total	2.93 (22.3%)	2.71 (21.1%)	2.82 (21.7%)

Table 17: AvgRelAE (and percentage of adjustments that improved accuracy) by adjusted magnitude and direction for the original M3 data.

Using the k-nTS+ ($k = 3, M = 1.5$) protected original data, we find that only 21.7% of the adjusted forecasts improved forecast accuracy (lower absolute error), which is significantly less than the reported 49.9% of judgmentally adjusted forecasts that improved accuracy in Khosrowabadi et al. (2022). Table 17 breaks down the results by adjustment magnitude and direction and displays the AvgRelAE and percentage of adjusted forecasts (in parentheses) that improved accuracy. The results show that most privacy adjusted forecasts degraded accuracy and the AvgRelAE is greater than one in all cases. Also, our results are contrary to the findings in the judgmental literature which shows that large adjustments and negative adjustments improve forecast

[9]AvgRelAE can be generalized to accommodate multiple forecasts for each series. See Davydenko & Fildes (2013) for the AvgRelMAE.

accuracy. We find that small privacy adjustments improved (33.5% of cases) forecast accuracy more frequently than large privacy adjustments (14.1% of cases). Furthermore, positive adjustments improved (22.3% of cases) forecast accuracy slightly more often than negative adjustments (21.1% of cases), but were more damaging on average to forecast accuracy.

		Direction		
		Positive	*Negative*	**Total**
Magnitude	*Large*	1.22 (42.2%)	1.23 (42.7%)	1.22 (42.4%)
	Medium	1.11 (42.4%)	1.03 (48.9%)	1.07 (45.7%)
	Small	1.07 (50.8%)	1.19 (37.5%)	1.13 (44.1%)
	Total	1.13 (44.4%)	1.11 (44.5%)	1.12 (44.5%)

Table 18: AvgRelAE (and percentage of adjustments that improved accuracy) by adjusted magnitude and direction for the M3 rate data.

Table 18 contains the same information as Table 17 but for the M3 rate data. The AvgRelAE and percentage of adjusted forecasts that improved accuracy are much lower and higher, respectively, than for the original M3 data, which is consistent with the forecast accuracy results in Section 5.4. About 44.5% of adjusted rate forecasts improved accuracy, and the average accuracy of an adjusted forecast was only 12% worse than on the unprotected data, where small positive and medium negative adjustments improved accuracy most often. However, on average, the adjusted rate forecasts reduced accuracy regardless of direction and magnitude.

Time series features could also affect whether a privacy adjusted forecast improves or degrades accuracy. For example, a negative slope in a series could cause positive adjustments to have a dampening effect on forecasts, and negative adjustments to overestimate the impact of the trend (Hyndman & Athanasopoulos, 2021). To investigate this further, Tables 19 and 20 display the AvgRelAE and the percentage of adjustments for time series with positive slopes vs. negative slopes based on the adjustment direction. To measure the slope, we calculate the slope coefficient of a simple linear regression that regresses the time series values on a continuous time variable. Our results show that the forecast accuracy for original time series (70% of which have positive slopes) with positive (negative) slopes improved most often when adjustments were made in the positive (negative) direction. However, this pattern reverses for the rate series (67% of which had negative slopes), with negative adjustments for series with positive slopes improving accuracy 13% more often than positive adjustments, and positive adjustments improving accuracy 6% more often than negative adjustments for series with negative slopes.

		Direction		
		Positive	*Negative*	**Total**
Slope	*Positive*	2.99 (21.9%)	2.88 (18.8%)	2.93 (20.3%)
	Negative	2.83 (23.1%)	2.24 (28.1%)	2.58 (25.1%)
	Total	2.93 (22.3%)	2.71 (21.1%)	2.82 (21.7%)

Table 19: AvgRelAE (and the percentage of adjustments that improved accuracy) by slope and direction for the original M3 data.

		Direction		
		Positive	*Negative*	**Total**
Slope	*Positive*	1.19 (37.4%)	1.04 (50.9%)	1.11 (44.0%)
	Negative	1.10 (47.9%)	1.15 (41.4%)	1.12 (44.7%)
	Total	1.13 (44.4%)	1.11 (44.5%)	1.12 (44.5%)

Table 20: AvgRelAE (and the percentage of adjustments that improved accuracy) by slope and direction for the M3 rate data.

Tables 21 and 22 measure the percentage of adjustments that improved accuracy and AvgRelAE for the original and rate M3 data sets categorized by the coefficient of variation of the unprotected series and whether k-nTS+ privacy protection increased, decreased, or maintained (within five percent) this coefficient of variation. We measure the coefficient of variation using the unprotected time series values since there was only one forecast horizon. We bin the coefficients of variation into high (> 0.75 quantile), low (< 0.25 quantile) and medium (≥ 0.25 quantile and ≤ 0.75 quantile). For the original data, we find that none of the coefficient of variation categories improve forecast accuracy compared to the unprotected data. Forecast accuracy degraded the most (least) when k-nTS+ was applied to time series with small (large) coefficients of variation. For the rate data, average degradation in forecast accuracy is similar for series with different coefficients of variation. There are some slight differences in forecast accuracy depending on the change in coefficient of variation after protection. For example, time series with small and medium coefficients of variation that maintained their coefficients had 17% better and 2% worse forecast accuracy on average, but only 0.2% and 4.9% of time series fall in these categories, respectively.

		Change in Coefficient of Variation			
		Decreased	*Maintained* (+/- 5%)	*Increased*	**Total**
Original Coefficient of Variation	*Large*	2.17 (26.7%)	1.85 (30.4%)	2.01 (30.7%)	2.04 (28.7%)
	Medium	2.40 (24.7%)	2.73 (23.2%)	3.10 (19.8%)	2.86 (21.6%)
	Small	4.86 (10.9%)	3.61 (16.1%)	3.73 (15.3%)	3.79 (15.1%)
	Total	2.41 (24.7%)	3.17 (19.3%)	2.50 (24.6%)	2.82 (21.7%)

Table 21: AvgRelAE (and percentage of adjustments that improved accuracy) by coefficient of variation of the unprotected series and the change in coefficient of variation in the protected series for the original M3 data.

		Change in Coefficient of Variation			
		Decreased	*Maintained* (+/- 5%)	*Increased*	**Total**
Original Coefficient of Variation	*Large*	1.11 (45.6%)	1.23 (38.5%)	1.12 (44.8%)	1.12 (45.4%)
	Medium	1.13 (43.9%)	1.02 (50.2%)	1.15 (42.8%)	1.13 (44.1%)
	Small	1.11 (44.3%)	0.83 (61.3%)	1.17 (43.1%)	1.11 (44.3%)
	Total	1.12 (44.5%)	1.02 (50.1%)	1.15 (43.0%)	1.12 (44.5%)

Table 22: AvgRelAE (and percentage of adjustments that improved accuracy) by coefficient of variation of the unprotected series and the change in coefficient of variation in the protected series for the M3 rate data.

Overall, our empirical results show that privacy adjustments affect forecast accuracy differently than judgmental adjustments. Specifically, for the original data, we found that privacy

adjustments had better forecast accuracy when the adjustments were small and negative, or when the coefficient of variation of the unprotected series was large. For the rate data, forecast accuracy under privacy adjustments was better when adjustments were small and positive or medium and negative, but the coefficient of variation of the unprotected series had little effect. On average for both types of data, forecast accuracy worsened for nearly every combination of magnitude, direction, and coefficient of variation. This is not surprising since a major motivation of judgmental adjustments is to improve forecast accuracy (Fildes et al., 2019) and judgmental adjustments have been shown to improve forecast accuracy by 5-10% on average (Davydenko & Fildes, 2013; Khosrowabadi et al., 2022). For our application, privacy adjustments blur the data for privacy reasons and are expected to reduce forecast accuracy. The secondary goal of our proposed privacy method is to maintain forecast accuracy, which the top performing method (k-nTS+ ($k = 3, M = 1.5$) swapping) did for only a subset of the original M3 time series with a 14.8% reduction in average forecast accuracy. Transforming the original time series into rates, however, allows all privacy methods to offer an improved trade-off between privacy and forecast accuracy, with k-nTS+ ($k = 3$) offering the best performance with a reduction in average forecast accuracy of 3.6% on the rate scale, and 19% when the rate forecasts are transformed to the original time series scale by the data owner.

8.5. Implementation Details

8.5.1. Preprocessing

The original time series were pre-processed by first ensuring that all time series values are greater than zero (this step actually has an effect on the data sets protected using additive noise and differential privacy) and then taking the log of the series. For the VAR model, we also took the first difference of the log series. For the global models (RNN and LGBM), we divided each series by its respective mean prior to taking the log.

The rate series were created from the original time series after applying a log transformation. We performed no additional pre-processing on the rate series except taking the first difference prior to forecasting with the VAR models and mean-scaling the rate series prior to forecasting with the global models.

8.5.2. Model Implementations

All local models (SES, DES, TES, and Auto-ARIMA) were implemented using the *sktime* forecasting module[10]. We used additive trend and seasonality components for DES and TES. We set Auto-ARIMA to apply a seasonal model when appropriate (SARIMA) with a maximum of fifteen iterations.

The VAR model from the *statsmodels* (Seabold & Perktold, 2010) python module was applied to subsets of approximately five time series at a time, ensuring that each subset consisted only of time series with the same length.

The LGBM and RNN models were implemented using the *darts* module (Herzen et al., 2022). For both models, we reserved a validation time period immediately prior to the desired forecast horizon. We used Bayesian optimization (Nogueira, 2014) to optimize the hyperparameters of each model to minimize the absolute forecast error (L1 loss) in the validation time period. We retrained each model using the optimized hyperparameters and the full training data (including the validation period) prior to generating forecasts for the desired forecast horizon.

[10]Link to *sktime* documentation.

The Bayesian optimizer was initialized for the LGBM models using five starting points and run for 25 iterations. For the RNN the optimizer was initialized using five starting points and run for 15 iterations. We limited the RNN to the last ten input-output window samples from each time series for computational efficiency and trained ten RNN models taking the median of the forecasts as the final forecast (Hewamalage et al., 2022). The hyperparameters for the LGBM and RNN models applied to the M3 data and their corresponding ranges provided to the optimizer are shown in Tables 9 and 10. The parameters used were identical for the M4 data, with the exception of the input window lengths which were as follows. For LGBM, 25 on Monthly and Hourly data, 10 for all other frequencies. For RNN, input window lengths were 25 for Monthly and Hourly data and 9 for all other frequencies, and training window lengths were 30 for Monthly and Hourly data and 10 for all other frequencies.

Hyperparameter	**Range**
Input Window Length	Monthly: 25, All other frequencies: 11
Learning Rate	(0.01, 0.1)
Number of Boost Rounds	(50, 1000)
Number of Leaves	(2, 100)
Bagging Frequency	(1, 5)
Bagging Fraction	(0.05, 1.00)
L2 Regularization Parameter	(0, 0.5)
Minimum Observations in Leaf	(3, 60)

Table 23: Hyperparameter ranges used for training LGBM models when forecasting for M3 data.

Hyperparameter	**Range**
Input Window Length	Monthly: 25; All other frequencies: 11
Training Length	Monthly: 30; All other frequencies: 13
Learning Rate	(0.001, 0.1)
Weight Decay	(0.0001, 0.0008)
Number of Layers	(1, 2)
Hidden Dimension	(20, 50)
Batch Size	(200, 700)
Number of Epochs	(3, 30)
Dropout Rate	(0.1, 0.5)

Table 24: Hyperparameter ranges used for training RNN models when forecasting for M3 data.

8.6. Mathematical Details of Identification Disclosure

To perform identification disclosure, we assume a third party possesses some unprotected data on a unit of interest in the protected data set. Denote this unprotected data $\mathbf{u}_i = (ID_i, u_i)$, which contains a direct identifier ID_i (*e.g.*, the identity of time series i) and unprotected data $u_i = (A_{i,t'}, \ldots, A_{i,t'+E})$ which contains a sequence of values that are components of the unprotected 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 $\mathbf{u}_i$ are components of the unprotected version of the 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,

$$\hat{M}_i = argmax_j \; P\left(M_i = j \,|\, u_i\right), \tag{18}$$

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

$$s\left(u_i, \widetilde{x}_j\right) = 1/\|u_i - x_j\|_2,\ \ j = 1, \ldots, J. \tag{19}$$

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

$$P\left(M_i = j \,|\, u_i\right) = \frac{s\left(u_i, \widetilde{x}_j\right)}{\sum_{j=1}^{J} s\left(u_i, \widetilde{x}_j\right)}, \tag{20}$$

and predicts $\hat{M}_i$ as in Equation 18.

To estimate the risk of identification disclosure, we perform simulations in which we sample E sequential values from each unprotected time series x_j, and we measure the average proportion of series which are identified. The sampled values are denoted $U = [\mathbf{u}_1, \ldots, \mathbf{u}_J]^T$. Each of the vectors $\mathbf{u}_i$ corresponds to one of the J unprotected time series, and we compute r_j conditional on the sampled u_i from series j. We repeat this simulation S times to obtain $\mathbf{U} = \{U_1, \ldots, U_S\}$, and compute the average proportion of correctly identified time series across all external data samples and unprotected time series,

$$\bar{P} = \frac{1}{J \times S} \sum_{s=1}^{S} \sum_{i=1}^{J} I\left(\hat{M}_i^s = j^*\right). \tag{21}$$

These simulations assume that the third party in possession of U predicts the match for each vector $\mathbf{u}_i$ 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 $\mathbf{u}_i$ and each attempts identification risk as described above. Overall, multiple vectors may be matched to the same protected time series.

8.7. Equations for Important Time Series Features from the Literature

8.7.1. Spectral Entropy

Suppose x_j is 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, 2013), 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

$$Spectral\ Entropy = -\int_{-\pi}^{\pi} \widehat{f_x}(\lambda)\, log \widehat{f_x}(\lambda)\, d\lambda, \tag{22}$$

where the maximum entropy occurs when $\Lambda \sim U(-\pi, \pi)$. In practice, estimates of the spectral entropy range from zero to one, where high values represent a low signal-to-noise ratio, indicating that x_j is difficult to forecast (Kang et al., 2017).

8.7.2. Hurst

Next, we consider a self-similarity feature quantified using the Hurst parameter (Wang et al., 2006), which measures the long-range dependence of a time series. Of the features examined by Spiliotis et al. (2020), changes in *Hurst* had the largest effect on forecast accuracy. We use the definition of self-similarity of a time series described by Willinger et al. (1998). Suppose that x_j is the increment process of y_j, *i.e.*, $x_{j,t} = y_{j,t+1} - y_{j,t}$. An aggregate sequence, denoted $x_j^{(m)}$, is created by averaging x_j over non-overlapping blocks of size m, where

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

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

$$x_j = m^{1-H} x_j^{(m)} \tag{23}$$

for all integers m. We use 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 $m \to \infty$ (Rose, 1996). 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; Yang & Hyndman, 2023) where

$$Hurst = H = d + 0.5. \tag{24}$$

Estimates of H range from zero to one, where $H = 0.5$ corresponds to a random walk (Sobolev, 2017), $H < 0.5$ corresponds to an anti-persistent or mean-reverting series, and $H > 0.5$ corresponds to a persistent time series that is more likely to maintain its current trend.

8.7.3. Skewness

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

$$Skewness = \frac{1}{n\sigma^3} \sum_{t=1}^{n} \left(x_j - \bar{x}_j\right)^3. \tag{25}$$

8.7.4. Kurtosis

We measure *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,

$$Kurtosis = \frac{1}{n\sigma^4} \sum_{t=1}^{n} \left(x_j - \bar{x}_j\right)^4 - 3, \tag{26}$$

where 3 is the *Kurtosis* of the standard normal distribution.

8.7.5. Error Autocorrelation Function (Error ACF)

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. (2023) to obtain

$$x_j = b_j + s_{1,j} + \ldots + s_{M,j} + e_t,$$

where b_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. (2020):

$$Error\ ACF = \frac{\sum_{t=2}^{T} \left(e_{j,t} - \bar{e}\right)\left(e_{j,t-1} - \bar{e}\right)}{\sum_{t=1}^{T} \left(e_{j,t} - \bar{e}\right)^2}, \tag{27}$$

which is a measure of the predictability of a time series after the trend and seasonality have been accounted for (Kang et al., 2017).

8.7.6. Trend and Seasonality

We also compute the strength of trend (*Trend*) and strength of the ith seasonal component ($Seasonality_i$) as follows,

$$Trend = 1 - \frac{Var\left(e_j\right)}{Var\left(f_j + e_j\right)}, \tag{28}$$

and

$$Seasonality_i = 1 - \frac{Var\left(e_j\right)}{Var\left(s_{i,j} + e_j\right)}. \tag{29}$$

In practice, the values of *Trend* and $Seasonality_i$ range from 0 to 1 (Yang & Hyndman, 2023).

8.7.7. Mean and Variance

The next two features are the *Mean* and *Variance*, also used by Bandara et al. (2020) to cluster similar time series for forecasting, which are written as follows,

$$Mean = \frac{1}{T} \sum_{t=1}^{T} x_{j,t}, \tag{30}$$

and

$$Variance = \frac{1}{T-1}\sum_{t=1}^{T}\left(x_{j,t} - \bar{x}_j\right)^2 . \tag{31}$$

We also included many other features from the *tsfeatures* package in R (Hyndman et al., 2023). We refer the reader to Yang & Hyndman (2023) for explanation of these features.