# Sparse Decomposition for Time Series Forecasting and Anomaly Detection*

Sunav Choudhary† Gaurush Hiranandani‡ Shiv Kumar Saini§

**Abstract**

Anomaly detection and forecasting are two fundamental problems in time series analysis that are relevant to a wide range of academic and industrial disciplines. Although these problems have been investigated in the literature previously, the assumptions therein are too restrictive for autonomous analysis. Common examples of limiting assumptions include perfect knowledge about the time series seasonality and/or presence of anomaly (spikes and level changes) free time windows. Current practice is to manually input this knowledge into anomaly detection and forecasting systems which negate any possibility of autonomous analysis. This paper relaxes these assumptions by jointly estimating the latent components (*viz.* seasonality, level changes, and spikes) in the observed time series without assuming the availability of anomaly-free time windows. The novel and flexible two stage approach proposed herein is based on (a) sparse modeling of the different latent components of the time series and (b) ARMA modeling for fitting the error. The approach leads to a solution for anomaly detection with control over type-I errors. Further, by design, the method is robust against anomalies in the observation window when it is used to solve the forecasting problem by extrapolation. Experiments are conducted with both synthetic and real datasets to demonstrate the efficacy of the proposed method. We compare our approach to various popular baselines. The presented approach outperforms baseline algorithms for anomaly detection in all our experiments and performs favorably for the forecasting task.

***Index terms***— Time series analysis, sparse decomposition, anomaly detection, forecasting

## 1 Introduction

Two problems of central importance in time series analysis are 'anomaly detection' and 'forecasting'. There is extensive academic literature from various disciplines that attempts to address these problems (see [13] and references therein). Further, several big technology companies, *viz.* Twitter, Adobe, Microsoft, Yahoo, *etc.* offer software solutions for model based anomaly detection and forecasting in time series data. What is often missing, from both the academic approaches and the commercial systems, is the ability to perform anomaly detection and forecasting robustly without human intervention. This is a significant limitation when considering datasets that are too large for manual processing.

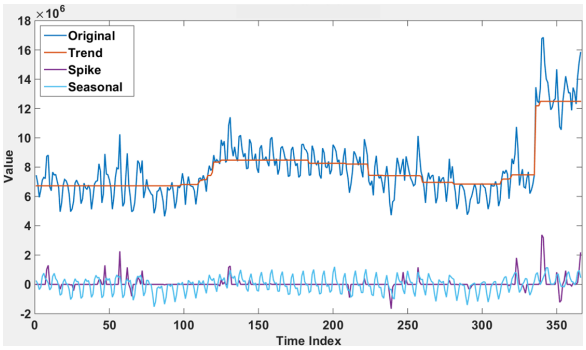


Figure 1: Daily website visits over a one-year period: Original time series and latent level changes, spikes and seasonality

We illustrate, with an example, the challenges involved in automating any anomaly detection and/or forecasting algorithm. Figure 1 plots the number of daily visits on a digital media publisher's website over a period of one year. This time series exhibits a complicated mix of seasonal variations, anomalous spikes in visits on some days, and occasional but sudden changes in the average number/level of visits as shown. Any automated algorithm for anomaly detection and/or forecasting needs to jointly estimate these latent seasonal patterns, spikes and level changes (henceforth simply referred to as 'latent components') with no human inputs. Both academic approaches and commercial systems tend to require significant hand tuning to achieve good results on time series with complicated effects like those in Figure 1. Such hand tuning can take the form of known seasonality

*In the spirit of reproducible research, all datasets and implementations in the paper are available upon request.

†Adobe Research, schoudha@adobe.com

‡UIUC, gaurush2@illinois.edu. Work done when the author was affiliated with Adobe Research.

§Adobe Research, shsaini@adobe.com

with one or two seasonal frequencies, knowledge about the number and position of anomalies [34], and/or availability of anomaly-free time windows for accurately estimating seasonal effects.

In this paper, we develop a decomposition method to separate a time series into its latent components that are easy to interpret in practice. The method relies on sparse modeling [12] of these latent components that are assumed to constitute the observed time series via linear superposition. The decomposition serves as a precursor to anomaly detection and forecasting, among other tasks in time series analysis. The sparse component estimation is cast as a convex optimization problem and the noise component is modeled as an ARMA process. We demonstrate the efficacy of this method by experimentation with synthetic and real datasets and conducting a comparative study *w.r.t.* popular freely available/open source, state-of-the-art baseline algorithms for anomaly detection (based on Twitter's Anomaly Detection [34]) and forecasting (based on `ETS` [17] and `ARIMA` [3, 17]). The real datasets used in the experiments are `NAB` [22], `M3` [27] and collected web metrics data from different websites. The experiments show that very straightforward rules for anomaly detection and forecasting, when applied to the decomposed time series components, result in superior performance *w.r.t.* widely used competing algorithms.

Overall, this paper makes the following contributions.

1. The proposed algorithm is novel in its use of sparse decomposition of the time series to aid forecasting and anomaly detection. The sparse and ARMA noise models each address the other's shortcomings, *viz.* sparsity modeling is able to easily remove seasonal effects that are problematic for ARMA models, and ARMA fitting is able to easily extract an uncorrelated noise process that is difficult to guarantee after sparse decomposition.
2. Our algorithm eliminates many restrictive assumptions found in popular anomaly detection methods enabling it to be completely automated over large datasets. In particular, our algorithm does not assume knowledge of seasonality or maximum number of anomalies or presence of anomaly-free windows. The sparsity assumptions in the algorithm are comparatively much weaker.
3. The sparse decomposition step implicitly makes the forecasting task robust to the presence of spikes and level changes in the observation window. In contrast, the modeling assumption of approximate stationarity in literature breaks easily in the presence of abrupt spikes and sudden level changes and hence leads to poor forecasts.
4. The algorithm is validated on real datasets including a collected web metrics dataset that contains time series with unknown seasonality, unexpected spikes and sudden level changes. Our approach achieves much better type-II error for anomaly detection in all our experiments without sacrificing type-I error or needing any side information about anomalies. Furthermore, our approach achieves the best mean squared forecasting error on a substantial fraction of both the M3 forecasting competition dataset [23, 27] and the collected web metrics dataset.

The rest of the paper is organized as follows. §2 describes the modeling assumptions and §3 presents the algorithmic details. §4 reviews prior art. §5 describes the evaluation datasets and experimental results. §6 concludes the paper.

*Notation:* Column vectors are in lowercase boldface alphabets. $\mathbf{0}$, $\mathbf{1}$ and $\mathbf{I}$ are respectively the all zero, all one and identity matrix/vector. $\|\cdot\|_1$ and $\|\cdot\|_2$ are $\ell_1$-norm and $\ell_2$-norm respectively.

## 2 Modeling Assumptions

While the general problem of sparse estimation from over-complete representation basis is NP-hard by reduction to the subset selection problem [8, 24], certain instances can be provably solved in time that is polynomial in the ambient dimension of the signal [5, 12]. We are guided by the key nature of these instances which are primarily two-fold:

1. The over-complete representation basis satisfies an incoherence property [12], *i.e.* the pairwise correlations between different atoms of the over-complete basis are small, and
2. The sparsity of the unknown signal is small *w.r.t.* its ambient dimension.

**2.1 Observation Model** Let $\boldsymbol{y}$ denote a time series. Notionally, we assume that $\boldsymbol{y} = \boldsymbol{s} + \boldsymbol{t} + \boldsymbol{d} + \boldsymbol{e}$ is a linear superposition of four simpler time series components illustrated in Figure 1, *viz.* seasonality $\boldsymbol{s}$, level $\boldsymbol{t}$, spikes $\boldsymbol{d}$, and noise $\boldsymbol{e}$. An analytic toy example is obtained by setting $\boldsymbol{s}(n) = \cos(2\pi n) + 3\sin(\pi n)$, $\boldsymbol{s}$ as a realization of a white noise process, $\boldsymbol{t} = 3\cdot\mathbf{1}(n > 5) - 2\cdot\mathbf{1}(n > 10)$, and $\boldsymbol{d}(n) = \delta(n-3) + 2\delta(n-7)$, where $\delta(n)$ is the Kronecker-delta function.

**2.2 Modeling for Sparse Decomposition** Given $\boldsymbol{y}$, we want to estimate the $\boldsymbol{s}$, $\boldsymbol{t}$, $\boldsymbol{d}$ and $\boldsymbol{e}$ series components. This will enable us to isolate the spikes in $\boldsymbol{y}$ into the $\boldsymbol{d}$ series by removing the interference due to $\boldsymbol{s}$, $\boldsymbol{t}$ and $\boldsymbol{e}$ components. Without any further assumptions, this decomposition problem is under-determined by a factor of four and is therefore unidentifiable. The toy

example in §2.1 suggests that $\boldsymbol{s}$, $\boldsymbol{t}$ and $\boldsymbol{d}$ look geometrically different. We can model these components as sparsely representable *w.r.t.* interpretable and pairwise incoherent bases [5] in a manner that preserves their respective interpretations:

1. The seasonal component $\boldsymbol{s}$ has a sparse representation in the discrete Fourier domain. By design, this basis is well suited to represent periodic signals.
2. The level component $\boldsymbol{t}$ is an infrequently changing piecewise constant. Thus, $\boldsymbol{t}(k+1) - \boldsymbol{t}(k)$ is zero for most values of $k$. This assumption allows us to capture shifts in the level.
3. The spikes occur infrequently, and thus $\boldsymbol{d}$ is sparse in the time domain.

While the piecewise constant assumption on $\boldsymbol{t}$ might seem restrictive, it strikes a good balance between simplicity of the model and over fitting to the data. From a theoretical standpoint, piecewise constant functions form a complete basis so there is no loss of generality in this assumption. From practical considerations, we are usually interested in slowly varying levels which are captured well by a sequence of infrequent level shifts (or equivalently, piecewise constant signals).

**2.3 ARMA Modeling for Error** We explicitly model the error series $\boldsymbol{e}$ as a $\{p, q\}^{\text{th}}$ order ARMA process represented by

$$\boldsymbol{e} = \mathscr{L}(\boldsymbol{\phi})\boldsymbol{e} + \mathscr{L}(\boldsymbol{\theta})\boldsymbol{\eta} + \boldsymbol{\eta} \tag{2.1}$$

where $\boldsymbol{\phi} \in (-1, 1)^p$, $\boldsymbol{\theta} \in (-1, 1)^q$, $\boldsymbol{\eta} \in \mathbb{R}^N$ is a zero mean white noise process, and $\mathscr{L}\colon \mathbb{R}^r \to \mathbb{R}^{N\times N}$ is the unique linear operator satisfying $\mathscr{L}(\boldsymbol{z}) \triangleq \sum_{k=1}^{r} \boldsymbol{z}(k)\boldsymbol{L}^k$ for every $\boldsymbol{z} \in \mathbb{R}^r$. Here, $\boldsymbol{L}$ denotes the unit lag operator in matrix form. The ARMA model helps to identify statistically significant spikes in $\boldsymbol{d}$ and extrapolate the error series $\boldsymbol{e}$ for forecasting.

## 3 Decomposition, Detection and Forecasting

Sparse signal recovery literature [5] suggests that appropriately formulated convex optimization based estimators could be used for decomposing $\boldsymbol{y}$ into the sparse components $\boldsymbol{s}$, $\boldsymbol{t}$ and $\boldsymbol{d}$. Once these components are estimated, we can

1. design a detection rule for statistically significant anomalies, and
2. write down forecasting rules.

These steps are described below. We define some notational shorthand for clarity of description. We assume that each time series is being considered over a finite sized fixed window of size $N$, *i.e.* $\boldsymbol{y}, \boldsymbol{s}, \boldsymbol{t}, \boldsymbol{d}, \boldsymbol{e} \in \mathbb{R}^N$. Let $\boldsymbol{F}$ denote the $N \times N$ discrete Fourier transform (DFT) matrix and let $\Delta\colon \mathbb{R}^N \to \mathbb{R}^{N-1}$ denote the first difference operator, *i.e.* $k^{\text{th}}$ element of $\Delta\boldsymbol{t}$ would be $\boldsymbol{t}(k+1) - \boldsymbol{t}(k)$.

**3.1 Decomposition via Convex Optimization** We consider the convex optimization problem

$$(\mathrm{P}_1) \qquad \begin{aligned} \underset{\boldsymbol{s},\boldsymbol{t},\boldsymbol{d},\boldsymbol{e}}{\text{minimize}} \quad & \|\boldsymbol{F}\boldsymbol{s}\|_1 + w_1\|\Delta\boldsymbol{t}\|_1 + w_2\|\boldsymbol{d}\|_1 \\ \text{subject to} \quad & \boldsymbol{y} = \boldsymbol{s} + \boldsymbol{t} + \boldsymbol{d} + \boldsymbol{e}, \\ & \|\boldsymbol{e}\|_2 \le \rho, \end{aligned}$$

where $w_1$, $w_2$ and $\rho$ are hyper-parameters and we let $\left(\hat{\boldsymbol{s}}, \hat{\boldsymbol{t}}, \hat{\boldsymbol{d}}, \hat{\boldsymbol{e}}\right)$ denote the solution obtained. We use $\hat{\boldsymbol{s}}$, $\hat{\boldsymbol{t}}$ and $\hat{\boldsymbol{d}}$ as the estimates for the true values of $\boldsymbol{s}$, $\boldsymbol{t}$ and $\boldsymbol{d}$.

Problem $(\mathrm{P}_1)$ is motivated from sparse recovery literature [5, 12]. Its objective function promotes sparsity on the different components of $\boldsymbol{y}$ using the $\ell_1$-norm according to the sparsity models in §2.2. For example, the frequency domain sparsity of the seasonal component $\boldsymbol{s}$ is encouraged by the $\|\boldsymbol{F}\boldsymbol{s}\|_1$ function in the minimization objective, where $\boldsymbol{F}\boldsymbol{s}$ is the DFT of $\boldsymbol{s}$. Likewise, the sparsity of the $\Delta\boldsymbol{t}$ vector is captured by the $\|\Delta\boldsymbol{t}\|_1$ term in the objective function. We assume the noise and the model fitting errors to be captured in $\boldsymbol{e}$ and we control its effect by upper bounding its energy with the constraint $\|\boldsymbol{e}\|_2^2 \le \rho^2$.

Problem $(\mathrm{P}_1)$ is a convex optimization problem and is equivalent to a second-order conic program (SOCP) [4]. Generic SOCP solvers like ECOS [11] or SCS [25] can solve this problem efficiently in theory and reasonably well for small to medium-sized instances (less than 1100 time points) in practice. Using SCS with DFT specific optimizations, the average CPU time per time series was under 5 seconds on the tested datasets. For longer time series, a more efficient implementation is needed that exploits special structures in the objective function. This is a topic of ongoing research.

**3.2 Statistical Testing for Anomalous Spikes** We use the following steps to test for significant anomalies.

1. Estimate the parameters $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ of the ARMA process
$$\hat{\boldsymbol{e}} = \mathscr{L}(\boldsymbol{\phi})\hat{\boldsymbol{e}} + \mathscr{L}(\boldsymbol{\theta})\boldsymbol{\eta} + \boldsymbol{\eta}. \tag{3.2}$$
2. Calculate the unconditional variance of $\hat{\boldsymbol{e}}$ by the formula
$$\boldsymbol{\Omega} = \boldsymbol{W}\,\mathrm{Var}[\boldsymbol{\eta}]\boldsymbol{W}^{\mathrm{T}}, \tag{3.3}$$
where $\boldsymbol{W} \triangleq [\mathbf{I} - \mathscr{L}(\boldsymbol{\phi})]^{-1}[\mathbf{I} + \mathscr{L}(\boldsymbol{\theta})]$ and $(\cdot)^{\mathrm{T}}$ denotes matrix transpose operator.
3. The anomalous spikes are at the indices $n$ for which one of $\hat{\boldsymbol{d}}(n) \le z_{\alpha/2}\sqrt{\boldsymbol{\Omega}(n,n)}$ or $\hat{\boldsymbol{d}}(n) \ge z_{(1-\alpha)/2}\sqrt{\boldsymbol{\Omega}(n,n)}$ is true. Here, $z_x$ is the $x^{\text{th}}$ percentile of a Standard Normal distribution.

This detection rule fits the estimated noise series $\hat{\boldsymbol{e}}$ to an ARMA model to determine threshold levels for declaring spikes in $\hat{\boldsymbol{d}}$ as significant. This upper bounds the type-I error (also known as the false alarm rate) by $\alpha\%$ by declaring spikes close to the noise floor as non-anomalous. An implicit assumption in the detection rule is that the distribution of $\hat{\boldsymbol{e}}$ approximates that of $\boldsymbol{e}$ and that $\hat{\boldsymbol{e}}$ is stationary and generated by a zero mean white Gaussian noise process $\boldsymbol{\eta}$ according to (2.1). We note that a more sophisticated algorithm than an ARMA model might give better statistical performance. However, the goal of this paper is to highlight the utility of the sparse decomposition approach by demonstrating substantial performance gains even with simple models for downstream tasks.

**3.3 Forecasting Rules** We forecast each of the four components $\boldsymbol{s}$, $\boldsymbol{t}$, $\boldsymbol{d}$ and $\boldsymbol{e}$ individually and get the forecast for $\boldsymbol{y}$ using the relation $\boldsymbol{y} = \boldsymbol{s} + \boldsymbol{t} + \boldsymbol{d} + \boldsymbol{e}$. The following rules are used:

1. The predicted future values of spikes and level change is set to zero, *i.e.* the level stays constant. This rule is motivated by the assumption that spikes and level changes are infrequent unpredictable anomalies and are unlikely to occur in the forecast period.
2. The predicted future values for the error series are given by extrapolating the fitted ARMA model. For this rule, we have implicitly assumed stationarity of the ARMA model parameters for $\boldsymbol{e}$ and approximated them by the corresponding parameters for $\hat{\boldsymbol{e}}$ (*viz.* $\hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{\eta}}$) that are estimated via (3.2).
3. The forecast for the seasonal component is a time domain extrapolation of the estimated seasonal frequencies in $\hat{\boldsymbol{s}}$, using the shift property of the DFT [26]. We have implicitly assumed the seasonal pattern to be unchanged within the forecast horizon.

As in §3.2 above, we acknowledge that a more sophisticated forecasting algorithm might give better statistical performance. Our goal in the paper is primarily to show the utility of the sparse decomposition approach by providing evidence of very good forecasting performance even with simple rules.

## 4 Related Work

The problem studied herein can be broadly associated with unsupervised anomaly detection and forecasting in time series and our solution approach draws heavily from the somewhat different field of sparse signal processing. We connect our approach to both these fields.

*Time Series Outlier and Change Point Detection:* Over the last few decades, a wide variety of statistical time series models have been proposed and explored in academic literature for outlier and change point detection [1, 9, 16] under a host of assumptions. A recent survey of techniques for detecting point outliers and change points is available in [13]. The broad approach consists of using variants of a cumulative sum [2] approach and/or a generalized likelihood ratio based test [14]. Since this family of approaches is sensitive to various effects like seasonality, the seasonal effects need to be filtered out for practical applicability. For example, [34] removes the seasonal effect using LOWESS smoothing [7] and then applies a Generalized Extreme Studentized Deviate (GESD) test [31] to bound the type-I error for outlier detection. A hybrid version of the GESD test for anomaly detection was developed more recently by Twitter [15]. The major drawback of such an approach is the difficulty of accurately estimating any seasonal effects beyond a single sinusoid, which translates to the need for externally specifying the seasonal effect for the approach to be broadly useful in practice. Although there have been various extensions to this broad approach to outlier and change point detection [6, 33] with consideration for non-stationarity [36] and structural shifts in econometrics [28–30], to the best of our knowledge, the drawback due to unknown seasonal effect was never addressed. The anomaly detection system at Yahoo [21] that leverages an ensemble of existing anomaly detection models to construct an anomaly filtering layer, also suffers from a similar drawback. The approach in this paper entirely eliminates this drawback.

*Time Series Forecasting:* Forecasting is a crucial aspect of time series analysis and two state-of-the-art algorithms in this field are ETS [17] and ARIMA [3, 17]. The `Theta` method [23] based on Simple Exponential Smoothing with drift [18] and the `B-J Automatic` method [23] based on ARIMA were among the best performers in the M3 forecasting competition [19]. Despite the wide applicability of ETS and ARIMA, they exhibit sensitivity to the presence of anomalies in the observation period which could lead to significantly inaccurate forecasts. In the econometric forecasting literature, this has been identified as a shock/structural break to a system [28, 29] and there have been some efforts towards filtering out the shock by sampling from recent history for better forecasts [30]. In experimental setups like collected web metrics (discussed in §5), the proportion of anomalies and change points tend to be moderately high, making it important for forecasting algorithms to be resilient. When shocks are level changes and anomalies are spikes, our approach circumvents the adverse effects of shocks easily, providing a great deal of robustness and performing at par with ETS and ARIMA in such settings.

*Sparse Signal Recovery:* Many of the drawbacks in prior literature on anomaly detection and forecasting could be addresses by jointly modeling and estimating the latent effects comprising of seasonality, spikes, and change points. Literature on sparse modeling and recovery algorithms [5, 8, 12] provides a convenient framework to jointly model such interpretable effects that admit sparse representations in known bases. Sparsity assumptions tend to be far less stringent than exact knowledge assumptions. As an example, while the exact seasonal effects may not be known in practice, it is broadly true that there are only a few seasonal effects of interest (may be daily, weekly, monthly and yearly). Somewhat surprisingly, this utility of sparse modeling to time series analysis has not been fully explored. One prior attempt in this direction consists of utilizing multi-scale wavelet representations for compactly representing a time series [20]. Since these wavelets are localized in frequency domain while spike anomalies are not, a wavelet representation cannot capture such non-localized anomalies succinctly. Further, the use of seasonal-ARIMA to extract the seasonality requires trial and error and is challenging to automate. Another work based on continuous wavelet transform with Gaussian kernels [32] can detect spike outliers by adjusting the bandwidth parameter of the kernel. However, this approach too falls short on detecting seasonality.

## 5 Experiments

**5.1 Datasets** This section describes the synthetic and real datasets used for experiments.

`SD` Dataset: This is a synthetic dataset consisting of 480 time series and it helps illustrate the salient properties of our algorithm. Each time series is a sum of four different components, *viz.* level, seasonal pattern, spikes, and error, all generated synthetically. We cover a wide variety of time series patterns - no seasonality to multiple seasonal patterns, no spike to 5% of observations as spikes, no change in level to multiple changes in level, and different sample sizes.

`SD-L` Dataset: This is derived from the `SD` dataset by ignoring the level changes in each time series in `SD`.

`NAB` Dataset: This is a dataset available from [22] and contains both real-world and synthetic time series. It contains 58 different scalar valued time series with labelled anomalies and is meant to provide a reference dataset for research into anomaly detection on streaming data. The real-world time series in this dataset come from different setups like Amazon Web Services (AWS) server metrics, Twitter posts volume, web advertisement click metrics, city traffic data, *etc.* These time series are very long with only a handful of data points tagged as anomalies. For example, the time series recording the Twitter volume of IBM contains about $1.6 \times 10^4$ data points, out of which only 2 data points are tagged as anomalous. Considering the limitations of the ECOS [11] and SCS [25] solvers, we have subset the time series to much smaller lengths while retaining approximately 95% of the labeled anomalies in the whole corpus.

`NAB-HR` Dataset: This dataset is created from the 8 distinct time series in the `NAB` dataset that are of hourly and half-hourly granularities. These time series tend to show daily as well as weekly seasonal patterns. We break these 8 time series into non-overlapping windows of approximately 1000 data points each resulting in a total of 37 hourly and half-hourly time series in this dataset.

`WM` Dataset: This is a real-world dataset with time series representing the number of daily visits on websites. Overall, the dataset comprises of 58 different websites spanning across a variety of industry verticals such as e-commerce, finance, publishing, automobile, *etc.* For each website, we choose two different window lengths, *viz.* 50 and 55 days and two distinct forecasting horizons of 7 and 10 days. This gives us 4 time series per website and a total of $58 \times 4 = 232$ web analytic time series. This dataset does not have labeled spikes or change points and hence we use it solely for studying forecasting performance.

A web metrics dataset is a very important test case for fully automated anomaly detection and forecasting since websites typically generate hundreds of thousands of time series metrics to measure traffic, consumer behavior, product and channel performance, *etc.* that could be tracked at multiple granularities, *viz.* hourly, daily, weekly, *etc.* based on the business application. It is typical for web analytic time series to show complex seasonality, unexpected spikes and sudden level changes.

`M3` Dataset: This dataset was used for the M3 forecasting competition [27] and contains 3003 time series. Each time series belongs to one of six different types, *viz.* micro, macro, industry, finance, demographic and other, and is recorded at one of four possible granularities, *viz.* yearly, quarterly, monthly and other. The forecast horizons in the M3 competition differed *w.r.t.* granularity and we choose to use the same specifications.

We use this dataset to study the robustness of forecasting when spikes are present in the observation window. Owing to absence of labelled anomalies, we introduce $x$% of spikes at random into each time series $\boldsymbol{y} \in$ `M3` as follows.

1. Select $x$% of the indices uniformly at random.
2. For each selected index $k$, change $\boldsymbol{y}(k)$ to a random draw from the uniform distribution over $[1.5\boldsymbol{y}(k), 3\boldsymbol{y}(k)]$.

We denote the modified dataset by $\texttt{M3-A}_x$ and we choose $x$ from $\{2.5, 5, 10\}$. Since $\texttt{M3-A}_x$ is randomly generated, forecasting performance subsequently presented in Table 3 are averaged over 10 different realizations of $\texttt{M3-A}_x$.

**5.2 Competing Algorithms** We denote our algorithm by `Alg-S` (for Sparse) and we have used a multilingual implementation since

1. the `CVXPY` package [10] (for disciplined convex programming in the Python programming language) is used to solve the SOCP, and
2. the `auto.arima` library [19] (for fitting ARMA models in the R programming language) is used to fit the ARMA model parameters.

We use $w_1 = 7$, $w_2 = 1$ and $\rho = 0.05 * \|\boldsymbol{y}\|_2$ across all our experiments. $w_1$ and $w_2$ were set by trial and error *w.r.t.* the performance of Problem ($\mathrm{P}_1$) on a small number of synthetically generated time series. $\rho$ was chosen to reflect that the energy of the error series need not be small in an absolute sense, but should be small relative to the energy of the observed time series $\boldsymbol{y}$.

We compare the performance of `Alg-S` against the following algorithms.

1. The Twitter algorithm [34] for anomaly detection, denoted by `Alg-T`. Since there is no forecasting component in this algorithm, we use it only for the anomaly detection part of the evaluation. We have used the implementation available at [35]. `Alg-T` is based on the Seasonal Hybrid Extreme Studentized Deviate (S-H-ESD) test that is built upon the GESD test. Like `Alg-S`, the algorithm decomposes the time series into various components, but it takes a different approach. The anomaly detection rule is created using statistical metrics like the median together with ESD. It should be noted that a limitation of `Alg-T` is that it requires one to specify the maximum number of anomalies in the input time series. Otherwise, `Alg-T` assumes that up to half of the data points could be anomalous. A second major limitation of `Alg-T` is that it needs the seasonal frequency of the input time series to be specified. This rules out multiple seasonal frequencies which is commonplace in time series data. For example, a daily time series is likely to follow a weekly and a monthly seasonal pattern. To make a meaningful comparison, we let `Alg-T` have the advantage of prior knowledge of the seasonal frequency in our experiments.
2. The ETS algorithm [17] designated as `Alg-ETS`. This is a popular, open source state-of-the-art algorithm for time series forecasting. It gives forecasts that are in the same ballpark as the best methods in the M3 competition. The `Theta` method [23], which was the top performer in the M3 competition, is a special case of Simple Exponential Smoothing with drift [18]. The implementation that we have used, available at [19], is a generalization of the these exponential smoothing family of algorithms. The ETS algorithm in [17] searches over more than 30 time series models accounting for additive and/or multiplicative error, additive and/or multiplicative seasonality, and additive and/or multiplicative level. In spirit, `Alg-ETS` has similarities to our approach in the way it does forecasting by first estimating the latent states via decomposing the time series.
3. The ARIMA forecasting algorithm [3,17] designated as `Alg-ARIMA`. This is another widely popular open source forecasting algorithm in the industry which performs at par to the best methods in the M3 competition. For example, a commercial software - `B-J Automatic` [23] had produced forecasts restricted to ARIMA models and was one of the best performers in the M3 competition. We have used the implementation available in [19].

**5.3 Anomaly Detection Performance** Since the anomaly detection problem is a decision between the presence or absence of an anomaly, a reasonable choice of performance metric is given by the duo defined by the type-I (false alarm) and type-II (mis-detection) errors. The performance of a traditional supervised classification task is evaluated using precision and recall. However, these measures are not well suited to reason about unsupervised anomaly detection tasks. Type-I error has a simple interpretation for anomaly detection that does not translate to precision and recall in an easily interpretable way. Hence, we will rely on type-I and type-II error values to compare the algorithms. Ideally, the estimated type-I error should be close to the choice of type-I error (5% in this case) and the best performing algorithm should have the lowest type-II error. Since the precision and recall values have been reported in past literature, we too have reported these metrics in our evaluation results.

We present the anomaly detection results for the `Alg-S` and `Alg-T` algorithms in terms of all four performance metrics for the datasets `SD`, `SD-L`, `NAB` and `NAB-HR` in Table 1. In all the competing algorithms, we have kept the level of statistical significance with which to accept or reject anomalies ($\alpha$ in §3.2) to be equal to 0.05.

A few comments about the results are in order. First, consider the results corresponding to `SD` dataset. The results suggest that `Alg-S` achieves a very healthy balance between type-I and type-II errors at 3.73% and 1.99% respectively. In contrast, not only does

Table 1: Evaluation Metrics (expressed as %) for Anomaly Detection. Lower type-I and type-II errors are better. Higher precision and recall are better. Best performance in boldface.

| Dataset | Method | Type-I | Type-II | Prec. | Rec. |
|---|---|---|---|---|---|
| SD | Alg-S | **3.73** | **1.99** | **24.09** | **98.01** |
| | Alg-T | 3.98 | 37.31 | 15.96 | 62.69 |
| SD-L | Alg-S | 7.90 | **1.03** | 13.14 | **98.97** |
| | Alg-T | **0.03** | 22.82 | **96.94** | 77.18 |
| NAB | Alg-S | **6.62** | **28.21** | **2.19** | **71.79** |
| | Alg-T | 6.96 | 29.49 | 2.05 | 70.51 |
| NAB-HR | Alg-S | 5.53 | **32.00** | **0.96** | **68.00** |
| | Alg-T | **5.29** | 36.00 | 0.94 | 64.00 |

Table 2: % of Wins for Forecasting on the WM Dataset. Higher is better. Best performance in boldface.

| | (window, forecast) | | | | |
|---|---|---|---|---|---|
| Method | (55,10) | (55,7) | (50,10) | (50,7) | Overall |
| Alg-S | **46.55** | **44.83** | **51.72** | **60.34** | **50.86** |
| Alg-ARIMA | 34.48 | 39.66 | 31.03 | 24.14 | 32.33 |
| Alg-ETS | 18.97 | 15.52 | 17.24 | 15.52 | 16.81 |

Alg-T require additional information in the form of specification of seasonal frequency, but it also *performs worse on both type-I and type-II errors* (despite this additional information being available) at 3.98% and 37.31% respectively.

Second, consider the results corresponding to the SD-L dataset. They suggest that absence of level changes in the dataset significantly helps the performance of Alg-T. In fact, the type-II error rate decreases to about 22% in the SD-L dataset from around 37% in the SD dataset for Alg-T. This could be attributed to the fact that Alg-T was not designed to account for level changes in the input dataset. Further, the type-I error for Alg-T is reduced to nearly zero on SD-L suggesting that the higher type-I error rate in the SD dataset might have mostly come from detecting a level change as an anomalous spike. For Alg-S, the absence of level changes results in type-II error rate getting reduced to 1% and type-I error rate increasing from 3.73% to 7.90% which is *close to the allowed 5% type-I error rate.*

Third, consider the results for the NAB dataset. We see that even for this dataset, Alg-S performs better than Alg-T. Although the type-I and type-II error rates for these two algorithms are close (with the numbers for Alg-T being slightly worse), recall that Alg-T requires the seasonality to be specified as additional information.

Table 3: % of Wins for Forecasting on the M3-$A_x$ Datasets. Higher is better. Best performance in boldface.

| Method | M3 | M3-$A_{2.5}$ | M3-$A_5$ | M3-$A_{10}$ |
|---|---|---|---|---|
| Alg-S | 20.00 | 32.17 | **35.93** | **41.20** |
| Alg-ARIMA | **42.86** | **34.83** | 32.55 | 30.13 |
| Alg-ETS | 37.53 | 33.00 | 31.52 | 28.67 |

Finally, for the NAB-HR dataset which contains time series that tend to show multiple seasonal patterns, the type-I error for Alg-S and Alg-T are fairly similar with Alg-S being much better at detecting outliers as evidenced by lower type-II error. Thus our approach is particularly better at handling multiple seasonal patterns with long time periods.

The comparison on the anomaly detection task shows that the approach proposed in this paper works better than the baselines even when the baseline algorithms are provided with some correct inputs based on ground truth. Note that our algorithm does not know the seasonal frequencies or maximum number of anomalies. The real benefit from the algorithm is revealed when complex seasonality and/or changes in levels are present. In such cases, our algorithm provides a large improvement over other baselines.

**5.4 Forecasting Performance** Since forecasting involves predicting the time series as accurately as possible within the forecast horizon, we have considered root mean squared error (RMSE) as a performance metric. Note that the observed values for the forecasting period are available for the WM dataset and the M3 competition corpus. For these two datasets, we calculate the forecasting RMSE for each algorithm on each of the 232 plus 3003 time series and pick the winner (*i.e.* the algorithm that achieves the least RMSE). Table 2 shows the comparative results presented as a percentage of the number of times each algorithm was the winner *w.r.t.* the WM dataset. For example, the last column of Table 2 says that Alg-S achieved the minimum forecasting RMSE on 50.86% of the 232 time series in the WM dataset. This improvement comes despite the fact that Alg-S required no input on the seasonal frequencies. The improvement is consistent across various time series lengths and forecasting horizons. The main reason behind the performance of Alg-S is the robustness of the algorithm against spikes.

To demonstrate the need for robustness against complex seasonality and spikes, we compare the forecasting performance of the three algorithms on the variants of the M3 dataset that consists of data at different granularities and hence different seasonalities. Recall that

we have added spikes to the original `M3` dataset to test how the results change in presence of spikes. Table 3 shows the comparative results presented as the percentage of times each algorithm was the winner on each variant of the `M3` dataset. Clearly, `Alg-S` is the winner on a substantial fraction of the $\texttt{M3-A}_x$ dataset for every $x \in \{2.5, 5, 10\}$. Further, `Alg-S` performs better relative to `Alg-ETS` and `Alg-ARIMA` with an increase in the percentage of anomalous spikes *w.r.t.* the total length of the time series.

These results reiterate the need for an algorithm that is capable of handling spike anomalies and complex seasonal patterns. Moreover, the results show that while the current algorithms in literature might work well on economic time series, a more robust approach is needed for large volumes of time series data that are getting generated today from sources like web analytic metrics. Our proposed method fills this gap.

## 6 Conclusions

We proposed a novel and flexible sparse separation based time series decomposition approach for anomaly detection and forecasting. The algorithm jointly estimates the latent but interpretable components (*viz.* seasonality, level changes, and spikes) in the observed time series and fits an ARMA model for the decomposition error. We demonstrated that our approach overcomes many limiting assumptions made by the existing time series analysis algorithms like knowledge of seasonality and presence of anomaly-free time windows. As a pleasant side effect, our approach leads to a forecasting algorithm that is robust to the presence of anomalies in the observation window, and further admits a greater degree of autonomy due to absence of any manual inputs (unlike existing algorithms that require seasonal pattern input). A further advantage of our approach is that it can simultaneously detect both point anomalies and change points as well as forecast, whereas competing approaches typically do only one of these at a time. We conducted experiments with synthetic and real-world datasets and compared our approach to popular state-of-the-art alternatives. We demonstrated that our approach convincingly outperforms competing anomaly detection algorithms under type-I error constraints and shows robustly better forecasting performance when compared to the state-of-the-art forecasting algorithms. As a future research effort, we shall explore theoretical properties of the proposed method and develop a scalable online version for of our algorithm to work with streaming data.

## Acknowledgements

We thank the anonymous reviewers whose comments greatly helped in better contextualization of the research presented in this paper.

## References

[1] C. C. Aggarwal, *Outlier analysis*, in Data mining, Springer, 2015, pp. 237–263.

[2] M. Basseville and I. V. Nikiforov, *Detection of Abrupt Changes: Theory and Application (Prentice Hall information and system sciences series)*, Prentice Hall, 4 1993.

[3] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control (Wiley Series in Probability and Statistics)*, Wiley, 5 ed., 6 2015.

[4] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[5] A. M. Bruckstein, D. L. Donoho, and M. Elad, *From sparse solutions of systems of equations to sparse modeling of signals and images*, SIAM Review, 51 (2009), pp. 34–81.

[6] C. Chen and L.-M. Liu, *Joint estimation of model parameters and outlier effects in time series*, Journal of the American Statistical Association, 88 (1993), pp. 284–297.

[7] R. B. Cleveland, W. S. Cleveland, and I. Terpenning, *STL: A seasonal-trend decomposition procedure based on loess*, Journal of Official Statistics, 6 (1990), p. 3.

[8] G. Davis, S. Mallat, and M. Avellaneda, *Adaptive greedy approximations*, Constructive Approximation, 13 (1997), pp. 57–98.

[9] J. G. De Gooijer and R. J. Hyndman, *25 years of time series forecasting*, International journal of forecasting, 22 (2006), pp. 443–473.

[10] S. Diamond and S. Boyd, *CVXPY: A Python-embedded modeling language for convex optimization*, Journal of Machine Learning Research, 17 (2016), pp. 1–5.

[11] A. Domahidi, E. Chu, and S. Boyd, *ECOS: An SOCP solver for embedded systems*, in 2013 European Control Conference (ECC), July 2013, pp. 3071–3076.

[12] D. L. Donoho and M. Elad, *Optimally sparse representation in general (nonorthogonal) dictionaries via $l^1$ minimization*, Proc. Natl. Acad. Sci. USA, 100 (2003), pp. 2197–2202 (electronic).

[13] M. Gupta, J. Gao, C. Aggarwal, and J. Han, *Outlier Detection for Temporal Data: A Survey*, IEEE Transactions on Knowledge and Data Engineering, 26 (2014), pp. 2250–2267.

[14] F. Gustafsson, *The marginalized likelihood ratio test for detecting abrupt changes*, IEEE Transactions on automatic control, 41 (1996), pp. 66–78.

[15] J. Hochenbaum, O. S. Vallis, and A. Kejariwal, *Automatic anomaly detection in the cloud via statistical learning*, ArXiv e-prints, abs/1704.07706 (2017).

[16] V. Hodge and J. Austin, *A survey of outlier detection methodologies*, Artificial intelligence review, 22 (2004), pp. 85–126.

[17] R. Hyndman, A. B. Koehler, J. K. Ord, and R. D. Snyder, *Forecasting with Exponential Smoothing: The State Space Approach (Springer Series in Statistics)*, Springer, 2008 ed., 7 2008.

[18] R. J. Hyndman and B. Billah, *Unmasking the Theta method*, International Journal of Forecasting, 19 (2003), pp. 287–290.

[19] R. J. Hyndman and Y. Khandakar, *Automatic time series forecasting: the forecast package for R*, Journal of Statistical Software, 26 (2008), pp. 1–22.

[20] T. W. Joo and S. B. Kim, *Time series forecasting based on wavelet filtering*, Expert Systems with Applications, 42 (2015), pp. 3868 – 3874.

[21] N. Laptev, S. Amizadeh, and I. Flint, *Generic and scalable framework for automated time-series anomaly detection*, in Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '15, New York, NY, USA, 2015, ACM, pp. 1939–1947.

[22] A. Lavin and S. Ahmad, *Evaluating Real-Time Anomaly Detection Algorithms–The Numenta Anomaly Benchmark*, in Machine Learning and Applications (ICMLA), 2015 IEEE 14th International Conference on, IEEE, 2015, pp. 38–44.

[23] S. Makridakis and M. Hibon, *The M3-competition: results, conclusions and implications*, International journal of forecasting, 16 (2000), pp. 451–476.

[24] B. K. Natarajan, *Sparse approximate solutions to linear systems*, SIAM Journal on Computing, 24 (1995), pp. 227–234.

[25] B. O'Donoghue, E. Chu, N. Parikh, and S. Boyd, *Conic optimization via operator splitting and homogeneous self-dual embedding*, Journal of Optimization Theory and Applications, 169 (2016), pp. 1042–1068.

[26] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing (3rd Edition) (Prentice-Hall Signal Processing Series)*, Pearson, 3 ed., 8 2009.

[27] K. Ord, M. Hibon, and S. Makridakis, *The M3-competition*, International journal of forecasting, 16 (2000), pp. 433–436.

[28] P. Perron, *The great crash, the oil price shock, and the unit root hypothesis*, Econometrica: Journal of the Econometric Society, (1989), pp. 1361–1401.

[29] ———, *Further evidence on breaking trend functions in macroeconomic variables*, Journal of econometrics, 80 (1997), pp. 355–385.

[30] M. H. Pesaran, D. Pettenuzzo, and A. Timmermann, *Forecasting time series subject to multiple structural breaks*, The Review of Economic Studies, 73 (2006), pp. 1057–1084.

[31] B. Rosner, *Percentage points for a generalized esd many-outlier procedure*, Technometrics, 25 (1983), pp. 165–172.

[32] Z. R. Struzik and A. P. Siebes, *Wavelet transform based multifractal formalism in outlier detection and localisation for financial time series*, Physica A: Statistical Mechanics and its Applications, 309 (2002), pp. 388 – 402.

[33] R. S. Tsay, *Time series model specification in the presence of outliers*, Journal of the American Statistical Association, 81 (1986), pp. 132–141.

[34] O. Vallis, J. Hochenbaum, and A. Kejariwal, *A Novel Technique for Long-term Anomaly Detection in the Cloud*, in Proceedings of the 6th USENIX Conference on Hot Topics in Cloud Computing, HotCloud'14, Berkeley, CA, USA, 2014, USENIX Association, pp. 15–15.

[35] ———, *Anomaly detection in R*. `https://github.com/twitter/AnomalyDetection`, 2015.

[36] K. Yamanishi and J.-i. Takeuchi, *A unifying framework for detecting outliers and change points from non-stationary time series data*, in Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2002, pp. 676–681.