Data Mining and Knowledge Discovery (2023) 37:1209–1229
https://doi.org/10.1007/s10618-023-00919-7

AA-forecast: anomaly-aware forecast for extreme events

Ashkan Farhangi[1] · Jiang Bian[2] · Arthur Huang[1] · Haoyi Xiong[2] · Jun Wang[1] · Zhishan Guo[1]

Received: 22 November 2021 / Accepted: 9 January 2023 / Published online: 13 March 2023

Abstract

Time series models often are impacted by extreme events and anomalies, both prevalent in real-world datasets. Such models require careful probabilistic forecasts, which is vital in risk management for extreme events such as hurricanes and pandemics. However, it's challenging to automatically detect and learn from extreme events and anomalies for large-scale datasets which often results in extra manual efforts. Here, we propose an anomaly-aware forecast framework that leverages the effects of anomalies to improve its prediction accuracy during the presence of extreme events. Our model has trained to extract anomalies automatically and incorporates them through an attention mechanism to increase the accuracy of forecasts during extreme events. Moreover, the framework employs a dynamic uncertainty optimization algorithm that reduces the uncertainty of forecasts in an online manner. The proposed framework demonstrated consistent superior accuracy with less uncertainty on three datasets with different varieties of anomalies over the current prediction models.

Responsible editor: Albrecht Zimmermann

✉ Ashkan Farhangi
ashkan.farhangi@ucf.edu

Jiang Bian
bianjiang03@baidu.com

Arthur Huang
arthur.huang@ucf.edu

Haoyi Xiong
xionghaoyi@baidu.com

Jun Wang
jun.wang@ucf.edu

Zhishan Guo
zhishan.guo@ucf.edu

1 University of Central Florida, Orlando, FL, USA

2 Baidu Research Lab, Beijing, China

Keywords Time series forecasting · Uncertainty optimization · Anomaly decomposition

1 Introduction

Time series forecasting during the presence of extreme events is a critical tool for resource allocation and resilience planning (Jing et al. 2021; Santos-Burgoa et al. 2018; Khan et al. 2021). Extreme events such as natural disasters are causing more than 400% economic damage in the U.S. compared to 1990s (Smith 2022). This requires us to develop highly accurate forecasts with low uncertainty to uncover the influence of external events on large-scale time series data (Adilova et al. 2021). Moreover, it is crucial to understand how different industries are influenced by and recover from such extreme events over time (Rolnick et al. 2019). Yet, it remains a challenge to develop such reliable and accurate forecasting models, as the real-world dataset often contains anomalies that are in their nature rare and random. Therefore, it is important to develop a forecast model that can leverage the previously seen extreme events and anomalies for their forecasts.

Although there have been considerable achievements in machine learning-based models, existing methods tend to overlook anomalies' special effects on real-world time series data. For instance, LSTMs (Hochreiter and Schmidhuber 1997) are widely used to address the vanishing gradient problem via gate mechanism and have the ability to capture complex temporal dependencies (Zhu and Laptev 2017; Laptev et al. 2017). However, Khandelwal et al. (2018) show that even LSTMs have a limited ability to capture long-term dependencies, and their awareness of context degrades as the length of the input sequence increases. Consequently, making them inefficient to capture and learn from rare occurrences or extreme events.

As an alternative, Li et al. (2019) considered the use of transformer models for time series forecasting. Transformers benefit from the self-attention mechanism which allows each observation in the feature sequence to attend independently to every other feature in the sequence. However, they have considerable computational and memory requirements that grow quadratically with respect to sequence length, making it computationally rigorous to train large-scale data (Li et al. 2019). Such deficiency makes them computationally unsuitable for extreme events that often appear in longer sequences than the transformer inputs. Moreover, it was not even clear from the design itself that transformers can be as effective as RNNs, whereas Zaheer et al. (2020) reported that the attention mechanism in transformers does not even obey the sequence order of time steps which is essential for the time series domain. Furthermore, non-transformer architectures (i.e. MLP) have been shown to perform competitively with transformers when designed and trained properly (Tolstikhin et al. 2021).

This lack of systematic strategy to handle anomalies and not provide forecasts with nontransparent uncertainty levels makes the current forecast model unreliable during the presence of extreme events. As a result, a key aspect of our knowledge in developing time series models for critical moments of extreme events will remain a puzzle unless the long-term effects of anomalies are well captured and utilized.

Contribution. This work proposes a novel and generalized anomaly-aware prediction framework, AA-Forecast, which automatically extracts and uses anomalies to optimize its probabilistic forecasting. Specifically,

- AA-Forecast extracts anomalies through a novel decomposition method and leverages them through an attention mechanism designed to optimize its probabilistic forecasting during extreme events. Also, AA-Forecast is able to perform zero-shot prediction for unseen time series and does not suffer from quadratic computational time and memory complexity of transformers.
- An online optimization procedure is proposed to minimize the prediction uncertainties of the AA-Forecast framework, which features applying the optimal dropout probability at each time step during the testing.
- Extensive experimental studies are conducted on three real-world datasets that are prone to extreme events and anomalies. The comparisons with state-of-the-art models illustrate the higher accuracy and less uncertainty in the AA-Forecast's prediction.

2 Problem formulation

In this study, we are interested in the task of time series forecasting under the influence of extreme events and anomalies. Given a dataset $\mathbf{D} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(K)}\}$ with K univariate time series, $\mathbf{x}^{(k)} = \{x_1^{(k)}, x_2^{(k)}, \ldots, x_T^{(k)}\}$ denotes a time series instance with length T, where $\mathbf{x}^{(k)} \in \mathbb{R}^T$. For every time step, the corresponding extreme events are aligned and labeled as covariates $\mathbf{e}^{(k)} = \{e_1^{(k)}, e_2^{(k)}, \ldots, e_T^{(k)}\}$. Extreme events are considered as the influence of external events that promote a dynamic occurrence within a limited time steps (Broska et al. 2020). Specifically, $e_t^{(k)} \in \mathbb{R}$ indicates the level of extreme event (e.g., hurricane category) at time t, otherwise, $e_t^{(k)} = 0$ indicates a non-extreme event condition for periods outside of the event. To this end, we denote the data with extreme events as a series of tuples $\widehat{\mathbf{x}}^{(k)} \triangleq \{(x_1^{(k)}, e_1^{(k)}), (x_2^{(k)}, e_2^{(k)}), \ldots, (x_T^{(k)}, e_T^{(k)})\}$. Particularly, given the previous τ observations $\widehat{\mathbf{x}}_{t-\tau+1:t}^{(k)} = \{(x_{t-\tau+1}^{(k)}, e_{t-\tau+1}^{(k)}), (x_{t-\tau+2}^{(k)}, e_{t-\tau+2}^{(k)}), \ldots, (x_t^{(k)}, e_t^{(k)})\}$, we aim to model the conditional distribution of the next observation:

$$p(x_{t+1}^{(k)} \mid \widehat{\mathbf{x}}_{t-\tau+1:t}^{(k)}; \mathbf{\Phi}), \tag{1}$$

where $\mathbf{\Phi}$ denotes the parameters of a nonlinear prediction model. We are also interested in reducing the uncertainty of predictions in an online setting, whereas uncertainty of prediction can be viewed as the variability of the distribution. Therefore, the optimization problem during the online settings is defined as follows:

$$\mathbf{\Phi}_{\text{on}}^* = \text{argmin}_{\mathbf{\Phi}}\, \mathcal{V}\left(p(x_{t+1}^{(k)} \mid \widehat{\mathbf{x}}_{t-\tau+1:t}^{(k)}; \mathbf{\Phi})\right), \tag{2}$$

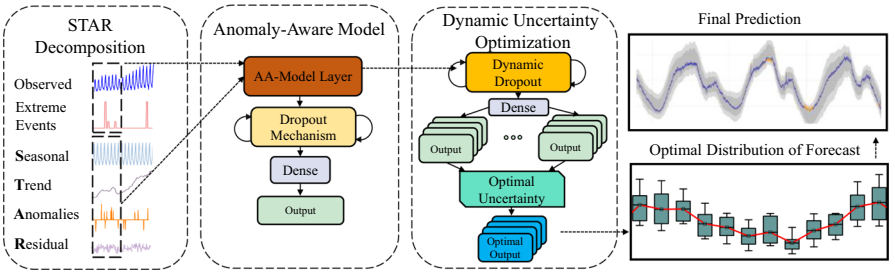

Fig. 1 Main components of AA-Forecast: (i) **STAR Decomposition** to automatically extract essential features such as anomalies, (ii) an **Anomaly-Aware Model** to leverage such extracted features, and (iii) a **Dynamic Uncertainty Optimization** to reduce the uncertainty of the network. The final predicted series contains confidence intervals with the least uncertainty

where $\mathcal{V}(\cdot)$ represents the variability of the probability distribution and $\mathbf{\Phi}^*_{\text{on}}$ is the optimal online parameters of the nonlinear prediction model that produces the least amount of uncertainty in each time step.

3 AA-forecast framework

The proposed AA-Forecast framework consists of three main components. Section 3.1 proposes a novel anomaly decomposition method that automatically extracts the anomalies and essential features of the time series data. Then, the extracted anomalies are fed into an anomaly-aware model detailed in Sect. 3.2. Specifically, it leverages an attention mechanism on anomalies and extreme events to produce the distribution of the forecasts. To further reduce the forecast uncertainty in an online manner, Sect. 3.3 proposes a dynamic uncertainty optimization algorithm.

3.1 STAR decomposition

STAR decomposition is used as a strategy to not only extract the anomalies and sudden changes of data but also decompose the complex time series to its essential components. Unfortunately, widely popular decomposition method such as STL (Cleveland et al. 1990) does not extract anomalies. Although recent works such as STR (Dokumentov and Hyndman 2020) and RobustSTL (Wen et al. 2019) are designed to be robust to the extreme effect of anomalies in their decomposition, they are not used to explicitly extract anomalies from the residual component.

To alleviate these issues, we propose STAR decomposition that decomposes the original time series $\mathbf{x}^{(k)}$ in a multiplicative manner to its **s**easonal ($\mathbf{s}^{(k)}$), **t**rend ($\mathbf{t}^{(k)}$), **a**nomalies ($\mathbf{a}^{(k)}$), and **r**esidual ($\mathbf{r}^{(k)}$) components:

$$\mathbf{x}^{(k)} = \mathbf{s}^{(k)} \times \mathbf{t}^{(k)} \times \mathbf{a}^{(k)} \times \mathbf{r}^{(k)} \tag{3}$$

Such decomposition is important due to increasing the dimensions of the original data and providing the model with automatic extraction of anomalies. As shown in

Fig. 1, we begin the decomposition by approximating the trend line $\mathbf{t}^{(k)}$ with the locally weighted scatterplot smoothing (i.e., LOESS Cleveland 1979). Then, we divide the original data $\mathbf{x}^{(k)}$ by the approximated trend line to derive the detrended time series.[1]

We then partition the detrended time series into periods of cyclic sub-series where the cycle size is determined by the time interval of the dataset. As an example, the cycle size for a monthly dataset would be 12 (one year as a cycle). Then we obtain the seasonal component ($\mathbf{s}^{(k)}$) by grouping the detrended series in each period and deriving the average value of each period across the time series. Subsequently, the residual component ($\mathbf{r}^{(k)}$) is derived by dividing the seasonal and trend segments from the original series.

Note that the anomaly component ($\mathbf{a}^{(k)}$) can be considered as the oddities of the dataset, which do not follow the extracted trend or seasonal components. Intuitively, the anomalies spread through the residual components, which also contain noise and other real-world effects. To distinguish anomalies from residual components, statistical metrics such as mean and variance are not the appropriate measure as they are highly sensitive to the severity level of anomaly values. As one expects, the severity of the anomalies can change the mean and variance values which are unwanted. To resolve this issue, we leverage the median of the residuals, which is immune to the severity of the outliers in the residual components. Next, we define robustness score $\rho_t^{(k)}$ for each observation at time t as:

$$\rho_t^{(k)} = \frac{|r_t^{(k)} - \dot{r}^{(k)}|}{\sqrt{\frac{\sum_{t=1}^{T} |r_t^{(k)} - \dot{r}^{(k)}|}{T-1}}} \tag{4}$$

where $\rho_t^{(k)}$ stands for the strength of the anomalies, $r_t^{(k)}$ is the residual at time step t and $\dot{r}^{(k)}$ is the median of the residuals.

Note that the larger ρ_t indicates that a drastic change has occurred in the trend and seasonal components. We then extract the anomalies from the residuals as follows:

$$\mathbf{a}_t^{(k)} = \begin{cases} 1, & \rho_t^{(k)} < \rho_c^{(k)} \\ r_t^{(k)}, & \rho_t^{(k)} > \rho_c^{(k)} \end{cases} \tag{5}$$

where $\rho_c^{(k)}$ is the constant threshold given by the value of a robustness score ranked in the p-value 0.05[2] while the values of elements in $\rho^{(k)}$ are ranked in descending order from large to small.

Notably, when the value of the anomaly component ($\mathbf{a}^{(k)}$) deviates further from the value 1, it indicates an abrupt change in the trend and the seasonal component (no sign of anomalies). On the contrary, when both anomaly and residual values are equal 1 ($\mathbf{r}_t^{(k)} = 1$ and $\mathbf{a}_t^{(k)} = 1$), it indicates that the observed signal at time t explicitly follows the trend and the seasonal component. Note that such important information

[1] We use the log transform of $\mathbf{x}^{(k)}$ to handle the situation that specific values of original data are zero.

[2] Adopted based on the choice of the p value (0.05) which is used as a standard level of statistical significance.

might not be automatically inferred when additive decomposition methods are being used. This is due to the fact that the values of residual components can differ from one dataset to another, which requires manual effort in their detection.

A sample result of anomaly decomposition is shown in Fig. 4, where the observed time series data is decomposed into its seasonal, trend, anomalies, and residual components respectfully. Each of these components holds essential information about the characteristics of the time series and will be leveraged to train the forecast model. To this end, we concatenate the derived decomposed vector of the time series with the input, which includes the observed time series and its labeled extreme event. Specifically, the STAR decomposition concatenates the original time series to $\widetilde{\mathbf{x}}^{(k)} = (\mathbf{x}^{(k)}, \mathbf{e}^{(k)}, \mathbf{s}^{(k)}, \mathbf{t}^{(k)}, \mathbf{a}^{(k)}, \mathbf{r}^{(k)})$ which can be leveraged by the anomaly-aware model described in the next section.

3.2 Anomaly-aware model

The Anomaly-Aware model is designed to explicitly incorporate extracted anomalies $\mathbf{a}^{(k)}$ and extreme event covariates $\mathbf{e}^{(k)}$ into the prediction. As these features are rare in the whole time series, feeding them directly into a regular RNN like LSTM (Hochreiter and Schmidhuber 1997) can be potentially ignored during the training of the model. Note that the extracted anomalies and previously experienced external events hold valuable information regarding the effect of extreme events that should be handled carefully.

Recent robust prediction models rely on the LSTMs or transformers architecture to provide robustness in their prediction. Although LSTMs are designed to obtain long-term dependencies, their ability to pay different degrees of attention to sub-window features within large time steps is inadequate (Zaheer et al. 2020). As an example, Khandelwal et al. (2018) showed that even though the LSTM model can have an effective sequence size of 200 observations, they are only able to sharply distinguish the 50 closest observations. This indicates that even LSTMs struggle to capture long-term dependencies. On the other hand, conventional transformers suffer from quadratic computation and memory requirements, which limits their ability to process long input sequences.

Even though such memory bottlenecks have been improved by using sparse-attention algorithms (Li et al. 2019), their performance improvement is not significant compared to a full-attention mechanism for real-world datasets (Lim et al. 2019). Given that extreme events and anomalies are rare and can appear at very long distances from each other, it is computationally infeasible to increase the input sequence to provide attention to all previously seen anomalies and extreme events.

To address such problems, one must pay attention to all the anomalies and extreme events throughout the dataset, no matter how far they have occurred. Intuitively, due to their rare nature, they are of greater importance in learning, given that the trend and seasonal patterns are often easier to predict by statistical or deep learning models. Ergo, we developed a novel attention mechanism explicitly for extreme events and

anomalies, which are considered the crucial time steps of time series data and often cause the biggest error in prediction.

Architecture design of AA-model. LSTMs and GRUs are suitable for predicting the recurring patterns with a fairly low computational time and memory complexity which suffer from the quadratic complexity of full-attention transformers. However, we enhance the long-term dependencies of these models through an attention mechanism that retains the effect of anomalies and extreme events for future predictions. Such a decision in architecture allows the model not only to be computationally feasible for handling large-scale datasets but also to take the critical moments of extreme events and anomalies into consideration.

Given the past τ time steps of observations as $\tilde{\mathbf{x}}_{t:t-\tau+1}$[3], we derive the hidden states of an RNN that deals with vanishing gradient problem (e.g., LSTM or GRU) as:

$$\mathbf{h}_{t:t-\tau+1} = \text{RNN}\left(\tilde{\mathbf{x}}_{t:t-\tau+1}\right), \tag{6}$$

where h_t is the hidden layer of RNN at time step t. Note that we are only paying attention to anomalies and extreme events which are naturally rare and belong to a small population of observations. Moreover, both could have different impacts on the prediction and based on the type of dataset, can be challenging to model. Hence, we design the attention mechanism to automatically incorporate extreme events and anomalies during their occurrence:

$$J = \{t \in \mathbb{Z}^+ | e_t \neq 0 \vee a_t \neq 1\}, \tag{7}$$

where J is the set of time steps including two possible circumstances: the presence of extreme events covariates ($e_t \neq 0$) or anomalies ($a_t \neq 1$). We then gather all the previous hidden states of the RNNs for all critical time steps in J and regularize them by the weights generated from the attention layer as v_t which follows:

$$v_t = \tanh(\mathbf{w}_\alpha^\top \mathbf{h}_t + b_\alpha), \quad \forall t \in J \tag{8}$$

where $\mathbf{w}_\alpha$ and b_α are the attention layer's weight and bias. Then, we derive the attention weights of all previous values as:

$$\alpha_t = \text{Softmax}\left(v_1, v_2, \ldots, v_t\right), \quad \forall t \in J \tag{9}$$

where α_t is the attention weight at the critical time steps. The generated attention weights are then used in the AA-Forecast layer as:

$$\mathcal{A}_t = \begin{cases} \mathbf{h}_t, & \forall t \notin J \\ \sum_{t \in J} \alpha_t \cdot \mathbf{h}_t, & \forall t \in J \end{cases} \tag{10}$$

where the attention values are only calculated in the presence of anomalies and extreme events as shown in Fig. 2. The value of the next time step is calculated through a dense

[3] To reduce the ambiguity of the AA-Forecast layer, we are omitting the superscript (k) from this section

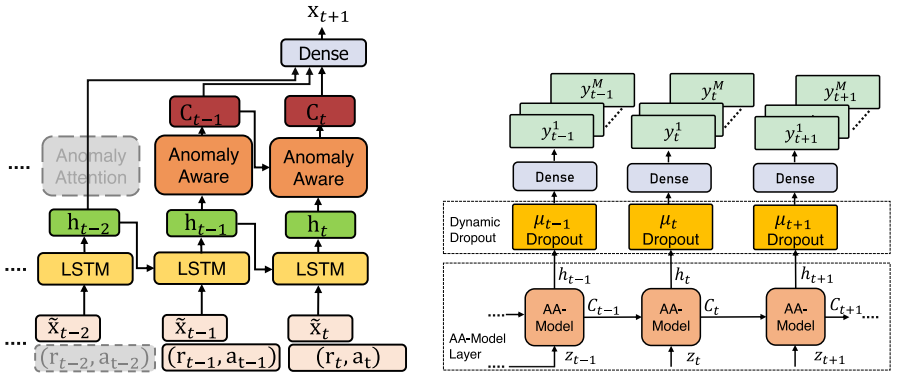

Fig. 2 **Left:** AA-model architectures. **Right:** Dynamic dropout μ_t determines the optimal probability of dropout at each time step during the online settings (i.e., inference). The output $\hat{y}$ consists of a distribution of predicted test values. The dropout optimization improves the certainty and accuracy at each time step t by determining how relevant the previous hidden state is for the next time step prediction

layer:

$$y_{t+1} = \mathbf{w}_d(\mathcal{A}_{t:t-\tau+1}) + b_d, \tag{11}$$

where $\mathbf{w}_d$ and b_d are the weights and biases of the dense layer. To train the network, we minimize the prediction loss $\mathcal{L}$ which is defined as follows:

$$\mathbf{\Phi}^*_{\text{off}} = \text{argmin}_{\mathbf{\Phi}}\, \mathcal{L}\left(\mathcal{F}_{\mathbf{\Phi}}(\widetilde{\mathbf{x}}), y\right), \tag{12}$$

where $\mathcal{F}_{\mathbf{\Phi}}$ is the anomaly-aware model and y is the training label, which is the ground truth of the next time step prediction. Note that $\mathbf{\Phi}^*_{\text{off}}$ represents the optimal model parameters after the offline training phase.

3.3 Dynamic uncertainty optimization

Although Monte Carlo (MC) dropout (Gal and Ghahramani 2016) probability is treated as a static hyperparameter in previous studies (Salinas et al. 2020; Laptev et al. 2017), it plays an important role in the prediction outcome and can be leveraged to reduce the uncertainty of the prediction during the testing phase (Wahab et al. 2020). Therefore, we rely on an automatic selection mechanism for optimal dropout in online settings. Such selection is based on the uncertainty of the prediction produced during the testing phase (Fig. 2).

Note that the model's uncertainty is desired to be the lowest and as stable as possible in real-world settings. Therefore, it is essential to optimize further the uncertainty of the model prediction both during the offline training and online testing phase. Specifically, we apply a dropout operation after every AA-Forecast layer with a specific probability (p).

AA-Forecast not only reports the prediction distribution, but also provides the point prediction (average of the distribution) and the prediction uncertainty (variability of

the distribution). Specifically, by producing M forecast for every time step in an online manner (test data $\widetilde{\mathbf{x}}^*$) from the previously trained model $\mathcal{F}_{\mathbf{\Phi}}(\widetilde{\mathbf{x}})$, we obtain M outputs y^* as a from the prediction distribution $\left\{y^*_{(1)}, \ldots, y^*_{(M)}\right\}$. Then, the average of the distribution is calculated as $\bar{y}^* = \frac{1}{M}\sum_{m=1}^{M} y^*_{(m)}$.

We represent uncertainty as to the variability of the prediction distribution —- the standard deviation (SD) of the probability distribution of future observations conditional on the information available at the time of forecasting. We further optimize the uncertainty of the framework by deriving the optimal dropout probability p at each time step. We derive the prediction error for the probability p between 0 and 1 with 0.1 increments. Notably, without such probability (i.e., $p = 0$) the model prediction deviates from probabilistic forecasting and does not provide a level of uncertainty in its prediction for each time step. The optimal uncertainty μ_t is then reported when it results in a minimal variance (i.e., SD) of the predicted values, thereby reducing the prediction uncertainty to its minimum during the testing phase. To this end, the prediction uncertainty is formulated as:

$$\sigma^2\left(\mathcal{F}_{\mathbf{\Phi}}(\widetilde{\mathbf{x}}^*)\right) = \sqrt{\frac{1}{M}\sum_{m=1}^{M}\left(y^*_{(m)} - \bar{y}^*\right)^2}. \tag{13}$$

Algorithm 1 Psuedecode for AA-Forecast

Input: data $\widetilde{\mathbf{x}}^{(k)} = (\mathbf{x}^{(k)}, \mathbf{e}^{(k)}, \mathbf{s}^{(k)}, \mathbf{t}^{(k)}, \mathbf{a}^{(k)}, \mathbf{r}^{(k)})$
1: Initialize parameters $\mathbf{\Phi}$
2: **for** $k = 1$ to K_{train} **do**
3: Sample $(\tilde{\mathbf{x}}^k, y^k)$ from training data:
4: **for** $b = 1$ to B **do**
5: $\mathbf{\Phi}_{e+1} \leftarrow \mathbf{\Phi}_e$ - $\xi \cdot \nabla\ \mathcal{L}(\mathcal{F}_{\mathbf{\Phi}}(\tilde{\mathbf{x}}^k), y^k)$
6: Update the optimal parameters:
$\mathbf{\Phi} = \text{argmin}_{\mathbf{\Phi}}\mathcal{L}(\mathcal{F}_{\mathbf{\Phi}}(\tilde{\mathbf{x}}^k), y^k)$
7: **end for**
8: **end for**
9: Dynamic Uncertainty optimization: $\mathbf{\Phi}^* \leftarrow \mathbf{\Phi}$
10: **for** $\delta = 0.1$ to 0.9 increment by 0.1 **do**
11: Update the optimal uncertainty:
$\mathbf{\Phi}^* = \text{argmin}_{\mathbf{\Phi}}\mathcal{V}(\mathcal{F}_{\mathbf{\Phi}}(x^k))$
12: **end for**

Algorithm 1 presents the pseudocode for AA-Forecast. Specifically, we sample $(\tilde{\mathbf{x}}^k, y^k)$ as a driving example which includes extracted anomalies $\mathbf{a}^{(k)}$ and extreme events $\mathbf{r}^{(k)}$. Next, we train the model by maximizing the overall prediction accuracy. Upon testing, the network leverage dynamic uncertainty optimization further optimizes the prediction uncertainty automatically in online testing so that it would not require any further training.

Note that the network's predictions during the testing phase cannot benefit from the supervised training. However, the control of variability is possible and ensures that the prediction uncertainty is minimal in each step of future predictions, regardless

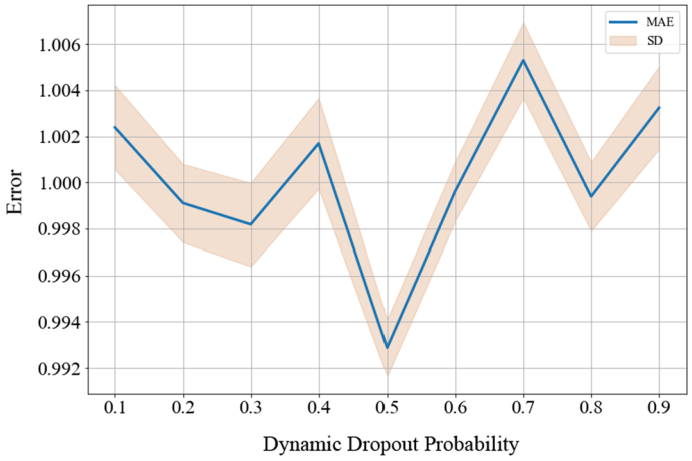

Fig. 3 Effects of dynamic uncertainty optimization on prediction error and uncertainty during the occurrence of an anomaly. The method automatically selects the optimal probability that yields the lowest uncertainty

of whether the labels are provided or not. Additionally, the algorithm testing time complexity is similar to other RNN-based models due to the use of dynamic uncertainty optimization during the test phase solely. This allows the model to provide the least amount of uncertainty during the presence of anomalies or extreme events where critical online decisions are being made.

As an example, Fig. 3 shows that optimal uncertainty results can occur when the standard deviation is the lowest. Intuitively, the network at $p = 0.5$ shows the highest confidence in its prediction (i.e., the lowest uncertainty) where unnecessary neurons are dropped out from the network. Therefore, the network automatically selects and reports the $p = 0.5$ probability as the best choice for this time step in the testing phase.

4 Experiments

This section reports multiple experiments comparing the proposed AA-Forecast framework with baseline models using different types of large-scale time series datasets.

4.1 Dataset and experimental settings

Three real-world time series with diverse structures and domains are gathered (Fig. 4).[4] Table 1 provides descriptive statistics and the detailed description are as follows:

- We gathered a new spatio-temporal benchmark dataset (`Hurricane`), which is suited for forecasting during extreme events and anomalies. The dataset is provided through the Florida Department of Revenue which provides the monthly sales revenue (2003-2020) for the tourism industry for all 67 counties of Florida which are prone to annual hurricanes. Furthermore, we aligned and joined the raw time

[4] All datasets are publicly available at https://github.com/ashfarhangi/AA-Forecast

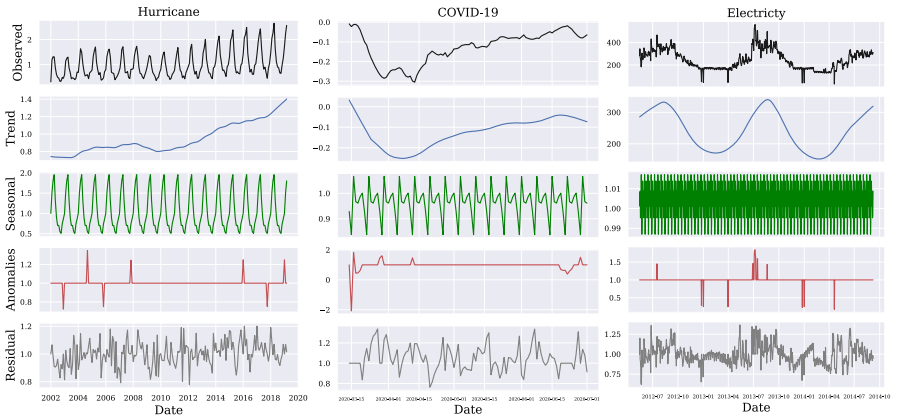

Fig. 4 The results of STAR Decomposition for three samples. The hurricane dataset sample is taken from Collier county in Florida, USA and the values are normalized in USD. The COVID-19 sample is taken from Florida state and the values show the changes (percentage) from the beginning of the pandemic. The Electricity sample is from MT-200 and values are in kW

Table 1 Descriptive statistics of the datasets

Dataset	Hurricane	COVID-19	Electricity
Time step	Monthly	Daily	Hourly
# Unique time series	9876	15,312	370
# Observation	9876	15,312	11,952,480
# Train	7900	12,250	9,561,984
# Test	1975	3062	2,390,496
# Regions	48	50	370
# Extreme events	88	100	–
# Anomalous points	102	124	672

series with the history of hurricane categories based on time for each county. More precisely, the hurricane category indicates the maximum sustained wind speed which can result in catastrophic damages (Oceanic 2022).

- The second dataset (`COVID-19`) showcases the changes in the number of employees based on one million employees active in the US during the COVID-19 pandemic and is gathered from Homebase (Bartik et al. 2020). We further enriched the data with the state-level policies as an indication of extreme events (e.g., the state's business closure order).
- The third dataset (`Electricity`) is a publicly available benchmark dataset that contains the electricity consumption of 370 consumers hourly from 2011 to 2014. Note that this benchmark dataset is anonymized and does not contain extreme event labels, yet AA-Forecast is able to automatically extract the anomalies, indicating abrupt changes in trend and seasonality.

Table 2 Hyperparameters of AA-forecast used for each dataset

Parameter	Hurricane	COVID-19	Electricity
Batch size	128	64	64
Learning rate	1×10^{-5}	3×10^{-5}	5×10^{-5}
Weight decay	1×10^{-6}	1×10^{-5}	1×10^{-4}
Number of epochs	40	40	40
Static dropout	0.5	0.4	0.6

We propose two sets of experiments for all baseline models. The first experiment follows a standard 80–20 dividing of the dataset to training and testing sets and $\tau = 12$ for window length. The second experiment evaluates the zero-shot prediction capability of the model based on various window search ranges in $\{3, 6, 12, 24\}$, and thus is more applicable for real-world settings when the newly added time series cannot train on a newly added time series. Hence, the second experiment evaluates the prediction accuracy of all models on a set of completely unseen time series.

The models are implemented using Python 3.7 and tested on a cloud workstation with two Intel Xeon 2.3 GHz CPUs, 64 GB RAM, and one Nvidia Tesla A100 GPU. We conduct a grid search over all tunable hyperparameters on the held-out validation set for baseline methods and our framework. The hyperparameters for each dataset are shown in Table 2. To provide a fair evaluation, all baseline models benefit from the essential features extracted by AA-Forecast except the ARIMA model which does not benefit from multidimensional features. Moreover, future known information is not included in all the models.

The training times of AA-forecast for all three datasets are reported in Table 3. We kept training to 40 iterations for all experiments. The reported values are the average of the observed error five times during the test stage. The hyperparameters of all baseline methods are tuned based on a grid search.

4.2 Methods for comparison

The baseline methods for comparison include:

- ARIMA (Box and Pierce 1970): A traditional autoregressive integrated moving average method for time series prediction and often used as a baseline.
- AE-LSTM (Sagheer and Kotb 2019): An LSTM network that uses an autoencoder for deep feature extraction and provides a deterministic prediction.
- SARIMAX (Tarsitano and Amerise 2017): An autoregressive model that can handle seasonality and exogenous features of time series.
- UberNN (Zhu and Laptev 2017): An LSTM-based model that uses Monte Carlo dropout to provide uncertainty and is able to extract deep features of time series through autoencoders.
- TSE-SC (Cai et al. 2020): was recently proposed as a Transformer-based Deep Learning model that can forecast abrupt changes accurately. (i) STAR Decomposition to automatically ex- tract essential features such as anomalies, (ii) an Anomaly-Aware Model to leverage such extracted features, and (iii) a Dynamic

Table 3 Runtime of the methods used in the study

Parameter	Hurricane	COVID-19	Electricity
ARIMA (Box and Pierce 1970)	2:32m	3:21m	13:56m
AE-LSTM (Sagheer and Kotb 2019)	9:57m	13:54m	42:15m
SARIMAX (Tarsitano and Amerise 2017)	3:25m	4:36m	14:56m
UberNN (Zhu and Laptev 2017)	10:52m	13:26m	44:52m
TSE-CE (Cai et al. 2020)	17:41m	21:55m	1:12:45h
AA-Forecast (GRU)	10:26m	14:23m	44:21m
w/o STAR decomposition	9:53m	12:43m	44:12m
w/o Uncertainty optimization	10:26m	14:26m	44:51m
w/o Anomaly attention	7:53m	10:12m	43:15m

Uncertainty Optimization to reduce the uncertainty of the network. The final predicted

- AA-Forecast (LSTM) is our proposed model with LSTM cells.
- AA-Forecast (GRU) is our proposed model with GRU cells.

4.3 Metrics

For providing a comprehensive evaluation, we adopted three different evaluation metrics. The first evaluation metric is the Continuous Ranked Probability Score (CRPS), which evaluates probabilistic forecasting. Formally defined as $\mathbf{CRPS} = \int_{-\infty}^{\infty}(F(y) - \mathbb{1}(y - \hat{y}))^2 \, \mathrm{d}y$ where F is the cumulative distribution function of its forecast distribution and $\mathbb{1}$ is the Heaviside step function. We also report the root mean square error (RMSE). Formally defined as $\mathbf{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_{t,(i)} - \hat{y}_{t,(i)}\right)^2}$ where y_t is the mean of the predicted distribution at time t and $\hat{y}_t$ is the observed value at time t. The third evaluation metric is the standard deviation (SD) that is correlated to the uncertainty of the prediction and is denoted as $\mathbf{SD} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_{t,(i)} - \tilde{y}_t\right)^2}$ where $\bar{y}$ is the mean of the predicted distribution.

4.4 Experimental results

We provide two comprehensive comparisons and evaluations of the proposed AA-Forecast framework: the aforementioned 80–20 testing where 20% of the data are unseen, as well as the testing on zero-shot prediction where the whole time series is unseen. In both cases, we calculate the CRPS, RMSE, and SD. Lastly, we provided an ablation study to discuss the effectiveness of different AA-Forecast components.

The 80 − 20 testing. We first used the 'older' 80% of each time series in training and tested the accuracy of prediction on the rest of 20%. Table 4 reports the loss of the networks under such 80 − 20 testing, where the SD of AA-Forecast (GRU) method is lower than all baseline methods, showing the model's high confidence in the forecasts.

Table 4 Performance comparison of our proposed framework and baseline models under 80 − 20 testing

Methods	Metrics	Dataset		
		Electricity	COVID-19	Hurricane
ARIMA (Box and Pierce 1970)	CRPS	1.150	0.103	0.761
	RMSE	1.520	0.114	0.802
	SD	0.225	0.011	0.106
AE-LSTM (Sagheer and Kotb 2019)	CRPS	0.895	0.086	0.531
	RMSE	1.296	0.087	0.576
	SD	0.215	0.009	0.102
SARIMAX (Tarsitano and Amerise 2017)	CRPS	0.911	0.098	0.532
	RMSE	1.285	0.108	0.578
	SD	0.195	0.009	0.093
UberNN (Zhu and Laptev 2017)	CRPS	0.633	0.071	0.442
	RMSE	1.015	0.081	0.453
	SD	0.134	0.007	0.073
TSE-SC (Cai et al. 2020)	CRPS	0.583	0.062	0.384
	RMSE	0.983	0.072	0.423
	SD	0.146	0.007	0.092
AA-Forecast (LSTM)	CRPS	0.546	**0.059**	0.237
	RMSE	0.949	**0.068**	0.274
	SD	0.095	**0.003**	0.060
AA-Forecast (GRU)	CRPS	**0.493**	0.063	**0.216**
	RMSE	**0.894**	0.073	**0.253**
	SD	**0.081**	0.003	**0.051**

Bold values indicate the best performance

Among the baseline methods, UberNN and TSE-SC have shown good accuracy but suffer from higher SD (uncertainty) compared to the AA-Forecast (LSTM-GRU) models. Considering that the extracted features are available for all the baseline methods, we believe the higher uncertainty of SD is due to their static dropout probability that is constant for all time steps. Therefore, the two proposed models, AA-Forecast (LSTM-GRU), consistently outperform state-of-the-art methods. Considering all three evaluation metrics, AA-Forecast (GRU) is the best-suited framework for our dataset as it provides higher accuracy and confidence.

Zero-shot prediction: Table 5 demonstrates the zero-shot prediction abilities for the selected models. Both AA-Forecast (LSTM-GRU) predictions follow the observed time series in general. The prediction errors are comparably low during the presence of extreme events (i.e., hurricanes). This is mainly due to the anomaly attention mechanism developed to further reduce the prediction error during extreme events. Moreover, extracted anomalies from STAR decomposition led to the recall of the hurricane effects on previously seen regions, thus providing predictions for unseen time series data with a lower error given the presence of anomalies. Figure 5 showcases a

Table 5 Performance comparisons of zero-shot prediction abilities of models using ten randomly selected counties' sales tax data where they have not been used in training entirely

Methods	Metrics	Input time window			
		3	6	12	24
ARIMA (Box and Pierce 1970)	CRPS	0.893	0.891	0.861	0.831
	RMSE	0.934	0.932	0.922	0.872
	SD	0.119	0.1154	0.115	0.113
AE-LSTM (Sagheer and Kotb 2019)	CRPS	0.663	0.661	0.651	0.601
	RMSE	0.708	0.706	0.696	0.646
	SD	0.115	0.112	0.111	0.109
SARIMAX (Tarsitano and Amerise 2017)	CRPS	0.664	0.662	0.662	0.602
	RMSE	0.714	0.712	0.712	0.652
	SD	0.106	0.102	0.102	0.100
UberNN (Zhu and Laptev 2017)	CRPS	0.547	0.545	0.535	0.485
	RMSE	0.585	0.583	0.573	0.523
	SD	0.086	0.082	0.082	0.08
TSE-SC (Cai et al. 2020)	CRPS	0.766	0.764	0.754	0.704
	RMSE	0.795	0.793	0.783	0.733
	SD	0.105	0.102	0.101	0.099
AA-Forecast (LSTM)	CRPS	0.362	0.361	0.351	0.301
	RMSE	0.406	0.404	0.394	0.344
	SD	0.073	0.071	0.069	0.067
AA-Forecast (GRU)	CRPS	**0.348**	**0.346**	**0.336**	**0.286**
	RMSE	**0.385**	**0.383**	**0.373**	**0.323**
	SD	**0.064**	**0.060**	**0.062**	**0.058**

Bold values indicate the best performance

sample of these predictions for each model where for every time step, the prediction uncertainty is the least.

Given that the network did not train on the selected time series directly, it's able to transfer its knowledge from previously seen extreme events (i.e., the effect of cat 4 hurricanes) and provide more accurate prediction when not provided with such ability.

4.5 Ablation study

In this section, we provide an extensive analysis of the performance of AA-Forecast, as well as the impact of each component on the performance of AA-Forecast. The results are shown in Table 6 where we removed each component and reported the changes in accuracy and uncertainty.

Influence of anomaly-aware decomposition: To demonstrate that the anomaly-aware decomposition can aid in improving the time series prediction, we fed the input series to the prediction model directly. This modification resulted in the worst performance in our ablation study. Note that AA-Forecast (GRU) still benefits from dynamic dropout

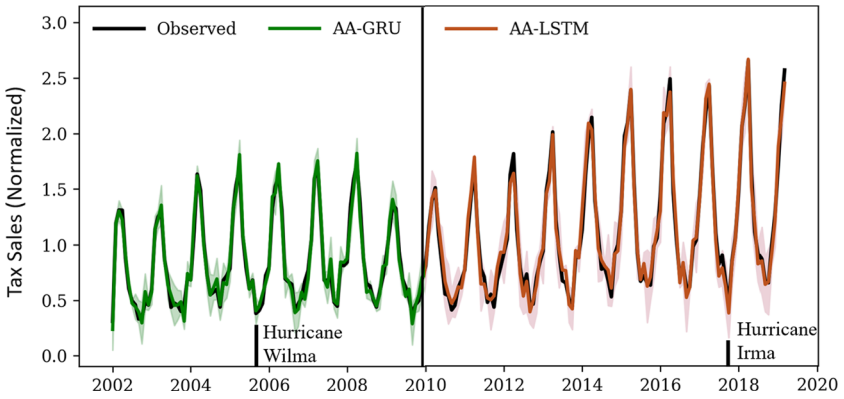

Fig. 5 Zero-shot prediction for hotel tax sales of Collier County, Florida, U.S. Both variations of AA-forecast are concatenated for demonstration

Table 6 Ablation study on AA-forecast (GRU) model using the sales tax dataset to show the effectiveness of its components

AA-Forecast (GRU)	Metrics	Time window			
		3	6	12	24
w/o STAR decomposition	CRPS	0.493	0.446	0.445	0.457
	RMSE	0.512	0.464	0.463	0.494
	SD	0.074	0.071	0.070	0.070
w/o Uncertainty optimization	CRPS	0.429	0.431	0.43	0.367
	RMSE	0.466	0.471	0.467	0.404
	SD	0.088	0.088	0.087	0.083
w/o Anomaly attention	CRPS	0.379	0.380	0.367	0.317
	RMSE	0.416	0.417	0.404	0.354
	SD	0.067	0.067	0.063	0.061
AA-forecast (GRU)	CRPS	**0.348**	**0.346**	**0.336**	**0.286**
	RMSE	**0.385**	**0.383**	**0.373**	**0.323**
	SD	**0.064**	**0.060**	**0.060**	**0.058**

Bold values indicate the best performance

optimization and extreme event labels, and the predicted uncertainty is optimized. However, the accuracy of AA-Forecast prediction (GRU) drops because of the limited number of features, indicating that the neural network does not have a strong ability to capture complex and nonlinear information. This can highlight the role of auxiliary features such as decomposed anomalies and extreme events for forecasting.

Influence of uncertainty optimization: We also used a static dropout throughout the experiments at every time step, which caused a substantial increase in SD. Uncertainty optimization of dropout plays a critical role in reducing the uncertainty of the forecast

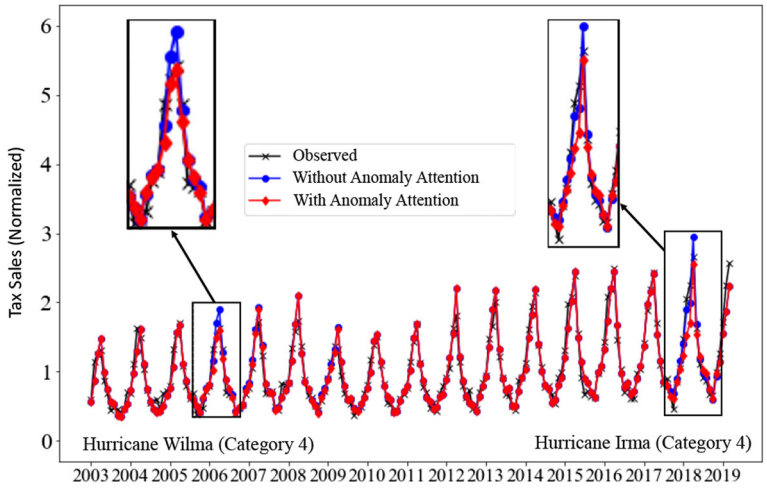

Fig. 6 Influence of anomaly attention on hurricanes. Two Category 4 hurricanes (Wilma and Irma) have caused similar annual sales losses. Anomaly-attention activation occurs during the presence of extreme events which makes it computationally efficient compared to the full-attention mechanism in transformers

intervals. Such modification also caused a higher error in the forecast, which is the model's inability to forecast with higher confidence.

Influence of anomaly attention: We conducted experiments to demonstrate the effectiveness of anomaly awareness through the network's attention mechanism. Specifically, we directly fed the extreme events and anomalies without the anomaly-attention mechanism described in Sect. 3.2. Such change makes limits AA-Forecast's knowledge about hurricanes and the severity of their effects. As an example, in Fig. 6 (right), the results show that the network's error during the presence of harder-to-predict time points (anomalies and extreme events) weakens.

Thus, removing the attention mechanism for anomalous/extreme event points of the dataset will reduce the performance of the model during the critical months of extreme events such as hurricanes. Simply relying on the previously seen dataset will not allow the network to handle external events and sudden changes effectively.

4.6 Discussion

Interpretation: The benefits of providing optimal uncertainty in prediction are twofold: first, it provides a systematic way to aid in resource allocation. Second, it further prepares the domain for interventions. For example, if one region receives more catastrophic extreme events, the resources can be transferred to that region. Moreover, government and industries can provide better-informed interventions and decisions (e.g., financial aid relief during COVID-19). As shown in the ablation study, including additional features such as extreme events and anomalous points can improve accuracy and better prime the model to handle predictions than deviate from trend or seasonality. Moreover, as shown in Fig. 6 without proper attention to these points, they result

in a large amount of error in forecasting. Given that such critical moments are of high importance during extreme events such as natural disasters, the performance of the model during critical time steps can be improved. Hence, it is essential to provide a thorough learning objective in our time series models to not only improve the overall performance but take critical moments into more consideration. Furthermore, allowing the model to provide its level of uncertainty establishes transparency and builds a level of trust for the users. Table 3 also showcases the runtime for the methods used in the study. Although the traditional method's accuracy and uncertainty are reported less than the deep learning methods, they still have better runtime efficiency. However, they contain few learnable parameters which result in lower models' capacity. Moreover, they are not able to share information sharing across regions for various time series.

Anomalies: As shown in Fig. 4, for the Hurricane case study, the anomalies start with the losses of the early 2000s Atlantic hurricane season. Interestingly, hurricane Irma 2017 did a similar catastrophic damage (77.16 billion) to the early 2000s season (Oceanic 2022). which allows the model to predict with higher accuracy when trained on previously sens effects of these anomalies. Similarly, for the COVID-19 dataset, the anomalies start by indicating the drastic changes in lockdown order which caused a great loss in the percentage of employment. These anomalies for each state play a critical role in future pandemics so that enough resources can be allocated to combat the losses (Selerio and Maglasang 2021). In the Electricity case study, note that the larger values of anomalies need to be carefully handled given that these points of high electricity load can lead to unplanned generation plant outages (Grace and Christiansen 2013).

Limitations & future directions: Although the dynamic dropout mechanism guarantees the least uncertainty in predictions, it cannot provide guarantees to do the same for prediction accuracy. This is due to the random nature of the dropout which we left as a future work where the dropout can appear for a predetermined distribution of the neurons. Therefore, maximizing the useful information contained in the multidimensional model serves to predict time series in extreme events. When it is not available, it's more reasonable to suggest methods that extract potential critical time steps such as anomalous points (e.g., STAR decomposition).

5 Related works

Anomalies in time series data often produce a high variance of uncertainty prediction that is difficult to predict, thus becoming a challenge for reliable model design (Zhu and Laptev 2017; Pang et al. 2017). To provide a more reliable forecast during the presence of anomalies, probabilistic forecasting methods are often studied, which can report a level of uncertainty (Li et al. 2019).

The majority of Bayesian Neural Networks in probabilistic forecasting require specific training and optimization methods and require additional model parameters that result in a larger amount of computation. Hence, MC dropout is preferred due to its practicability and its out-of-the-box solution (Zhu and Laptev 2017). Applying

standard dropout to Bayesian Neural Networks often results in poor performance on account of dropout noise preventing the network from maintaining long-term memory (Labach et al. 2019). Gal and Ghahramani (2016) proposed the MC dropout, in which the dropout can be interpreted as a sampling method that is equivalent to a variational approximation of a deep Gaussian process. MC dropout that is used for recurrent layers has proved to be successful and is commonly used in practice by applying dropout to recurrent connections in a way that can preserve long-term memory (Labach et al. 2019). In previous studies, static MC dropout was used throughout their experiments, which suffers the model's robustness toward the effect of anomalies. Given that probabilistic models still require an overall great accuracy of their forecasts, optimizing the uncertainty in prediction intervals remains a challenging question

6 Conclusion

We propose an anomaly-aware time series prediction framework, namely AA-Forecast, to capture and leverage the effect of extreme events and anomalies for the time series prediction task. It features a novel anomaly decomposition method that also extracts the essential features of the data. We also proposed an anomaly-aware model to leverage the extracted anomalies through an attention mechanism. Moreover, we reduced the uncertainty of the network without any further training so that the prediction uncertainty is minimal through the testing state. We compare our framework with several statistical and deep learning models using three real-world time series datasets. The results show that the AA-Forecast framework outperforms these models in prediction error and uncertainty. For future work, the prediction performance could be further improved if we target specific groups of neurons (e.g., the neurons containing unnecessary details of the time series dynamics) for dynamic dropout optimization.

Declarations

Conflict of interest We disclose the following potential conflicts of interest by declaring all the institutions with their corresponding email domains: University of Central Florida, USA @ucf.edu; North Carolina State University, USA @ncsu.edu; Baidu Inc., China, @baidu.com.

References

Adilova L, Chen S, Kamp M (2021) Novelty detection in sequential data by informed clustering and modeling. arXiv preprint arXiv:2103.03943

Bartik AW, Bertrand M, Lin F, Rothstein J, Unrath M (2020) Labor market impacts of covid-19 on hourly workers in small-and medium-sized businesses: four facts from homebase data. Institute for Research on Labor and Employment

Box GE, Pierce DA (1970) Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. J Am Stat Assoc 65(332):1509–1526

Broska LH, Poganietz WR, Vögele S (2020) Extreme events defined—a conceptual discussion applying a complex systems approach. Futures 115:102490

Cai L, Janowicz K, Mai G, Yan B, Zhu R (2020) Traffic transformer: capturing the continuity and periodicity of time series for traffic forecasting. Trans GIS 24(3):736–755

Cleveland RB, Cleveland WS, McRae JE, Terpenning I (1990) Stl: a seasonal-trend decomposition. J Off Stat 6(1):3–73
Cleveland WS (1979) Robust locally weighted regression and smoothing scatterplots. J Am Stat Assoc 74(368):829–836
Dokumentov A, Hyndman RJ (2020) Str: A seasonal-trend decomposition procedure based on regression. arXiv preprint arXiv:2009.05894
Gal Y, Ghahramani Z (2016) Dropout as a bayesian approximation: representing model uncertainty in deep learning. In: international conference on machine learning, pp. 1050–1059
Grace D, Christiansen T (2013) Risk-based assessment of unplanned outage events and costs for combined-cycle plants. J Eng Gas Turbines Power **135**(2)
Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
Jing M, Ng KY, Mac Namee B, Biglarbeigi P, Brisk R, Bond R, Finlay D, McLaughlin J (2021) Covid-19 modelling by time-varying transmission rate associated with mobility trend of driving via apple maps. J Biomedi Inform 122:103905
Khan A, Bibi S, Lyu J, Latif A, Lorenzo A (2021) Covid-19 and sectoral employment trends: assessing resilience in the us leisure and hospitality industry. Curr Issues Tour 24(7):952–969
Khandelwal U, He H, Qi P, Jurafsky D (2018) Sharp nearby, fuzzy far away: How neural language models use context. arXiv preprint arXiv:1805.04623
Labach A, Salehinejad H, Valaee S (2019) Survey of dropout methods for deep neural networks. arXiv preprint arXiv:1904.13310
Laptev N, Yosinski J, Li LE, Smyl S (2017) Time-series extreme event forecasting with neural networks at uber. In: International Conference on Machine Learning, vol. 34, pp. 1–5
Li L, Yan J, Yang X, Jin Y (2019) Learning interpretable deep state space model for probabilistic time series forecasting. In: IJCAI, pp. 2901–2908
Li S, Jin X, Xuan Y, Zhou X, Chen W, Wang YX, Yan X (2019) Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. In: Advances in Neural Information Processing Systems, pp. 5243–5253
Lim B, Arik SO, Loeff N, Pfister T (2019) Temporal fusion transformers for interpretable multi-horizon time series forecasting. arXiv preprint arXiv:1912.09363
Oceanic N, Administration A Nhc data archive. https://www.nhc.noaa.gov/data/. (Accessed on 08/20/2022)
Pang J, Liu D, Peng Y, Peng X (2017) Anomaly detection based on uncertainty fusion for univariate monitoring series. Measurement 95:280–292
Rolnick D, Donti PL, Kaack LH, Kochanski K, Lacoste A, Sankaran K, Ross AS, Milojevic-Dupont N, Jaques N, Waldman-Brown A, et al (2019) Tackling climate change with machine learning. arXiv preprint arXiv:1906.05433
Sagheer A, Kotb M (2019) Unsupervised pre-training of a deep lstm-based stacked autoencoder for multivariate time series forecasting problems. Sci Rep 9(1):1–16
Salinas D, Flunkert V, Gasthaus J, Januschowski T (2020) Deepar: probabilistic forecasting with autoregressive recurrent networks. Int J Forecast 36(3):1181–1191
Santos-Burgoa C, Sandberg J, Suárez E, Goldman-Hawes A, Zeger S, Garcia-Meza A, Pérez CM, Estrada-Merly N, Colón-Ramos U, Nazario CM et al (2018) Differential and persistent risk of excess mortality from hurricane maria in puerto rico: a time-series analysis. Lancet Planet Health 2(11):e478–e488
Selerio E, Maglasang R (2021) Minimizing production loss consequent to disasters using a subsidy optimization model: a pandemic case. Struct Chang Econ Dyn 58:112–124
Smith AB (2022) 2021 us billion dollar weather and climate disasters in historical context including new county-level exposure, vulnerability and projected damage mapping. In: 102nd American Meteorological Society Annual Meeting. AMS
Tarsitano A, Amerise IL (2017) Short-term load forecasting using a two-stage sarimax model. Energy 133:108–114
Tolstikhin IO, Houlsby N, Kolesnikov A, Beyer L, Zhai X, Unterthiner T, Yung J, Steiner A, Keysers D, Uszkoreit J et al (2021) Mlp-mixer: an all-mlp architecture for vision. Adv Neural Inform Process Syst 34:24261–24272
Wahab H, Jain V, Tyrrell AS, Seas MA, Kotthoff L, Johnson PA (2020) Machine-learning-assisted fabrication: Bayesian optimization of laser-induced graphene patterning using in-situ raman analysis. Carbon 167:609–619

Wen Q, Gao J, Song X, Sun L, Xu H, Zhu S (2019) Robuststl: A robust seasonal-trend decomposition algorithm for long time series. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 5409–5416

Zaheer M, Guruganesh G, Dubey A, Ainslie J, Alberti C, Ontanon S, Pham P, Ravula A, Wang Q, Yang L, et al (2020) Big bird: transformers for longer sequences. arXiv preprint arXiv:2007.14062

Zhu L, Laptev N (2017) Deep and confident prediction for time series at uber. In: 2017 IEEE International Conference on Data Mining Workshops (ICDMW), pp. 103–110. IEEE