eForecaster: Unifying Electricity Forecasting with Robust, Flexible, and Explainable Machine Learning Algorithms

Zhaoyang Zhu[1*], Weiqi Chen[1*], Rui Xia[1], Tian Zhou[1], Peisong Niu[1], Bingqing Peng[1], Wenwei Wang[1], Hengbo Liu[1], Ziqing Ma[1], Qingsong Wen[2], Liang Sun[2†]

[1]DAMO Academy, Alibaba Group, Hangzhou, China
[2]DAMO Academy, Alibaba Group, Bellevue, WA, USA
{zhuzhaoyang.zzy, jarvus.cwq, guangyao.xr, tian.zt, niupeisong.nps, pengbingqing.pbq, duoluo.www, yangyang.lhb, maziqing.mzq, qingsong.wen, liang.sun}@alibaba-inc.com

Abstract

Electricity forecasting is crucial in scheduling and planning of future electric load, so as to improve the reliability and safeness of the power grid. Despite recent developments of forecasting algorithms in the machine learning community, there is a lack of general and advanced algorithms specifically considering requirements from the power industry perspective. In this paper, we present eForecaster, a unified AI platform including robust, flexible, and explainable machine learning algorithms for diversified electricity forecasting applications. Since Oct. 2021, multiple commercial bus load, system load, and renewable energy forecasting systems built upon eForecaster have been deployed in seven provinces of China. The deployed systems consistently reduce the average Mean Absolute Error (MAE) by 39.8% to 77.0%, with reduced manual work and explainable guidance. In particular, eForecaster also integrates multiple interpretation methods to uncover the working mechanism of the predictive models, which significantly improves forecasts adoption and user satisfaction.

Introduction

What will be the electricity consumption of a city in the next few days if extreme weather occurs (e.g., temperature dramatically increases, or heavy rain comes)? What will be the electricity consumption of an office building in the face of weather conditions and holidays? How much power will a wind farm generate given the fluctuating wind speed and maintenance plan of a few wind turbines tomorrow? These questions are typical examples of the well-known electricity forecasting problem (Nti et al. 2020), which is crucial for reliable and efficient operation, management, and planning of a power grid system.

In particular, electricity forecasting can be viewed as an instance of time series forecasting problem, and a large number of researches have focused on addressing both general time series forecasting (Salinas et al. 2020) and electricity forecasting (Song et al. 2017; Hooi et al. 2018). Despite the advances, these methods are brutal to utilize straightforwardly in practice, as they are often formulated, solved, and implemented as ideal mathematical problems and may not acknowledge the complexities in real-world scenarios. Thus, orchestrating electricity forecasting applications (e.g., bus load forecasting, system load forecasting, and renewable energy forecasting) in the real world is quite challenging. Moreover, decision-making in the electricity system is high-risked since the prediction results influence both future assessments and the balance of subsequent power dispatch. Hence, a more explainable forecast is necessary to narrow the trust gap between algorithms and experts.

Challenges. In summary, our main goal is to design a comprehensive, systematic, and unified AI platform to facilitate the development and deployment of various real-world electricity forecasting applications with minimum human interference. We summarize the challenges as follows:

- Multi-source electricity(-relevant) data are suffused with outliers, noises, and missing values due to sensing, acquisition, and recording errors, hindering the accuracy and reliability of the downstream forecasting.
- Electricity forecasting applications are diversified due to heterogeneous scenarios and complicated data characteristics. Thus, they entail an optimal selection of huge design space of data modeling.
- The deployed models often prioritize forecasting performance but lack explanations of how predictions are made. In such a high-risked domain, making the forecasts trustworthy and user-friendly is essential.

Contributions. We tackle the aforementioned three challenges simultaneously within the proposed eForecaster (see Figure 1), a unified AI platform including robust, flexible, and explainable machine learning algorithms covering major electricity forecasting applications. Specifically, 1) eForecaster supports a collection of temporal/static and structured/unstructured electricity(-relevant) data, accompanied by an elaborated data preprocessing module, especially for electricity time series, where data errors often occur; 2) eForecaster provides plenty of functional modules, ranging from feature engineering and forecasting models to domain-knowledge-infused postprocessing. These modules can be assembled into a pipeline using manual selection or optimization interface; and 3) eForecaster also integrates multiple interpretation techniques to uncover the working mech-

*These authors contributed equally.
†Corresponding author.

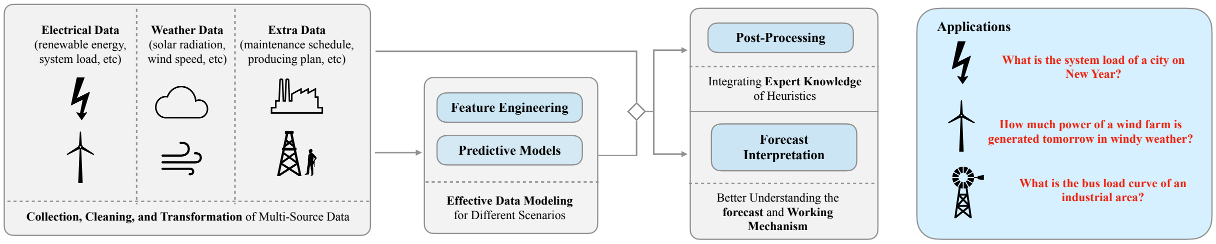

Figure 1: eForecaster Overview. A typical solution formulation for a specific instance of electricity forecasting problem involves (1) collecting, cleaning, transforming multi-source data, (2) performing effective data modelling, (3) explaining the results and integrating domain-knowledge into the forecasts. Abstracting out the problem commonalities, eForecaster provides a unified approach for solving different electricity forecasting problems.

anism of the forecasting models, with various presentation forms designed for different applications.

Deployment Status and Payoff. Since Oct. 2021, we have deployed electricity forecasting systems supported by eForecaster for seven provinces in China, including Shandong, Zhejiang, Jiangxi, Shanxi, Henan, Hunan, and Hebei. These systems now provide bus load forecasting for more than 1000 buses, system load forecasting for more than 35 cities which consumes near 9000 billion kilowatt hours of electric power per year for a population over 200 million, and renewable energy forecasting for more than 400 wind farms and 400 photovoltaic plants across these seven provinces. These renewable energy plants generate more than 1.3×10^{11} kWh per year which is about 13% of the total renewable energy generated in the year 2021 in China. Furthermore, our systems have significantly improved the forecasting accuracy in all scenarios, compared to the previous approaches used in practice. For instance, the deployed bus load forecasting system has replaced manual predictions by experts in Shandong province, and the average forecasting accuracy has increased from 97.1% to 97.6% (which is a significant improvement for the particular accuracy metric). In Zhejiang province, the percentage of daily forecast accuracy of wind farms below the assessment standard has been reduced from 20.7% to 8.4%, which helps in managing power generation and improving the safeness of the power grid. As a unified electricity forecasting solution, Alibaba Cloud released DAMO Academy's Load and Renewable Energy Forecast Product at the 2022 Global Digital Economy Industry Conference[1] concluded recently.

Electricity Forecasting Problem

The goal of electricity forecasting is to predict the future electricity consumption or generation $\mathbf{x}_{t+1:t+T}$ given the previous observations $\mathbf{x}_{t-T'+1:t}$ and the corresponding covariates (e.g., weather conditions, time stamps, holidays, external events, etc.), which can be divided into temporal covariates $\mathbf{c}_{t-T'+1:t+T}$ and static covariates $\mathbf{s}$. Formally, the electricity forecasting aims to learn a function $f(\cdot)$ that

$$[\mathbf{x}_{t-T'+1:t}; \mathbf{c}_{t-T'+1:t+T}; \mathbf{s}] \xrightarrow{f(\cdot)} \mathbf{x}_{t+1:t+T},$$

where the temporal covariates $\mathbf{c}_{t-T'+1:t+T}$ span the past and future as they can be predicted (e.g., weather conditions) or scheduled (e.g., holidays), and t is the current time step as well as T' and T stand for the input and output horizon, respectively. Note that the function $f(\cdot)$ can be either a machine learning model or a complicated composite including data transformation, feature engineering, and data modeling. A plethora of methods-based researches (Song et al. 2017; Hooi et al. 2018) focuses on the former route while may not acknowledge the complexities in real-world electricity forecasting applications as in our discussion.

eForecaster Platform

As aforementioned, solving a real-world electricity forecasting problem is prohibitive due to the complexity of processing incomplete data and the difficulty of modeling and understanding the forecast. eForecaster aims to overcome these challenges by providing a composable, flexible, unified AI platform for developing different electricity forecasting applications. Specifically, as shown in Figure 2, with eForecaster, developers can implement an end-to-end forecasting pipeline composed of three phases, namely pre, in, and post-modeling. In the pre-modeling phase, robust preprocessing methods (e.g., time series decomposition and data imputation) are provided for data cleaning and exploratory data analysis (EDA). Afterwards, state-of-the-art deep time series forecasting models and various statistical machine learning models, accompanied with domain-specific feature engineering modules, are used to solve data modeling problems for different scenarios. Next, the post-processing module is used to refine the model's forecasts using heuristic rule-based methods. At last, the forecast interpretation module helps users to better understand the forecasts. In addition, the pipeline composer solves the pipeline selection and configuration problem and organizes individual components into a complete end-to-end pipeline for a specific application.

[1]https://m.gdecexpo.com/

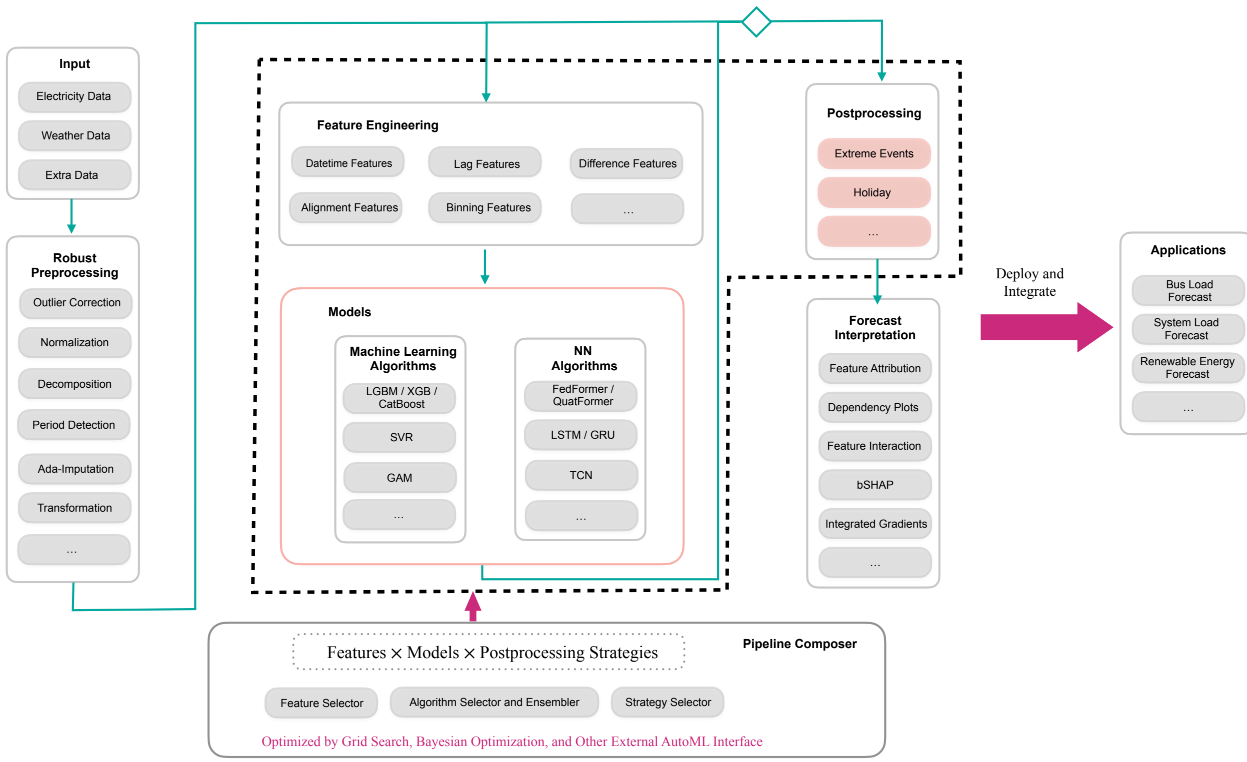

Figure 2: eForecaster Framework. The grey boxes depict the data modeling pipeline for electricity forecasting, $\mathrm{data\ input} \rightarrow \mathrm{preprocessing} \rightarrow \mathrm{feature\ engineering} \rightarrow \mathrm{models} \rightarrow \mathrm{postprocessing} \rightarrow \mathrm{forecast\ interpretation}$. The blue box is an illustration of the pipeline composer, which is used to automatically or manually select feature engineering methods, models, and postprocessing strategies, and compose all selected modules into an eForecaster pipeline.

Pre-Modeling

Input Data The input data module contains data loaders that are responsible for reading, parsing, combining raw data, and converting them into pandas DataFrame objects (McKinney et al. 2011). Raw data includes temporal data such as electricity load, weather data (e.g. temperature, wind speed, and solar radiation), and operation data (e.g., electrical equipment's maintenance schedule and factories' producing plan), as well as static data (e.g., installed capacity of a wind farm and types of areas). Our deployed forecasting systems have access to the data in a production environment and input them to this input data module.

Robust Preprocessing The robust preprocessing module is designed for exploratory data analysis and data cleaning. It contains periodicity detection, robust seasonal-trend decomposition, trend filtering, outlier detection methods such as RobustPeriod (Wen et al. 2021), RobustSTL (Wen et al. 2019b, 2020; Yang et al. 2021), RobustTrend (Wen et al. 2019a), ℓ_1 Trend Filtering (Kim et al. 2009), Hampel Filters (Pearson et al. 2016), etc. Note that many of these algorithms are developed by our team to build the automatic pipeline. Furthermore, the above data analysis methods can be combined with normalization and imputation methods to provide adaptive and effective data preprocessing. Specifically, for different electricity time series data, providing the detected period length, missing values are imputed according to their period information using different strategies (e.g., periodic imputation, which fills missing values using the values from the previous week if weekly periodicity exists).

In-Modeling

Feature Engineering The feature engineering module generalizes standard time series feature engineering and domain-specific feature engineering methods that integrate data characteristics and domain knowledge. It is worth noting that, besides commonly used time series feature engineering including lag, difference, and rolling statistics features, we introduce an alignment feature engineering that aligns each timestamp to a group of similar time intervals given a set of reference time series (e.g., similar bus series in the same city), a sequence of rules (e.g., weather conditions, time index) and any measurement of similarity. These feature engineering techniques provide informative, discriminative understanding of diversified problems.

Models An abundant set of models are provided and maintained: 1) State-of-the-art deep learning models, including internally developed ones (e.g., Quatformer (Chen et al. 2022) and FEDformer (Zhou et al. 2022)), and popular ones (e.g., TCN (Bai, Kolter, and Koltun 2018) and LSTM (Hochreiter and Schmidhuber 1997)); 2) Traditional machine learning models such as CatBoost (Dorogush, Ershov, and Gulin 2018), LightGBM (Ke et al. 2017), and XGBoost (Chen and Guestrin 2016). It is worth noting that models, which specialize in modeling problems in specific scenarios, such as a highly explainable generalized additive model (GAM) for extreme weather generalization and exponential moving average on history time intervals for a volatile, fluctuating factory load, are also included. Different models can easily be aggregated using voting, stacking, and other assembling methods in the pipeline composer (which will be described in detail later).

Post-Modeling

Postprocessing Postprocessing is a major module in eForecaster. It blends domain knowledge with model forecasts by acting directly on the model outputs. This module integrates *Rule-based Methods* containing domain rules distilled from years of experience of domain experts (e.g. special production plans for a certain type of factories during holidays). With prior knowledge of the problem and a set of domain rules, predictions become more reasonable under realistic settings that satisfy certain constraints (e.g., power outages), aiding forecast quality and interpretability.

Forecast Interpretation The forecast interpretation module provides both model-agnostic and model-specific explanation methods to uncover the working mechanism of the learned black-box model. We elaborate on two primary forecast interpretation strategies.

Feature Attribution. We take inspiration from the manual forecasting routine where a similar historical date is firstly chosen, and its corresponding load is then modified towards the prediction according to changes in weather conditions or time features. The difference between a prediction and a historical observation is distributed back to each input feature by reversing the procedure. Particularly, each feature attribution $\{\phi_i\}_{i=1}^D$ would add up to be the difference $\sum_i \phi_i = \Delta = f(\mathbf{x}_t) - y_b$, where f is the learned predictive model, $\mathbf{x}_t$ is the input feature of the explained point, and y_t is the load observed at some historical time. Using a totally randomized tree and its connection to Shapley Value (Sutera et al. 2021) combined with sampling strategies, the difference Δ can be efficiently decomposed to each feature.

Dependency Plots. Apart from local explanations where each instance has its own decomposition, the global trend of each feature is also necessary. Since the correlations with other variables cannot be ignored, a single plot of dependency would be crude and with high variance. Hence, the interaction between variables is achieved by filtering such that a plot is drawn under certain conditions satisfied by correlated features.

With these strategies mentioned above, users can comprehend how the learned model gives a particular prediction and why sometimes they are inconsistent with human decisions. Local explanations like feature attribution provide detailed cause analysis and break down the prediction at a particular time and situation, while global explanations like dependency plots summarize model behaviors from a broader range in an accessible and concise way.

Pipeline Composer

As a unified approach to various electricity forecasting applications, the main goal of the pipeline composer is to help developers efficiently and effectively select an optimal pipeline from large design space for a specific application. On the one hand, several default pipelines are built for typical applications including 1) industrial, residential, and commercial load forecasting, 2) system load (the total electricity consumption of a large area, such as a city, which is a mixture of different types of load) forecasting, and 3) renewable energy forecasting of photovoltaic (PV) power plants and wind farms. On the other hand, Grouping Feature Selector, Model Selector, Ensembler, and Postprocessing Strategy Selector are integrated for automating pipeline selection.

Grouping Feature Selector. As different electricity data have different types of impact factors (e.g., the residential load is correlated with the weather, but the industrial load is probably not), we can process a group of features of the same type[2] simultaneously, which effectively prunes the search space. Thus we enhance existing feature selection methods (e.g., recursive feature elimination, sequential feature selection, etc.) with grouping mechanism.

Model Selector and Ensembler. The predictive model is either a single model or model ensembles. Model selector can select models according to validation results and model ensembler implements ensembling strategies including voting and stacking.

Postprocessing Strategy Selector. This selector helps to select suitable postprocessing strategies for specific business targets of interest.

As candidates of each selector or ensembler are set, models, features, and hyperparameters can be jointly optimized using grid search or Bayesian optimization for the optimal pipeline. Moreover, pipeline composer can be wrapped with any autoML toolkit for additional flexibility.

Remarks. The design of different modules of eForecaster follows closely with the *fit-transform-predict* paradigm of scikit-learn (Varoquaux et al. 2015; Jarrett et al. 2021). Moreover, new implementations of algorithms can be integrated into the framework simply by inheriting specific Base classes in eForecaster. We introduce eForecaster as the foundation of various electricity forecasting applications which integrates abundant machine learning algorithms considering both the commonality and individuality of diversified scenarios. An eForecaster pipeline can be implemented straightforwardly with only a few configurations (e.g., determining specific components at each phase and defining search spaces for selectors of pipeline composer). The resulting pipeline also provides training, inference, and forecast interpretation interfaces, which are used in deployed forecasting systems. In the next section, we will elaborate on the development and deployment of real-world forecasting systems for typical applications, which are significantly facilitated by eForecaster.

[2]We define features generated from the same data source using the same set of feature engineering methods to be the same type.

Deployed Forecasting Systems

We have developed and deployed electricity forecasting systems with user interface (UI) for different electricity applications. These forecasting systems have access to the power grid data center to synchronize real-time electricity load data, and invoke the training and inference of eForecaster pipelines. The user interface is designed to display various data and forecasts, and assist users with data management (e.g., load-transfer schedule, maintenance schedule, and industrial producing plan). A deployed forecasting system for a specific electricity application has three layers: 1) a data access layer that provides an interface for collecting load data, weather prediction data, and extra operation data; 2) an application layer that provides extra functions such as user management tools and data visualization tools; and 3) an algorithm layer supported by eForecaster, which acts as the foundation for the system and endows the system the ability to forecast by using eForecaster pipelines.

We utilize the portability and flexibility of docker containers and containerize eForecaster pipelines into various docker containers to provide different functions (i.e., model training, inference, and forecast interpretation). These docker containers with different functions are arranged into scheduling tasks managed by distributed scheduling framework (e.g., XXL-JOB). Kubernetes is used to manage these docker images and containers, and MySQL is used as our database management system. In general, the deployment procedure of forecasting systems in the production environment consists of three steps: 1) install necessary components (i.e., Docker, Kubernetes, MySQL); 2) deploy scheduled tasks including training, inference (predicting), and forecast interpretation; and 3) retrieve, parse, and store results from tasks and display them on the user interface. These forecasting systems are developed for various applications.

Application 1 (System Load Forecasting) The task of system load forecasting is to predict the total electricity consumption of a large area, such as a city, for specific days in the future. We often encounter noisy or missing data problems due to environmental or software faults during the data collection. These problems disturb the distribution of training data and hurt the performance of most of our machine learning-based models.

In deployed system load forecasting systems, the preprocessing module is incorporated to perform the data imputation and outlier removal. Specifically, we decompose the time series into trend, season and residual components using seasonal-trend decomposition methods aforementioned. A Gaussian filter is performed on the residuals to avoid outliers and a periodic imputation is introduced to fill the missing values if periodicity exists.

Another major challenge is the out-of-distribution forecasting. For instance, unprecedented extreme weather leads to unprecedented electricity consumption characteristics, e.g., the significantly higher electricity consumption caused by the intense cold wave. Thus, we use an ensemble model combining the gradient boosting methods (e.g., XgBoost and LightGBM) and deep neural network models, as these two categories of models are often complementary in empirical deployment. Gradient boosting methods are inferior at dealing with out-of-distribution forecasting, although they have excellent performance on regular days. However, deep learning models can directly model the relationship between temperature and electricity consumption with great extrapolation. The ensemble model can utilize different models' advantages in this situation and improve our accuracy in load forecasting tasks. Besides, for the gradient boosting methods, a series of delicated feature engineerings are designed to incorporate some practical domain knowledge.

Since early 2022, our system load forecasting systems have been deployed over 5 provinces in China including Hubei, Hunan, Henan, Jiangxi, and Zhejiang. They are used by the province and city-level authorities to assist their daily decisions.

Application 2 (Bus Load Forecasting) The task of bus load forecasting is quite different from system load prediction. This task describes a more microscopic problem of forecasting electricity consumption of a relatively small region of different functions (e.g., industrial, residential, and commercial area), which is more sensitive to various external factors including industrial producing plan, power outage, load-transfer, and fine-grained weather conditions. Practically, we put more efforts to access and process these different types of data in bus load forecasting. Grid-wise numeric weather prediction, maintenance schedules of electrical equipments, and topological structure of the power grid are involved, some of which rely on regular data import by users. Moreover, bus load forecasting is challenging as characteristics from different buses are quite different. As a result, developers need to perform onerous experiments for data modeling previously, while eForecaster can resolve this issue using its automated pipeline composer.

In deployed bus load forecasting systems, we have used separate modeling strategies where one eForecaster pipeline has been built specifically for a bus based on its characteristics. In the pre-modeling phase, we perform data preprocessing on bus load times series similar to system load including outlier correction and missing data imputation, while the difference is that we keep those outliers corresponding to extreme events and special holidays, and only remove those introduced during data sensing and collection. Due to the substantial number of buses and limited computational resources in the production environment, we only consider ensembles of non-deep machine learning models (e.g., LightGBM, CatBoost, and Random Forests) during automatic model selection. Then, post-modeling (i.e., Post-processing) phases are automatically selected and combined with pre and in-modeling phases into end-to-end pipelines using pipeline composer. Thus, all deployed eForecaster pipelines are generated automatically. To better manage the topological structure of the power grid, we have used graph database Neo4j, and to boost the training process of eForecaster pipelines, we have used distributed storage filesystem Glusterfs.

Since Oct. 2021, our bus forecasting systems have been deployed across three provinces of China (i.e., Shanxi, Zhejiang, and Shandong) covering more than 1000 buses. They have been used by city and county-level authorities to reduce the majority of manual works and help them to make decisions.

Application 3 (Renewable Energy Forecasting) In this application, we focus on short-term centralized wind and solar power and distributed solar power forecasting with forecasting horizons from several hours to 10 days. While renewable energy can effectively alleviate human dependence on fossil fuels, the roaring capacity installment of renewable energy has posed tremendous challenges to power grid management due to its intermittent nature. Therefore, accurate short-term forecasting is essential to power systems management and security. Unlike traditional time series forecasting problems, historic patterns generally are less informative in days ahead forecasting scenarios considering the high dependency on weather conditions (e.g., wind speed and cloud coverage). Instead, numeric weather prediction (NWP) becomes the most predictive input, which to some extent simplifies the forecasting task to the regression problem mainly dependent on NWP.

In terms of deployment, the system first automatically selects the latest available NWP as input features. Outliers and noise are filtered using methods from preprocessing module. The NWP is then fed into deployed models stored in a distributed file system or cloud storage which is typically Aliyun OSS (Amazon S3 alternative). For centralized forecasting, we have trained both tree-based models with numerous features powered by the Feature Engineering module and neural networks free of feature engineering for each site offline. We adopt end-to-end neural networks for distributed solar forecasting, using various CNNs rolling over area NWP grids as spatial embeddings. Moreover, the system enables users to upload such schedules ahead and makes proper adjustments to the final forecast.

The renewable energy forecasting systems have been operating smoothly in Zhejiang and Shandong provinces since Jan. 2022. The upcoming 8 days' forecasts at various aggregation levels are disseminated every morning at 6 am local time. Visualized through the User Interface, operators can download the forecasts for reporting and dispatching.

Maintenance

As time goes by, the inherent data distribution may drift and the performance of models trained with early data degrades (namely, model aging problem), which poses a huge challenge for the maintenance of such applications (You et al. 2021). To tackle this issue, we introduce data elimination and pipeline retraining on top of the electricity forecasting application, which has been set as scheduled tasks (weekly or monthly) in deployed systems. In real deployed systems, we hold a continuous history of two years. We use a window that contains one month of the most recent data to represent recent temporal patterns (e.g., 2022-09), and a reference window that contains one month of the oldest data of the same period to represent old temporal patterns (e.g., 2020-09). If they differ dramatically in temporal characteristics with respect to a measure of distance (e.g., DTW), we would remove the history up to and including the data in the reference window. Then, the rest of the history is used for retraining the eForecaster pipelines. When pipeline retraining is performed, besides retraining the whole pipeline, fine-tuning would also be triggered to carry out minor revisions of underlying model hyperparameters.

Experiment

In this section, we conduct experiments to answer the following questions:

- **Q1. Forecasting performance:** Can eForecaster promote the prediction performance in various applications (i.e., bus load forecasting, system load forecasting, and renewable energy forecasting)?
- **Q2. Effectiveness of major modules:** How effective are major modules of eForecaster, including Robust Preprocessing and Postprocessing?
- **Q3. Data insights and forecast interpretation:** How can eForecaster help developers perform data analysis and modeling and facilitate users to understand the forecasts?

All results of eForecaster are real performances retrieved from deployed systems. We reproduce experiments of baselines offline with training data of the same period and compare their performances with online performances of eForecaster pipelines. The retrieved datasets are described in the following section.

Data

- **Bus Load** This dataset contains three typical challenging bus load time series (Industrial, Residential, and Commercial) from Shandong Grid, ranging from January 1, 2020, to July 31, 2022.
- **System Load** This dataset contains four system load time series from different provincial power grids (Jiangxi, Henan, Hunan, Hubei) and the load of the whole four provinces (Central China), ranging from January 1, 2020, to July 31, 2022.
- **Renewable Energy** This dataset contains wind power data from a wind farm and distributed solar power data from a city in Zhejiang province, ranging from January 1, 2021 to June 30, 2022.

For the bus load forecasting, we evaluate performance on the last three months. For system load and renewable energy forecasting, the results are evaluated on the last month. The time frequency is 15 minutes for all tasks.

Comparison Experiments (Q1)

In this section, we investigate the performance of eForecaster pipeline compared with commonly used electricity forecasting methods in different applications.

Baselines To demonstrate the effectiveness and robustness of eForecaster under different settings, we consider three common baselines: LightGBM (LGBM) with feature engineering (Ke et al. 2017), LSTM (Hochreiter and Schmidhuber 1997), and TCN (Bai, Kolter, and Koltun 2018).

	eForecaster		LSTM		TCN		LightGBM	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Industrial Load	**15.8**	**26.2**	23.8	28.6	25.3	29.4	21.385	26.7
Commercial Load	**5.8**	**9.0**	15.1	19.1	18.6	22.0	19.5	23.3
Residential Load	**7.1**	**10.6**	12.5	15.3	14.6	18.4	11.8	15.3

Table 1: Bus load forecasting results on three typical, challenging bus load time series. A lower MAE and RMSE indicate better performance. The best results are highlighted in bold.

	eForecaster		LSTM		TCN		LightGBM	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Jiangxi	**81.5**	**105.5**	356.9	438.7	355.8	439.3	468.4	504.7
Henan	**278.9**	**373.6**	983.1	1203.1	736.4	908.2	598.2	726.4
Hunan	**118.1**	**155.7**	485.0	598.4	480.9	591.0	544.9	602.8
Hubei	**157.8**	**219.4**	649.1	804.8	534.4	662.6	431.9	544.6
Central China	**462.8**	**589.1**	2378.8	2916.6	1797.6	2220.7	1412.5	1693.1

Table 2: System load forecasting results on five different regions with prediction length 96 (one day). A lower MAE and RMSE indicate better performance. The best results are highlighted in bold.

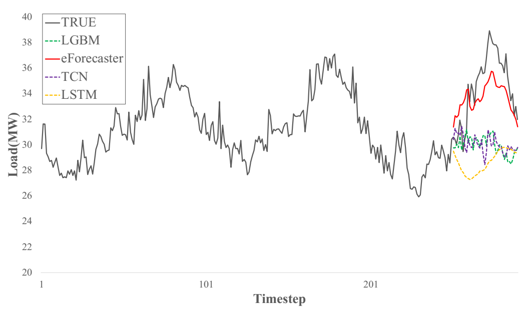

Figure 3: eForecaster (*red solid line*) can capture overall pattern under challenging industrial bus load scenario. All baselines are trained, tuned on two years of history and evaluated on the last 48 timesteps. The eForecaster pipeline result was retrieved from real deployed forecasting system.

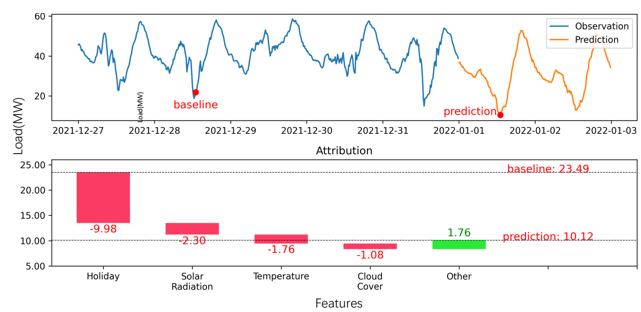

Figure 4: Upper: Load observation before new year and prediction after new year. Lower: Feature attribution between load prediction at 2022/01/02 13:00:00 and observation at 2021/12/28 13:00:00. *Holiday* acts as the most important feature that pulls down load for 9.98 MW.

Experiment settings

- **Bus Load** We train separate baseline models for each bus load time series. LSTM, TCN are trained using Adam optimizer (Kingma and Ba 2014) and L2 loss.
- **System Load** We similarly train baseline models (LSTM, TCN, LightGBM) for each provincial power grid and Central China system load time series.
- **Renewable Energy** We apply the same baseline models, including LightGBM, LSTM, and TCN.

All baseline models are tuned on the validation set, as this tuning procedure is also implemented in deployed systems. For evaluation metrics, we use Root Mean Square Error (RMSE) and Mean Absolute Error (MAE).

Results The results of comparison studies are shown in Table 1, 2, and 3. In all scenarios, eForecaster outperforms the baselines by a large margin, with at least 39.8% lower MAE in the bus load dataset, 77% lower MAE in the system load dataset, and 2.2% lower MAE in the renewable energy dataset. These results indicate that eForecaster pipelines can surely enhance forecasting performance under these three challenging settings.

Ablation Studies(Q2)

Using methods from the robust preprocessing module, we can detect and correct both point outliers and sub-sequence noises, and also impute missing values. Table 4 verifies that robust preprocessing module can promote predictive performance. Moreover, we can observe that this is even more effective for Industrial Load (45% reduction in MAE) than Commercial Load (13% reduction in MAE) and Residential Load (21% reduction in MAE). This is mainly because there are more noises in Industrial data due to volatility and uncertainty in industrial production.

Postprocessing For system load time series, postprocessing methods for extreme events are selected to process extremely high-temperature dates. We examine the rationality of these postprocessing methods in two representative regions (Hunan, Central China) and compare the performance with and without these methods. Table 5 verifies the effec-

	eForecaster		LSTM		TCN		LightGBM	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Wind	**7.2**	**99.1**	7.6	104.6	7.6	102.9	8.0	107.4
Solar	**89.21**	**150.86**	101.18	167.16	91.25	151.93	102.58	170.58

Table 3: Day ahead renewable energy forecasting results on a wind farm and regional photovoltaic. A lower MAE and RMSE indicate better performance. The best results are highlighted in bold.

	Preprocessing	MAE	RMSE
Industrial Load	W/O	28.8	41.1
	With	**15.8**	**26.2**
Commercial Load	W/O	6.7	11.2
	With	**5.8**	**9.0**
Residential Load	W/O	9.0	13.1
	With	**7.1**	**10.6**

Table 4: Ablation studies of Robust Preprocessing.

	Postprocessing	MAE	RMSE
Hunan	-	145.1	184.2
	Extreme Events	**126.1**	**169.9**
Central China	-	600.0	738.7
	Extreme Events	**554.5**	**701.4**

Table 5: Ablation studies of Postprocessing.

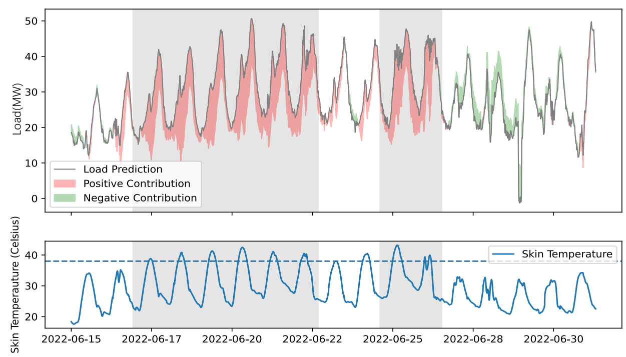

Figure 5: Contribution of Skin Temperature. Upper: Original load prediction is shown as a grey line, and the contribution from skin temperature is shown in highlighted red/green region. Lower: Trend of skin temperature. Two highlighted grey regions represent extreme events of high temperature.

tiveness of the postprocessing module. Designing and selecting suitable postprocessing strategies are helpful for reliable and accurate forecasting.

Case Studies in Deployed Systems (Q3)

Electricity Load Forecasting In this section, we illustrate the predictability of eForecaster pipeline using a challenge industrial bus load time series from Shandong Grid. As shown in Figure 3, traditional algorithms alone (LightGBM, LSTM, TCN) fail to capture the overall trend of this highly fluctuated data. Furthermore, they tend to forecast averages that are of no use in real-world scenarios. However, our eForecaster pipeline captures the overall pattern and provides more compelling forecasts.

Forecast Interpretation In this section, we provide two case studies for the forecast interpretation methods that have been used in the deployed system. The first study focuses on the effect of special holidays as shown in Figure 4. The load prediction for the new year starting from Jan. 1 is particularly lower than usual. The difference between load prediction at 1 p.m., Jan. 2, and observation at 1 p.m., Dec. 28 is decomposed and presented as a waterfall chart in the lower graph of Figure 4. It is obvious that the feature *Holiday* contributed most to the difference, as large as 9.98 MW, reflecting that model has successfully captured a decrease in load during holidays.

The second study is an analysis of extreme weather events as shown in Figure 5. In the lower graph, trend of skin temperature is shown and two grey regions are highlighted where the temperature exceeded 38 Celsius and considered as extreme events of high temperature. In the upper graph, the original load prediction is shown as a grey line, and the highlighted red(green) region indicates a decrease(increase) of prediction if temperatures are set to an average value. The large red region during extreme events of high temperature justifies that a higher temperature leads to a higher load because of the increasing usage of air-conditioning is learned by our model. The degree of increase is quantitatively shown and can be modified by users aiming to reach a better performance.

Conclusions and Future Works

We propose eForecaster, an integrated AI platform to tackle electricity forecasting problems using robust, flexible, and explainable machine learning algorithms. Furthermore, we have deployed electricity forecasting systems based on eForecaster to provide data-driven, explainable, and intelligent bus load, system load, and renewable energy forecasts in seven provinces in China (Shandong, Zhejiang, Jiangxi, Shanxi, Henan, Hunan, and Hebei). In the hearts of these systems, eForecaster improves the forecast accuracy and interpretability, while minimizing the cost of deployment and manual works. Consequently, eForecaster provides domain experts with truth-worthy guidelines for decision-making and enables fast, large-scale deployments.

In future, we will focus on alleviating concept drift (Ditzler et al. 2015) or the model aging (You et al. 2021) problem in electricity forecasting. Our goal is to design an interface that endows pipeline abilities including drift detection, drift understanding, and drift adaptation in the real-world deployed environment with minimum human interference.

References

Bai, S.; Kolter, J. Z.; and Koltun, V. 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*.

Chen, T.; and Guestrin, C. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785–794.

Chen, W.; Wang, W.; Peng, B.; Wen, Q.; Zhou, T.; and Sun, L. 2022. Learning to Rotate: Quaternion Transformer for Complicated Periodical Time Series Forecasting. In *KDD 2022*.

Ditzler, G.; Roveri, M.; Alippi, C.; and Polikar, R. 2015. Learning in nonstationary environments: A survey. *IEEE Computational Intelligence Magazine*, 10(4): 12–25.

Dorogush, A. V.; Ershov, V.; and Gulin, A. 2018. CatBoost: gradient boosting with categorical features support. *arXiv preprint arXiv:1810.11363*.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation*, 9(8): 1735–1780.

Hooi, B.; Song, H. A.; Pandey, A.; Jereminov, M.; Pileggi, L.; and Faloutsos, C. 2018. Streamcast: Fast and online mining of power grid time sequences. In *Proceedings of the 2018 SIAM International Conference on Data Mining*, 531–539. SIAM.

Jarrett, D.; Yoon, J.; Bica, I.; Qian, Z.; Ercole, A.; and van der Schaar, M. 2021. Clairvoyance: A Pipeline Toolkit for Medical Time Series. In *the 9th International Conference on Learning Representations*.

Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; and Liu, T.-Y. 2017. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30.

Kim, S.-J.; Koh, K.; Boyd, S.; and Gorinevsky, D. 2009. ℓ_1 Trend Filtering. *SIAM Review*, 51(2): 339–360.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

McKinney, W.; et al. 2011. pandas: a foundational Python library for data analysis and statistics. *Python for high performance and scientific computing*, 14(9): 1–9.

Nti, I. K.; Teimeh, M.; Nyarko-Boateng, O.; and Adekoya, A. F. 2020. Electricity load forecasting: A systematic review. *Journal of Electrical Systems and Information Technology*, 7(1): 1–19.

Pearson, R. K.; Neuvo, Y.; Astola, J.; and Gabbouj, M. 2016. Generalized hampel filters. *EURASIP Journal on Advances in Signal Processing*, 2016(1): 1–18.

Salinas, D.; Flunkert, V.; Gasthaus, J.; and Januschowski, T. 2020. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3): 1181–1191.

Song, H. A.; Hooi, B.; Jereminov, M.; Pandey, A.; Pileggi, L.; and Faloutsos, C. 2017. Powercast: Mining and forecasting power grid sequences. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 606–621. Springer.

Sutera, A.; Louppe, G.; Huynh-Thu, V. A.; Wehenkel, L.; and Geurts, P. 2021. From global to local MDI variable importances for random forests and when they are Shapley values. *Advances in Neural Information Processing Systems*, 34: 3533–3543.

Varoquaux, G.; Buitinck, L.; Louppe, G.; Grisel, O.; Pedregosa, F.; and Mueller, A. 2015. Scikit-learn: Machine learning without learning the machinery. *GetMobile: Mobile Computing and Communications*, 19(1): 29–33.

Wen, Q.; Gao, J.; Song, X.; Sun, L.; and Tan, J. 2019a. RobustTrend: a Huber loss with a combined first and second order difference regularization for time series trend filtering. In *IJCAI 2019*, 3856–3862.

Wen, Q.; Gao, J.; Song, X.; Sun, L.; Xu, H.; and Zhu, S. 2019b. RobustSTL: A robust seasonal-trend decomposition algorithm for long time series. In *AAAI Conference on Artificial Intelligence (AAAI)*, 5409–5416.

Wen, Q.; He, K.; Sun, L.; Zhang, Y.; Ke, M.; and Xu, H. 2021. RobustPeriod: Time-Frequency Mining for Robust Multiple Periodicity Detection. In *ACM International Conference on Management of Data (SIGMOD)*, 2328–2337.

Wen, Q.; Zhang, Z.; Li, Y.; and Sun, L. 2020. Fast RobustSTL: Efficient and Robust Seasonal-Trend Decomposition for Time Series with Complex Patterns. In *KDD 2020*, 2203–2213.

Yang, L.; Wen, Q.; Yang, B.; and Sun, L. 2021. A Robust and Efficient Multi-Scale Seasonal-Trend Decomposition. In *IEEE ICASSP*, 5085–5089.

You, X.; Zhang, M.; Ding, D.; Feng, F.; and Huang, Y. 2021. Learning to learn the future: Modeling concept drifts in time series prediction. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2434–2443.

Zhou, T.; Ma, Z.; Wen, Q.; Wang, X.; Sun, L.; and Jin, R. 2022. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *ICML 2022*.