SAINT-PETERSBURG UNIVERSITY

Published in manuscript form

Zhang Yuyi

Explainable Artificial Intelligence in Time Series Forecasting

Scientific specialty 1.2.2.
Mathematical modeling, numerical methods and software packages

DISSERTATION

Thesis for a degree candidate of technical sciences

Supervisor:
Doctor of physical and mathematical sciences, professor
Petrosian Ovanes Leonovich

Saint Petersburg

2024

Contents

Introduction

Relevance of thesis topic

In recent years, artificial intelligence (AI) [1–3] models such as ensemble learning and deep learning have demonstrated notable success in time series forecasting, particularly for long-term forecasting [4–6]. In various competitions, these machine learning techniques have consistently outperformed traditional statistical methods. A significant milestone was reached when LightGBM [7] won the M5 competition [8–11], drawing widespread attention to the capabilities of artificial intelligence. Prof. S. Makridakis, the founder of the M competition series[1] and an eminent authority in the field of forecasting, has recently compared the effectiveness of statistical algorithms, machine learning, and deep learning [12]. The experimental results affirm the substantial potential of AI for long-term time series forecasting, surpassing conventional statistical approaches. Figure 1 illustrates the winning solutions from major time series forecasting competitions over recent years, further emphasizing the prevailing trend towards AI-driven methods.

Competitions	Year	Winning solution	Types
Schneider competition	2018	LightGBM	AI
M4 competition	2020	ES-RNN	Statistics + AI
M5 competition	2021	LightGBM	AI
M6 competition	2022	Neural network	AI

Figure 1: Summary of winning solutions for mainstream competitions in time series forecasting

AI techniques have proven highly effective in modeling complex patterns and

[1]M competition homepage

generating accurate forecasts, making them indispensable across various industries such as energy [13, 14], healthcare [15, 16], and finance [17, 18]. However, the inherent opacity of these models presents a significant challenge. The lack of transparency in these black-box models undermines trust and limits their broader acceptance. This issue is particularly critical in applications where understanding the decision-making process is as important as the forecasting results. To address this challenge, there has been increasing interest in Explainable AI (XAI) [19–21, 23, 26, 28–31]. XAI aims to make these black-box models more transparent and trustworthy by elucidating their decision-making processes. This technology holds substantial implications for advancing artificial intelligence and ensuring its safer application in society [32–34]. Figure 2 illustrates two distinct types of explanations in the context of time series forecasting.

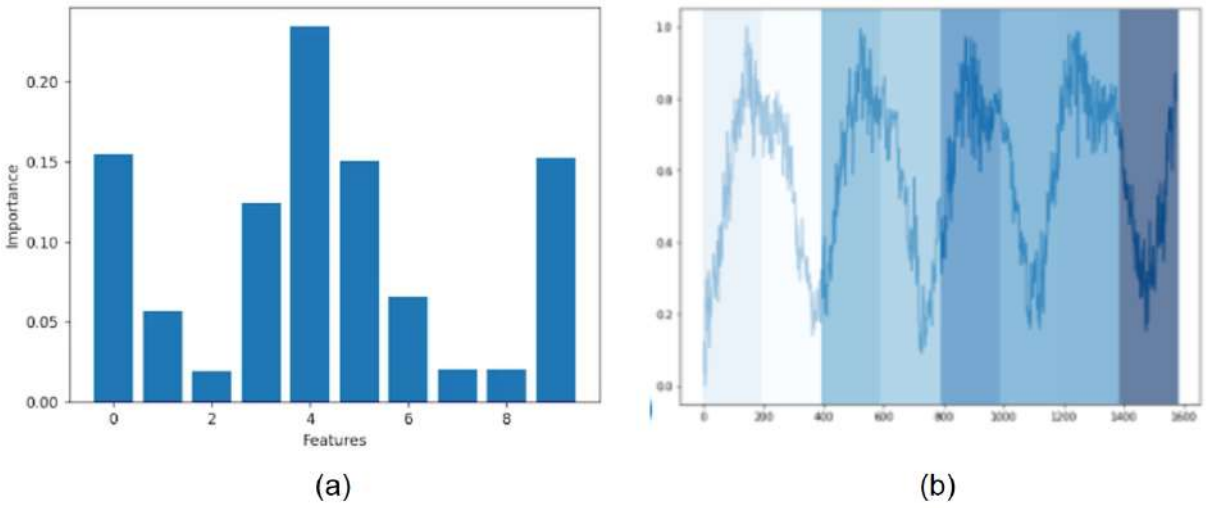


Figure 2: Examples of explanation results in time series forecasting. (a) shows how each feature affects the target variable; (b) shows how historical data affects the target variable.

Figure 2(a) presents an explanation from a feature perspective, showing how each variable influences the target variable [19–23, 28–31]. This type of explanation is particularly useful for multivariate time series forecasting tasks, where multiple variables are involved. In contrast, Figure 2(b) provides an explanation from a temporal perspective. It highlights how historical data impacts the forecasting results [26]. This approach is applicable to both univariate and multivariate time series forecasting tasks, making it versatile across different types of datasets. These visualizations underscore the importance of explainability in AI models used for time series forecasting. By

understanding how features and historical data contribute to forecasts, stakeholders can gain valuable insights, leading to more informed decision-making.

In recent years, the rapid advancement of artificial intelligence (AI) models in time series forecasting has underscored the need for Explainable AI (XAI) solutions in this field. Understanding and interpreting these complex models is crucial for their adoption and trustworthiness. Consequently, the exploration of XAI within the context of time series forecasting holds significant importance [35–44]. The modern concept of XAI can be traced back to 2007 with V. Schetinin's research [45] on the explainability of Bayesian decision trees. Since then, substantial academic progress has been made, including numerous theoretical frameworks, algorithms [46–48], and supporting code libraries[2].

The XAI algorithms can be broadly categorized into **model-specific**, **model-agnostic**, and **hybrid approaches**. Model-specific algorithms, such as the feature importance (FI) [49, 50] embedded in boosting algorithms, are tailored to specific types of black-box models. On the other hand, model-agnostic algorithms like SHAP [51] and LIME [52] offer versatility by being applicable to all types of black-box models, providing strong generalizability. Hybrid approaches combine an interpretable module [53, 54] within the forecasting model, where the module's output explains the overall model's behavior. While these algorithms have produced considerable research outcomes for classification and regression tasks, their application to time series data remains limited. This limitation highlights several key problems that need to be addressed:

Problem 1: How to evaluate the effectiveness of the explanation results in time series forecasting?

Unlike tasks such as regression, where real target values are available for evaluation, the field of Explainable AI (XAI) faces challenges in obtaining labeled values in practical applications [23, 55, 56]. This lack of accessible labeled data complicates the objective assessment of XAI quality. Consequently, unlike well-established metrics in other fields, such as mean squared error (MSE) and

[2]SHAP official homepage

area under the curve (AUC), there is a scarcity of universally accepted quantitative metrics within the XAI domain. This absence makes it difficult to standardize the evaluation of XAI algorithms.

Problem 2: How to build XAI algorithms that explain the impact of historical data for all types of AI models?

Recent research has underscored the growing importance of XAI in the field of time series forecasting. Much of this research has focused on model-specific algorithms to individual models [35, 38, 39]. Typically, these studies involve developing a new time series forecasting model and then adding interpretable modules to claim it as an explainable model. While this methodology has its merits, we argue that the adoption of model-agnostic algorithms is crucial, especially in engineering applications. The generalizability of these methods, regardless of the forecasting model or explanation method used, holds significant value.

Problem 3: How to build XAI algorithms based on new hybrid mechanisms in time series forecasting?

Currently, the hybrid method involves incorporating an attention mechanism [42–44] into forecasting models to improve their overall explainability. Although this approach has been applied in time series forecasting, the inherent explainability of the attention mechanism itself is still a topic of ongoing discussion.

Problem 4: How to translate explanation results into real economic value?

Many previous studies present the concept of explainability as a new idea but often overlook its practical applications for current challenges [35–44].

These algorithms developed in this work are primarily based on the Shapley value, a concept introduced by L. Shapley in 1951. The Shapley value [57] is a foundational method in cooperative game theory, designed to fairly allocate payoffs among participants. This principle has been effectively adapted for XAI, particularly in the form of the SHAP algorithm developed by S. Lundberg and Su-In Lee. While the traditional SHAP method calculates variable contributions to model outputs, it lacks the capability to provide explanations from a temporal perspective. Addressing this limitation is essential for advancing interpretability in time series forecasting. Significant contributions to

attribution algorithms have also been made by researchers such as S. Bach, A. Charnes, A. Datta, S. Lipovetsky, M. T. Ribeiro, A. Shrikumar, E. Strumbelj, and H. P. Young . These advancements collectively enhance our understanding and application of fair and reasonable allocation methods in the context of XAI.

Goals and tasks of thesis work

This thesis seeks to advance the field of XAI in time series forecasting. As black-box AI models continue to be prevalent, our aim is to create XAI algorithms specifically designed for this domain while investigating their economic applications. Achieving this goal requires addressing four primary challenges in the field. First, obtaining labeled data for XAI in practical applications remains difficult, leading to a lack of universally accepted quantitative metrics. This thesis aims to propose new evaluation metrics that objectively assess XAI algorithm quality. Second, employing model-agnostic approaches is essential, particularly in engineering contexts where general applicability across various forecasting models is desired. We intend to develop model-agnostic XAI algorithms to enhance flexibility and usability. Third, while attention modules are commonly used in models, their interpretability is often questioned. This research will explore alternative mechanisms to develop hybrid approaches, thereby improving the explainability of time series forecasting models. Finally, many studies emphasize explainability without fully addressing its specific purposes or contributions to current challenges in the field. This thesis aims to fill this gap by conducting comprehensive exploratory research, applying XAI algorithms to real-world scenarios, and elucidating their benefits. To achieve these objectives, the following tasks will be undertaken:

1. Establishing Quantitative Evaluation Metrics: Develop common evaluation metrics based on recognized theorems to select applicable XAI algorithms for different AI models. This foundation will enable the development of new algorithms rooted in the most suitable XAI methodologies.

2. Developing Model-Agnostic Algorithms: Create a model-agnostic algorithm for time series forecasting that can be applied to any forecasting model. The

algorithm will explain the significance of historical data over different time periods, subsequently enhancing forecasting accuracy.

3. Hybrid Approach Mechanism: Implement a combination of model-agnostic and model-specific algorithms as a new mechanism for hybrid approaches. This will expand the technical scope of XAI algorithms, providing variable importance explanations and improving forecasting accuracy.

4. Exploring Economic Value: Investigate the economic impact of XAI by applying it to practical issues such as online adaptation, improving forecasting accuracy, and enabling impact factor analysis.

Through these tasks, the thesis aims to link theoretical insights with tangible outcomes in real-world scenarios, thereby contributing both academically and practically to the field.

Scientific novelty

The challenge of non-explainability in AI algorithms significantly hinders their broader application in time series forecasting. This paper addresses this issue by exploring Explainable AI (XAI) techniques within the context of time series forecasting, a field where AI algorithms have diverse applications. The novelties of this work are as follows:

1. In contrast to time series forecasting tasks, where algorithm performance is assessed using metrics like mean squared error, evaluating AI model explanations lacks true benchmark values. This complicates the assessment of high-quality explanatory methods. Our research introduces a quantitative framework designed to evaluate these explanatory results, allowing superior XAI methods to be identified and highlighted.

2. Different from the original SHAP, our algorithm - ShapTime achieves explanation in the time series dimension, i.e., it is able to output the importance of historical data for the forecasting results, which is not possible with other model-agnostic algorithms.

3. Combination of the model-agnostic and the model-specific enables the explanation of the results to be more informative implicitly, thus helping to optimise the accuracy of the forecasting.

4. Explanatory results from XAI methods are merely presented without practical application in previous work. Our paper bridges this gap by demonstrating how these insights can be translated into tangible economic value. We highlight the utility of the explanatory results by applying them to real-world problems. This includes analysis of influencing factors, forecasting performance improvement, and online adaptation problems.

Through these innovations, this work aims to advance the reliability and applicability of AI in time series forecasting by making AI models more interpretable and practically beneficial.

Theoretical and practical significance

This research holds significant theoretical and practical implications. Theoretically, by comparing the performance of boosting models and neural network models in time series forecasting tasks, we provide a foundation for selecting the most appropriate model to be explained. This indirectly highlights the necessity of developing general model-agnostic algorithms. Furthermore, through the development of model-agnostic algorithms for time series forecasting (ShapTime) and a hybrid approach combining feature importance and SHAP algorithms (FI-SHAP), this study expands the realm of explainable AI technology in time series forecasting and introduces new perspectives for explanation.

Practically, this study offers valuable guidance for model selection and deployment, black-box model explanation, decision support, and risk management. By applying XAI algorithms, we can analyze influencing factors and provide robust decision support, thereby driving the development and application of AI technology. Overall, this research positively impacts the comprehensibility and trustworthiness of artificial intelligence, promoting its widespread use in real-world scenarios.

In conclusion, this study enhances the explainability and trustworthiness of artificial intelligence, facilitating its extensive application in practical settings. By introducing novel approaches and perspectives in explainable AI for time series forecasting, we are better equipped to address challenges and meet demands in real-world contexts. This research not only improves the effectiveness of decision support systems but also lays the groundwork for the broad application of AI technology across various domains. Through a deep understanding and application of XAI algorithms, we ensure transparency, explainability, and trustworthiness in advancing artificial intelligence, thus fostering its broad adoption and societal benefits.

Structure of dissertation

The first chapter provides a comparative analysis of artificial intelligence models, focusing on both theoretical aspects and forecasting performance. Section 1.1 clarifies the fundamental characteristics of various AI models. In Section 1.2, we compare different AI methods to identify the most effective model for various tasks.

In the second chapter, we introduce an explainable AI (XAI) technique designed specifically for time series forecasting, adopting a model-agnostic approach. Section 2.1 addresses the challenges faced in developing this temporal perspective. Section 2.2 details the structure of the proposed ShapTime algorithm and explains how it mitigates these challenges.

The third chapter focuses on improving existing XAI techniques by combining model-agnostic and model-specific algorithms. This integration results in a novel XAI approach called FI-SHAP, tailored for boosting models.

Chapter fourth showcases practical applications of XAI techniques through simulations, including factor analysis in forecast modeling and addressing online adaptation problems. Section 4.1 and 4.2 utilizes available XAI techniques to explain the optimal model and perform factor analyses on variables related to solar power generation and PM2.5 concentrations, providing recommendations for optimal siting of solar power plants. In Section 4.3, the XAI method is applied to manage concept drift in online adaptation scenarios.

Research methods

This study employs literature analysis as a theoretical approach to investigate the application of artificial intelligence (AI) methods in time series forecasting tasks, as well as the historical development and current challenges faced by explainable AI techniques in this domain. Empirically, we conducted comparative research to compare various forecasting methods and different explainable AI technologies. Additionally, experimental studies were carried out to demonstrate the validity of the algorithms. Furthermore, simulation studies were performed to simulate the application scenarios of Explainable AI methods in time series forecasting using real-world data.

Approval of obtained results

The results presented in the thesis were reported and **approved** at the following international conferences and seminars (with a sufficient number of foreign participants):

- NeuroNT 2021-2022: 2nd and 3rd International conference on neural networks and neuro-technologies, St. Petersburg (Russia).
- IntelliSys 2022-2023: Intelligent Systems Conference, Amsterdam (The Netherlands).
- MLIS 2022: Machine Learning and Intelligent Systems, Daegu, South (Korea).

During the author's PhD study, she participated in a collaborative project between St. Petersburg State University and a chinese commercial company. This project focused on energy forecasting and control, and findings from this thesis work were partially implemented in the project, leading to the successful achievement of the expected outcomes.

The results of research have been repeatedly reported in co-authorship by Professor Petrosian Ovanes Leonovich, Ph.D student Jinying Zou, Feiran Xu, Ruimin Ma, Jing Liu, and Ph.D candidate Dongfang Qi and Qiushi Sun.

Publications. The author completed over 13 scientific papers [19–31], including 5 papers [19, 21–23, 26] in the periodicals indexed by The Web of Science or Scopus databases. Among them, 9 papers [19–23, 26, 27, 30, 31] are on the thesis topic, including 4 papers [19, 21, 22, 27] in the periodicals from the list of peer-reviewed journals recommended by the Higher Attestation Commission of the Russian Federation, and 4 papers [19, 21, 22, 27] in the periodicals indexed by The Web of Science or Scopus databases scientific results submitted for defense were published in the following peer-reviewed periodicals. The main scientific results submitted for defense were published in the following peer-reviewed periodicals: [23, 27] - item 1; [26] - item 2; [20] - item 3; [19, 21, 22] - item 4.

All of author's research papers have received a total of 300 citations on Google Scholar[3] during PhD studies (form 2020 to 2024 year). Further, all the code is published on GitHub[4].

Personal contribution of author

This work was done at St. Petersburg State University. Some of the research was done jointly with Petrosian Ovanes Leonovich, Jinying Zou, Feiran Xu, Ruimin Ma, Jing Liu, Qiushi Sun, Dongfang Qi. Most of the research results presented in the dissertation were published with co-authorship; to avoid ambiguity in the dissertation, the corresponding references are labelled with a complete list of names. Meanwhile, the results of the thesis defence belong to the authors only.

Main scientific results

- The XAI evaluation metric - MDMC is created, in order to measure the accuracy of the explanatory results so that the respective most suitable explanatory methods for different AI algorithms can be filtered out. It is a generalised evaluation metric, i.e. it can be used for any task including time series forecasting tasks, see work [23, 27] and Chapter 1 (P.22) in this work (with at least 80% individual contribution).

[3]The Google Scholar homepage of Zhang Yuyi
[4]The GitHub homepage of Zhang Yuyi

- The new XAI algorithm - ShapTime is created, in order to visualise the importance of historical time periods for the forecasting results. This algorithm divides time steps into the time periods in the time dimension and calculates the shapley values for each time period as its importance for the forecasting results, see work [26] and Chapter 2 (P.39) in this work (with at least 80% individual contribution).

- The new XAI hybrid mechanism - FI-SHAP is created, in order to improve the explanatory accuracy of the current SHAP algorithm. This mechanism enables the explanatory results to contain richer information compared to previous SHAP methods through combining model-agnostic algorithms and model-specific algorithms, see work [20] and Chapter 3 (P.56) in this work (with at least 80% individual contribution).

- Developed XAI methods and approaches are applied in analyzing influencing factors, solving concept drift in adaptation problems and improving the accuracy of time series forecasting, see work [19, 21, 22], and Chapter 4 (P.71) in this work. (with at least 80% individual contribution).

Results submitted for defense

- Quantitative evaluation of XAI methods - MDMC. An evaluation metric for the accuracy of the explanatory results is constructed, which is able to show the most suitable XAI method for a given black-box AI model. In this way, the optimal XAI method can be determined, and in this work, the SHAP method based on the shapley value is confirmed to be optimal. Therefore, subsequent development of new XAI algorithms are based on the shapley value.

- Generalised XAI method for the time series dimension - ShapTime, which calculates the shapley values for different time periods in the time dimension, which represents the contribution of different time periods to the forecasting results. ShapTime is more suitable for time series forecasting tasks than previous XAI methods that output the contribution of variables. The results

of ShapTime are also consistent with the MDMC metrics, which greatly proves its effectiveness.

- XAI method based on the new hybrid mechanism - FI-SHAP. In boosting algorithms, feature importance and SHAP are two common explanatory methods, the former contains information from the model itself, while the latter contains information from the dataset. FI-SHAP combines both, so that it contains more information, and correspondingly, the feature engineering based on it achieves better performance enhancement results.

- Explore scenarios where explainable AI can be used in real-life applications. This is crucial for AI to better serve the society, and is key to improving the transparency of AI as well as the trust of human beings. This includes influencing factor analysis as well as solving online adaptation problems.

Literature review

Current approaches to explainable time series forecasting can be divided into three categories: model-agnostic methods, model-specific methods and hybrid methods.

Model-agnostic methods. It is realized by perturbing the input data set and inducing the change of the output, and finally attributing this change to the input features, so as to realize the explanation of the model. This is one approach that is widely used compared to others because of its generality(e.g [19, 20, 35–37]). However, this type of XAI method was originally developed based on classification and regression tasks, so they often output feature importance instead of the importance of time itself, that is, they cannot output $\Phi(X_{T_i})$.

Model-specific methods. This is a specific method for the development of time series forecasting models, that is, to embed the interpretation function into the model(e.g [38, 40]), in order to achieve better interpretation effect for time series forecasting. Although some works pay attention to the explanation of the temporal dimension (e.g ([39, 41]), they are still based on features, that is, the XAI approach outputs the feature importance at each moment and stitches them together to achieve the explanation of the temporal dimension. Strictly speaking, such an explanation also fails to output $\Phi(X_{T_i})$, and seriously lacks generality.

Hybrid methods. This is an approach to achieve explanation effects by hybridizing modules with a certain degree of interpretation function into time series forecasting model. The most representative approach is to hybridize the attention mechanism into the time series forecasting model(e.g [42–44]), and achieve the explanation through the explainability of the attention mechanism. Some works have discussed the explainability of the attention mechanism. Even though there are some controversies(e.g [66]), researchers still hold a positive

attitude towards its explainability (e.g [67, 68]. The hybrid methods suffers from the same problem as above approaches, that is, even if there is an explanation in the temporal dimension, this explanation is dependent on features. On the other hand, it also requires the development of new models and thus lacks generality.

Machine learning and deep learning techniques have demonstrated superior performance in the domain of time series forecasting, particularly for long-term forecasting tasks. Ensemble learning [69] and deep learning techniques have proven to be highly effective for handling non-linear and non-stationary data. Ensemble learning involves combining the predictions of multiple models in order to improve accuracy, while deep learning utilizes intricate network structures to identify relationships between input and output variables. These approaches hold particular significance in domains such as solar and wind energy forecasting [70], where nonlinear relationships and patterns are frequently observed. Ensemble learning encompasses Boosting and Bagging algorithms, with Boosting algorithms such as XGBoost [71], LightGBM [72], and CatBoost CatBoost [73] being widely utilized. On the other hand, Deep Learning can be classified into three distinct types based on their network structure: Artificial Neural Networks (ANN) [27, 74], Convolutional Neural Networks (CNN) [75], and Recurrent Neural Networks (RNN) [76].

XGBoost, LightGBM, and CatBoost have exhibited remarkable efficacy in time series forecasting competitions. Notably, XGBoost has been extensively employed in the M4 competition [77]. LightGBM has proven its effectiveness in handling vast datasets and achieving rapid training times, rendering it a popular selection in diverse competitions. In fact, LightGBM surpassed other frameworks and emerged victorious in the M5 forecasting competition [78], thus highlighting its exceptional performance in real-world scenarios. CatBoost is also widely utilized in Kaggle time series forecasting competitions. All three frameworks have demonstrated their potency for long-term time series forecasting, with each possessing unique advantages that render them suitable for distinct use cases [79] [80]. Bae DJ, et al. [81] proposed an XGBoost-based load forecasting algorithm, which enhanced accuracy by 21% and 29% in 2019 and 2020, respectively, when compared with previous models. Zhang Y, et al. [19] assessed mainstream prediction methods across various datasets,

including solar power generation. The experimental results indicated that LightGBM was the superior algorithm overall, outperforming others in the three datasets. As another notable boosting algorithm, CatBoost also exhibits evident potential in solar power generation forecasting [82]. Deep learning techniques are frequently employed for time series forecasting, particularly artificial neural network (ANN) [83], recurrent neural network (RNN)-based models (RNN [84], LSTM [85], GRU [86]), and bidirectional RNN-based models [87] (Bi-RNN, Bi-LSTM, Bi-GRU). These models possess the capacity to automatically extract pertinent features from input data, providing an advantage over ensemble learning, which relies on manual feature engineering. ANN excels at capturing intricate non-linear patterns in scenarios where the relationships between input features and target variables are unclear or non-linear. While RNN-based and Bi-RNN-based models were initially devised for natural language processing [88], they have become extensively adopted in solar power generation forecasting due to their ability to capture temporal dependencies. Given that different models may possess varying abilities to learn from data, it is imperative to conduct comprehensive comparisons and analyses tailored to specific cases [19].

Ensemble learning and deep learning represent prominent methodologies for time series forecasting, particularly in the context of intricate and long-term predictions. Nevertheless, their opaque nature poses a challenge to discerning the factors that impact the accuracy of these forecasts [89]. Conversely, the identification of such factors assumes great importance when optimizing solar energy systems. Factors such as weather conditions [90], shading [91], and equipment characteristics [92] assume a pivotal role in system performance, cost reduction, and site selection. Scrutinizing these factors can mitigate risks, optimize energy generation, and facilitate the selection of the most suitable location for a solar power plant. Consequently, successful solar power generation necessitates the critical analysis of influential factors alongside effective forecasting techniques.

However, the lack of explainability presents a significant obstacle to their further development. Non-interpretability refers to the inability of a model to be comprehensible to human. Explainability [93–95] encompasses the following aspects:

- Interpretability of the patterns
- Interpretability of the parameters

If learned patterns and parameters can be readily understood by humans, it is referred to as a "white-box model." Conversely, models whose internal mechanisms are opaque to human interpretation are termed "black-box models." Notably, it is these black-box models that have increasingly exhibited superior performance in time series forecasting tasks, especially for long-term forecasting problems.

Despite the superior performance of black-box models, represented by machine learning and deep learning techniques, in terms of long-term forecasting, their inherent non-interpretability raises two significant concerns: the crisis of trust and the lack of knowledge provision [96–98].

- The crisis of trust arises from the inability of humans, whether users or developers, to comprehend the patterns and decision-making processes within these black-box models. This lack of transparency raises concerns about potential discrimination, bias, or other undesirable factors embedded within the model's architecture. Furthermore, the absence of a clear understanding of the model's inner workings hinders subsequent improvement and optimization, leaving them largely reliant on engineering experience. Collectively, these factors exacerbate human distrust in black-box models.

- The lack of knowledge provision is another critical issue. While black-box models can produce relatively accurate results in time-series forecasting tasks, they merely generate outputs without providing additional information or insights. In contrast, traditional mathematical and statistical methods, with their interpretable models and parameters, offer supplementary knowledge and information to aid human decision-making processes. These methods can identify the variables and time periods that exert the greatest influence on the predicted outcomes, thereby assisting humans in making more informed and reasonable decisions.

As a potential resolution to the controversies surrounding opaque artificial intelligence (AI) models, the concept of Explainable Artificial Intelligence

(XAI) [32–34, 46–48] has been proposed. XAI aims to enhance the interpretability and transparency of complex "black-box" models, rendering them comprehensible to human users, even if only partially or to a certain degree. By doing so, XAI endeavors to provide humans with as much information and knowledge as feasible, thus fostering a better understanding of the underlying decision-making processes and rationale employed by these models.

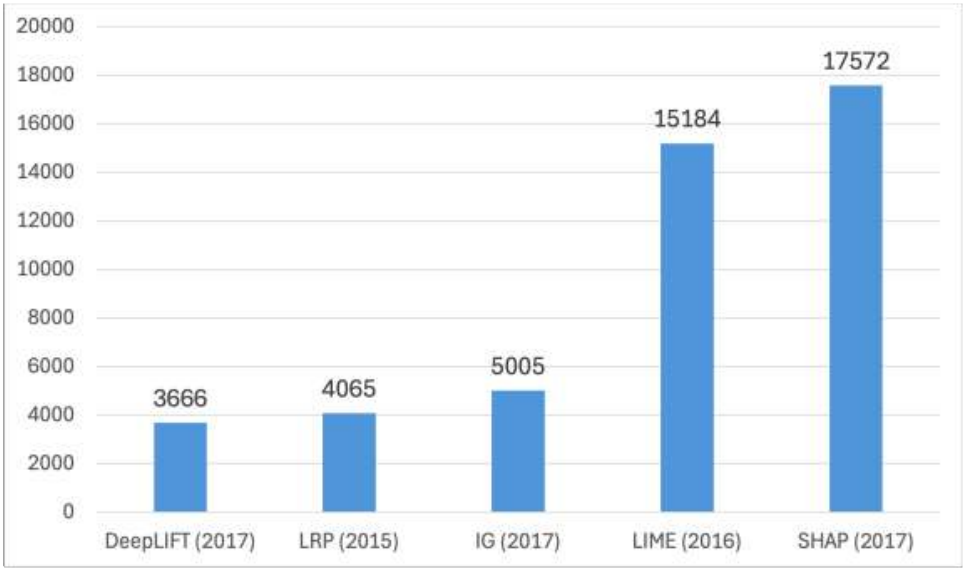


Figure 3: Paper citations for the mainstream algorithms with a data collection date of 07.12.2023.

The pursuit of interpretability in machine learning models has led to the development of various algorithms, including LIME (Local Interpretable Model-agnostic Explanations) [52], SHAP (SHapley Additive exPlanations) [51], Deep-LIFT [99, 100], Integrated Gradients (IG) [101], and LRP (Layer-wise Relevance Propagation) [102]. Notably, the Shapley value, upon which SHAP is based, is the only algorithm that satisfies certain deterministic properties, contributing to its widespread adoption in studies involving interpretability. Figure 3 illustrates the indexing of the original papers for these approaches, revealing a clear lead for SHAP. It is crucial to emphasize that the efficient implementation of SHAP in engineering technology leverages the ideas of LIME, directly leading to LIME's high level of interest. Furthermore, the concepts underpinning LIME have inspired the key algorithms employed in this study.

It is an established fact that SHAP is a widely adopted method for

interpreting black-box models [51, 103, 104]. However, its direct application to time series forecasting tasks is not feasible. In classical SHAP, the assumption of sample independence is made, implying that any temporal dependencies between samples are disregarded. Consequently, when dealing with data exhibiting temporal relationships, the explanation results produced by SHAP are limited to feature importance and fail to capture the intrinsic dynamics governing the time series. For instance, in the context of temperature forecasting [105], traditional SHAP can elucidate the relative importance of factors such as solar irradiation and humidity on temperature. Nonetheless, it falls short in elucidating the influence of previous time steps' temperature values on subsequent time steps, which is a crucial aspect in interpreting time series forecasting models.

Chapter 1

Comparison of forecasting and explainable AI algorithms for time series forecasting

Explainable AI (XAI) focuses on making AI models more interpretable, so comparing the performance of these models is essential. In this chapter, we compare mainstream AI models to identify the most suitable ones for various tasks, and **our results have been published in academic journals [19, 21, 27]**. It's important to note that different XAI algorithms can yield varying results even when applied to the same AI model. Therefore, measuring the quality of these explanation results is crucial. To address this, we developed an evaluation framework for XAI called MDMC, which helps filter out the most appropriate XAI algorithms for different AI models, and **our results have been published in conference [23]**.

1.1 Comparison of explainable artificial intelligence algorithms

1.1.1 Forecasting algorithms

Neural networks and ensemble models are considered exemplary representatives of black-box models within the field. Notably, artificial neural networks (ANN) [27, 74] as well as ensemble models like Random Forest (RF) [106] and LightGBM [78–80] demonstrate outstanding predictive capabilities.

The realm of predictive modeling encompasses various black-box models, prominently comprising neural network models and ensemble models. Among these, the Artificial Neural Network (ANN) model stands as a representative of

the neural network paradigm. Meanwhile, LightGBM and Random Forest are deemed as representatives within the ensemble model domain. LightGBM, being founded upon the boosting algorithm, epitomizes an ensemble model, whereas Random Forest is rooted in the bagging algorithm. Both boosting and bagging techniques are essentially composed of multiple simplistic tree models. In the absence of interconnections between these tree models, the bagging algorithm employs a "voting" mechanism for deriving the final output result. On the other hand, when strong interdependencies exist among these tree models, they form a boosting algorithm wherein the output of preceding tree models influences subsequent ones. Additionally, linear regression and white-box model-decision trees are also employed in prediction tasks for comparative analysis.

The **Neural Network** can be described as an adaptive nonlinear dynamic system that encompasses numerous basic units, commonly referred to as neurons. These neurons are interconnected through activation functions, giving rise to a complex web of interactions. Although the structure and function of each individual neuron are relatively straightforward, the collective behavior of the entire network becomes exceedingly intricate and challenging to elucidate. The ANN architecture is typically organized into three distinct layers: input, hidden, and output. This layered arrangement enables the network to process information in a sequential manner. The input layer receives external stimuli or data, which is then transmitted to the hidden layer. Within the hidden layer, the intermediate computations take place, allowing for complex transformations and feature extraction. Finally, the processed information flows to the output layer, which generates the network's response or prediction.

GBDT [78–80] holds a position of utmost significance in the domain of machine learning. At its core, this model revolves around leveraging weak classifiers, specifically decision trees, to progressively train and acquire an optimal model. The utilization of such an approach renders GBDT highly desirable due to its exceptional training efficacy and ability to circumvent overfitting. In line with GBDT, LightGBM, also known as Light Gradient Boosting Machine, emerges as a remarkable framework that implements the GBDT algorithm. This framework offers several noteworthy advantages, including accelerated training speed, reduced memory consumption, and

enhanced accuracy. LightGBM's ability to deliver these benefits further reinforces its value as an essential tool within the field of machine learning.

The **Random Forest** is a powerful ensemble learning technique that combines multiple decision trees to improve the accuracy and stability of predictions. By merging these decision trees, the random forest classifier leverages the aggregate knowledge of all individual classifiers, while utilizing the hyperparameters associated with the bagging classifier to regulate its overall structure. It is important to mention that in addition to classification, the concept of a random forest can also be applied to regression problems, wherein a random forest regressor is employed.

1.1.2 Tests of forecasting performance

In order to visually ascertain the advantages of the black-box model in prediction, we employ two traditional models, linear regression (LR), and the classic white-box prediction model-decision tree (DT), as a control group. These models are used alongside artificial neural networks (ANN), LightGBM, and random forest (RF) to predict the Boston housing data set. The prediction results in terms of quality are presented in the following table 1.1, considering a standard data set [1].

Table 1.1: The Quality Statistics

$Model$	R^2	MSE	MAE
LR	0.64856	28.40585	3.69136
DT	0.74361	20.72322	3.05065
ANN	0.79477	16.58769	2.57977
LGBM	0.80417	15.82852	2.53292
RF	0.81888	14.63918	2.35044

The analysis of metrics reveals that artificial neural networks (ANN), LightGBM, and random forests (RF) exhibit substantial advantages when utilized for prediction tasks. Nevertheless, in contrast to linear regression and decision trees, ANN, LightGBM, and RF pose difficulties for human comprehension due to their intricate internal structures. Consequently, even though black-box models with enhanced accuracy surpass traditional models

[1]Boston housing dataset

and white-box models with reduced accuracy across various domains, they cannot entirely supplant them. Consequently, it becomes imperative to advance effective approaches aimed at comprehending and elucidating the workings of these opaque models.

1.1.3 Explainable artificial intelligence algorithms

According to Figure 3, an evident superiority of attention is observed in SHAP and LIME compared to the other algorithms. Hence, for the purpose of this section, we shall concentrate on these two approaches, employing quantitative evaluation metrics to assess their performance.

SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) represent two prominent methodologies utilized in the realm of interpreting machine learning models and comprehending their predictions. In the case of SHAP, its primary objective lies in assigning significance values to the input features of a given model to shed light on its output. Employing game theory principles and Shapley values, SHAP effectively computes feature attributions. By exhaustively considering all feasible combinations of features and corresponding outcomes, this approach accurately gauges the contribution made by each feature in predicting an instance. Notably, this methodology embraces an inclusive perspective on feature interactions, thereby furnishing explanations that are both locally precise and globally coherent. SHAP values introduce a comprehensive framework for interpreting diverse machine learning algorithms encompassing deep neural networks, tree-based models, and linear models. Consequently, it facilitates our comprehension of how individual features impact the model's predictions, granting us insights into the rationale underlying specific decisions.

In the framework of SHAP, the variables $\mathbf{x} = (x_1, x_2, \cdots, x_n)$ are considered as the set of players, and the black-box model f is replaced by an explainer g, such that $f(\mathbf{x}) = g(\mathbf{x}) = \phi_{x_0} + \sum_{i=1}^{n} \phi_{x_i}$. This implies that SHAP assumes any black-box model to be represented by this linear expression, where ϕ_i represents the contribution of the corresponding variable to the prediction, and Φ_{x_0} represents the model's output when all variables are ineffective. In fact, when modeling

real-world problems, it is nearly impossible to list all influencing factors. The contributions generated by these variables not included in the analysis scope are represented by Φ_{x_0} in a physical sense. Within this framework, the mathematical expressions of all black-box models are assumed to be the aforementioned linear expression, and the contribution of each variable represented by Φ_{x_i} serves as the explanation result, thereby achieving the generality of XAI. The contribution of each variable is calculated using the Shapley value formula,

$\phi(x_i) = \sum_{\mathbf{x_s} \subseteq \{x_1 \cdots x_n\} \backslash \{x_i\}} \frac{|s|!(n-|s|-1)!}{n!}(v(\mathbf{x_s} \cup \{x_i\}) - v(\mathbf{x_s}))$, which considers the prediction results of a black-box model $f(\mathbf{x}_s)$ (where s is a subset of n) as the profit function $v(\mathbf{x}_s)$. This allows us to compute the contribution value $\phi(x_i)$ for each variable x_i, serving as an explanation for the black-box model.

In comparison, LIME places its emphasis on providing explanations at a local level for individual predictions rather than striving for global interpretability. It achieves this by approximating intricate machine learning models with interpretable models that are comparatively simpler to comprehend. LIME's approach involves perturbing the input data surrounding a particular instance of interest and observing the resulting changes in predictions. By constructing a fresh dataset through sampling instances from the original data, LIME fits a local model to this sampled data, thereby generating explanations.

The approximated model produced by LIME sheds light on significant features and their influence on the prediction for that specific instance. A notable aspect of LIME is its model-agnostic nature, enabling its application to any black-box model without necessitating an understanding of the internal workings of said model. One of LIME's strengths lies in providing human-understandable explanations, such as feature importances or highlighting pertinent parts of the input. This capability contributes to the development of trust in machine learning models.

Both SHAP and LIME serve as potent tools for interpreting complex machine learning models. While SHAP delivers global explanations that account for feature interactions, LIME primarily focuses on supplying local explanations by employing a simplified approximated model. The choice between these methodologies depends on the particular task at hand and the specific requirements of the interpretability analysis.

1.1.4 Evaluation framework for explainable AI algorithms

The establishment of the XAI evaluation framework is based on a premise: *deleting or changing the large contribution (calculated by the XAI method) features in the data set will result in a significant decrease in the accuracy of the model's prediction.* Therefore, based on this premise, the degree of change in the metrics (R2, MSE, MAE) can be used as the core of the evaluation framework.

Algorithm 1 Mean degree of metrics change (MDMC)

Input: Original original metric: R_0^2, MSE_0, MAE_0; Changed metric: R_i^2*, MSE_i*, MAE_i*; Total number of features n_0

Output: MDMC value

1: Let $n = \frac{2}{3}n_0$.
2: Let $D = (R_0^2 - R_i^2) + (MSE_i - MSE_0) + (MAE_i - MAE_0)$.
3: **for** i in range (n) **do**
4: $\quad MDMC = \frac{1}{n}\sum_{i=1}^{n} D$.
5: **end for**
6: **return** MDMC value

Mean degree of metrics change (MDMC)

In perturb the original data according to the output of the XAI method, and input the modified data into the prediction model to obtain a new metrics (R^{2*}, MSE*, MAE*), then the degree of change of the metrics (D) can be is defined as:

$$D = f(M - M^*) \tag{1.1}$$

M is the original metric, and M* is the changed metric.

Combining the degree of change of all metrics, the final evaluation framework (MDMC) can be defined as:

$$\begin{aligned} MDMC &= \frac{1}{n}\sum_{i=1}^{n} D = \frac{1}{n}\sum_{i=1}^{n} f(M - M^*) \\ &= \frac{1}{n}\sum_{i=1}^{n} \left[(R_0^2 - R_i^2) + (MSE_i - MSE_0) + (MAE_i - MAE_0)\right] \end{aligned} \tag{1.2}$$

It should be noted that the values of MSE and MAE in Equation 1.2 need to be used after normalization. In theory, the larger the value of MDMC, it means that the black-box prediction model has made significant changes to the data set, thus proving the effectiveness of XAI.

Visualisation of qualitative evaluation

In order to understand the MDMC framework intuitively, take the ANN model as an example to evaluate the XAI algorithms. The changes in the metrics are shown in Figure 1.1. It is worth noting that if the stability of the ANN model itself is lacking, then randomly deleting features will also cause a decrease in the accuracy of the model's prediction. In order to eliminate this possibility, in the evaluation process, the data set of "randomly deleted features" was used as a comparison of the XAI algorithms to prove the stability of the ANN model we constructed.

As shown in Figure 1.1, compared to randomly deleting features, the perturbation based on the results of the XAI algorithms significantly reduces the accuracy of the ANN prediction model.

Metrics for quantitative evaluation

Visualization can intuitively evaluate the XAI method, but for those tasks with high-precision requirements, a quantitative evaluation matrix is necessary. The process of the XAI algorithms on LightGBM and Random Forest is similar to that of ANN and follows the same calculation formula. The values in Table 1.2 are calculated based on Equation 1.2. The larger the value, the more effective the XAI method is.

Table 1.2: Evaluation Matrix

MDCD	ANN	$LightGBM$	$RandomForest$
Random	0.8333	0.8222	0.6242
SHAP	1.4154	**1.5004**	1.3286
LIME	**1.6506**	1.3995	**1.6069**

The experimental results show that in the black-box model, the perturbation based on the XAI algorithms is more effective than the random perturbation.

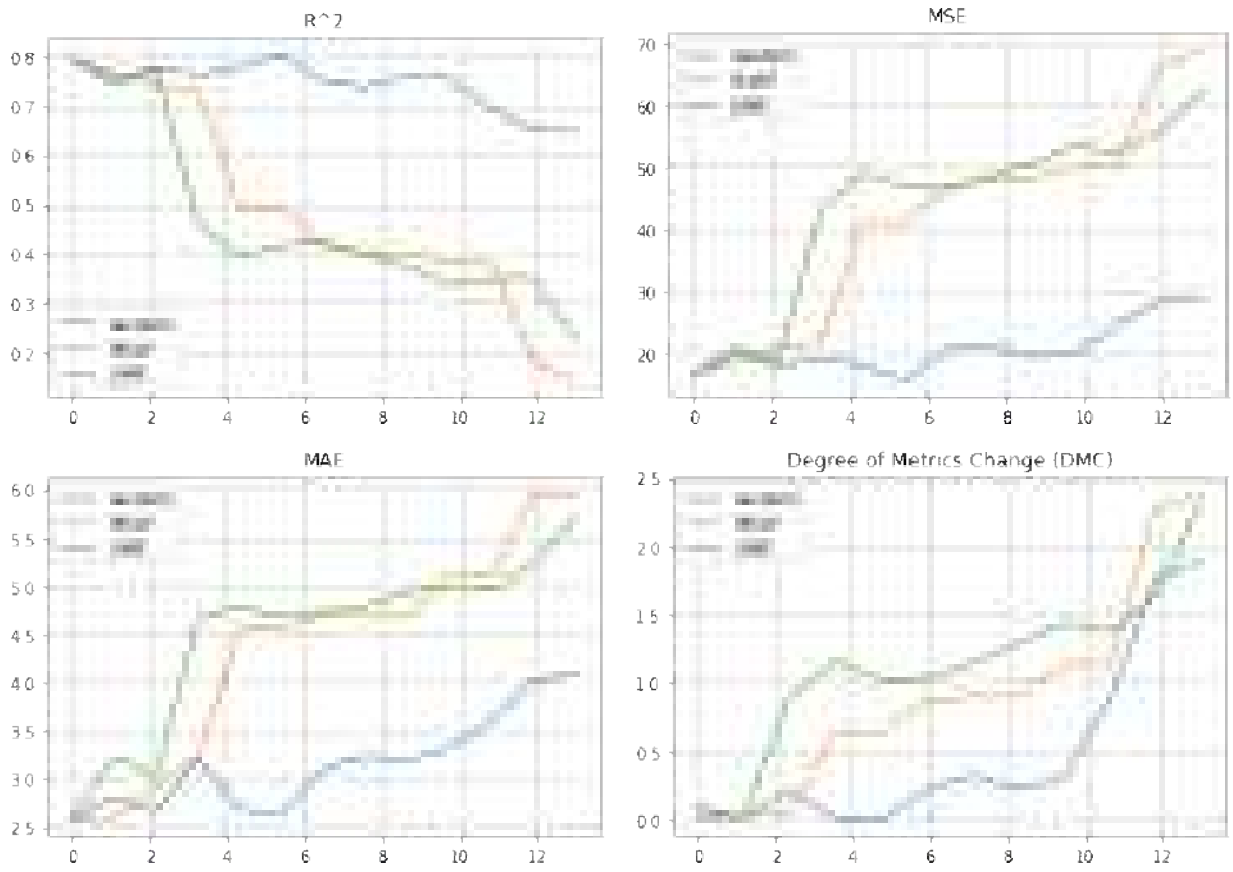


Figure 1.1: Visualization of the changes in the metrics for ANN

On the other hand, within the XAI method, LIME performs better in the ANN model and the random forest model, and SHAP is more effective for the LightGBM model.

1.2 Comparison of forecasting algorithms based on artificial intelligence

1.2.1 Comparison of energy time series forecasting

Data description

The Schneider competition places significant emphasis on the fundamental role of planning and forecasting for achieving effective operation in the energy sector. Therefore, accurate time series forecasting becomes imperative within this domain. Concurrently, the American Society of Heating, Refrigerating, and Air-Conditioning Engineers (ASHRAE) highlighted in their building energy prediction competition on Kaggle that the prevailing estimation algorithms lack

cohesion and fail to offer scalability. Consequently, the absence of a standardized approach hinders our ability to ascertain the optimal model for time series forecasting. A description of the energy dataset used in the prediction competition is as follows:

- ASHRAE: great energy predictor III[2]
  The ASHRAE dataset comprises energy usage data from a diverse range of sources within buildings, encompassing chilled water, electric, hot water, and steam meters. This comprehensive dataset spans over three years and encompasses more than 1000 buildings. In order to facilitate accurate predictions, the dataset incorporates 15 distinct features. These features include both internal characteristics of the buildings, such as building ID, usage type, area, year of completion, and floor count, as well as external factors like wind speed, wind direction, temperature, and cloud cover. Precise estimation of energy-saving investments is of paramount importance as it garners increased attention from stakeholders, particularly financial institutions. This heightened focus on the field ultimately drives advancements in building efficiency.

- Power laws: forecasting energy consumption[3]
  The dataset provided by Schneider encompasses a comprehensive set of 14 features, including building ID, temperature, and information regarding holidays and weekends, which spans a period of approximately three years. The primary objective of this competition is to enhance the accuracy of estimating global energy consumption in buildings. By leveraging this dataset, participants aim to refine and improve the existing methodologies used for estimating these consumption patterns.

- Solar power generation data[4]
  In our research, we incorporated solar power plant data as an additional resource. The primary objective of this project is to enhance grid management by accurately forecasting the near-term solar power generation

[2]ASHRA data set
[3]Schneider data set
[4]Solar Power data set

capacity. The dataset utilized encompasses five distinct features, namely device number, direct current, alternating current, temperature, and radiation. These variables were carefully selected to capture essential aspects that influence solar power generation.

In this study, a distinct time point is utilized as the demarcation between the training set and the test set, aligning with the inherent logic of the time series. Approximately 70% of the elements are required to be incorporated into the training set.

Objective function

Various criteria have been developed to assess the performance of machine learning models. Two widely used metrics include Coefficient of Determination (R^2) ans Mean Squared Error (MSE), They can be calculated as follow:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{1.3}$$

$$MSE = \frac{1}{n}\sum_{i=0}^{n}(y_i - \hat{y}_i)^2 \tag{1.4}$$

where y_i is the true value; $\hat{y}_i$ is the forecast value, $\bar{y}$ is the average of all true values, and n is the number of observations.

Comparative results

In this study, we present the loss value results of the ASHRAE competition as examples, focusing on their simplicity. To ensure efficient computation, a subset of the data was selected based on the computing power of the computer. Specifically, the training set consisted of 242,214 samples, while the test set contained 79,514 samples. To facilitate analysis, a total of 98 features were constructed using this data.

The learning curves of all models exhibited a consistent decline, as depicted in Figure 1.2. Notably, the Bi-RNN model displayed superior performance on the validation set compared to the training set. In order to provide a comprehensive comparison, it is essential to include a detailed quantitative index analysis.

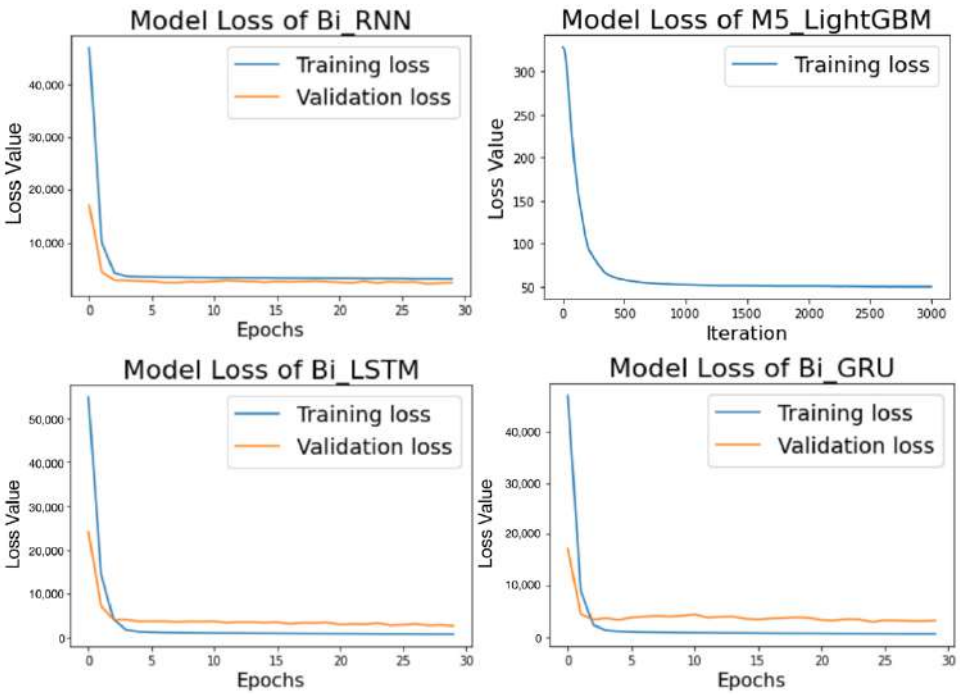


Figure 1.2: Visualisation of the training process. During the training process, the errors of all models decreased steadily, which means that these models are effective.

It is evident that all forecasting models have demonstrated exceptional performance on the ASHRAE dataset, as depicted in Figure 1.2 and Table 1.3. Among these models, Bi-RNN stands out with the most impressive performance. However, it is noteworthy that M5 LightGBM exhibits significantly shorter running times compared to both random forest and neural network models. This implies that M5 LightGBM can achieve accuracy levels similar to those of neural network models in the ASHRAE context while requiring less computational time.

Table 1.3: Forecast Quality of Power ASHRAE

ASHRAE	R^2	MSE	TimeSpent
M5LightGBM	0.9676	2516.83	39.5 s
Random Forest	0.9673	2538.70	3 min 8 s
Bi-RNN	0.9688	2426.72	6 min 33 s
Bi-LSTM	0.9655	2678.17	13 min 47 s
Bi-GRU	0.9568	3358.26	12 min 11 s

Following our initial application of these forecasting models, we proceeded to utilize them on various datasets, encompassing the Schneider competition as

well as the solar power plant dataset made available on Kaggle. In the Schneider dataset, we carefully selected a portion of the data for analysis, consisting of a training set comprising 271,803 instances and a test set containing 78,380 instances. Subsequently, we crafted a comprehensive array of 28 features based on this dataset.

In the solar power dataset, we employed the entirety of the available data (which was relatively small in size). For Plant 1, the training set contained 2,393 instances, while the test set comprised 864 instances. Similarly, for Plant 2, the training set encompassed 2,293 instances, with the test set consisting of 862 instances. Based on this dataset, we created a total of 19 features to facilitate our analysis.

To assess the effectiveness of the forecasting models, it is crucial to measure their performance (refer to Table 1.4). Accuracy is evaluated within each dataset, while stability is examined across all datasets. Notably, the target variable's data values in both the Schneider and Kaggle datasets are considerably large, rendering the loss optimization of all forecast models nearly ineffective. Consequently, a prudent approach is adopted during the forecasting process where the target variable and feature data undergo separate processing using the MinMax algorithm.

Table 1.4: Forecast Quality of Power Schneider

Schneider	R^2	MSE	TimeSpent
M5LightGBM	0.9381	0.0001	42.1s
Random Forest	0.9297	0.0001	1min 34s
Bi-RNN	0.8595	0.0003	55.2 s
Bi-LSTM	0.9146	0.0001	1min 19s
Bi-GRU	0.9165	0.0001	1min 48s

In relation to the solar power datasets, the performance of forecasting models for power plant 1 is consistently excellent, as evidenced by Table 1.5. Specifically, the M5 LightGBM model stands out for its accuracy and efficiency, surpassing all other models in terms of both metrics.

When it comes to power plant 2, the primary objective is to maintain high accuracy, achieving a commendable value of 0.9005. In this regard, the M5

Table 1.5: Forecast Quality of Solar Power Generation—Plant 1

Solar-1	R^2	MSE	TimeSpent
M5LightGBM	0.9928	0.0008	0.74 s
Random Forest	0.9723	0.0034	0.99 s
Bi-RNN	0.9665	0.0041	6.17 s
Bi-LSTM	0.9887	0.0013	12.9 s
Bi-GRU	0.9865	0.0016	8.36 s

LightGBM model emerges as the most optimal choice, ensuring efficient operation while minimizing time costs. These findings are presented in Table 1.6 for reference.

Table 1.6: Forecast Quality of Solar Power Generation—Plant 2

Solar-2	R^2	MSE	TimeSpent
M5LightGBM	0.9005	0.0081	0.16 s
Random Forest	0.8689	0.0107	1.19 s
Bi-RNN	0.9329	0.0055	6.39 s
Bi-LSTM	0.8917	0.0088	13.7 s
Bi-GRU	0.9185	0.0066	8.76 s

The results indicate that M5 LightGBM exhibits distinct advantages. In terms of accuracy, M5 LightGBM outperforms other algorithms in both datasets (refer to Table 1.4 and Table 1.5). Additionally, it demonstrates significantly shorter processing time, especially when compared to neural network algorithms. While M5 LightGBM may not achieve the best performance in certain datasets (see Table 1.3 and Table 1.6), it still attains a comparable forecasting accuracy ($R^2 > 0.9$) at a lower computational cost. Consequently, after conducting a comprehensive comparison, we conclude that M5 LightGBM serves as a superior forecasting model.

Ensemble learning algorithms, despite their seemingly simplistic nature, often deliver exceptional performance in practical applications. This can be attributed to their ability to amalgamate multiple fundamental algorithmic ideas into a coherent framework, bolstering strengths and minimizing weaknesses. Consequently, this optimization enhances the robustness and generalization capacity of the original basic algorithms, thereby facilitating stable model

operations. To provide visual representation of the forecast results, please refer to Figure 1.3.

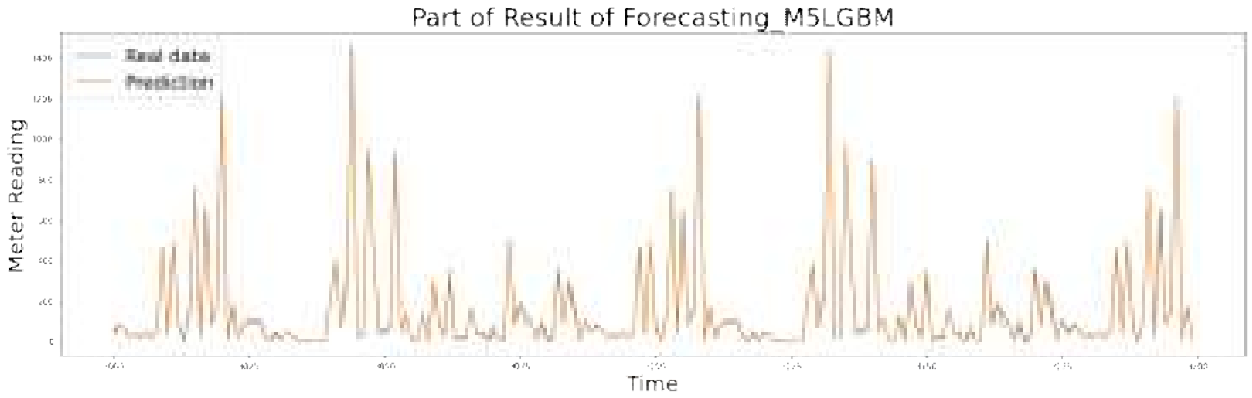


Figure 1.3: Visualisation of M5LightGBM forecasts on the ASHRAE dataset

1.2.2 Comparison of PM2.5 time series forecasting

Data description

The dataset used in this study comprises PM2.5 concentration data collected at the Beijing Olympic Sports Centre Gymnasium[5]. The data spans a period from 1 March 2013 to 28 February 2017, with measurements recorded at hourly intervals.

The dataset used in the study consists of 31,815 observations. It encompasses 11 variables, comprising both meteorological conditions and emission factors. The emission factors encompass particulate matter ($PM10$, ug/m^3), sulfur dioxide (SO_2, ug/m^3), nitrogen dioxide (NO_2, ug/m^3), carbon monoxide (CO,ug/m^3), and ozone (O_3,ug/m^3). Among the meteorological conditions, there are continuous variables such as temperature (TEMP, degree Celsius), pressure (PRES, hPa), dew point (DEWP, degree Celsius), rainfall (RAIN, mm), wind speed (WSPM, m/s), and one discrete variable, wind direction (WD). Wind direction is coded using a natural order system with 16 directions: NNW, N, NW, NNE, ENE, E, NE, W, SSW, WSW, SE, WNW, SSE, ESE, S, SW.

Objective function

Given that our focus in this section pertains to long time series prediction as a more intricate task, it becomes imperative to employ comprehensive evaluation

[5]Beijing PM2.5 data set

metrics for effectively assessing the performance disparities among various AI models. Therefore, we have specifically selected objective functions such as R^2 (coefficient of determination), RMSE (root mean squared error), and MAE (mean absolute error) to serve as the basis for our assessments. These metrics will enable us to thoroughly evaluate the efficacy of distinct AI models in tackling this complex prediction challenge.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{1.5}$$

$$RMSE\% = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}}{n} \times 100 \tag{1.6}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{1.7}$$

where y_i is the true value; $\hat{y}_i$ is the forecast value, $\bar{y}$ is the average of all true values, and n is the number of observations.

Comparative results

Long-term forecasting based on statistical algorithms are challenging due to non stationarity and noise. Deep learning [107–111] and ensemble learning [112–114] are both powerful techniques for handling non-linear and non-stationary data. They offer effective ways to uncover relationships in complex datasets. Ensemble learning accomplishes this by combining the predictions of multiple models, leveraging their collective wisdom to enhance accuracy. Boosting algorithms [113, 114], such as XGBoost, LightGBM, and CatBoost, are widely used in ensemble learning, along with Bagging algorithms. On the other hand, deep learning utilizes a complex network structure to identify intricate patterns and dependencies between input and output variables. This technique is categorized into three types based on network structure: Artificial Neural Networks (ANN) and Recurrent Neural Networks (RNN). These algorithms find significant applications in domains like long-term PM2.5 forecasting [4–6], where nonlinear relationships and patterns are prevalent. By employing ensemble learning and deep learning, researchers and practitioners can avoid the problem of non-linear and non-stationary data in these fields more effectively.

We applied emission factors and meteorological conditions as inputs to black-box models for PM2.5 forecasting and evaluated the performance of these models at different forecast horizons (30 days, 90 days, and 180 days). The results are summarized in Table 1.7.

Horizon	30			90			180		
Metrics	R^2	$RMSE$	MAE	R^2	$RMSE$	MAE	R^2	$RMSE$	MAE
XGBoost	0.9623	24.180	0.0144	0.9376	28.167	0.0239	0.9349	28.471	0.0210
LightGBM	0.9639	23.653	0.0148	0.9433	26.863	0.0227	0.9414	27.001	0.0203
Catboost	**0.9734**	**20.310**	**0.0128**	0.9297	29.893	0.0252	0.9349	28.471	0.0214
ANN	0.9278	33.298	0.0218	0.9498	26.305	0.0256	0.9304	28.036	0.0264
RNN	0.9614	24.462	0.0149	0.9639	21.418	0.0206	0.9564	23.274	0.0185
LSTM	0.9659	22.973	0.0148	0.9521	24.677	0.0216	0.9531	24.149	0.0182
GRU	0.9327	32.307	0.0227	0.9513	24.878	0.0237	0.9520	24.433	0.0189
Bi-RNN	0.9603	24.807	0.0165	0.9570	23.375	0.0262	0.9576	22.970	0.0189
Bi-LSTM	0.9419	30.005	0.0212	**0.9661**	**20.740**	**0.0196**	0.9576	22.949	0.0185
Bi-GRU	0.9648	23.345	0.0153	0.9656	20.912	0.0214	**0.9589**	**22.601**	**0.0182**

Table 1.7: Comparison of forecasting performance. The optimal model is filtered through the validation set and compared on the test set. The performance metric values reported here are taken from the test set.

The analysis reveals that Catboost performs best for a 30-day forecast horizon, while Bi-LSTM yields the highest accuracy at 90 days, and Bi-GRU excels at 180 days. When considering model types, ensemble learning models demonstrate superior performance for shorter forecast horizons, whereas recurrent neural network models, including their derived bi-directional counterparts, exhibit better accuracy for longer-term forecasting. Conversely, the multilayer perceptron model did not deliver noteworthy results in our study.

1.3 Conclusion of chapter 1

After comparing with the accuracy of traditional linear regression and white-box models, black box models such as neural networks and ensemble models including boosting algorithms and bagging algorithms have absolute advantages in prediction, however, the unexplainable nature makes the black-box model an

obstacle in the process of practical application. The XAI algorithms can obviously alleviate this problem. By identifying features with relatively high contributions, users can understand the black box model more clearly when using the black box model to predict, thereby increasing trust. However, because many XAI algorithms have different principles and characteristics, different XAI algorithms output different results for the same black-box model. Therefore, the establishment of an evaluation framework for XAI algorithms is necessary. The XAI algorithms are evaluated through the established XAI evaluation framework - MDMC. The results show that LIME is more suitable for ANN model and random forest model based on bagging algorithm, and SHAP is more suitable for LightGBM based on boosting algorithm.

Chapter 2

Explainable AI algorithms for calculating importance of time periods

The application of Explainable AI (XAI) in time series forecasting has gradually attracted attention, given the widespread implementation of machine learning and deep learning. ShapTime - A general XAI approach based on shapley value specially developed for explainable time series forecasting, which can explore more plentiful information in the temporal dimension, instead of only roughly applying traditional XAI approaches to time series forecasting as in previous works. **The research results have been published in the conference paper [23, 26]**. The **novelty** of our method is that it achieves explanation in the time series dimension, i.e., it is able to output the importance of historical data for the forecasting results, which is not possible with other model-agnostic methods.

2.1 Description of problems in calculating importance of time periods

2.1.1 Lack of generalisability

Numerous time series forecasting competitions including M4 [77] and M5 [78] have shown that ML and DL perform significantly better than traditional statistical methods, especially for more complex tasks. This has led to research on the application of Explainable AI (XAI) in time series forecasting. explainable time series forecasting aims to improve the trustworthiness of ML and DL in fields such as Finance, Energy and Meteorology. There are two main approaches to apply XAI in time series forecasting models: (1) directly using the existing model-

agnostic method with high generality; (2) developing a model-specific method specifically for the model. These two approaches directly caused two key problems.

Bottleneck 1*: In time series forecasting, the existing model-agnostic method is roughly applied, resulting in insufficient explanation.* The essential reason for the insufficient explanation is that most of the existing model-agnostic methods are feature attribution methods, given that XAI was originally developed based on regression and classification tasks, such as SHAP [51], LIME [52], etc. In contrast to time-series data, there is no temporal relationship among data instances for regression and classification, so the corresponding model-agnostic method pays more attention to feature importance (or contribution). However, this is not sufficient for time series forecasting. In time series forecasting models $y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \omega_t$; $\omega_t = b + k_1 x_1 + k_2 x_2 + \cdots + k_n x_n$. y_{t-i} is the historical data of the target variable, and ω_t is composed of features x_i and intercept terms b. These traditional model-agnostic methods can only output feature importance k_i, but not the importance of time itself ϕ_i. Actually, due to human cognitive inertia, it is common to use feature importance as the explanation result even in model-specific methods.

Bottleneck 2*: The specially developed model-specific method needs to be embedded in the model, resulting in low generality and high application cost.* Several works have noticed problem 1 and developed some explainable time series forecasting models that can show periodicity as well as trend, which acts on the temporal dimension. However, these methods are still not able to output ϕ_i, and the degree of explanation is limited. In addition to the above problems, there is also a common problem in the field of XAI, which is listed as problem 3 in this work.

Bottleneck 3*: Lack of application scenarios.* In works involving XAI, usually, explanation results are only presented as an innovation. In a large number of previous works, the application scenarios of XAI are only mentioned in the introduction part, including helping users trust the model and developers debugging the model, but these scenarios are not implemented.

Overall, on the one hand, the current model-agnostic method with high generality cannot fully explain the time series forecasting task, that is, it cannot realize the explanation of the temporal dimension. On the other hand, the

model-specific method that can achieve temporal dimension explanation to a certain extent also has limitations, and its low generality also increases the cost of use. Therefore, a general XAI approach to explainable time series forecasting is needed, given the increasing importance of ML and DL in time series forecasting.

ShapTime[1] realizes the attribution of time by calculating the shapley value [57] on the temporal dimension, and finally outputs the importance of time itself ϕ_i. Therefore, ShapTime can realize the explanation in the temporal dimension, and it belongs to the model-agnostic method, which means that it can be deployed on any forecasting model at a lower cost. The foundation of ShapTime is the shapley value, which is the basis for numerous attribution methods including SHAP. shapley value comes from cooperative game theory, which studies how to reasonably distribute the benefits to the players in the alliance, and it has been proved to have some good properties. Therefore, in recent years, the development of XAI methods around shapley value is trying to become a stable path [103], and our ShapTime is exploring explainable time series forecasting as a branch on this path. Its contributions include:

- It realizes time attribution in the temporal dimension, that is, the importance of time itself ϕ_i can be obtained
- As a highly general model-agnostic method, it can be deployed in any forecasting model
- Its explanation results can be used as a guide to improve the forecasting performance of the model

2.1.2 High computational complexity

The introduction of feature attribution methods such as SHAP aims to address the interpretability issue of black-box models. Among various feature attribution methods, SHAP stands out as the only method that satisfies several desirable properties. Furthermore, the corresponding software library is highly compatible with the Gradient boosting algorithm. As a result, the combination of SHAP

[1]The github library of ShapTime

and Gradient boosting has gained widespread attention in the industry due to its convenience. Each year, a significant number of application-oriented research works based on this combination are published, focusing on tasks related to time series forecasting.

However, the initial development of SHAP was primarily focused on regression and classification tasks. It assumes that samples are independent and identically distributed (i.i.d.), thus it can only explain the influence of features X_i on y_i (Figure 1), i.e., the forecast value $\hat{y}_i = \Phi_i X_i = \phi_1 x_1 + \cdots + \phi_S x_S$, but not the impact of historical data $Y_L = \{y_{i-L}, \cdots, y_{i-L}\}$ on y_i.

This problem can be addressed by transposing Y_L from the temporal dimension to the feature dimension. In other words, the original data is processed as $t_i = \{X_i, Y_L, y_i\}$. By doing this, the historical data Y_L of y_i is integrated into the feature dimension. Consequently, using SHAP, forecast value $\hat{y}_i$ can be attributed as $\hat{y}_i = \Phi_i X_i + \Psi_L Y_L$, $\Psi_L Y_L = \psi_{i-L} y_{i-L} + \cdots + \psi_{i-1} y_{i-1}$, allowing for explanation in the temporal dimension. However, when we need to explain a longer historical data range in the temporal dimension, it leads to high-dimensional data. For instance, with a time frequency of 1 hour and wanting to investigate the impact of the past year's historical data on the current value y_i, i.e., $L = 24 \times 365 = 8760$, the resulting dimensionality of the processed temporal data becomes $8760 + S$. Similarly, with a time frequency of 15 minutes, L will reach $96 \times 365 = 35040$. Such high-dimensional data poses two challenges for achieving explainability:

Challenge 1: It requires substantial computational resources, to the point where computation may become infeasible. The core algorithm of SHAP is the shapley value, and its computational complexity is 2^F (where F represents the total number of features). Despite the existence of some variant algorithms in the current SHAP library that can reduce the complexity, high-dimensional data still poses a significant burden on their calculations. The introduction of SHAP and its variants can be found in Section 3.1.

Challenge 2: Extracting meaningful information from high dimensional explanatory results becomes difficult. Assuming negligible computational resource consumption, we obtained the temporal explanation of this high-dimensional data. Taking a time frequency of 1 hour and $L = 8760$ as an example, the final explanatory results $\Psi_L = \{\psi_{i-8760}, \cdots, \psi_{i-1}\}$ will be

outputted for the historical data of y_i. In fact, it is challenging to extract useful information from such dense data, and the presence of abnormal fluctuations in time series data may still prevent human understanding of these explanatory results.

In short, to overcome the bottleneck of SHAP in time series forecasting, the two aforementioned challenges must be addressed. Therefore, it is necessary to have a reasonable dimensionality reduction framework for Y_L that can preserve temporal information. figure 2.1 intuitively demonstrates the approach to addressing these challenges by reducing Y_L to Y_K.

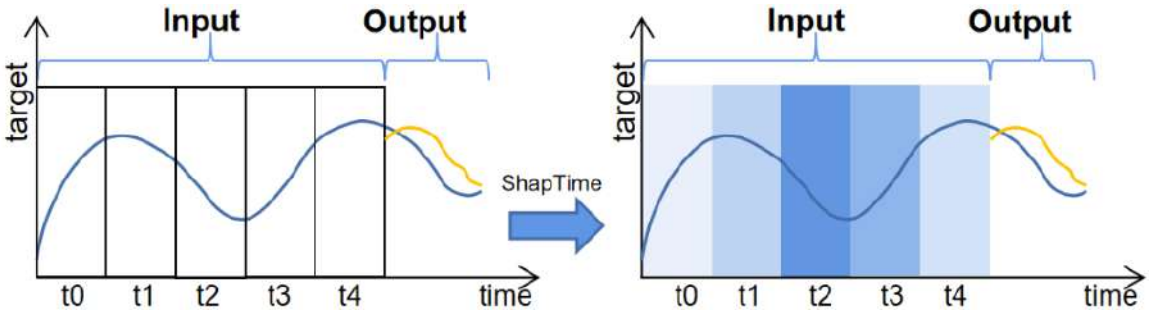


Figure 2.1: Dimensionality reduction. In this example, K = 5

The choice of K depends on practical requirements, for example, if we are interested in determining the day within the past week that had the greatest impact on the current y_i, with a time frequency of 1 hour, then $L = 7 \times 24 = 168$ and $K = 7$. This results in a new dataset $t_i = \{X_i, Y_K, y_i\}$, $Y_K = \{y_{T_0}, y_{T_1}, \cdots, y_{T_6}\}$. By calculating SHAP, $\hat{y}_i$ can ultimately be decomposed into $\hat{y}_i = \Phi_i X_i + \Psi_K Y_K$, $\Psi_K Y_K = \psi_{T_0} y_{T_0} + \cdots + \psi_{T_6} y_{T_6}$, thus achieving our explanatory goal.

2.2 ShapTime: explainable AI algorithm with generalizability and low computational complexity for calculating importance of time periods

Shapley value is one of the classic theories in cooperative games, which aims to distribute the benefits fairly to the players in the alliance. There is the correspondence between shapley value and model explanation: the features used for training correspond to "players", and the model's predictions correspond to

"revenues". Therefore, the assignment achieved by shapley value can attribute the prediction result to the features, that is, the contribution (importance) of each feature to the prediction result. The assignment is achieved by the following formula:

$$k_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(n - |S| - 1)!}{n!} \times (v(S \cup \{i\}) - v(S)). \tag{2.1}$$

where N is the set of all players (features) (1,2,3...i...n), which is the complete set; S is a subset of N, in which removes the explained feature i, with a total of 2^N; v is the gain function ($v(S) = E_{\hat{D}}[f(x) \mid x_S]$, where $\hat{D}$ is the empirical distribution of the training data and f is the black-model).

The reason why shapley value has become the basis of many XAI approaches is that it has some desirable properties, including: efficiency, symmetry, linearity and null player, and among many attribution methods, it is the only mapping ($v : 2^N \to \mathbb{R}$) that can satisfy the above properties.

shapley value has two key elements: "player" and "gain function". When both are determined, shapley value has the possibility to be calculated theoretically. Accordingly, shapley value is the attribution of "player".

In order to achieve attribution on the temporal dimension, time points should be considered as "players". Corresponding to the time series forecasting model, y_{t-k} is regarded as a "player", and after defining the gain function, the attribution of y_{t-k}, namely ψ_k, can be realized through shapley value. The modified formula (ShapTime) is:

$$\begin{aligned} \psi_{t_k} = \sum_{S' \subseteq T \setminus t_k} & \frac{|S'|!(|T| - |S'| - 1)}{|T|!} \\ & \times (v'(S' \cup t_k) - v'(S')). \end{aligned} \tag{2.2}$$

where t_k is super-time (players), T is the set of all super-time t_k. S' is a subset of T, in which removes the explained super-time t_k, with a total of 2^T. v' is the gain function of ShapTime.

2.2.1 Super-time: method to reduce computational complexity

The so-called temporal attribution is to attribute the forecasting results to time, which means treating the time points as "players", which will inevitably cause the dimension explosion and the system crash. A similar problem is encountered in the XAI approach for image recognition - a large number of pixels makes the computational cost increase exponentially. In order to solve this problem, [52] proposed the concept of super-pixel, which is to cluster pixels with high similarity into a whole, and then participate in the calculation of pixel contribution, thereby greatly reducing the calculation cost.

Algorithm 2 Super-time

Input: Original data set: X

Parameter: The number of super-time: n

Output: All the super-time t_i

1: Let $L = \text{int}(\text{length}(X)/n)$.
2: Let start = length(X) - $L \times n$.
3: Let $X_{used} = X\,[start :, :]$.
4: **for** i in range (n) **do**
5: $\quad t_i = X\,[i \times L : (i+1) \times L]$.
6: **end for**
7: **return** all the t_i

In view of the similarity of the problem, we refer to his method to construct super-time in the temporal dimension, that is, the data set is divided into n super-times t_i according to the temporal dimension.

$$t_i = \{y_{t-i}, y_{t-i-1}, y_{t-i-2}, \cdots\} \tag{2.3}$$

In ShapTime, super-time is equivalent to "player". Crucially, although super-time controls the computational cost within an acceptable range, considering the complexity of $O(2^n)$ in the ShapTime, we recommend that n not exceed 10 or 11. Algorithm 4 shows the construction process of Super-time.

2.2.2 Redefinition of functions for generalizability

In ShapTime, the attributed object is super-time, which is the collection of time points within a period of time, that is, the "player" is no longer a time point at

this time, but a time period.

Algorithm 3 Gain Function of ShapTime

Input: Explained model: f; Super-time: t_i

Output: All the gain value: v'

1: Let $S' \subseteq T \setminus t_i$.
2: Let $T = \{t_0, t_1, \cdots, t_n\}$
3: **for** S' in T **do**
4: **if** length(S') == 1 **then**
5: $S' = \{t_i\}$.
6: $v' = \text{sum}(f(t_i))/\text{length}(t_i)$.
7: **else**
8: $S' = \{t_i, t_j, \cdots\}$.
9: $S_c = concat(t_i, t_j, \cdots)$
10: $v'(S_c) = \text{sum}(f(S_c))/\text{length}(S_c)$.
11: **end if**
12: **end for**
13: **return** 2^T gain values: v'

Correspondingly, our forecasting target y_t is also the collection of time points. However, the calculation of shapley value requires that for each "player" combination, the corresponding gain function outputs a value. Therefore, here, the averaged model f forecasting results are taken as the gain function v' of ShapTime (lines 6 and 10 in Algorithm 3). The formula is:

$$v'(S') = avg(f_{S'}(x_{S'})) \tag{2.4}$$

Eventually, the gain value v' of all combinations of super-time will be obtained, a total of $2^{|T|}$. In this way, according to equation (7), the forecasting result can be attributed to each super-time t_i, that is, ϕ_{t_i}, so as to realize the explanation in the temporal dimension.

2.2.3 Visualisation of importance of time periods

Currently, there is the lack of recognized correct labels in the XAI research field, which poses challenges for the evaluation of XAI methods. On the other hand, our ShapTime is the model-agnostic method for the explanation of the temporal dimension. In the field, this form of explanation is scarce, resulting in the lack of

objects for comparison, so evaluation by contrast is difficult to achieve. Therefore, we propose some reasonable evaluation criteria to evaluate ShapTime effectively to a certain extent. It also provides a basic benchmark for subsequent research.

Property 1*: Under the premise of using the same data set, the explanation results of the XAI approach to similar types of models should be consistent to a certain extent.*

Property 2*: The XAI approach should be subject to sensitivity analysis, that is, when vital "players" are perturbed, there is the significant drop in forecasting performance.*

XAI's evaluation criteria are critical for users to build trust in machines. The most important of these is that the explanation results of XAI should maintain a certain degree of stability (Property 1), which is the basis for building trust. On this basis, the XAI method also guarantees validity (Property 2), that is, when important "players" are changed, the performance of the model will decline significantly. This can prove to a certain extent that the explanation results of this XAI approach are valid. We applied ShapTime on the above 8 models and generated the explanation results (figure 2.2: For the sake of brevity, we only show the explanation results of ShapTime for XGBoost and LSTM in 5 data sets, and the complete explanation results are in the Github).

In explanation results, the daily climate dataset (Figure.2.2(a)(f)) is taken as the example. In this example, the number n of super-time is set to 8, then the corresponding theoretical modified time series model is:

$$y_t = \phi_{t_0} t_0 + \phi_{t_1} t_1 + \cdots + \phi_{t_7} t_7 + \omega_t \tag{2.5}$$

After the training is completed, we use ShapTime to explain XGBoost and LSTM, so as to attribute the forecasting results of the model to each super-time, and then get each ϕ_{t_i} value, and display them visually through the heat map. We can clearly see that both models capture the most recent super-time t_7 as the most important input. In the energy consumption dataset, the two models regard t_7 as the most important super-time; in gold price it is t_9; in tesla stock it is t_{10}. Different situations appear in solar generation. The super-time captured by the two models is different. XGBoost captures t_0 as the most important, while LSTM captures t_5.

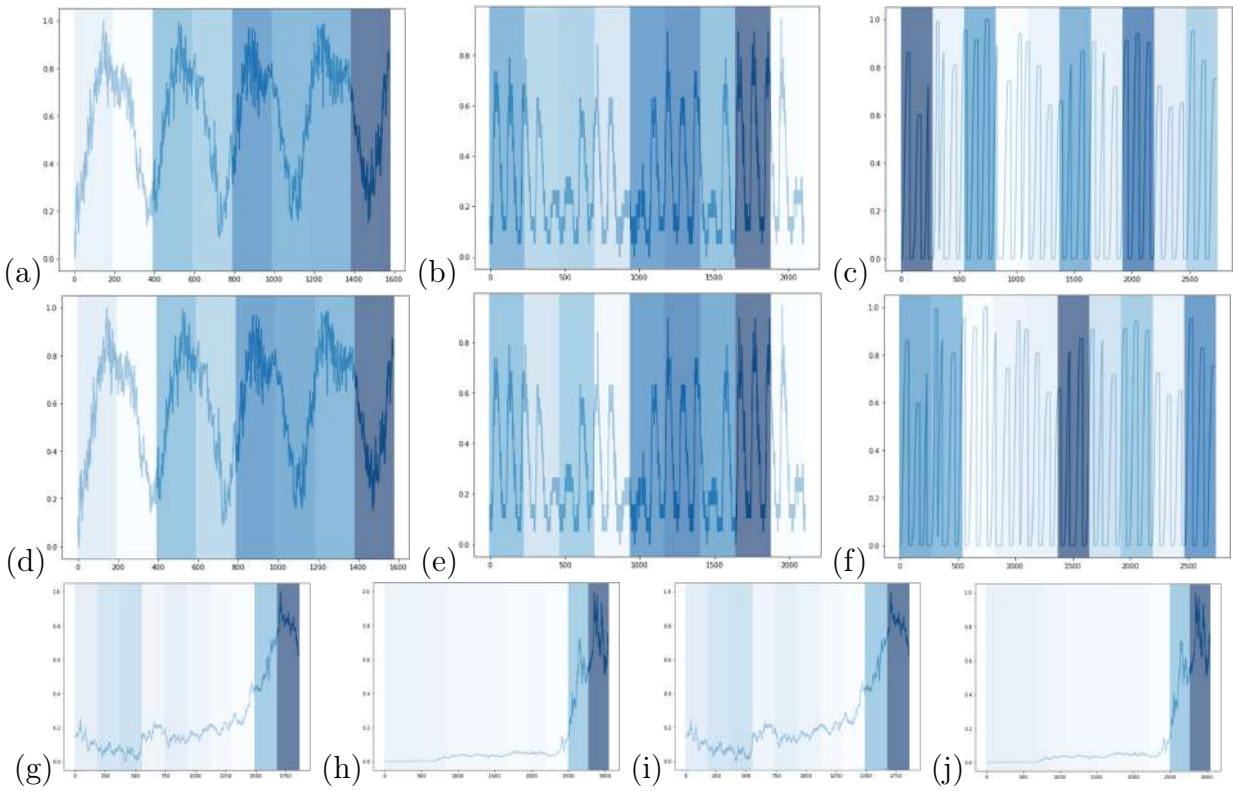


Figure 2.2: Example of explanation results from ShapTime. This is ShapTime's explanation of XGBoost and LSTM. a: XGBoost-Climate; b: XGBoost-Energy; c: XGBoost-Solard; d: LSTM-Climate; e: LSTM-Energye; f: LSTM-Solar; g: XGB-Gold; h: XGB-Tesla; i: LSTM-Gold; j: LSTM-Tesla. The heat map is used to visualize the explanation results of the input data. The darker the color, the more important the super-time. This means that the model pays more attention to this super-time during the training process. It can be seen from figures that the explanation of trend data by LSTM and XGBoost is basically consistent, and there is a certain degree of difference in the explanation of periodic data, especially in the Solar Generation data set.

Looking at all the 5×8 explanation results, the most important super-time captured by all models is the last one in the trend data, but in the periodic data, there is no obvious rule. However, if we analyze the forecasting performance of the model (Table.2.1), we can still find some potential rules, that is, the Boosting model is more suitable for periodic data, while the RNN-based model is more suitable for periodic data when the explanation results are generally consistent.

According to the above evaluation criteria, we evaluate the explanation results of ShapTime, and the evaluation examples are shown in figure 2.3. Similarly, for simplicity, we only display the evaluation results on the Boosting model. figure 2.3(a)(d) represents the explanations of ShapTime for XGBoost and LightGBM,

respectively. It can be intuitively seen that the explanation of ShapTime can be maintained all the time for the same type of models, which is a common pattern in all the explanation results. This is key to the trust of users from all industries. Imagine if the explanation results change frequently during use, which will directly lead to user distrust or even abandonment.

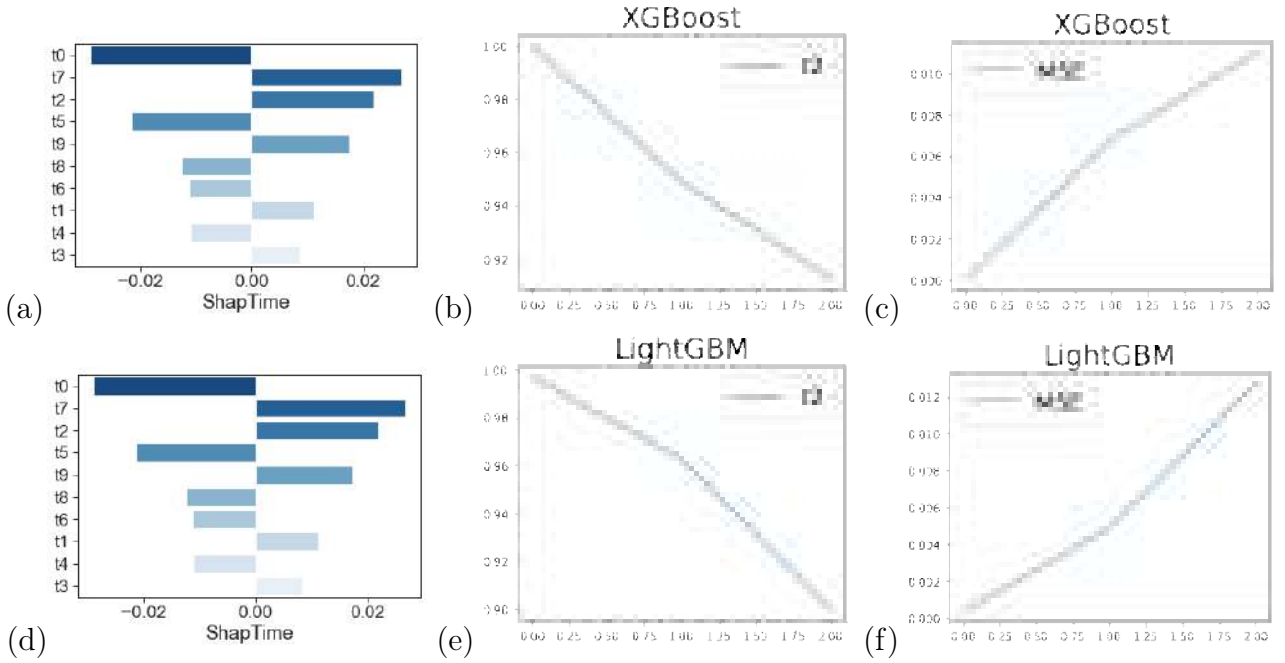


Figure 2.3: Evaluation example of ShapTime explanation results. a: XGB-Solar; b: XGB-r^2; c: XGB-MSE; d: LGB-Solar; e: LGB-r^2; f: LGB-MSE. (a) and (b) satisfy property 1, (b)(c) and (e)(f) satisfy property 2.

This is exactly the purpose for which Property 1 was defined. However, in case of different types of models, in principle, differences in the explanation results need to be allowed, since models of varying architectures are not equally sensitive to the nature of the distribution of the data. As in figure 2.2 (c)(h), the explanation results present differences in the various models.

On the other hand, theoretically, if we perturb the training data according to the explanation results, i.e., the important super-time is replaced with the least important, then the forecasting performance of the model will be significantly decreased. In this example, the most important t_0 is replaced with the least contributing t_3, as well as t_7 is replaced with t_4. Correspondingly, the r^2 and MSE of XGBoost and LightGBM exhibit significant and gradual performance degradation (figure 2.3(b)(c)(e)(f)). This evaluation schema is the relatively

classical evaluation method in the field of XAI, and it is also known as sensitivity analysis. In this work, it is summarized in property 2.

2.2.4 Improvement of forecasting accuracy using ShapTime

Even though research on explainability has received increasing attention in recent years, however, in general, explanation results are simply presented as samples. This phenomenon does not only occur within the domain of explainable time series forecasting, but is the pervasive problem for the entire domain of Explainable AI (Bottleneck 3). To explore the application scenarios of the XAI approach, we try to use ShapTime's explanation results as a guideline, aiming to improve the performance of time series forecasting.

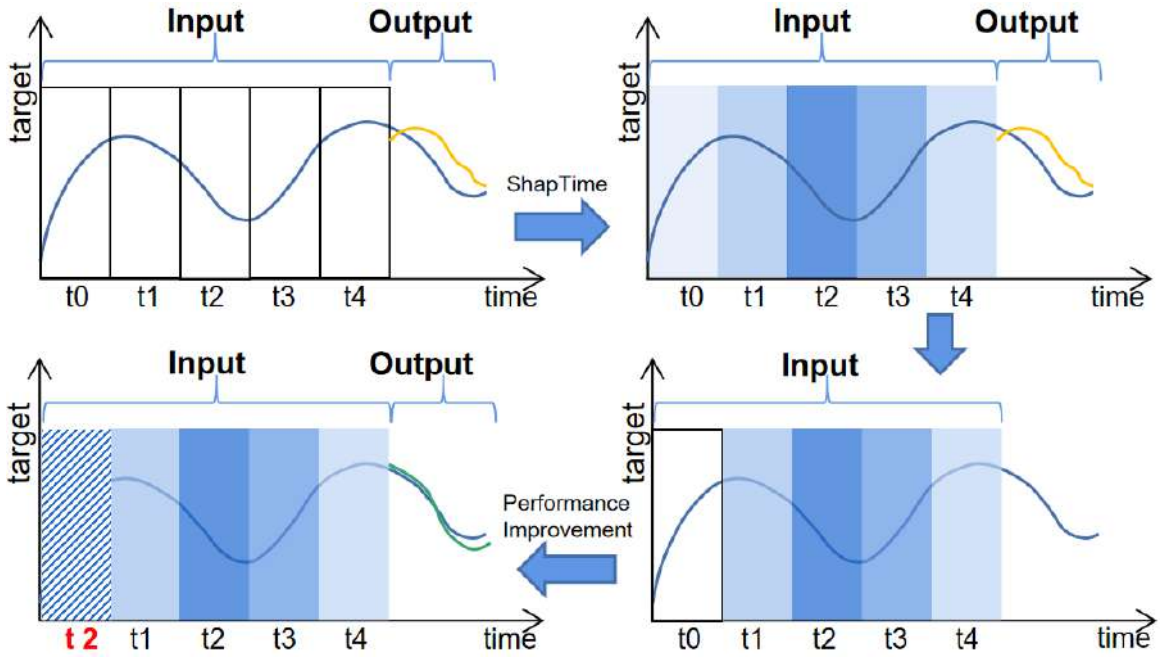


Figure 2.4: ShapTime-based performance improvement process. The blue line represents the original data set, and the orange line represents the forecasting results of the model. The blue background represents the explanation result of ShapTime. The darker the color, the more important the t_k is, and vice versa. After obtaining the importance in the temporal dimension, the t_k with the lowest value is replaced by highest one. The green line represents the improved forecasting results.

First of all, the input data set is divided equally into several super-times, and then, the importance of each super-time is calculated using ShapTime and visualized by the heat map, where the darker color means more important and the lighter color is less important. The way to improve performance is to replace

low-importance t_k with high-importance t_k, and use this newly constructed data set to retrain the model. The reason for constructing such the improvement scheme (figure 2.4) is that ShapTime based on shapley value is the model-agnostic method, so its operation logic is based on the data set. In detail, the disturbance of the input data will change the forecasting result, and this type of method realizes attribution through this change. In fact, $(v'(S' \cup t_k) - v'(S'))$ represents this change. Therefore, dataset-specific refinements are naturally adopted when utilizing the explanation results output by this method.

Based on our developed approaches, FI-SHAP and ShapTime, we provide explanations of the model from two different perspectives. The former focuses on the feature (or variable) level and improves performance through enhanced feature engineering techniques. On the other hand, the latter operates at the temporal level and enhances performance through data augmentation methods.

Experiments

In order to verify the practical application ability of ShapTime, we selected 5 real data sets, including: climate data, energy consumption, solar power, gold price, tesla stock. Among them, the first three are periodic data, and the last two are trend data, so as to test the performance of ShapTime under different data types.

On the other hand, there are two types of black box models involved in training: Boosting model and RNN-based model. The former consists of XGBoost and LightGBM; the latter includes RNN, LSTM, GRU (RNN-based) and Bi-RNN, Bi-LSTM, Bi-GRU(Bi-RNN-based). These forecasting models basically include the current mainstream methods in competition and practice. Their forecasting performance measures are shown in table 2.1

Improvement results

To create valuable XAI application scenarios, we use the explanation results of ShapTime as our guide for achieving improved forecasting performance, and the improvement process is shown in figure 2.4. table 2.2 shows the improved performance metrics (compared to table 2.1), and the results show that Bi-RNN-based and Boosting still maintain their original advantages in trending

Data	Climate	Energy	Gold	Solar	Tesla
XGB	0.7756	0.7570	**0.7988**	0.9307	0.7632
	(0.0081)	0.0095	0.0006	0.0055	0.0029
LGB	**(0.7814)**	**(0.8290)**	0.7779	**0.9322**	**0.7777**
	0.0082	**(0.0071)**	**0.0006**	**0.0053**	**0.0023**
RNN	0.7170	0.6574	0.7018	0.9587	0.8133
	0.0104	0.0128	0.0011	0.0051	0.0022
LSTM	**0.7507**	0.6249	**0.8182**	0.9544	**0.8586**
	0.0081	0.0153	**0.0005**	0.0057	**0.0017**
GRU	0.6719	**0.7130**	0.7096	**(0.9661)**	0.8176
	0.0117	**0.0119**	0.0009	**(0.0035)**	0.0025
Bi-R	**0.7257**	0.6804	0.8048	0.9312	0.7990
	0.0102	0.0120	0.0008	0.0099	0.0029
Bi-L	0.6903	**0.7122**	**(0.8791)**	0.9326	**(0.8689)**
	0.0103	**0.0119**	**(0.0005)**	0.0082	**(0.0016)**
Bi-G	0.7136	0.5927	0.8664	**0.9481**	0.7055
	0.0109	0.0154	0.0006	**0.0056**	0.0045

Table 2.1: Metrics to forecasting performance. forecasting models are divided into three categories: Boosting; RNN-based; Bi-RNN-based. r^2 and MSE are used as forecasting performance metrics, for each model, the first row is r^2 and the second row is MSE. Within each class of models, the best model is marked (in bold); among all models, the best model is additionally marked (in brackets). As can be seen from the table, Bi-RNN-based performs best in trending data (Gold Price and Tesla Stock), and Boosting performs best in periodic data (Daily Climate, Energy Consumption, and Solar Generation).

and periodic data, respectively.

The measure of the degree of improvement is presented in figure 2.5 (a), which shows the average degree of improvement for each type of model in each data set. The percentages are calculated based on the degree of improvement in table 2.2 compared to table 2.1. It can be intuitively seen that ShapTime shows the maximum improvement for Bi-RNN-based, and the most significant improvement for solar generation. Overall, the effect of ShapTime on boosting is less significant than that of other types of models. Specifically, ShapTime has the most significant improvement on Bi-GRU. In the original forecasting performance (table 2.1), Bi-GRU does not possess the best performance in all

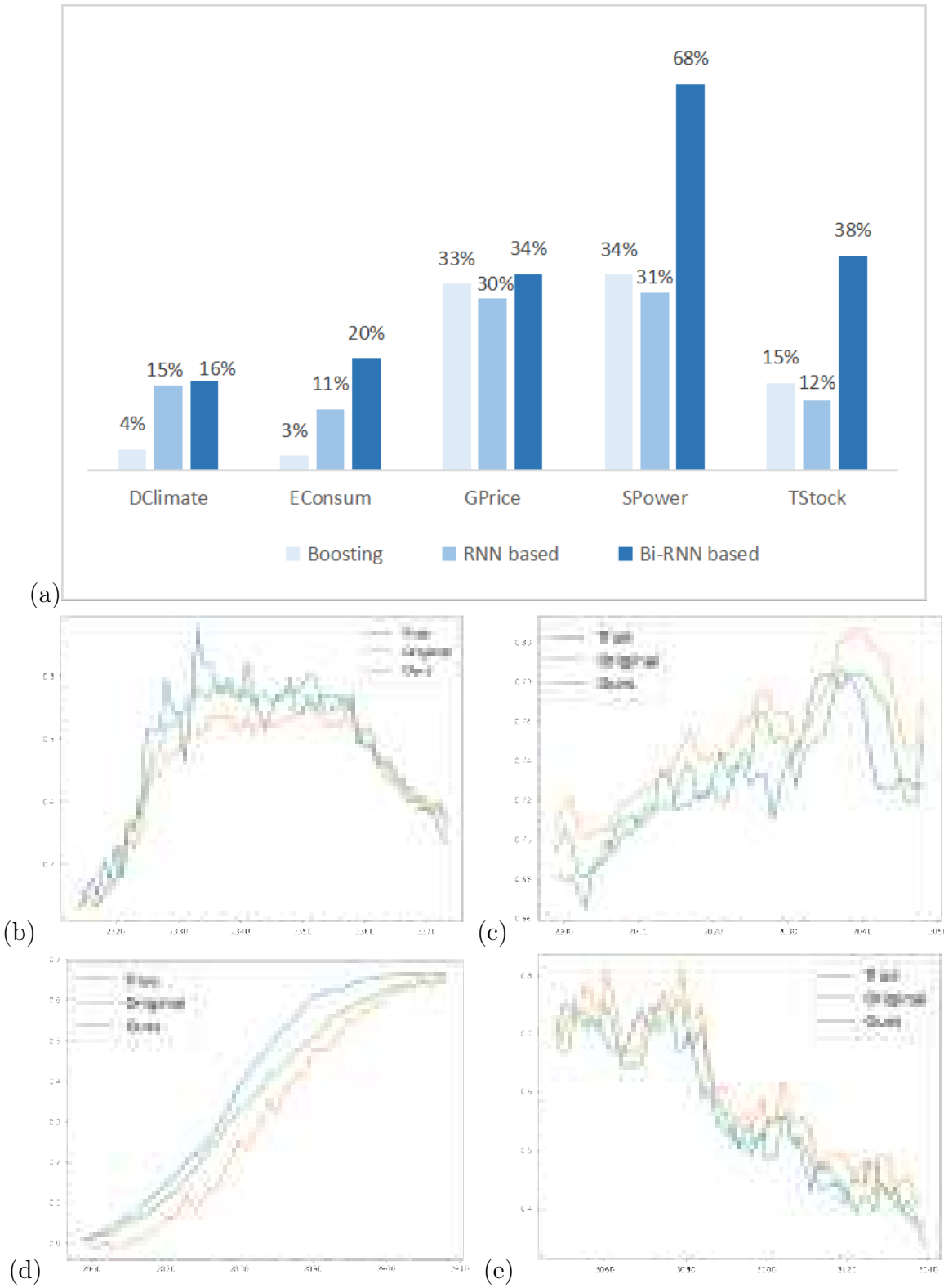


Figure 2.5: Visualization of performance improvement. a: Improvement results of forecasting performance; b: Improve Energy; c: Improve Gold; d: Improve Solar; e: Improve Tesla. "Original" is the output of the original forecasting model, and "Ours" is the output of the ShapTime-guided forecasting model.

Data	Climate	Energy	Gold	Solar	Tesla
XGB	**(0.7898)**	0.7975	**0.8880**	0.9521	**0.8198**
	0.0075	0.0093	**(0.0004)**	0.0038	**0.0021**
LGB	0.7847	**(0.8445)**	0.8715	**0.9573**	0.8038
	0.0082	**(0.0069)**	0.0004	**0.0033**	0.0022
RNN	0.7255	**0.7572**	**0.8535**	0.9701	0.8385
	0.0096	0.0116	0.0006	**0.0024**	0.0020
LSTM	**0.7751**	0.6412	0.8287	0.9597	**0.8812**
	(0.0070)	0.0142	**0.0005**	0.0047	**0.0016**
GRU	0.7470	0.7376	0.7825	**0.9740**	0.8498
	0.0092	**0.0101**	0.0006	0.0026	0.0020
Bi-R	0.7428	0.7425	0.8980	0.9693	0.8932
	0.0082	0.0101	0.0004	0.0036	0.0016
Bi-L	**0.7532**	**0.7604**	0.9002	0.9742	**(0.9080)**
	0.0084	**0.0100**	0.0004	0.0028	**(0.0013)**
Bi-G	0.7412	0.7092	**(0.9043)**	**(0.9840)**	0.8505
	0.0100	0.0112	**0.0004**	**(0.0014)**	0.0021

Table 2.2: ShapTime guided forecasting performance improvement. r^2 and MSE are used as forecasting performance metrics, for each model, the first row is r^2 and the second row is MSE.

datasets, while after the improvement (table 2.5), the best forecasting performance of gold price and solar generation appears in Bi-GRU.

The partial visualization of the improvement effect is shown in figure 2.5 (b,c,d,e). From the forecasting effect, the forecasting results after ShapTime improvement still maintain approximately the same pattern as the original forecasting results, however, they can be much closer to the original data, thus achieving the performance improvement.

2.3 Conclusion of chapter 2

In this work, the XAI approach specifically oriented to time series forecasting is developed, and we name it ShapTime since its computation is based on shapley value. It enables attribution in the temporal dimension, thus explaining the importance of time itself, which differs from previous works. On the other hand,

with the help of ShapTime explanation, we have been able to achieve the improved performance in time series forecasting. By replacing data in times of low contribution with high ones, performance improvements can be achieved to some extent.

Chapter 3

Explainable AI algorithms for calculating feature importance

The aforementioned research findings indicate that both ensemble learning, represented by boosting, and deep learning, represented by neural networks, are important in the field of time series forecasting, especially for long-term sequence forecasting. However, they are both black-box models that cannot be understood by humans. In this chapter, we focus on the boosting model as a representative of ensemble learning. In the field of time series forecasting, the application of boosting models often requires feature engineering. Therefore, explanations for boosting models can provide effective guidance for feature engineering, leading to performance improvement.

Boosting algorithm (BA) is state-of-the-art in major competitions, especially in the M4 and M5 time series forecasting competitions. However, the use of BA requires tedious feature engineering work with blindness and randomness, which results in a serious waste of time. In this work, we try to guide the initial feature engineering operations in virtue of the explanation results of the SHAP technique, and meanwhile, the traditional feature importance (FI) method is also taken into account. **The results of the research have been successfully published [20]**. The **novelty** of our method is that the combination of the two methods enables the explanation of the results to be more informative implicitly, thus helping to optimise the accuracy of the forecasting.

3.1 Feature engineering based on feature importance

What feature engineering needs to achieve is to enrich the information of the data set, so that the prediction model can learn more "knowledge" and show better performance [115]. Existing feature engineering methods in time series forecasting tasks fall into two categories: exogenous and endogenous. The exogenous scheme is to add features that may affect the time series; the endogenous scheme is to enrich the information of the dataset by extracting the hidden features of the original features. In this work, only the endogenous scheme is discussed.

Feature engineering does not have a fixed execution route. It often requires practitioners to design features based on their professional knowledge, so it is extremely dependent on human experience. This is unreliable and time consuming. To alleviate this problem, a large number of automatic feature engineering methods have been developed, such as [116–120]. These mainstream frameworks use multiple feature groups contained in the original dataset to discover new relevant features, and they all focus more on classification tasks. Even though these methods can be partially used in time series forecasting, they do not reflect the temporal features. In contrast, our feature engineering method based on XAI is specially designed for time series data and can reflect temporal features.

Feature engineering methods for time series forecasting include [121–123]. They essentially add lag features on the basis of the above methods, that is, introduce autoregressive features, so as to ensure that enough time features are involved in training. In addition, feature selection is performed by constructing a large number of features (including autoregressive features) in advance, and calculating the feature importance of the model after forecasting, such as [124–126]. However, this process has a certain degree of blindness and randomness. However, our XAI-based feature engineering method is able to quantitatively calculate the order of the required lags, thereby eliminating this blindness and randomness.

3.1.1 Construction of lagged features

For feature engineering in time series forecasting, the construction of lag features is extremely critical. Firstly, time series forecasting, theoretically, is an

autoregressive task, that is, using its own historical data to predict future data, so the construction of lag features is indispensable for time series forecasting tasks. Secondly, the more lag features are not the better, and too many lag features will cause the decline of the prediction performance. This is because the autoregressive process will cause the accumulation of errors, which means that the more lag features, the more errors will accumulate. Therefore constructing a suitable number of lag features is especially critical in time series forecasting. Automatic feature engineering developed by us will focus on the construction of lag features in time series forecasting.

3.1.2 Disadvantages of existing algorithms for calculating feature importance for feature engineering

The Feature importance in the boosting model is an important part of feature engineering. On the one hand, it has perfect mathematical theoretical knowledge [127]; on the other hand, it is applicable to almost all tree models and is extremely convenient. Feature importance is essentially Information Gain, which is used for feature selection when the decision tree is split, and its calculation is based on Shannon entropy. Features with larger information gain are considered important features. The calculation process of the information gain is defined as:

Expected information (Shannon entropy):

$$Info(X) = -\sum_{i=1}^{n} P(x_i) log_2 P(x_i) \tag{3.1}$$

Information Gain:

$$Gain = Info(X) - \sum_{i=1}^{n} \frac{|X^i|}{X} Info(X^i) \tag{3.2}$$

X is a random variable, P is the probability of all cases.

As mentioned above, in order to construct the right amount of lag features, we must know the importance of the features in advance. In addition to the FI function that comes with the boosting model, XAI is also a way worth trying.

The explanation method achieves the purpose of explanation by constructing a simpler and easier-to-understand model and allowing it to continuously approximate the model that needs to be explained. This kind of method is called Post-Hoc, which is different from Intrinsic, which integrates interpretable

functions into the black-box model. The former rarely relies on the architecture of the black-box model and can be widely used in models that have been trained. Among them, Generalized Additive Models [129], Bayes Rule List [128], and Neural Additive Model [130] belong to Intrinsic, and their operation is closely related to the black box model. SHAP [51] and LIME [52] are two exceedingly popular model-agnostic (Post-Hoc) explanations. In this work, we only consider SHAP due to its complete code repository and rigorous mathematical theory.

The creation of SHAP is based on the Shapley value, which treats each feature as a "player" to build a system where "single-player (single feature)" and "alliance (feature combination)" participate in the "game (black-box model)". SHAP actually attributes the output value to the shapely value of each feature. In other words, it calculates the Shapley value of each feature and based on this, measures the impact of the feature on the final output value. For linear models with independent features, the sum of the contributions of all the features of the sample is equal to the predicted value minus the average predicted value, but this is obviously not true for boosting algorithms. Therefore, for the features of the Boost model, the Shapley value needs to be calculated for all possible feature combinations (including different orders) and then weighted and summed, which is defined as:

$$\phi(val) = \sum_{S \subseteq \{x_1 \cdots x_p\} \setminus \{x_j\}} \frac{|S|!(p-|S|-1)!}{p!} (val(S \cup \{x_j\}) - val(S)) \tag{3.3}$$

where S is a subset of the features used in the model, x is the vector of feature values of the sample to be explained, p is the number of features, and $val(S)$ refers to the model output value under the feature combination S. We can quantitatively construct lag variables based on the explanation results of XAI, enabling automatic feature engineering. The overall process is shown in Figure 3.1.

Actually, since a lot of resources are used in feature engineering when machine learning is performing tasks, the explanation of boosting Algorithm has been considered as early as its development. Feature Importance (FI), as an integrated attribute of boosting Algorithm itself, has been applied to feature selection for a long time. However, as an explanation of boosting Algorithm, FI has many shortcomings that cannot be ignored, including:

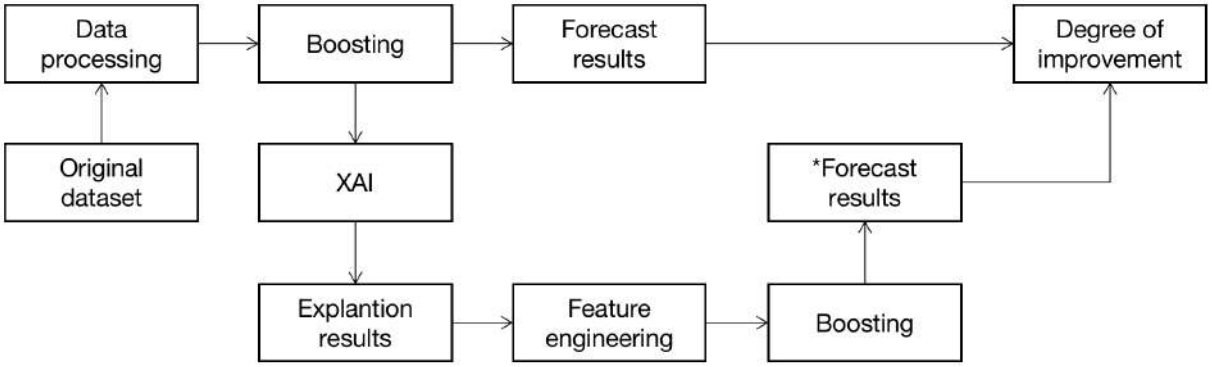


Figure 3.1: The process of feature engineering that XAI uses for time series forecasting. In theory, the XAI part can be replaced with other methods that can output feature importance (contribution). The interpretation result is the feature importance (contribution), which is also the basis for feature engineering processing in our method framework.

- FI cannot reflect whether the influence of features on the forecast results is positive or negative.
- FI cannot reflect the interrelationship between features and target variables. In essence, this means that the interpretation is not ideal

Different from the popular Model-Agnostic Approximations in the Explainable AI field, FI is an attribute of boosting Algorithm itself. Therefore, FI should be paid attention to in the framework of constructing boosting Algorithm explanation, even if it has extremely poor performance for user-oriented explanation.

The popularity of machine learning and deep learning has led to a wider focus on explainable artificial intelligence (XAI). Machine learning models and deep learning models are generally regarded as "black boxes" with internally unknown characteristics. Therefore, when these models are applied, it is very important to gain human's trust, clarify the specific meaning of their errors, and the reliability of their predictions.

3.2 FI-SHAP: explainable AI algorithm with hybrid mechanism for calculating feature importance

3.2.1 Description of hybrid mechanism

A hybrid explanation method combining FI and XAI was constructed, with the purpose of trying to combine feature engineering and Explainable AI to improve the forecast performance of boosting Algorithm in time series forecasting. Especially for energy and other time series that are susceptible to external interference. Taking the two Explainable AI methods mentioned above: SHAP and FI as examples, the new hybrid explanation method is defined as:

$$\varphi_{x_j} = \phi_{x_j} \times \frac{FI(x_j)}{\sum_{i=1}^{p} FI(x_i)} \tag{3.4}$$

φ_{x_j} represents the contribution of j feature to the forecast result under the explanation framework of SHAP.

FI-SHAP combines traditional feature engineering with the emerging Explainable AI. On the one hand, it enhances XAI's specificity for boosting Algorithm, and on the other hand, it provides users with more comprehensive explanation results.

Algorithm 4 FI-SHAP

Input: The feature importance from LightGBM: $FI(x)$; The feature contribution from SHAP ϕ_x

Parameter: The number of feature: p

Output: FI-SHAP value φ_{x_j}

1: **for** i in range (p) **do**
2: $\quad \varphi_{x_j} = \phi_{x_j} \times \frac{FI(x_j)}{\sum_{i=1}^{p} FI(x_i)}$
3: **end for**
4: **return** FI-SHAP value φ_{x_j}

3.2.2 Visualisation of feature importance

The energy time series data used in this work comes from Kaggle, which was collected from two solar power plants located in India with a time period of 34 days (every 15 minutes). The data set consists of two parts, the first part is the

power generation data set, which is generated by the inverter, including direct current, alternating current, daily yield, and total yield. The second part is the data collected by the sensors, including temperature and solar radiation. The inverters are named *sourcekey*, and each power plant has 22 inverters, for a total of 44 inverters. These inverters all generate power generation data at the same time, resulting in a dataset with a total of 136,476 rows and 7 columns (within 34 days).

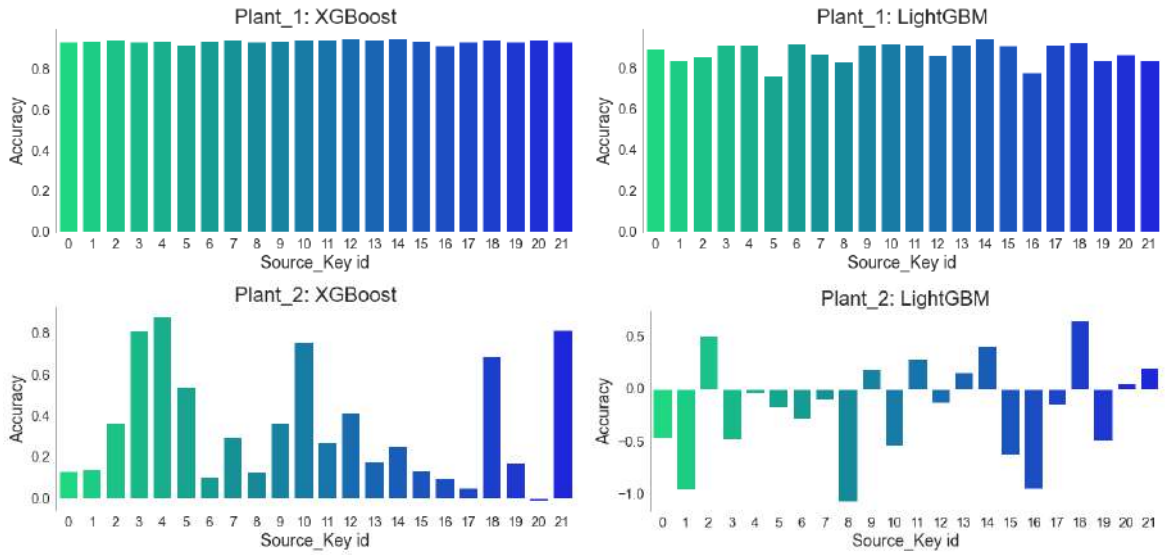


Figure 3.2: Raw Forecasting Results (No Feature Engineering). The results show that the data set of power plant 1 is inherently of high quality, while the data set of power plant 2 contains a large number of anomalies, and even LightGBM almost fails for it.

We adopted a multi-threaded processing scheme, that is, according to different inverter ids (*sourcekey*), the data table is divided into 44 data sets with about 3000 rows and 7 columns (within 34 days). We use both raw boosting models and boosting models with feature engineering on these datasets to highlight the best performing explanation methods.

We use XGBoost with LightGBM to separately make forecasting on 22 small datasets from the solar power plant. The forecasting results are shown in Figure 3.2. It should be noted that LightGBM has significantly outperformed XGBoost in time series competitions in recent years, because the scale of the datasets used in the competition is large. The basis of LightGBM is still XGBoost, and the addition of algorithms such as histogram enables LightGBM to train faster than XGBoost and to ensure accuracy. However, on small datasets, the advantages

of LightGBM disappear accordingly. This is also the reason why LightGBM's performance is poor compared to XGBoost in this work.

From such these data sets, we test the improvement performance of these explanation methods using the power plant 1 dataset and the repair performance of these explanation methods using the power plant 2 dataset.

Compared with Feature Importance, the explanation results (Figure 4) output by SHAP can show both the positive and negative effects of the feature, and can provide users with more comprehensive explanation information. Here, we only show the explanation results of a data set produced by an inverter as an example.

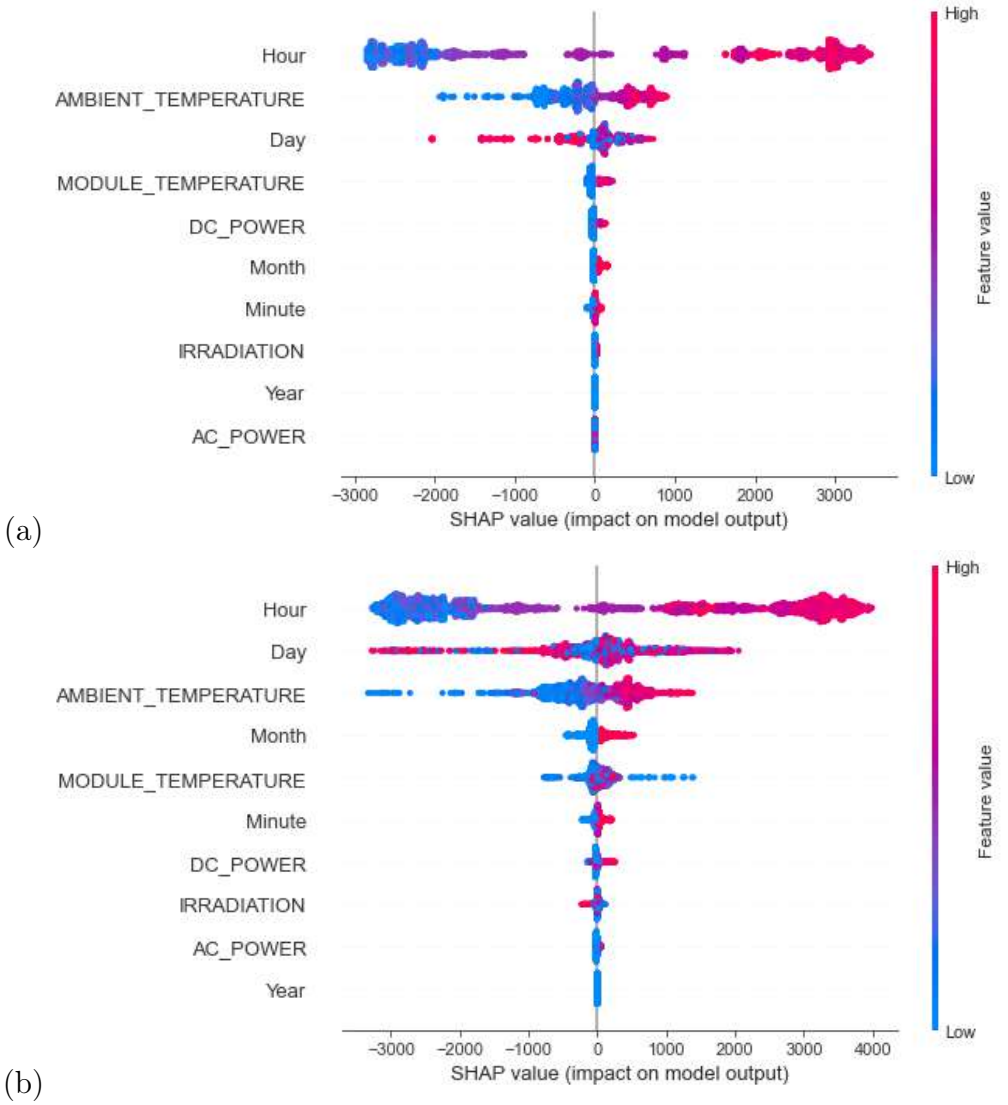


Figure 3.3: SHAP Explanation (Plant 1, Source Key id: 0): a: XGBoost; b: LightGBM

It can be seen from the results that the impact of "Hour" is both positive and negative, and the overall effect is relatively balanced. However, the impact of "AMBIENT TEMPERATURE" on the forecasting results is biased negatively,

that is, within a certain range, the rise in temperature will have a certain positive impact on the power generation, and other features are also explained according to the same logic. For Feature Importance, such an interpretation effect cannot be achieved, and FI can only show the ranking of feature importance, as shown in Figure 3.4.

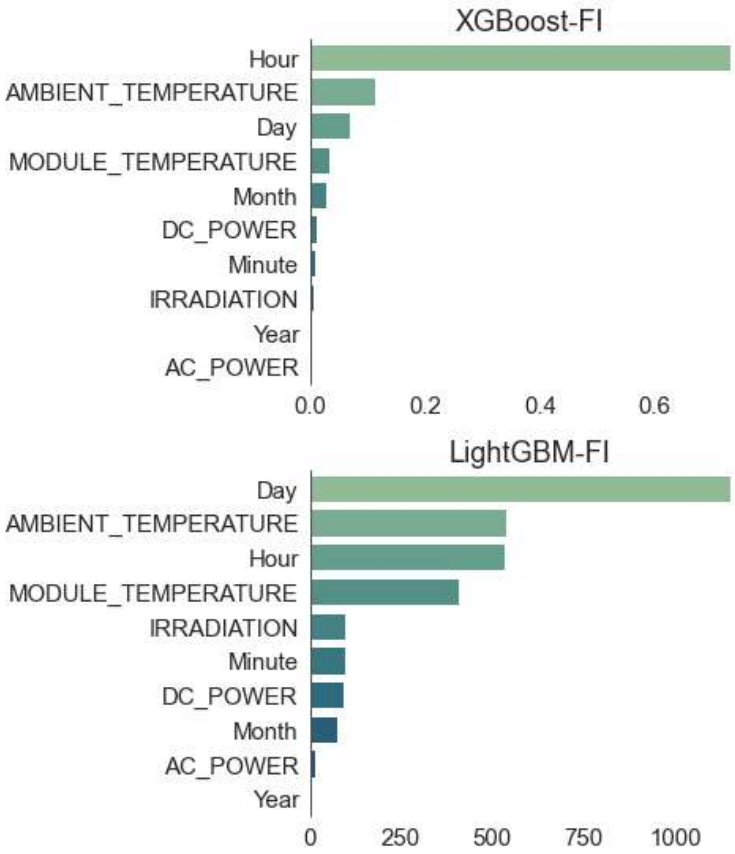


Figure 3.4: Feature Importance (Plant 1, Source Key id: 0)

3.2.3 Improvement of forecasting accuracy using FI-SHAP

As mentioned above, the core of our automatic feature engineering is to construct a suitable amount of lag features to achieve the purpose of feature engineering improvement. Initially, we used the features of the original data to forecast directly using the boosting model, i.e. without feature engineering. Explain the model after the model is trained to obtain the importance of the features, and calculate the weight of each feature, which is defined by the following formula:

$$Weight = \frac{F_j}{\sum_{j=1}^{n} F_j} \tag{3.5}$$

F_j represents the explanation value of each feature.

For different explanation methods, the representation of F_j is also different. For SHAP, F is $\phi(val)$ (Equ.3.3); for FI, F is Gain (Equ.3.2), and for our FI-SHAP, F is $\varphi(val)$ (Equ.3.4).

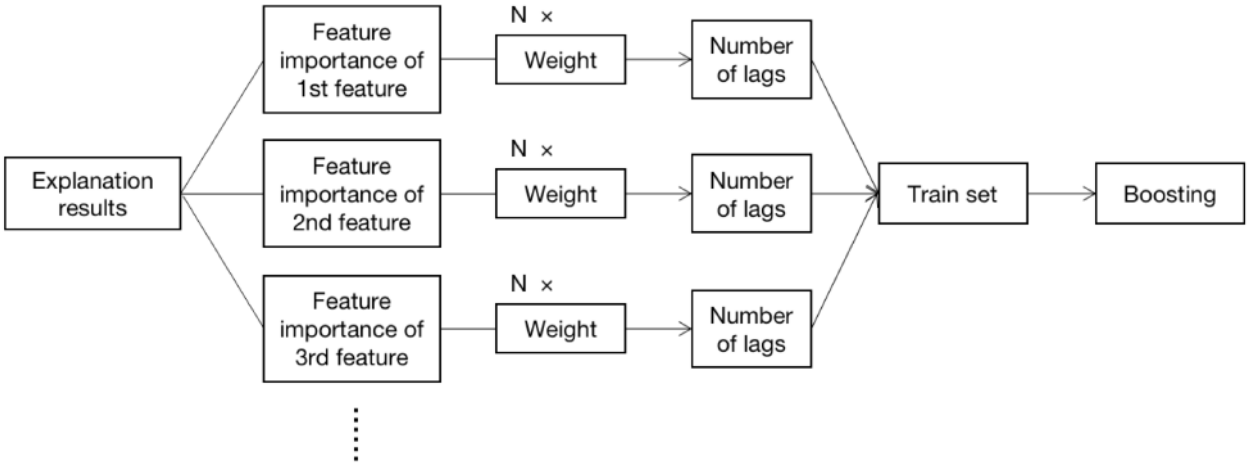


Figure 3.5: The process of feature engineering works by explanation results. N is the total number of features, which can be set by the user according to actual needs.

Calculate the weight of each continuous feature through the explanation results of SHAP and Feature Importance (FI) output, and set the total number of features to be constructed, so that lag features can be constructed according to the weight (Figure 3.5). After obtaining the weights, we also need to set the total number N of features to be constructed, and construct lag features for the continuous features in the original dataset according to the result of N*weight. However, in fact, we do not know the exact value of N, so the automatic feature engineering we provide takes an iterative approach to construct lag features. For example, when the user sets N to 50, our framework will start from 1 to 50, and select the best performing result. Therefore, theoretically, if the computing power of the computer allows, the value of N can be set larger. However, based on actual performance, we recommend setting the value of N not to exceed 200. See this code library for details[1].

In order to effectively measure the improvement effect of these explanation methods on feature engineering, we use the "average lag explanation" as the baseline of the experiment. In the "average lag explanation", the explained value

[1]The github library of FI-SHAP

of all features is artificially set to 1 (or any value, as long as the explained value of all features is guaranteed to be equal), that is, the weight of each feature is equal.

Table 3.1: Forecast Quality (R^2) of Power Plants 1

Source key	0	1	2	3	4	5	6	7	8	9	10
XGBoost	0.934	0.940	0.942	0.934	0.938	0.920	0.938	0.941	0.935	0.940	0.941
avg.	0.934	0.941	0.947	0.941	0.939	0.923	0.942	0.946	0.940	0.944	0.944
FI	0.938	0.953	**0.953**	0.942	**0.944**	0.927	0.938	0.951	0.951	0.942	0.944
SHAP	**0.940**	0.952	0.952	**0.944**	**0.944**	0.932	**0.943**	**0.952**	0.950	**0.943**	**0.945**
FI-SHAP	**0.940**	**0.954**	0.951	**0.944**	0.942	**0.933**	0.941	**0.952**	**0.953**	**0.943**	**0.945**
Source key	11	12	13	14	15	16	17	18	19	20	21
XGBoost	0.941	0.945	0.940	0.945	0.936	0.916	0.934	0.942	0.934	0.944	0.932
avg.	0.944	0.948	0.943	0.946	0.941	0.920	0.942	0.944	0.937	0.948	0.941
FI	**0.946**	0.953	0.943	0.948	0.942	0.926	0.938	0.946	0.947	0.953	0.949
SHAP	0.945	0.953	0.943	0.948	0.942	0.934	**0.939**	0.946	0.950	0.953	0.950
FI-SHAP	0.944	**0.954**	0.943	0.948	0.942	**0.935**	**0.939**	0.946	**0.951**	0.953	**0.951**
Source key	0	1	2	3	4	5	6	7	8	9	10
LightGBM	0.892	0.836	0.856	0.914	0.914	0.760	0.919	0.867	0.831	0.913	0.920
avg.	0.911	0.885	0.889	0.930	0.934	0.764	0.933	0.888	0.877	0.932	0.934
FI	0.928	0.889	0.893	0.940	0.936	0.774	**0.936**	0.895	0.879	0.935	0.940
SHAP	0.925	0.889	0.899	**0.944**	**0.939**	0.774	0.935	0.893	0.878	0.941	0.941
FI-SHAP	**0.931**	**0.892**	**0.900**	**0.944**	0.938	**0.859**	0.934	**0.904**	**0.888**	**0.942**	**0.945**
Source key	11	12	13	14	15	16	17	18	19	20	21
LightGBM	0.912	0.857	0.914	0.939	0.909	0.779	0.914	0.923	0.838	0.864	0.838
avg.	0.924	0.890	0.922	0.948	0.929	0.793	0.929	0.935	0.889	0.893	0.882
FI	0.935	0.891	0.936	0.947	0.926	0.827	0.937	0.940	0.881	0.887	0.880
SHAP	0.937	0.894	**0.942**	**0.953**	**0.932**	0.903	0.939	**0.944**	0.883	0.890	0.888
FI-SHAP	**0.939**	**0.904**	0.938	**0.953**	**0.932**	**0.915**	**0.942**	0.943	**0.890**	**0.894**	**0.893**

So the effect it has is that each feature needs to build the same number of lag features. For example, when N = 30 and the original number of features is 5, the "average lag explanation" is to construct 6 lag features for each feature. Shap might assign 20 lag features to the first feature, 15 features to the second, and no lag features to features with low explanatory values. Finally, compare the

effect of these methods on the improvement of forecasting performance, which also represents the improvement effect of these methods on feature engineering.

Table 3.2: Forecast Quality (R^2) of Power Plants 2

Source key	0	1	2	3	4	5	6	7	8	9	10
XGBoost	0.132	0.141	0.365	0.810	0.878	0.535	0.102	0.297	0.124	0.367	0.755
avg.	0.494	0.159	0.372	0.869	0.904	**0.556**	0.153	0.327	0.174	0.449	0.821
FI	**0.569**	0.209	**0.465**	0.934	0.924	0.544	0.192	0.375	0.227	**0.554**	0.889
SHAP	0.568	**0.309**	0.384	0.946	**0.943**	0.538	**0.216**	0.394	0.362	0.398	0.890
FI-SHAP	0.558	0.252	**0.497**	**0.951**	0.937	0.536	0.212	**0.433**	**0.375**	0.401	**0.896**
Source key	11	12	13	14	15	16	17	18	19	20	21
XGBoost	0.271	0.415	0.176	0.251	0.134	0.093	0.051	0.689	0.173	-0.011	0.813
avg.	**0.318**	0.598	0.321	0.311	0.347	0.281	0.324	0.788	0.216	0.061	0.822
FI	0.301	0.608	0.456	**0.357**	0.449	0.343	0.360	0.938	0.247	0.337	0.826
SHAP	0.303	0.656	0.462	0.316	0.445	0.305	0.352	0.947	**0.258**	0.073	0.827
FI-SHAP	0.302	**0.682**	**0.541**	0.355	**0.465**	**0.471**	**0.368**	**0.948**	**0.249**	**0.296**	**0.828**
Source key	0	1	2	3	4	5	6	7	8	9	10
LightGBM	-0.472	-0.962	0.502	-0.486	-0.045	-0.176	-0.292	-0.107	-1.073	0.188	-0.546
avg.	-0.042	-0.314	0.576	0.654	0.405	0.080	0.004	0.318	-0.586	**0.398**	0.191
FI	**0.278**	-0.061	0.570	0.714	0.884	**0.205**	**0.169**	**0.396**	-0.179	0.367	0.500
SHAP	0.248	**0.206**	**0.597**	**0.770**	**0.903**	0.047	**0.022**	0.324	-0.008	0.309	0.615
FI-SHAP	0.253	0.074	0.585	0.688	0.889	0.135	0.117	0.356	0.022	0.324	**0.670**
Source key	11	12	13	14	15	16	17	18	19	20	21
LightGBM	0.288	-0.134	0.155	0.407	-0.628	-0.955	-0.148	0.653	-0.487	0.049	0.202
avg.	**0.327**	0.110	0.213	0.442	-0.051	-0.210	0.013	0.708	-0.288	0.193	0.258
FI	0.317	0.168	0.177	0.429	0.127	0.012	0.162	0.729	-0.224	0.177	0.325
SHAP	0.307	0.107	**0.503**	**0.477**	0.000	**0.026**	**0.190**	**0.901**	-0.290	**0.224**	0.313
FI-SHAP	0.290	**0.188**	0.403	0.475	**0.142**	-0.025	0.161	0.884	**-0.222**	0.188	**0.345**

We create different kinds of lag features based on different explanation results, thereby enriching feature engineering to improve the performance of forecasting models. The improvement effect of the explanation method is tested by the data set of Power Plant 1, and the repair effect of the explanation method is tested by the data set of Power Plant 2.

The final improvement results are shown in Table 3.1 and Table 3.2, Table 3.1

is the upgrade of Power Plant 1, and Table 3.2 is the repair of Power Plant 2.

According to the results, in the dataset of higher quality Power Plant 1, all explanation methods show improved effect. On the whole, the improvement effect of FI-SHAP is the best, especially in LightGBM. The second is SHAP, and FI has a general effect on improving high-quality data. In the low-quality Power Plant 2 dataset, synthetically, almost all explanation methods have insignificant repair effects. On the one hand, it means that only the construction of autoregressive features is not enough to achieve good performance. On the other hand, the results also show that for small datasets, the adaptability of LightGBM is not as good as that of XGBoost.

The performance improvement brought by more lag features is still not negligible, because when we are doing actual feature engineering, there are still many ways to participate, as described earlier. However, in this work, we only focus on the construction of lag features, in order to study time series forecasting tasks more professionally. In the repair of low-quality data, it can be clearly seen that the repair effect of FI-SHAP is more obvious for XGBoost, and the repair effect of SHAP is more obvious for LightGBM.

3.3 Conclusion of chapter 3

In this chapter, we propose a novel hybrid explanation method, termed FI-SHAP, for the boosting algorithm, which integrates both model-specific and model-agnostic techniques. By leveraging the feature importance scores obtained from the FI-SHAP output, we perform feature engineering on the most influential features to enrich the feature set. Our objective is to enhance the performance of time series forecasting tasks through this feature enrichment process.

Chapter 4

Applications of explainable AI

In this chapter, we apply Explainable AI techniques to real-world tasks, including the analysis of influencing factors, improvement of time series forecasting performance, and online adaptation problem. **Relevant results were published in the paper [21, 22]**. The results of section 4.2 [30] and section 4.3 [31] are under review by the journal. The results of section 4.1 [21, 22] have been published in journals. The **novelty** is that previous works only show explanation results without exploring the application scenarios, our research proposes three economically valuable application scenarios in practice.

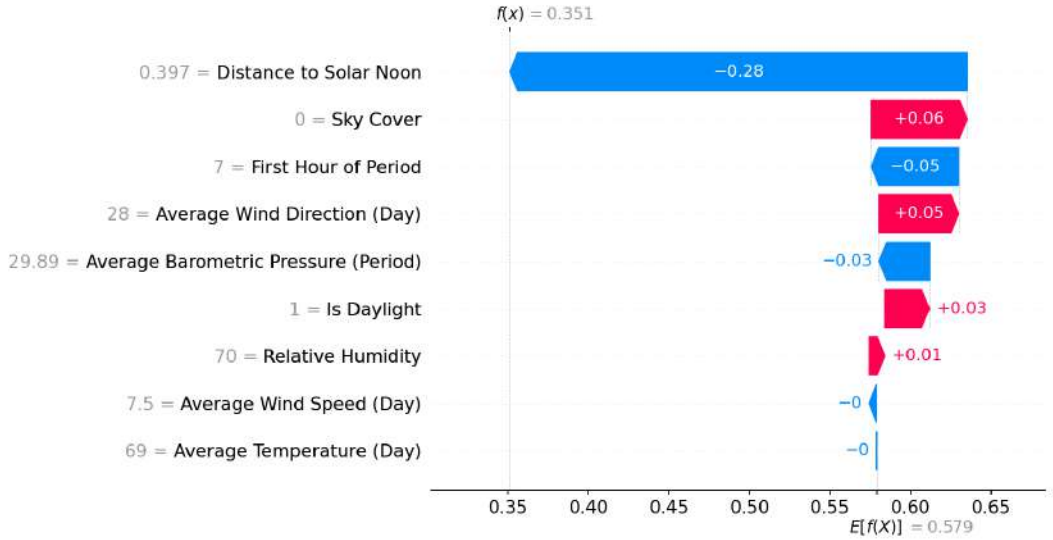
4.1 Analysis of factors affecting solar generation and air quality

The novelty of our method is that information on the importance of features is obtained based on the XAI method and this is used to obtain the location of the solar panel installation

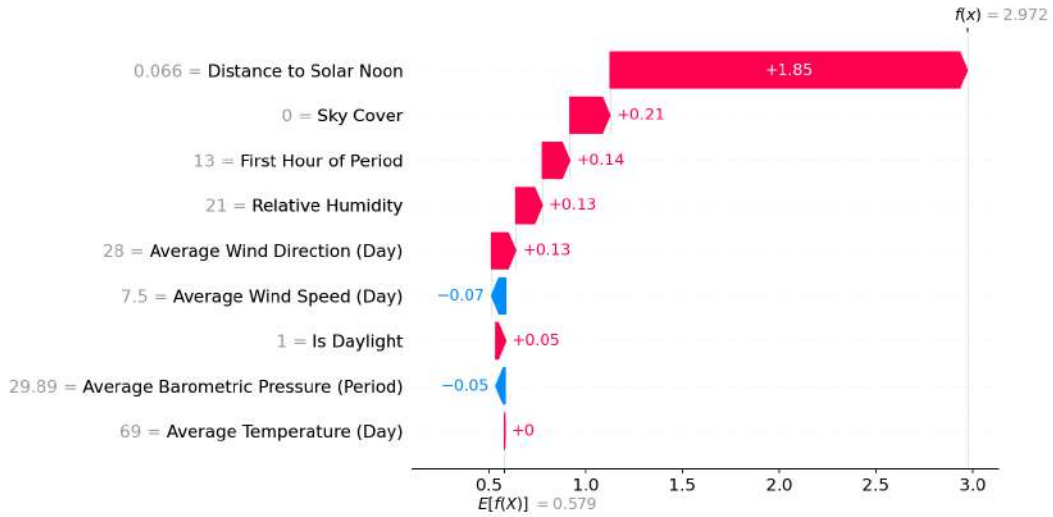
4.1.1 Analysis of factors affecting solar generation

Although more suitable forecasting models were found for the solar power generation dataset through comparative experiments, these models are different from traditional statistical models in that their output logic is not transparent to us. Rather, we only know that nonlinear relationships exist within them. This lack of transparency prevents us from fully relying on their forecast results and limits our ability to analyze the factors that influence these forecasts. Influencing factor analysis is an effective tool for forecasting and managing the

capacity of solar power systems.



(a)



(b)

Figure 4.1: Example of local explanation results. (a): 01 September 2008 7:00; (b): 01 September 2008 13:00. The horizontal axis represents the SHAP values, while the vertical axis displays the names and values of each variable. The numerical value of $f(x)$ represents the forecasting of LightGBM at that moment. In practice, it is not possible to enumerate all the factors that affect solar power generation. Therefore a concept of BASELINE is assumed in SHAP, represented by $E\left[f(x)\right]$, with which those influences that are not taken into account are represented. Here, it is represented by the average of all forecast values.

The identification of key factors that impact system performance, such as weather patterns, geographical location, and shading, can enable more precise forecast of energy generation and optimization of system performance through improved management strategies. Furthermore, historical data and trends derived from influencing factor analysis may support informed decisions regarding investments in solar technology and infrastructure.

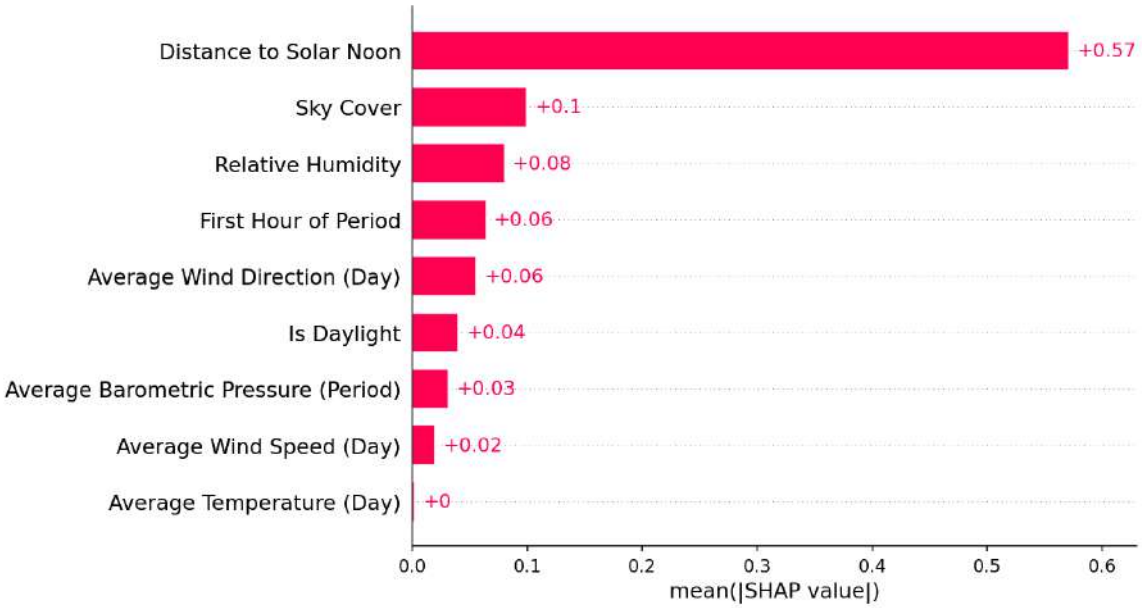


Figure 4.2: Example of global explanation results.

Influencing Factors Analysis

The study employs the SHAP algorithm as a general method for XAI to perform an influence factor analysis. The SHAP algorithm, which calculates Shapley values, is described in detail in Section 2.2. We utilizes LightGBM as an exemplar to showcase the implementation of SHAP. This choice is informed by SHAP's source code library support for ensemble learning and superior visualization capabilities, which surpass those of deep learning. It is important to acknowledge that this disparity in technical capabilities is confined to the code domains only, and SHAP retains its role as an explanation scheme for all black-box models within the theoretical framework.

It is important to emphasise that the SHAP value of a variable represents the degree of its contribution to the forecasting results, and the larger the value indicates that its corresponding variable is more important. We will use Figure 4.1 as an example to show in detail the analysis of influencing factors using SHAP as the basis of the technique.

Figure 4.1 shows the SHAP explanations for two sample points with timestamps of "September 1, 2008, 7:00" and "September 1, 2008, 13:00". The baseline, denoted as $E[f(x)]$, represents unconsidered variables and is substituted with the average value of all forecasting values. In this study, $E[f(x)] = 0.579$. For Figure 11(a), starting from the bottom, the model's baseline is 0.579. The inclusion

of average temperature and average wind speed does not cause a change in the forecasting value $f(x)$. However, once relative humidity is included, $f(x)$ begins to vary, and each subsequent variable contributes to the forecasting. The calculation process is as follows: $f(X) = 0.579 - 0 - 0 + 0.01 + 0.03 - 0.03 + 0.05 - 0.05 + 0.06 - 0.28 = 0.351$.

Thus, SHAP assigns the forecasting value $f(x) = 0.351$ to each variable. The same process applies to Figure 11(b). By comparing Figures 11(a) and 11(b), it can be observed that when the time changes from 7:00 to 13:00, most variables' contributions to the predicted value turn positive. Particularly, as the distance to solar noon decreases, its contribution significantly rises to +1.85, exerting a decisive influence. However, these are only the explanations for two sample points. In practice, it is impossible to analyze each data point individually. Therefore, SHAP also provides global explanations for variables by taking the absolute values of local explanation results and averaging them. This average serves as the global explanation, presented in Figure 4.2.

Global explanation can comprehensively assess the importance of these variables. Figure 4.2 illustrates the ranking of variable importance for forecasting outcomes, with variables arranged from top to bottom in descending order based on their importance. For the entire dataset, the "distance to the solar noon" is the most important, with its significance surpassing that of the remaining variables by a large margin. Subsequently, sky cover and relative humidity take precedence, while other variables such as wind direction, wind speed, and average temperature do not stand out in this comprehensive assessment. Up to this point, the aforementioned analysis is solely based on the static explanation of SHAP values, without incorporating variable values. Next, we will explore the SHAP values under variable value changes to achieve dynamic analysis. The global dynamic explanation is presented in Figure 4.3.

The analysis of Figure 4.3 helps us examine the presence of a strong monotonic relationship between SHAP values and variable values. The key aspect of analyzing Figure 4.3 lies in observing the clear distinction between the blue and red regions. For instance, considering the most important variable - distance to the solar noon, when this variable has lower values, it is represented by blue dots located on the right side. This indicates that lower values of this

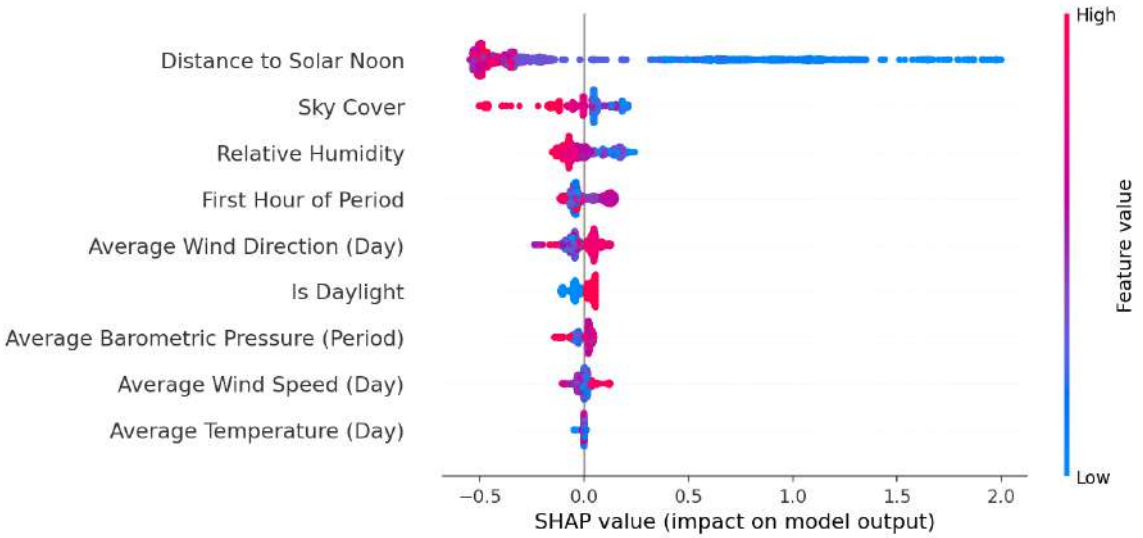


Figure 4.3: Global dynamic explanation. The left vertical axis shows the name of the variable and the right colour bar from blue to red represents the variable's value from small to large. The horizontal axis represents the SHAP value, indicating the importance or contribution of the variable to the forecasting results.

variable have a positive impact on the forecasting solar power generation. Conversely, the red dots representing higher variable values are concentrated on the left side, suggesting that larger values of this variable have a negative effect on the forecasting. The more distinct the separation between the red and blue regions, the stronger the monotonic relationship between variable values and their corresponding SHAP values. In addition, SHAP provides an interactive explanation plot (Figure 4.4, 4.5) which not only presents such monotonic relationships in detail but also illustrates the interaction effects among variables.

We begin by conducting an analysis on continuous variables. As shown in Figure 4.4, the trend exhibited by the data points represents the relationship between variable values and SHAP values, while the color indicates the variable value that has the most significant interaction effect with the given variable. The influential factors based on interactive plots are analyzed as follows:

Distance to Solar Noon (figure(4.4a)). The trend displayed by the data points indicates a strong monotonic relationship, wherein the SHAP value decreases significantly as the variable value increases. This implies a considerable reduction in its contribution to the predicted value. Additionally, when the variable value increases to approximately 0.3, its SHAP value remains

at a relatively low level (around 0.5); however, when the value reaches around 0.4, its contribution becomes negative.

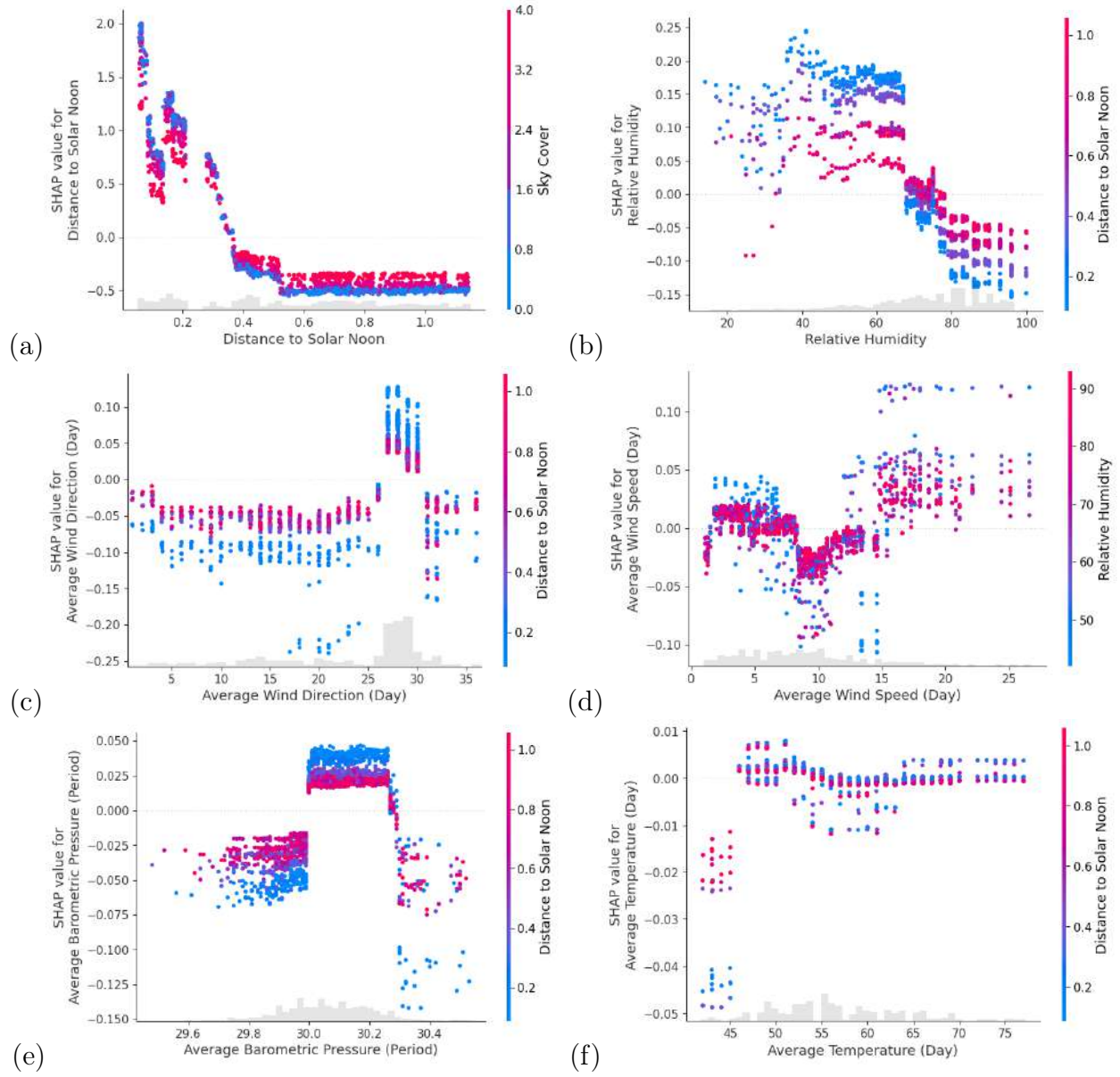


Figure 4.4: Interaction effects of continuous variables on forecasting results. (a): Distance to Solar Noon; (b): Relative Humidity; (c): Average Wind Direction; (d): Average Wind Speed; (e): Average Barometric Pressure; (f): Average Temperature. The grey shading demonstrates the distribution of the corresponding variables

On the other hand, the right side of the plot demonstrates the variable that exhibits a notable interaction effect, which in this case is sky coverage. It can be observed that even when the distance is within 0.3 km, the red data points representing high sky cover can still lower the SHAP value to a lower level. However, once the distance surpasses 0.3 km, due to a significant decrease in

power generation, such interaction effects lose their analytical value. Overall, it can be inferred that when the distance does not exceed 0.3 km and the sky cover remains below 2, the model can generate higher forecasting values. Finally, the results of the SHAP analysis indicate that the distance to the solar noon within 0.3 km, combined with a sky cover rate not exceeding 60%, are the most crucial environmental factors for high-quality solar power generation.

Relative Humidity (figure(4.4b)). As the relative humidity increases, its corresponding SHAP value gradually decreases, indicating a diminishing contribution to the forecasting values. As observed from the figure, when the distance to the solar noon is within 0.3km, relative humidity below 60% exhibits a relatively positive impact on solar power generation.

Average Wind Direction (figure(4.4c)). In regard to the dataset, it can be observed that when the distance to the solar noon is within 0.3km, the average wind direction falls between 25° and 30°, thus favoring solar power generation. However, it should be emphasized that this conclusion is only applicable to the given dataset.

Average Wind Speed (figure(4.4d)). In reference to this data set, when the relative humidity remains below 60% and the average wind speed ranges from 15m/s to 25m/s, a significant increase in corresponding SHAP values is observed. Consequently, this exerts a positive influence on solar power generation.

Average Barometric Pressure (figure(4.4e)). Concerning this dataset, when the distance to the solar noon is within 0.3 km, an average barometric pressure ranging from 30 inHg to 30.2 inHg can generate positive SHAP values, indicating a positive effect on the forecasting value. However, the magnitude of this influence is significantly lower than that of the aforementioned four environmental factors.

Average Temperature (figure(4.4f)). Overall, in comparison with other environmental factors, the average temperature has a relatively minor impact on solar power generation. Based on the results, most data points fluctuate around SHAP=0, indicating zero contribution to solar power generation. The only notable observation is that when the average temperature drops below 45°F, it does exert a negative influence on solar power generation. However, above this temperature threshold, its impact remains limited.

Figure 4.5 illustrates the analysis of discrete variables, with a particular focus on the presentation of "Is Daylight" for the sake of analytical completeness. In practice, when this value is 0, indicating nighttime, the solar power generation decreases to zero. Conversely, when the value is 1, representing daytime, the solar power generation starts to increase.

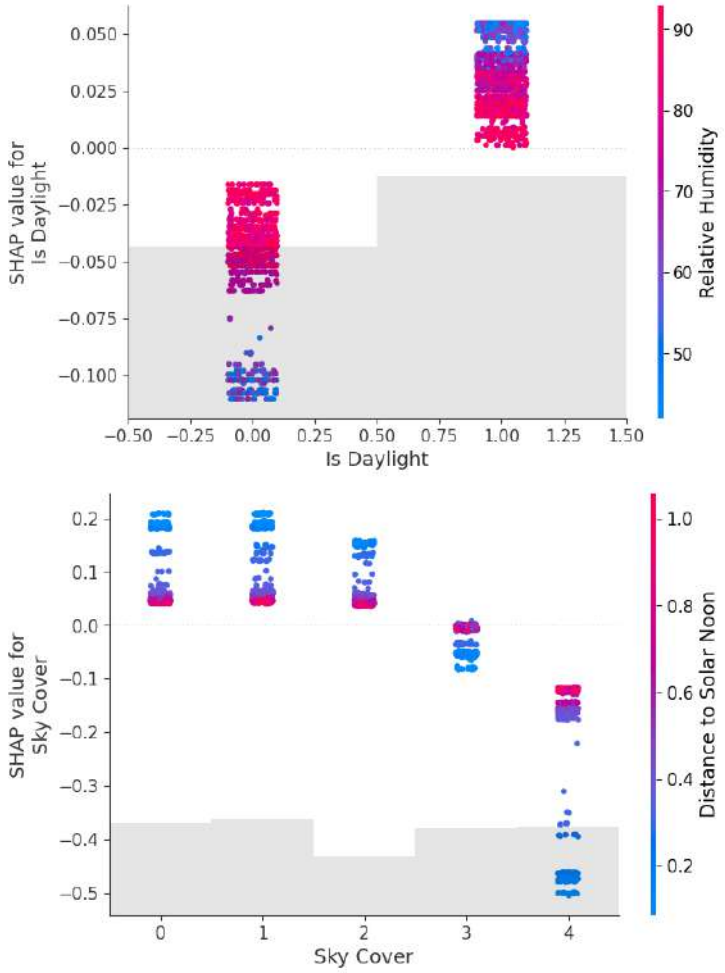


Figure 4.5: Interaction effects of discrete variables on forecasting results

The analysis of sky cover reveals that when the distance to the solar noon is within 0.3km, the sky cover does not exceed 2, corresponding to 60%. This situation positively affects the solar power generation capacity by enhancing its efficiency. However, conversely, it leads to a significant negative impact on solar energy generation.

Application Example

Taking these factors into consideration, we propose two potential sites for the power plant based on the topographic map of Berkeley (Figure 4.6). We have identified Location 1 with an average elevation of 340m and Location 2 with an

average elevation of 370m. Both locations are recommended as suitable options for the solar power plant site selection.

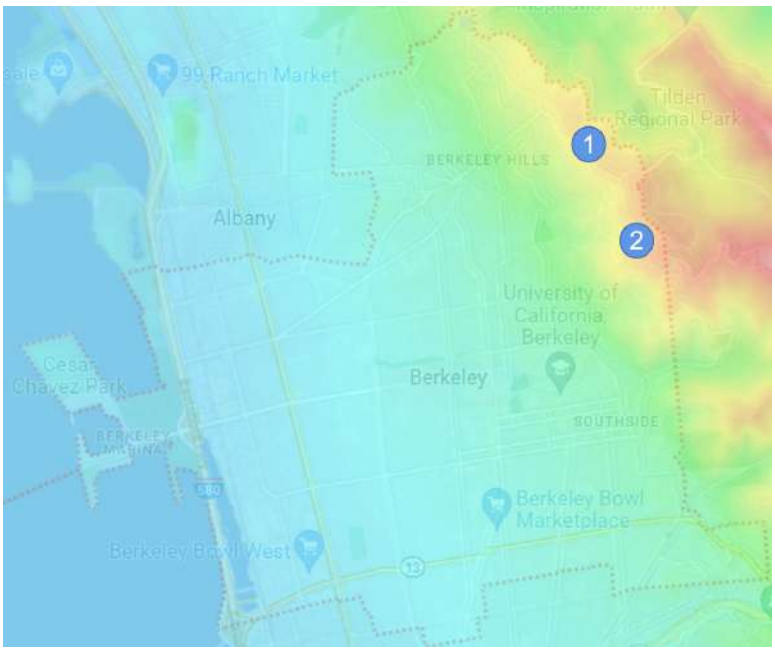


Figure 4.6: Suggested locations for power plant siting.

The following conclusions can be derived from the feature importance output obtained through modeling using LightGBM and explainable artificial intelligence (XAI) techniques represented by SHAP. It is evident that the distance from the sun at noon is the most significant environmental factor, forming the foundation for positive influences of other environmental factors. Specifically, it is crucial to maintain a distance within 0.3 km from the sun at noon in order to maximize solar power generation. Additionally, the following conditions further contribute to an increase in solar energy production: relative humidity below 60%, average wind direction between 25° and 30°, average wind speed ranging from 15 m/s to 25 m/s, average barometric pressure between 30 inHg and 30.2 inHg, average temperature below 45°F, and cloud cover not exceeding 60%.

4.1.2 Analysis of factors affecting air quality

In this study, we selected the ensemble learning model that was best supported by SHAP in order to analyze the factors influencing PM2.5. We focused our comparison within the type of ensemble model. Specifically, we utilized Catboost to examine the factors affecting PM2.5 with a 30-day forecast horizon,

and LightGBM for 90 and 180-day analyses. The SHAP explanation results offered valuable insights into how variables influenced the forecasting outcomes, capturing both individual variable effects and interactions between variables. To visually represent each variable's contribution, we utilized mean value plots, as shown in Figure 4.7.

In this analysis, SHAP's explanation of the three forecasting horizons maintains consistency. It reveals that PM10 has the highest contribution to the PM2.5 results across all horizons, indicating its significant impact. Following PM10, CO also exhibits a higher level of significance in the forecasted results for all horizons. Regarding emission factors like O_3 and SO_2, their impact is considered acceptable. However, among all the emission factors, NO_2 has the least significant effect. Concerning meteorological conditions, the dew point demonstrates the most pronounced effect on PM2.5, with temperature being the next influential factor. In fact, these two factors outweigh all emission factors except PM10 and CO. Pressure and wind speed hold a minor influence on PM2.5 concentrations. Since the analysis focuses solely on the PM2.5 concentration in a specific area, the impact of wind direction is limited. Furthermore, in this dataset, the variable rain representing rainfall amount is generally recorded as 0 in most cases. Consequently, its effect on PM2.5 is minimal, which is emphasized due to Beijing's climatic characteristics.

The analysis presented above focuses on individual variables, but the scatterplot provides additional insights into the relationship between variables and SHAP values. The scatterplot (figure 4.7 upper) shows high variable values depicted as red dots, while low values are represented by blue dots. For instance, with respect to PM10, a high PM10 value corresponds to a high SHAP value on the horizontal axis, indicating a positive effect. In other words, an increase in PM10 (red dots) contributes to the increase in PM2.5, whereas a decrease in PM10 (blue dots) inhibits the increase in PM2.5. Applying the same analytical process, we observed similar patterns for CO, dew point, and ozone. However, the effect of temperature exhibits a different correlation mechanism. An increase in temperature inhibits the increase of PM2.5, whereas a decrease promotes its increase. This phenomenon is likely influenced by the presence of centralised winter heating in the Beijing area. As the primary heating method during winter involves thermal power,

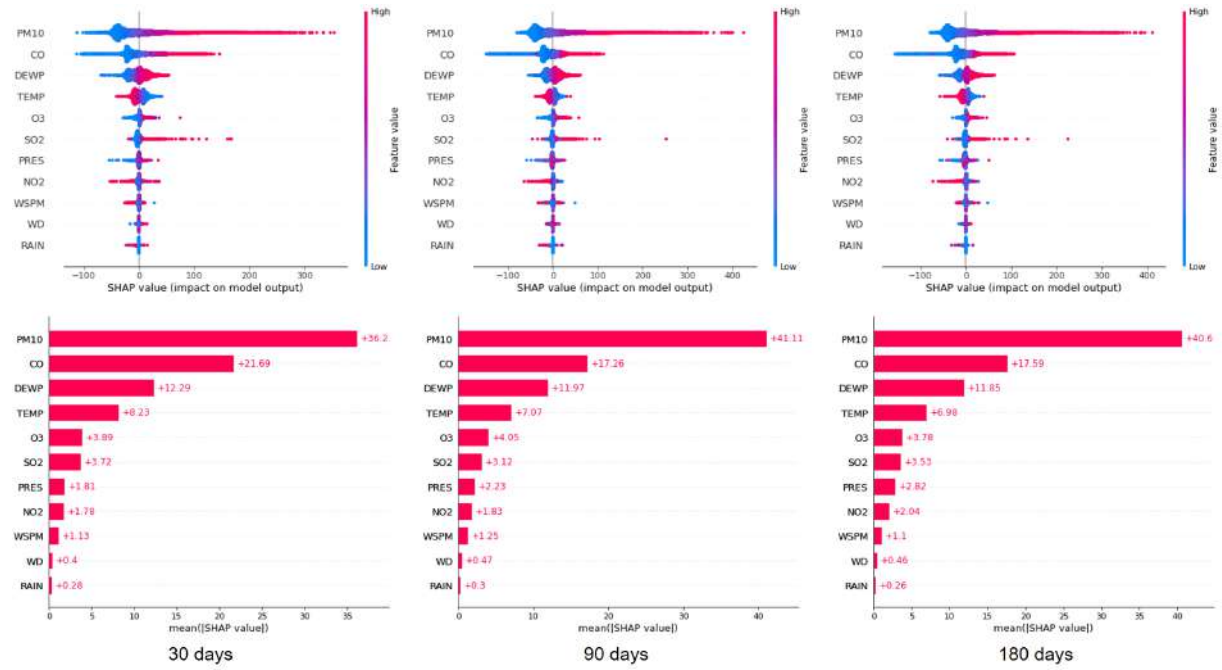


Figure 4.7: Univariate explanation results of SHAP for the Catboost (left: 30 days), for the LightGBM (middle: 90 days, right: 180 days). The scatterplot of the explanation results is shown above, and the mean plot is shown below. For each variable in each observation, SHAP calculates its contribution to the forecasted outcome, all contributions are processed in absolute values and averaged for each variable, which ultimately results in the global contribution value of the variable. Where the SHAP value of each data point is displayed as a point in the scatter plot and the final global contribution of the variable is plotted in the mean plot. In the scatterplot, the horizontal axis represents the SHAP Value, where a larger value represents a larger value of the target variable, and the vertical axis is a ranking of the contribution of all variables to the forecast, where a higher ranked variable represents its greater influence on the final forecast result. The blue and red transitions represent variable values from small to large.

a significant amount of fossil fuels is burned at lower temperatures, leading to higher concentrations of PM2.5.

Interaction analysis of factors

In light of the significance of the interaction explanation plot presentation, we have specifically chosen to focus our analysis on four variables that demonstrate a substantial influence on the results. These variables comprise two emission factors, namely PM10 and CO, along with two meteorological conditions, namely dew point and temperature. The SHAP interaction explanation plot offers valuable insights, particularly regarding the interplay among variables. It unveils the mechanisms behind variable influences on forecast results, even in cases where there is covariance between them. Figure 4.8 visually illustrates how

two crucial factors, PM10 and CO, impact the PM2.5 concentration.

The relationship between PM10 concentration and its contribution to the forecasting result, as indicated by the SHAP value, demonstrates that an increase in PM10 leads to a corresponding increase in PM2.5. The interaction effect of CO on PM10 is particularly significant. To analyze this interaction effect, we represent CO concentration using colors, with red indicating high CO concentration and blue indicating low CO concentration. Examining the data reveals that at high CO concentrations, there is a linear promotion of PM2.5 with increased levels of PM10 (red dots). Conversely, a decrease in CO concentration significantly inhibits the contribution of PM10 to PM2.5 concentrations (blue dots). Similarly, analyzing the CO interaction plots reveals that increasing CO also promotes PM2.5, although not to the same extent as PM10. Furthermore, at high PM10 concentrations, both PM10 and CO contribute to higher PM2.5 levels, but the suppression of this trend by low PM10 levels is not significant (only a few blue dots are observed).

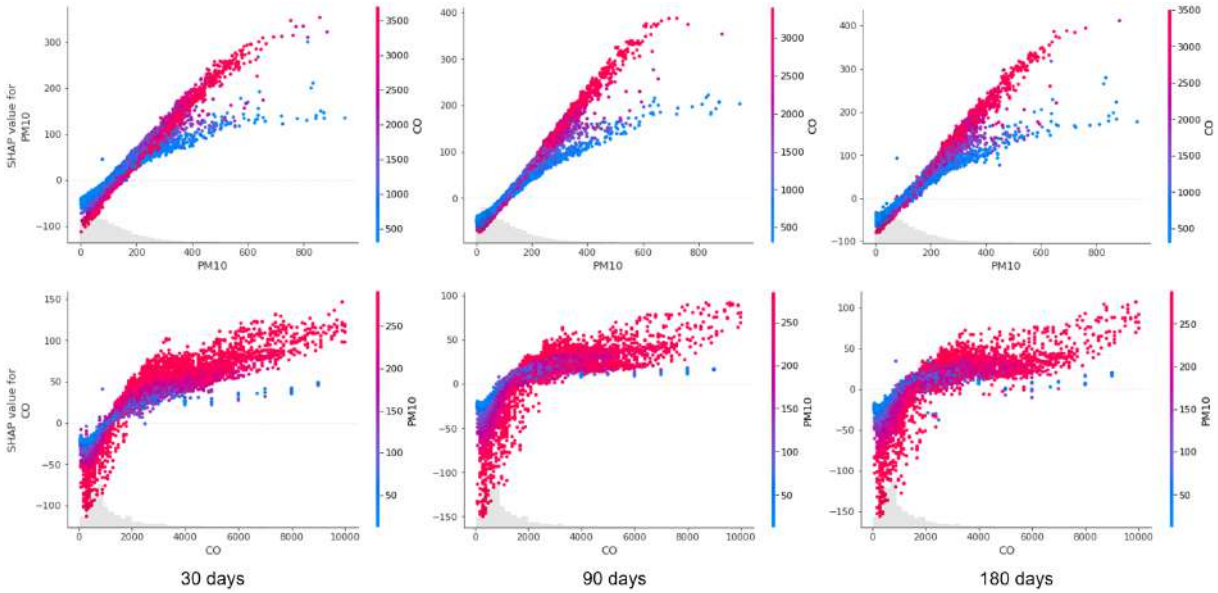


Figure 4.8: SHAP interaction effects plot of PM10 and CO for the Catboost (left: 30 days), for the LightGBM (middle: 90 days, right: 180 days). The horizontal axis represents the value of the variable, the shaded area represents the data distribution for this variables. the left vertical axis represents the SHAP value, and the right vertical axis displays the variable with which the variable has the most obvious interactions, representing the smallest to the largest of its values by transitioning from blue to red.

The SHAP interaction plot provides insights that are not attainable through the

scatter plot and average plot alone. While PM10 ranks highest in importance, this is observed specifically at high CO concentrations. At low CO concentrations, an increase in PM10 concentration tends to stabilize its impact on PM2.5. Therefore, it can be deduced that CO plays a crucial role in driving these effects.

Figure 4.9 illustrates the impact of a single variable on the forecasted outcomes. The variable exhibiting the most pronounced interaction with this variable is displayed on the right vertical axis. The interaction plots provide clearer depictions of the linear associations between changes in these variables and alterations in PM2.5 levels. Specifically, the dew point exhibits a positive correlation with PM2.5, while temperature generally show a negative correlation. On the other hand, for PM2.5 forecasting with a horizon of 30 days, "month" has the most obvious interaction with "dew" and "press", respectively. However, since "month" does not have a linear relationship with the forecast results, it does not show a clear red-blue boundary here either.

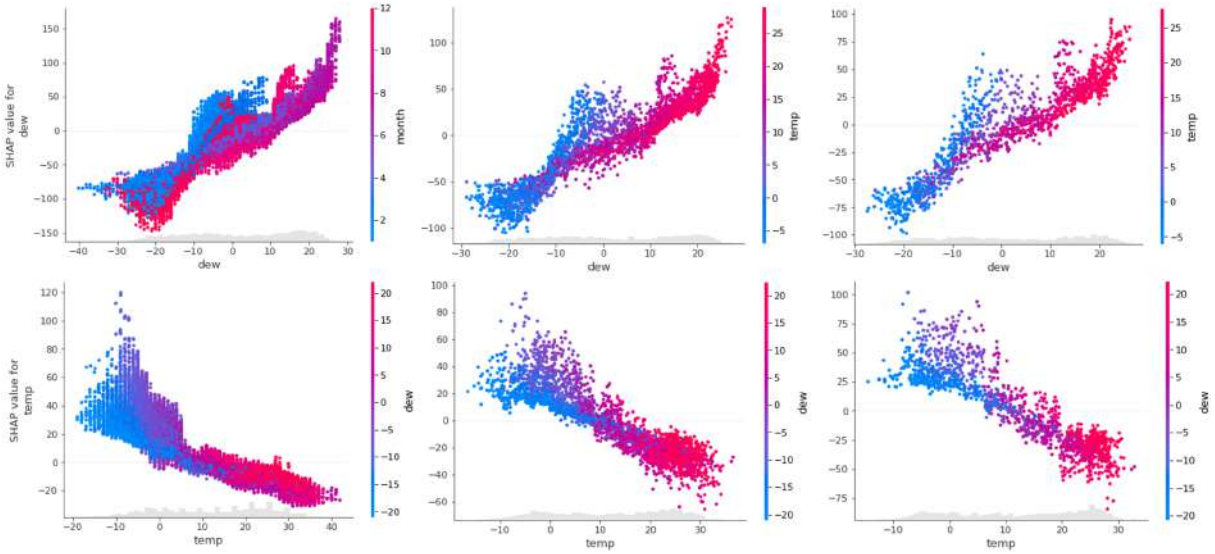


Figure 4.9: SHAP interaction effects plot of dew point and temperature for the Catboost (left: 30 days), for the LightGBM (middle: 90 days, right: 180 days).

For the remaining horizons, there exists a significant interaction between temperature and dew point. It is evident that higher dew points correspond to higher temperatures, indicating a positive correlation between dew point and temperature. This relationship can be further confirmed by examining the "temperature" variable, where high dew points (represented by red dots) are observed in the higher temperature range. Additionally, an increase in dew point

coincides with an increase in the SHAP value, suggesting a rise in PM2.5 concentration. Conversely, high temperatures accompanying elevated dew points lead to a decrease in PM2.5 concentration. However, numerically, when the dew point exceeds 25 degrees Celsius, it may even contribute to a 100 $\mu g/m^3$ increase in PM2.5 concentration, while a corresponding temperature increase only results in a decrease of about 60 $\mu g/m^3$. Therefore, the impact of a high dew point on PM2.5 concentration can be considered reasonably reliable. Consequently, it is crucial to underscore the positive effect of a high dew point on PM2.5 concentrations.

Findings from the analysis

The findings of the analysis indicate that "PM10" has the strongest impact on the prediction of PM2.5, followed by "CO," "dew point," and "temperature." These influential factors exhibit varying degrees of linear association with PM2.5. In particular, temperature displays a negative correlation with PM2.5, whereas PM10, CO, and dew point generally exhibit positive correlations with PM2.5. Additionally, the influence of PM10 on PM2.5 diminishes at lower concentrations of CO, while the influence of CO on PM2.5 appears to be largely unaffected by PM10. Theoretically, these correlated variables can influence PM2.5 concentration mutually. However, numerically, exceeding 25 degrees Celsius in dew point leads to a significant increase of up to 100 $\mu g/m^3$ in PM2.5 concentration, compared to a maximum reduction of 60 $\mu g/m^3$ at high temperatures and 20-40 $\mu g/m^3$.

4.2 Development of automated feature engineering for time series forecasting tasks

This study presents an automated feature engineering framework for time series forecasting that leverages explainable artificial intelligence (XAI). By integrating the XAI module, the framework enhances explainability and directs feature engineering efforts to improve forecasting accuracy. We employ LightGBM (Light Gradient-Boosting Machine) and a hybrid model that incorporates the Exponential Smoothing (ES) algorithm to address trend extrapolation issues inherent in tree-based models. The effectiveness of various XAI methods within

the framework is assessed through performance evaluation. Experimental results indicate that a hybrid XAI method, combining model-agnostic and model-specific approaches, delivers the most substantial performance improvements. However, the consistency of these improvements varies between the hybrid model and the original LightGBM, suggesting limitations in the framework. From an engineering perspective, this work demonstrates that XAI is a highly economical solution for enhancing forecast accuracy in time series data. On the artificial intelligence front, our findings highlight the potential of the hybrid XAI approach as an effective technical strategy, evidenced by the superior results achieved with our developed hybrid method.

4.2.1 Framework for automated feature engineering for time series forecasting

For univariate time series data, we focus on developing two columns of features. The first column represents the timestamp information, while the second column corresponds to the target variable. The automated feature engineering framework proposed in this study is highly applicable to regression models, specifically when applied to time series forecasting tasks, particularly univariate multi-step time series forecasting tasks. The framework, as illustrated in Figure 4.10, comprises three integral components: generating lagged features , generating time feature, and selection of the optimal lag.

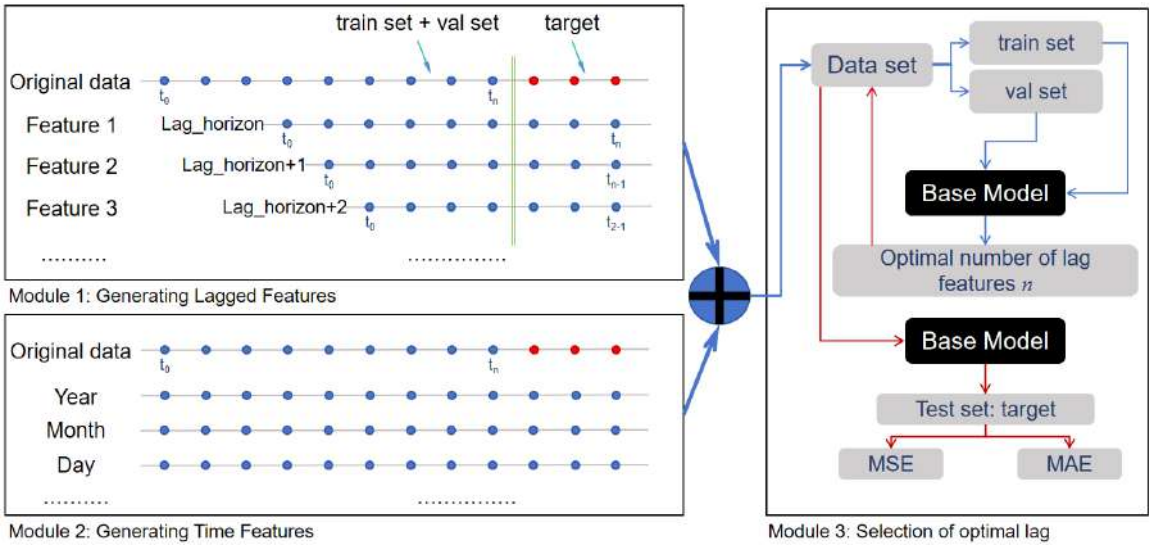


Figure 4.10: The automated feature engineering framework adapted for time series forecasting

Module 1: Generating lagged features

Within this module, only one column, specifically the target variable, is utilized to generate lagged features according to the desired forecasting horizon. As exemplified in Figure 4.10, when the forecasting horizon is set to 3, the initial feature generated is the 3rd order lag feature of the target variable. Subsequently, the 4th order lag feature is produced, followed by the 5th order lag feature, and so forth. Therefore, it becomes essential to define the maximum number of lag features, denoted as N. In this work, we set N equal to 100.

Module 2: Generating time features

This module entails generating relevant time features based on the timestamp information - a widely adopted feature engineering technique in the realm of time series forecasting. For instance, given a timestamp such as '2020- 01-01 15:30', the corresponding time feature would consist of various components: year=2020; month=01; day=01; hour=15; min=30.

Module 3: Selection of optimal lag

The data generated by Module 1 encompasses a total of 100 lag features. In this module, the 100 lagged features extracted from the dataset are incorporated individually into the model during the training phase. Through systematic evaluation on the validation set, the optimal number of lagged features, denoted as n, is determined. Subsequently, the trained model incorporating this optimal configuration is utilized for the forecasting task on the test set.

4.2.2 Automated feature engineering framework

The proposed framework, as illustrated in Figure 4.10, allows for the reasonable application of the regression model to the task of time series forecasting. Subsequent experimental results demonstrate the efficacy of the framework in terms of forecasting performance. However, this effectiveness does not absolve the framework from inherent limitations, namely the generation of redundant features due to the large number of lagged variables. The removal of these redundant features is crucial for improving forecasting performance.

The XAI-guided automatic feature engineering framework, depicted in Figure 4.11, addresses this challenge and comprises three distinct modules:

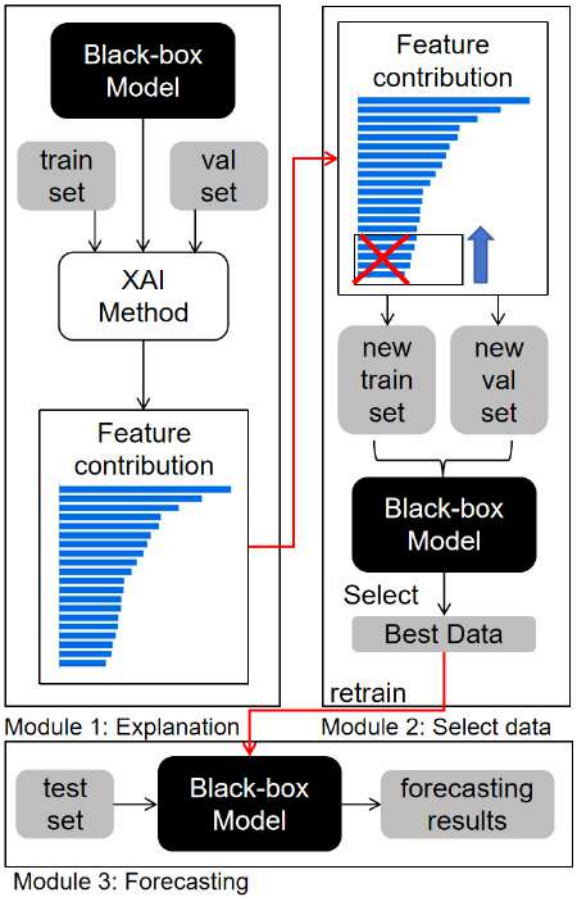


Figure 4.11: Framework for XAI-guided automated feature engineering

Module 1: Explanation

In this module, the XAI method is employed to explain the black-box model and the dataset, generating the contribution or feature importance of each feature. Features with lower contributions are considered redundant.

Module 2: Data selection

After obtaining the feature importance ranking, the determination of the number of features to be removed is necessary. In this module, we commence by iteratively removing features with the lowest feature importance, continuously forming a new dataset. Ultimately, we select the optimal number of features to be removed, resulting in the best dataset.

Module 3: Forecasting

In this module, the list of features from the best dataset is saved and deployed on the test set to obtain the prediction results.

The proposed XAI-guided framework aims to leverage the power of XAI techniques to identify and eliminate redundant features, thereby enhancing the

forecasting performance of the regression model in the context of time series forecasting tasks.

4.2.3 Improvement of forecasting accuracy

Data description

This study utilizes four diverse datasets obtained from various sources to investigate distinct phenomena. The datasets encompass solar power generation, traffic volume, temperature recordings, and egg sales data. A comprehensive description of each dataset is provided below, highlighting their origin, temporal coverage, frequency, and sample size.

Dataset 1 (D1): Solar Power Generation. The solar power generation dataset is sourced from a solar power system in Berkeley, California, USA. It records the energy output from this renewable energy source over a one-year period, from September 1, 2008, to August 31, 2009. The data is collected at three-hour intervals, providing a total of 2,920 samples.

Dataset 2 (D2): Traffic Volume. The traffic volume dataset originates from the westbound traffic flow on interstate highways in Minnesota, USA. It captures the vehicular flow on these major transportation arteries over an extensive period, spanning from October 2, 2012, to September 30, 2018. The data is recorded at hourly intervals, resulting in a substantial sample size of 48,204 observations.

Dataset 3 (D3): Temperature. The temperature dataset is obtained from the Weather Detection Centre in Delhi, India. It consists of daily temperature recordings over a period of approximately four years, from January 1, 2013, to April 24, 2017. The dataset comprises a total of 1,575 samples, providing insights into temperature fluctuations in the region.

Dataset 4 (D4): Egg Sales. The egg sales dataset is derived from a local shop in Sri Lanka, spanning a 30-year period. To ensure the data's relevance and accuracy, the first 5,000 samples, which dated back to 1993, were excluded from the analysis. The modified dataset covers a period from September 9, 2006, to December 31, 2021, with daily frequency, resulting in a total of 5,592 samples.

Comparison of forecasting performance

The feature importance rankings produced by various explanation methods are inconsistent. This inconsistency suggests that the forecasting results generated under the XAI framework will vary depending on the explanation method used, leading to different performance enhancement effects. Consequently, the feature lists generated by each explanation method are initially screened using the validation set. Once the optimal feature lists are determined, the corresponding feature engineering's performance enhancement effect on time series forecasting, guided by different explanation methods, is tested on the test set. The enhancement results for LightGBM in figure 4.12.

Models		LightGBM		FI		FI-SHAP		TreeSHAP		KernelSHAP		Partition		Additive		Permutation	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
D1	28	20.72	0.265	20.88	0.252	20.72	0.265	21.19	0.258	20.72	0.265	20.88	0.252	23.87	0.271	20.72	0.265
	56	11.02	0.199	14.24	0.228	12.49	0.209	12.49	0.209	12.49	0.209	12.50	0.210	12.62	0.217	12.49	0.209
	96	15.96	0.236	17.89	0.241	16.22	0.237	15.72	0.231	15.72	0.231	15.66	0.232	18.97	0.248	15.72	0.231
	avg	15.90	0.233	17.67	0.240	16.47	0.237	16.46	0.234	16.31	0.235	16.34	0.236	18.48	0.245	16.31	0.235
D2	48	10.04	0.201	11.34	0.217	9.831	0.198	10.24	0.201	12.15	0.245	9.831	0.198	10.20	0.199	10.03	0.198
	168	8.595	0.212	8.393	0.210	8.088	0.203	8.100	0.206	8.457	0.206	8.984	0.221	8.088	0.203	8.577	0.206
	240	8.908	0.220	8.595	0.223	8.595	0.223	8.595	0.223	8.848	0.221	8.699	0.217	8.916	0.226	8.595	0.223
	avg	9.181	0.211	9.442	0.216	8.838	0.208	8.978	0.210	9.818	0.224	9.171	0.212	9.068	0.209	9.067	0.209
D3	7	1.213	0.882	1.213	0.882	1.213	0.882	1.213	0.882	1.213	0.882	1.213	0.882	1.213	0.882	1.213	0.882
	15	5.583	1.966	5.583	1.966	5.583	1.966	5.583	1.966	5.583	1.966	5.583	1.966	5.583	1.966	5.583	1.966
	30	7.586	2.411	7.586	2.411	7.224	2.282	8.210	2.432	7.224	2.282	8.080	2.413	7.483	2.284	7.224	2.282
	avg	4.794	1.753	4.794	1.753	4.673	1.710	5.002	1.716	4.674	1.760	4.959	1.753	4.760	1.710	4.674	1.710
D4	24	13.29	0.299	10.17	0.259	8.426	0.232	9.119	0.262	8.426	0.232	23.22	0.420	23.33	0.423	8.426	0.232
	48	10.50	0.249	9.402	0.246	9.402	0.246	9.402	0.246	9.402	0.246	9.402	0.246	9.402	0.246	11.12	0.275
	120	9.233	0.258	9.233	0.258	7.431	0.222	9.233	0.258	9.666	0.250	7.431	0.222	9.233	0.258	9.666	0.250
	avg	11.00	0.268	9.601	0.254	8.419	0.233	9.251	0.255	9.164	0.242	13.35	0.296	13.98	0.309	9.737	0.252

Figure 4.12: Comparison results of time series forecasting. The shaded box shows the optimal average performance.

In summary, the use of XAI significantly enhances forecast accuracy across all datasets except Dataset 1. However, the model-specific explanation method, FI, which is incorporated within LightGBM, fails to improve forecast accuracy satisfactorily. This shortcoming arises because the FI method does not account for correlations between features, leading to inaccuracies in its explanatory results.

Although the improvement of FI is not readily apparent, the enhancement of FI-SHAP, which integrates it with TreeSHAP in LightGBM, is particularly significant. This is because the TreeSHAP method does not rely on the independence assumption. Furthermore, the combination of model-specific and

model-agnostic methods offers more effective explanatory information, which can provide valuable guidance for performance improvement. This guidance, based on XAI methods, can effectively eliminate redundant features, fundamentally differing from the mitigation of the trend extrapolation problem. Mitigating the trend extrapolation problem is demonstrated by the upward shifting of LightGBM predictions, meaning the predicted values by LightGBM increase to some extent. Conversely, removing redundant features is unrelated to the shifting of prediction results. Instead, it aims to alleviate the overfitting problem, bringing the prediction results closer to the actual values.

4.3 Handling concept drift in online adaptation problems

In time series forecasting tasks, the solution to concept drift is almost focused on the development of online model that can be updated in real-time. The implementation of these models mostly depends on the update of the training set and model parameters. We propose a new simple adaptation framework based on online model to further solve the concept drift in time series forecasting - the Tracker. In our adaptation framework, features with low feature contribution will be deleted immediately based on the feature contribution ranking obtained from XAI, so as to achieve dynamic improvement. Compared with the previous online models that focus on parameters and training set updates, within our adaptation framework, the dimensions of the training set are also updated in real-time. The experimental results prove that our framework has obvious improvement. The **novelty** of our method is that Based on the XAI methods, features can be dynamically updated in an online forecasting model. Previously, online adaptation could only dynamically update parameters and incorporate new data. As a result, our method is able to better forecast performance. The code for the experiment can be viewed in this link[1].

4.3.1 Concept drift

Concept drift [131] is a thorny problem encountered in time series forecasting tasks, which means that the accuracy of a trained model will gradually decrease

[1]The github library of XAI-adaptation framework

over time until fails. The reason [132] for this is that the distribution of the data stream changes over time, causing the static model that has been trained to fail. Therefore, the popular solution is to use the updated data set and parameters to train the model in real time, so that the forecasting model can capture the information of the data distribution change in time (Figure 4.13). These solutions are all from the perspective of the number of instances of the data set and the model itself, and we are more inclined to try the dimensional perspective. On the basis of these previous solutions, the addition of "dimension change" makes the unimportant features in the original data removed.

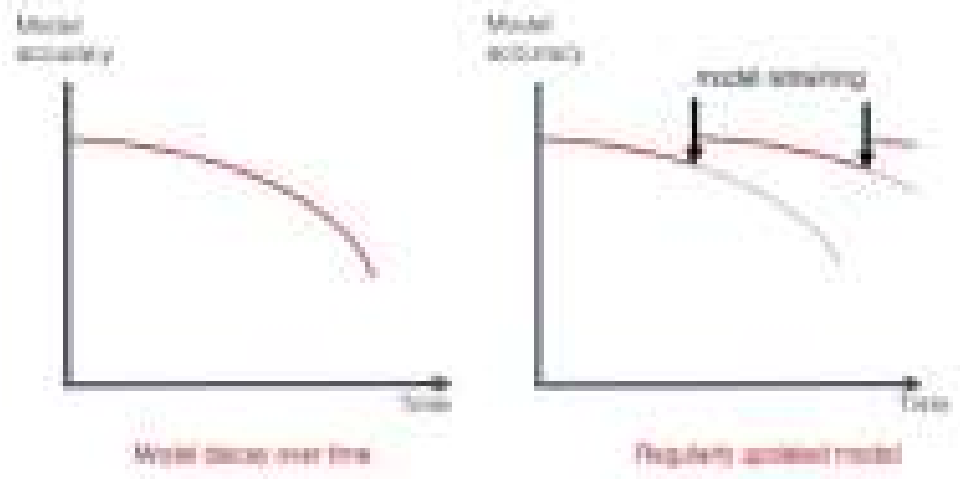


Figure 4.13: Visualization of concept drift and its solutions

Explainable AI [32–34], also called XAI, broadly speaking, is a kind of artificial intelligence designed to let humans understand the working principle of the black-box model through its explanation results. In view of different user identities, the human-computer interaction experience realized by XAI is also different: For users in various professional fields , its method tends to post-hoc explanation, that is, to measure the change of the prediction result through the "perturbation feature" to calculate the contribution of the feature; For model-oriented developers, the method tends to intrinsic, that is, to provide interpretation for the developers in the process of model development, leading to the optimization of the model. In recent years, the latter has gradually developed into Interpretable AI, which pays more attention to developing models that are inherently interpretable.

Considering that the purpose of this work is to construct a universal and generalized explainable adaptation framework, the post-hoc explanation is

adopted by us, which can ensure the application of the framework to other black box models. The simple adaptation framework built by us for concept drift in time series forecasting, the Tracker, not only covers the previous concept drift processing solutions but also introduces new perspectives, dimensional changes, to deal with the problem of concept drift, thereby increasing the upper limit of the performance of the forecasting model.

4.3.2 Online adaptation framework

The frame we designed is shown in Figure 4.14. After the explanation of each updated forecasting model is obtained, the low contribution features of the updated data set will be deleted. According to the result of the explanation, the forecasting model is retrained based on the data set whose dimensionality has changed. Finally, we found that the upper limit of the forecasting performance of the updated model was boosted with appropriate parameter adjustments.

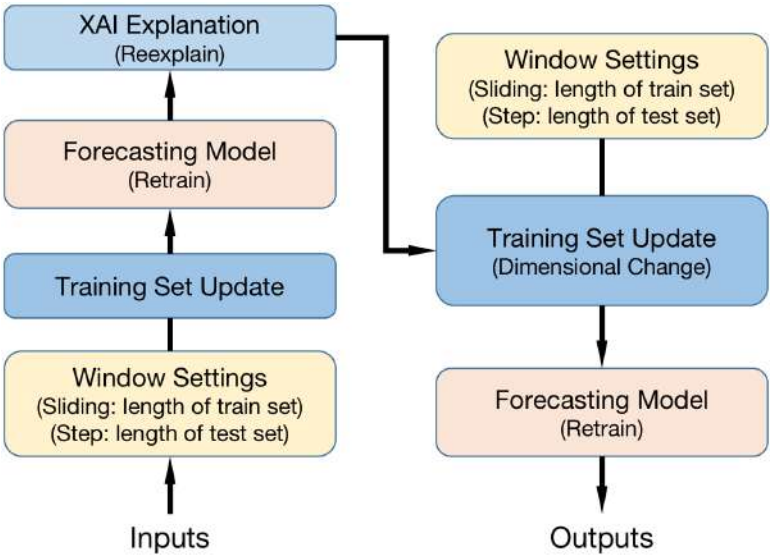


Figure 4.14: The Tracker: XAI-Based Adaptation Framework

Training set update

In our multivariate time series forecasting task, $X = \{x_0, x_1, \cdots, x_m\}$ is the feature matrix, where feature x contains time series instances accompanied by time $T = \{t_0, t_1, \cdots, t_n\}$, namely $x = \{a_{t_0}, a_{t_1}, \cdots, a_{t_n}\}$. $Y = \{y_0, y_1, \cdots, y_n\}$ is the target variable in the task, as shown in Figure 4.15. In order to make our presentation more concise, we stipulate $t = \{a_{0,t}, a_{1,t}, \cdots, y\}$. In actual

tasks, the data distribution of X and Y will change over time. As a result, the information of the data distribution change cannot be effectively captured in the static forecasting model, resulting in a continuous decrease in prediction accuracy.

Since concept drift is caused by changes in data distribution, continuously updating the training set has significant performance in solving concept drift. The training set can be updated by continuously adding newly generated data to the training set (increasing t_n), and adjusting the weight appropriately, that is, reducing the weight of the past time (w_0, w_1, w_1, etc.), and increasing the weight of the recent time (w_n, w_{n-1}, etc.). t is the time information, which is a row of training set at a point in time (Equation 4.1).

$$X_{expansion} = \{w_0 t_0, w_1 t_1, w_2 t_2, \cdots, w_n t_n\} \tag{4.1}$$

It can also be updated through a sliding window. After setting the window length L, the training set is updated through the step size S. i is the number of times the window slides, therefore, the value of i is the number of updates of the forecasting model (Figure 4.15).

The process can be summarized as the equations

$$\begin{aligned} X_{window} &= [i \times S, L + (i \times S)], i = 0, 1, 2, \cdots, k; L > S \\ X_{w0} &= \{t_0, t_1, \cdots, t_l\}, i = 0; l < n \\ X_{w1} &= \{t_s, t_{1+s}, \cdots, t_{l+s}\}, i = 1; l + s < n \\ &\cdots \\ X_{wl} &= \left\{t_{i \times s}, t_{1+(i \times s)}, \cdots, t_{l+(i \times s)}\right\}, l + (i \times s) < n \end{aligned} \tag{4.2}$$

In this work, the sliding window approach is adopted, taking into account its more excellent effectiveness, that is, it can maximize the retention of recent data information in real-time at a small computational cost.

Forecasting model

Since updating the forecasting model requires a lot of repetitive calculations, we are more inclined to the faster forecasting model, that is, the forecasting model that can achieve better accuracy with faster calculation efficiency.

Our previous experimental research results show that LightGBM [19] is a better forecasting model that can take both into account at the same time, in addition,

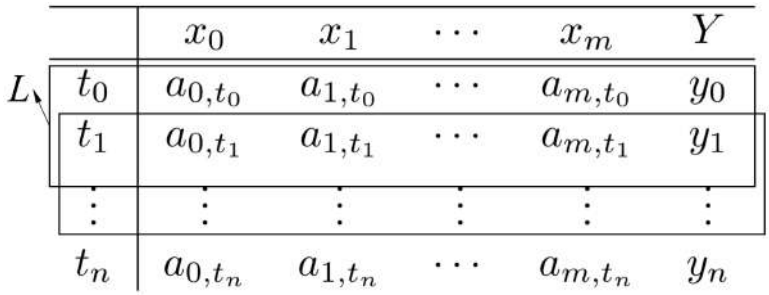


Figure 4.15: The process of updating forecasting. Every time the window of length L moves backward by S steps, the forecasting model will be retrained and the corresponding prediction value matrix will be output.

the title of the M5 prediction competition [8–10] champion also proved its excellent performance.

Therefore, within this framework, LightGBM is selected by us as the forecasting model. In fact, the forecasting model is not fixed, theoretically, it can be replaced with any type of model, as long as it is beneficial to the experimental goal.

The update of the forecasting model depends on the update of the training set, so the number of updates of the forecasting model is also i:

$$f_i(X_{window}) = \left\{\hat{Y_0}, \hat{Y_1}, \hat{Y_2}, \cdots, \hat{Y_k}\right\}, k < n \tag{4.3}$$

Among them, f is the forecasting model and $\hat{\mathrm{Y}}$ is the prediction value matrix.

As shown in Figure 4.14, with the continuous update of the training set, even if the distribution of the data changes, the forecasting model will capture this change in time with the help of the continuously updated training set.

XAI explanation

XAI technology is considered to be used in the explanation of this issue. We try to use the SHAP [51] method to explain the process, and implement further performance improvements on the updated forecasting model based on the explanation results.

The calculation of the SHAP value (V) is based on the Shapley value [57] in a cooperative game (Equation 4.4).

$$V_{x_i} = \sum_{U \subseteq M \setminus \{i\}} \frac{|U|!(M - |U| - 1)!}{M!} \left[F_x(U \cup \{i\}) - F_x(U)\right] \tag{4.4}$$

M is the dimension of all features, U is the dimension of the feature subset, and F is the feature function. The calculation of V, which represents the SHAP value, is the difference between the feature values assigned by the Shapley principle.

We are not satisfied to get the i numbers of predicted value matrix with more accuracy, but want to know more information about the forecasting model, including the feature information in the forecasting process, for example, the contribution of each feature to the forecasting result. In this way, we have a certain degree of understanding of the forecasting model that belongs to the black-box model, that is, the model explanation is obtained, and the explanation result provides us with the possibility to improve the robustness, safety, and performance of the forecasting model.

Explainable AI is divided into local and global. Both are the contribution V of features to the forecasting results. The difference is that the scope of local explanation g is limited to one instance (one row), while the scope of global explanation G is within a period of time (several rows), which is essentially the average weight of local explanations over a period of time.

$$\begin{aligned} G_i &= SHAP(f_i(X_{window})) \\ &= \{V_{x_0}, V_{x_1}, \cdots, V_{x_m}\} \\ g &= \{V_{a_0,t}, V_{a_1,t}, \cdots, V_{a_m,t}\} \end{aligned} \tag{4.5}$$

Therefore, theoretically, for the additive model, once the local explanation g is obtained, the global explanation G will be calculated accordingly.

$$V_{x_z} = \sum_{i=0}^{n} V_{a_z,t_i}, z = [0, m], z \subset N \tag{4.6}$$

In time series tasks, in order to make the terminology more reasonable, the "explanation" below all represents the global explanation G, and the local explanation g we use "real-time explanation" instead. "explanation" is the contribution of a subset of T to the forecasting result, and "real-time explanation" is the contribution of t to the forecasting result.

In the specific XAI methods, SHAP is given priority to us, not only because it has a complete resource library that can be directly called, but more importantly,

this explanation method based on the Shapley value has a mathematical proof foundation of game theory.

Dimensional change

The sliding of the window causes the forecasting model to be updated, and the explanation results are updated accordingly. Based on each explanation result, the original feature matrix is improved, that is, features with low contribution values are removed, and the forecasting model is retrained. Essentially, this real-time dimensional change is a noise reduction operation, thereby improving the efficiency of the data set. The process of dimension change is shown in Algorithm 5.

Algorithm 5 Tracker Algorithm

Input: Total number of features: M; Explainable results: G_i; Window length L: Move step: S

Output: The optimised forecast results: $f_i\left(X_{new}\right)$

for $i = 0$ to k **do**
  $SORT(G_i) : ascending$
  $X_{window} = [i * S, L + (i * S)]\,,\ i = 0, 1, 2, ..., k$
  Set $h = number(h < M)$
  $Interval = [0, h]$
  $Featuredeleted = SORT(G_i).Interval$
  $X_{new} = X_{window}.DROP(Featuredeleted)$
  $f_i\left(X_{new}\right)$
end for
$f_i\left(X_{new}\right)$

Since the removal of dimensions is based on the contribution of the feature, which is the result of the explanations, in the first step, we need to sort the feature contribution result G_i. Here, the ascending order is considered by us. Set a value h $(h < k)$, and construct an interval from 0 to h, and then extract the features of 0 to h sorted from G_i. Finally, the features with low contribution values are removed from the original training set X_{window} to form a new training set X_{new}, and the forecasting model is retrained.

4.3.3 Improvement of forecasting accuracy

In this work, the main process of optimization is to use the explanation results output by XAI to improve the features of the original data. Therefore, theoretically, any technology that can output feature importance (contribution) [113, 114] can achieve the purpose of optimization, for example, for LightGBM (that is, the forecasting model used in this work), its own Feature Importance (FI) function can output feature importance.

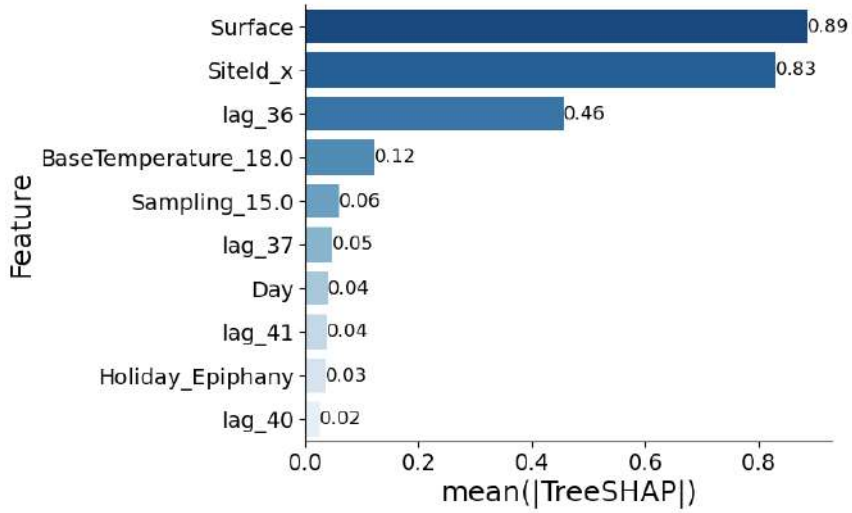


Figure 4.16: Explanation Results.

However, FI has more obvious limitations than the XAI technology we are referring to [103]. For example, it cannot output local explanation, which means that it cannot achieve real-time explanation in time series-related tasks. On the other hand, its explanation result is the global average importance of each feature, so it is not positive or negative, which means that users cannot use it to understand whether the impact of the features on the forecasting model is positive or negative.

In addition, in this work, our goal is to build an explainable and adaptive framework, so generalization is very important for all parts of the framework. This prompts us not to consider the unique explanation methods of a certain model or a certain type of model such as FI. In general, we are more inclined to XAI technology, which can achieve model-agnostic local and global explanations.

We train the static forecasting model by using part of the data extracted from the original data. We are not satisfied with simply getting the results of the forecasting, we hope to get more information about the internal workings of the

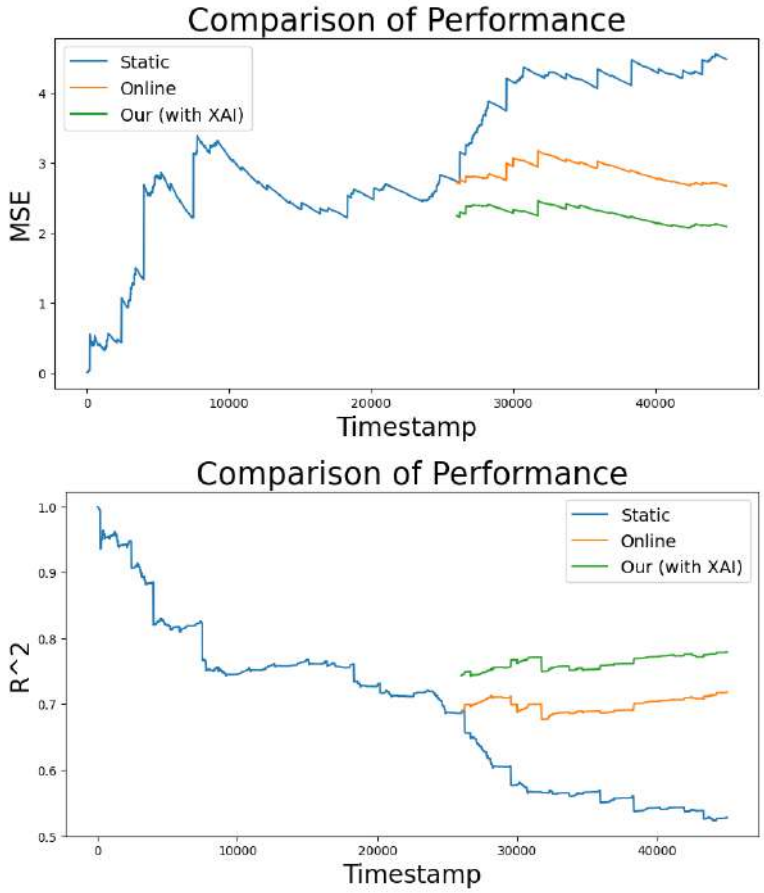


Figure 4.17: Simulation Results

forecasting model. Therefore, we use $SHAP$ as an explanation tool for the forecasting model. In this work, the explanation results are shown in Figure 4.16.

Some obvious features are clearly shown in Figure 4.16, such as Surface, site id, and lag features, which can significantly affect energy demand. The increase in Surface will naturally lead to an increase in energy demand.

From the final result (see Figure 4.17), our adaptation algorithm more or less has an optimization effect on the updated model.

From the perspective of the optimized trend, the trend is consistent with the original updated model. Therefore, this can also indicate that our adaptation framework is an optimization of the updated forecast model. If the updated forecast model does not work well (the performance of the updated model is not as good as the static model), the adaptation framework is also optimized on this basis.

4.4 Conclusion of chapter 4

This section delves into the practical application of eXplainable Artificial Intelligence (XAI) methods and emphasizes their economic value. In order to provide a comprehensive understanding, we conducted an in-depth analysis of numerous influential factors and examined their respective impacts. As a result, we were able to enhance our forecasting accuracy significantly through the integration of XAI techniques. Moreover, our research has made a noteworthy contribution by addressing the challenge of concept drift in online adaptation scenarios. This accomplishment is particularly significant as it paves the way for a solution that effectively tackles this persistent problem. By combining these pivotal elements, our study underscores the immense potential of XAI methods to be implemented profitably and with remarkable efficacy in real-world applications.

By shedding light on the practical implications and economic benefits of XAI, our findings encourage the adoption of these methods to unlock new opportunities for businesses and industries. Not only can XAI improve the accuracy and reliability of forecasts, but it also provides valuable insights into the decision-making process, offering transparency and interpretability. This not only instills trust among stakeholders but also enables organizations to make informed choices based on understandable and justifiable AI-driven predictions.

Conclusions

In this work, the XAI approach specifically oriented to time series forecasting is developed, and we name it ShapTime since its computation is based on Shapley Value. It enables attribution in the temporal dimension, thus explaining the importance of time itself, which differs from previous works.

With the explanation of ShapTime, we are able to understand the forecasting model to some extent. In trending time series data, all models focus on the most recent data as the most important learning object, while in periodic time series data, this pattern does not obviously exist and different models do not necessarily focus on the same time period for learning.

On the other hand, with the help of ShapTime explanation, we have been able to achieve the improved performance in time series forecasting. By replacing data in times of low contribution with high ones, performance improvements can be achieved to some extent. The improved performance metrics show that the Boosting model and the Bi-RNN-based model are still able to maintain their original advantages in periodic data and in trending data, respectively. In summary, ShapTime showed the most significant improvement for the Bi-RNN-based model, with the average improvement of 35%. In particular, ShapTime showed the most significant improvement for the Bi-GRU model, with the 73.87% improvement in the Solar Generation dataset.

On the other hand, we confirm that the construction of lagged features can improve the performance of forecasting models in time series forecasting tasks, but for lower quality data, lagged features are not enough. Simulation results also show that our constructed automatic lag feature method: FI-SHAP's improvement effect is the most stable in higher quality data. In lower quality data, FI-SHAP still has a significant repairing effect on the forecasting of XGBoost. Synthetically, XGBboost outperforms LightGBM for small datasets

and adapts better to poorer quality data. Most of the existing feature engineering methods focus on classification tasks, while feature engineering methods for time series forecasting also pay little attention to lag feature construction. In this work, reasonable lag feature construction is proven to be critical.

Acknowledgments

I would like to express my deepest gratitude to Ovanes Petrosian, my supervisor, for his unwavering guidance, support, and invaluable insights throughout this research endeavor. His expertise and constant encouragement played a pivotal role in shaping this thesis. I am also indebted to the members of my thesis committee, for their time and valuable feedback. Their expertise and diverse perspectives greatly enhanced the quality of this work.

Special thanks go to Shixiang Zhao, Dongfang Qi, Jing Liu, Ruimin Ma, Chi Zhao, Qiushi Sun, Jinying Zou, Feiran Xu for their assistance in various stages of this project. Their contributions and willingness to share their knowledge were instrumental in overcoming challenges and achieving the objectives of this research. I would like to extend my gratitude to for China Scholarship Council providing the necessary resources, facilities, and funding that made this research possible.

I am deeply grateful to my Mom and dear Miss Mae for their unconditional love, understanding, and encouragement throughout my academic pursuits. Their unwavering belief in me has been a constant source of strength. Last but not least, I want to acknowledge the countless unnamed individuals who have indirectly contributed to this research through their work, publications, or shared knowledge. Your contributions have been invaluable in shaping my understanding and methodology.

In conclusion, this thesis would not have been possible without the support and contributions of the aforementioned individuals and many others who have played a crucial role in my academic journey. Thank you all from the bottom of my heart.

Bibliography

[1] Janiesch C., Zschech P., Heinrich K. Machine learning and deep learning //Electronic Markets. - 2021. - Vol. 31, No. 3. - P. 685-695.

[2] Moein M. M. et al. Predictive models for concrete properties using machine learning and deep learning approaches: A review //Journal of Building Engineering. - 2023. - Vol. 63, - P. 105444.

[3] Choi R. Y. et al. Introduction to machine learning, neural networks, and deep learning //Translational vision science and technology. - 2020. - Vol. 9, No. 2. - P. 14.

[4] Kaytez F. A hybrid approach based on autoregressive integrated moving average and least-square support vector machine for long-term forecasting of net electricity consumption //Energy. - 2020. - Vol. 197, - P. 117200.

[5] Wu H. et al. Autoformer:Decomposition transformers with autocorrelation for long-term series forecasting //Advances in neural information processing systems. - 2021. - Vol. 34, - P. 22419-22430.

[6] Zhou T. et al. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting //International conference on machine learning. - PMLR, - 2022. - P. 27268-27286.

[7] Wellens A. P., Udenio M., Boute R. N. Transfer learning for hierarchical forecasting: Reducing computational efforts of M5 winning methods //International Journal of Forecasting. - 2022. - Vol. 38, No. 4. - P. 1482-1491.

[8] Makridakis S., Petropoulos F., Spiliotis E. Introduction to the M5 forecasting competition Special Issue //International Journal of Forecasting. - 2022. - Vol. 38, No. 4. - P. 1279.

[9] Makridakis S., Spiliotis E., Assimakopoulos V. The M5 competition: Background, organization, and implementation //International Journal of Forecasting. - 2022. - Vol. 38, No. 4. - P. 1325-1336.

[10] Makridakis S., Petropoulos F., Spiliotis E. The M5 competition: Conclusions //International Journal of Forecasting. - 2022. - Vol. 38, No. 4. - P. 1576-1582.

[11] Makridakis S. et al. The M6 forecasting competition: Bridging the gap between forecasting and investment decisions // arXiv preprint arXiv:2310.13357. - 2023.

[12] Makridakis S. et al. Statistical, machine learning and deep learning forecasting methods: Comparisons and ways forward //Journal of the Operational Research Society. - 2023. - Vol. 74, No. 3. - P. 840-859.

[13] Schraagen J. M. Responsible use of AI in military systems: Prospects and challenges //Ergonomics. - 2023. - Vol. 66, No. 11. - P. 1729.

[14] Zhang Y. L. et al. Application of artificial intelligence in military: From projects view //2020 6th International Conference on Big Data and Information Analytics (BigDIA). - IEEE, 2020. - P. 113-116.

[15] Reddy S. et al. A governance model for the application of AI in health care //Journal of the American Medical Informatics Association. - 2020. - Vol. 27, No. 3. - P. 491-497.

[16] Morley J. et al. The ethics of AI in health care: a mapping review //Social Science and Medicine. - 2020. - Vol. 260, - P. 113172.

[17] Cao L. Ai in finance: challenges, techniques, and opportunities //ACM Computing Surveys (CSUR). - 2022. - Vol. 55, No. 3. - P. 1-38.

[18] Goodell J. W. et al. Artificial intelligence and machine learning in finance: Identifying foundations, themes, and research clusters from bibliometric

analysis //Journal of Behavioral and Experimental Finance. - 2021. - Vol. 32, - P. 100577.

[19] Zhang Y. et al. Comparison and explanation of forecasting algorithms for energy time series //Mathematics. - 2021. - Vol. 9. - No. 21. - P. 2794.

[20] Zhang Y. et al. FI-SHAP: explanation of time series forecasting and improvement of feature engineering based on boosting algorithm //Intelligent Systems Conference. - Cham : Springer International Publishing, - 2022. - P. 745-758.

[21] Zhang, Y., Sun, Q., Liu, J. et al. Long-Term Forecasting of Air Pollution Particulate Matter (PM2. 5) and Analysis of Influencing Factors //Sustainability. - 2023. - Vol. 16, No. 1. - P. 19.

[22] Petrosian.O, and Yuyi Zhang. Solar Power Generation Forecasting in Smart Cities and Explanation Based on Explainable AI. //Smart Cities. - 2024. - Vol. 7, No. 6. -P. 3388-3411.

[23] Zhang Y. et al. XAI evaluation: evaluating black-box model explanations for prediction //2021 II International Conference on Neural Networks and Neurotechnologies (NeuroNT). - IEEE, 2021. - P. 13-16.

[24] Zou J. et al. High-dimensional explainable AI for cancer detection //International Journal of Artificial Intelligence. - 2021. - Vol. 19. - No. 2. - P. 195.

[25] Sun Q. et al. Resource Allocation in Heterogeneous Network with Supervised GNNs //International Conference on Swarm Intelligence. - Cham : Springer Nature Switzerland, 2023. - P. 350-361.

[26] Zhang Y. et al. ShapTime: A General XAI Approach for Explainable Time Series Forecasting //Intelligent Systems Conference. - Cham : Springer Nature Switzerland, 2023. - P. 659-673.

[27] Zhao, S., Petrov, Y. V., Zhang, Y. et al. Modeling of the thermal softening of metals under impact loads and their temperature-time correspondence //International Journal of Engineering Science. - 2024. - Vol. 194, - P. 103969.

[28] Ma, R., Zhang, Y., Liu, J. et al. Prediction of Next App in OS //2022 III International Conference on Neural Networks and Neurotechnologies (NeuroNT). - IEEE, 2022. - P. 28-31.

[29] Ma R, Zhang Y, Liu J, et al. Forecasting and XAI for Applications Usage in OS //Machine Learning and Artificial Intelligence. - IOS Press, 2022. - P. 17-27.

[30] Zhang Y. et al. Automated feature engineering based on explainable artificial intelligence for time series forecasting // Engineering Applications of Artificial Intelligence. - Under review.

[31] Zhang Y. et al. XAI-Based Explainable Adaptation Framework for Handling Concept Drift in Time Series Forecasting //Knowledge-based systems. - Under review.

[32] Dwivedi R. et al. Explainable AI (XAI): Core ideas, techniques, and solutions //ACM Computing Surveys. - 2023. - Vol. 55, No. 9. - P. 1-33.

[33] Saeed W., Omlin C. Explainable AI (XAI): A systematic meta-survey of current challenges and future opportunities //Knowledge-Based Systems. - 2023. - Vol. 263, - P. 110273.

[34] Lundberg S. M. et al. From local explanations to global understanding with explainable AI for trees //Nature machine intelligence. - 2020. - Vol. 2, No. 1. - P. 56-67.

[35] Pan Q., Hu W., Chen N. Two Birds with One Stone: Series Saliency for Accurate and Interpretable Multivariate Time Series Forecasting //IJCAI. - 2021. - P. 2884-2891.

[36] Ozyegen O., Ilic I., Cevik M. Evaluation of interpretability methods for multivariate time series forecasting //Applied Intelligence. - 2022. - P. 1-17.

[37] Jabeur S. B., Mefteh-Wali S., Viviani J. L. Forecasting gold price with the XGBoost algorithm and SHAP interaction values //Annals of Operations Research. - 2024. - Vol. 334. - No. 1. - P. 679-699.

[38] Oreshkin B. N. et al. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting //arXiv preprint arXiv:1905.10437. - 2019.

[39] Wang J. et al. Multilevel wavelet decomposition network for interpretable time series analysis //Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. - 2018. - P. 2437-2446.

[40] Shen Q. et al. Visual interpretation of recurrent neural network on multi-dimensional time-series forecast //2020 IEEE Pacific visualization symposium (PacificVis). - IEEE, 2020. - P. 61-70.

[41] Guo T., Lin T., Antulov-Fantulin N. Exploring interpretable LSTM neural networks over multi-variable data //International conference on machine learning. - PMLR, 2019. - P. 2494-2504.

[42] Lim B. et al. Temporal fusion transformers for interpretable multi-horizon time series forecasting //International Journal of Forecasting. - 2021. - Vol. 37. - No. 4. - P. 1748-1764.

[43] Ding Y. et al. Interpretable spatio-temporal attention LSTM model for flood forecasting //Neurocomputing. - 2020. - Vol. 403. - P. 348-359.

[44] Zhou B. et al. Interpretable temporal attention network for COVID-19 forecasting //Applied soft computing. - 2022. - Vol. 120. - P. 108691.

[45] Schetinin V. et al. Confident interpretation of Bayesian decision tree ensembles for clinical applications //IEEE Transactions on Information Technology in Biomedicine. - 2007. - Vol. 11, No. 3. - P. 312-319.

[46] Speith T. A review of taxonomies of explainable artificial intelligence (XAI) methods //Proceedings of the 2022 ACM conference on fairness, accountability, and transparency. - 2022. - P. 2239-2250.

[47] Dwivedi R. et al. Explainable AI (XAI): Core ideas, techniques, and solutions //ACM Computing Surveys. - 2023. - Vol. 55, No. 9. - P. 1-33.

[48] Arrieta A. B. et al. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI //Information fusion. - 2020. - Vol. 58. - P. 82-115.

[49] Alsahaf A. et al. A framework for feature selection through boosting //Expert Systems with Applications. - 2022. - Vol. 187. - P. 115895.

[50] Upadhyay D. et al. Gradient boosting feature selection with machine learning classifiers for intrusion detection on power grids //IEEE Transactions on Network and Service Management. - 2020. - Vol. 18, No. 9. - P. 1104-1116.

[51] Lundberg S. M., Lee S. I. A unified approach to interpreting model predictions //Advances in neural information processing systems. - 2017. - Vol. 30.

[52] Ribeiro M. T., Singh S., Guestrin C. " Why should i trust you?" Explaining the predictions of any classifier //Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. - 2016. - P. 1135-1144.

[53] Lim B. et al. Temporal fusion transformers for interpretable multi-horizon time series forecasting //International Journal of Forecasting. - 2021. - Vol. 37, No. 4. - P. 1748-1764.

[54] Lin Y., Koprinska I., Rana M. Temporal convolutional attention neural networks for time series forecasting //2021 International joint conference on neural networks (IJCNN). - IEEE, 2021. - P. 1-8.

[55] Lopes P. et al. XAI systems evaluation: A review of human and computer-centred methods //Applied Sciences. - 2022. - Vol. 12, No. 19, - P. 9423.

[56] Schlegel U. et al. Towards a rigorous evaluation of XAI methods on time series //2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW). - IEEE, 2019. - P. 4197-4201.

[57] Lloyd S Shapley. A value for n-person games. In: Contributions to the Theory of Games. - 1953. - Vol. 2, No. 28, - P. 307-317.

[58] Bach S. et al. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation //PloS one. - 2015. - Vol. 10, No. 7, - P. 0130140.

[59] Charnes A. et al. Extremal principle solutions of games in characteristic function form: core, Chebychev and Shapley value generalizations //Econometrics of planning and efficiency. - 1988. - P. 123-133.

[60] Datta A., Sen S., Zick Y. Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems //2016 IEEE symposium on security and privacy (SP). - IEEE, 2016. - PŸ. 598-617.

[61] Lipovetsky S., Conklin M. Analysis of regression in game theory approach //Applied stochastic models in business and industry. - 2001. - Vol. 17, No. 4, - P. 319-330.

[62] Shrikumar A., Greenside P., Kundaje A. Learning important features through propagating activation differences //International conference on machine learning. - PMLR, 2017. - P. 3145-3153.

[63] Shrikumar A. et al. Not just a black box: Learning important features through propagating activation differences //arXiv preprint arXiv:1605.01713. - 2016.

[64] Strumbelj E., Kononenko I. Explaining prediction models and individual predictions with feature contributions //Knowledge and information systems. - 2014. - Vol. 41, - P. 647-665.

[65] Young H. P. Monotonic solutions of cooperative games //International Journal of Game Theory. - 1985. - Vol. 14, No. 4, - P. 65-72.

[66] Jain S., Wallace B. C. Attention is not explanation //arXiv preprint arXiv:1902.10186. - 2019.

[67] Serrano S., Smith N. A. Is attention interpretable? //arXiv preprint arXiv:1906.03731. - 2019.

[68] Wiegreffe S., Pinter Y. Attention is not not explanation //arXiv preprint arXiv:1908.04626. - 2019.

[69] Zhou Z. H., Zhou Z. H. Ensemble learning. - Springer Singapore, 2021. - P. 181-210.

[70] da Silva R. G. et al. A novel decomposition-ensemble learning framework for multi-step ahead wind energy forecasting //Energy. - 2021. - Vol. 216. - P. 119174.

[71] Qiu R. et al. Generalized Extreme Gradient Boosting model for predicting daily global solar radiation for locations without historical data //Energy Conversion and Management. - 2022. - Vol. 258. - P. 115488.

[72] Ribeiro F., Gradvohl A. L. S. Machine learning techniques applied to solar flares forecasting //Astronomy and Computing. - 2021. - Vol. 35. - P. 100468.

[73] Fan J. et al. Predicting daily diffuse horizontal solar radiation in various climatic regions of China using support vector machine and tree-based soft computing models with local and extrinsic climatic data //Journal of Cleaner Production. - 2020. - Vol. 248. - P. 119264.

[74] Cabaneros S M, Calautit J K, Hughes B R. A review of artificial neural network models for ambient air pollution prediction //Environmental Modelling and Software. - 2019. - Vol. 119. - P. 285-304.

[75] Kiranyaz S, Avci O, Abdeljaber O, et al. 1D convolutional neural networks and applications: A survey //Mechanical systems and signal processing. - 2021. - Vol. 151. - P. 107398.

[76] Cossu A, Carta A, Lomonaco V, et al. Continual learning for recurrent neural networks: an empirical evaluation //Neural Networks. - 2021. - Vol. 143. - P. 607-627.

[77] Makridakis S., Spiliotis E., Assimakopoulos V. The M4 Competition: 100,000 time series and 61 forecasting methods //International Journal of Forecasting. - 2020. - Vol. 36. - No. 1. - P. 54-74.

[78] Makridakis S., Spiliotis E., Assimakopoulos V. M5 accuracy competition: Results, findings, and conclusions //International Journal of Forecasting. - 2022. - Vol. 38. - No. 4. - P. 1346-1364.

[79] Al Daoud E. Comparison between XGBoost, LightGBM and CatBoost using a home credit dataset //International Journal of Computer and Information Engineering. - 2019. - Vol. 13. - No. 1. - P. 6-10.

[80] Hong J. et al. An application of XGBoost, LightGBM, CatBoost algorithms on house price appraisal system //Housing Finance Research. - 2020. - Vol. 4. - P. 33-64.

[81] Bae D. J., Kwon B. S., Song K. B. XGBoost-based day-ahead load forecasting algorithm considering behind-the-meter solar PV generation //Energies. - 2021. - Vol. 15. - No. 1. - P. 128.

[82] Aksoy N., Genc I. Predictive models development using gradient boosting based methods for solar power plants //Journal of Computational Science. - 2023. - Vol. 67. - P. 101958.

[83] Pazikadin A. R. et al. Solar irradiance measurement instrumentation and power solar generation forecasting based on Artificial Neural Networks (ANN): A review of five years research trend //Science of The Total Environment. - 2020. - Vol. 715. - P. 136848.

[84] Vu B. H., Chung I. Y. Optimal generation scheduling and operating reserve management for PV generation using RNN-based forecasting models for stand-alone microgrids //Renewable Energy. - 2022. - Vol. 195. - P. 1137-1154.

[85] Neshat M. et al. Short-term solar radiation forecasting using hybrid deep residual learning and gated LSTM recurrent network with differential covariance matrix adaptation evolution strategy //Energy. - 2023. - Vol. 278. - P. 127701.

[86] Liu Y. et al. An attentionвЂђbased categoryвЂђaware GRU model for the next POI recommendation //International Journal of Intelligent Systems. - 2021. - Vol. 36. - No. 7. - P. 3174-3189.

[87] Peng T. et al. An integrated framework of Bi-directional long-short term memory (BiLSTM) based on sine cosine algorithm for hourly solar radiation forecasting //Energy. - 2021. - Vol. 221. - P. 119887.

[88] Alshemali B., Kalita J. Improving the reliability of deep neural networks in NLP: A review //Knowledge-Based Systems. - 2020. - Vol. 191. - P. 105210.

[89] Liang Y. et al. Explaining the black-box model: A survey of local interpretation methods for deep neural networks //Neurocomputing. - 2021. - Vol. 419. - P. 168-182.

[90] Alshawaf M., Poudineh R., Alhajeri N. S. Solar PV in Kuwait: The effect of ambient temperature and sandstorms on output variability and uncertainty //Renewable and sustainable energy reviews. - 2020. - Vol. 134. - P. 110346.

[91] Belhaouas N. et al. A new approach of PV system structure to enhance performance of PV generator under partial shading effect //Journal of Cleaner Production. - 2021. - Vol. 317. - P. 128349.

[92] Tu J. et al. Experimental study on the influence of bionic channel structure and nanofluids on power generation characteristics of waste heat utilisation equipment //Applied Thermal Engineering. - 2022. - Vol. 202. - P. 117893.

[93] Vilone G., Longo L. Notions of explainability and evaluation approaches for explainable artificial intelligence //Information Fusion. - 2021. - Vol. 76. - P. 89-106.

[94] Burkart N., Huber M. F. A survey on the explainability of supervised machine learning //Journal of Artificial Intelligence Research. - 2021. - Vol. 70. - P. 245-317.

[95] Heuillet A., Couthouis F., DΓaz-RodrΓguez N. Explainability in deep reinforcement learning //Knowledge-Based Systems. - 2021. - Vol. 214. - P. 106685.

[96] Vale D., El-Sharif A., Ali M. Explainable artificial intelligence (XAI) post-hoc explainability methods: Risks and limitations in non-discrimination law //AI and Ethics. - 2022. - Vol. 2. - No. 4. - P. 815-826.

[97] Colin J. et al. What i cannot predict, i do not understand: A human-centered evaluation framework for explainability methods //Advances in neural information processing systems. - 2022. - Vol. 35. - P. 2832-2845.

[98] Ferrario A., Loi M. How explainability contributes to trust in AI //Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency. - 2022. - P. 1457-1466.

[99] Shrikumar A. et al. Not just a black box: Learning important features through propagating activation differences //arXiv preprint arXiv:1605.01713. - 2016.

[100] Shrikumar A., Greenside P., Kundaje A. Learning important features through propagating activation differences //International conference on machine learning. - PMLR, 2017. - P. 3145-3153.

[101] Sundararajan M., Taly A., Yan Q. Axiomatic attribution for deep networks //International conference on machine learning. - PMLR, 2017. - P. 3319-3328.

[102] Bach S. et al. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation //PloS one. - 2015. - Vol. 10. - No. 7. - P. e0130140.

[103] Sundararajan M., Najmi A. The many Shapley values for model explanation //International conference on machine learning. - PMLR, 2020. - P. 9269-9278.

[104] Owen A. B., Prieur C. On Shapley value for measuring importance of dependent inputs //SIAM/ASA Journal on Uncertainty Quantification. - 2017. - Vol. 5. - No. 1. - P. 986-1002.

[105] Ghafarian F. et al. Application of extreme gradient boosting and Shapley Additive explanations to predict temperature regimes inside forests from standard open-field meteorological data //Environmental Modelling and Software. - 2022. - Vol. 156. - P. 105466.

[106] Altman N, Krzywinski M. Ensemble methods: bagging and random forests //Nature Methods. - 2017. Vol. 14. - No. 10. - P. 933-935.

[107] Benidis K. et al. Deep learning for time series forecasting: Tutorial and literature survey //ACM Computing Surveys. - 2022. - Vol. 55, No. 6. - P. 1-36.

[108] Mahmoud A., Mohammed A. A survey on deep learning for time-series forecasting //Machine learning and big data analytics paradigms: analysis, applications and challenges. - 2021. - P. 365-392.

[109] Sherstinsky A. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network //Physica D: Nonlinear Phenomena. - 2020. - Vol. 404. - P. 132306.

[110] Hewamalage H., Bergmeir C., Bandara K. Recurrent neural networks for time series forecasting: Current status and future directions //International Journal of Forecasting. - 2021. - Vol. 37. - No. 1. - P. 388-427.

[111] Orvieto A. et al. Resurrecting recurrent neural networks for long sequences //International Conference on Machine Learning. - PMLR, 2023. - P. 26670-26698.

[112] Lewis R. J. An introduction to classification and regression tree (CART) analysis //Annual meeting of the society for academic emergency medicine in San Francisco, California. - San Francisco, CA, USA : Department of Emergency Medicine Harbor-UCLA Medical Center Torrance, 2000. - Vol. 14.

[113] Schapire R. E. et al. A brief introduction to boosting //IJCAI. - 1999. - Vol. 99, No. 999. - P. 1401-1406.

[114] Mayr A. et al. The evolution of boosting algorithms //Methods of information in medicine. - 2014. - Vol. 53, No. 06. - P. 419-427.

[115] Guyon I., Elisseeff A. An introduction to feature extraction //Feature extraction: foundations and applications. - Berlin, Heidelberg : Springer Berlin Heidelberg, 2006. - P. 1-25.

[116] Kanter J. M., Veeramachaneni K. Deep feature synthesis: Towards automating data science endeavors //2015 IEEE international conference on data science and advanced analytics (DSAA). - IEEE, 2015. - P. 1-10.

[117] Katz G., Shin E. C. R., Song D. Explorekit: Automatic feature generation and selection //2016 IEEE 16th International Conference on Data Mining (ICDM). - IEEE, 2016. - P. 979-984.

[118] Kaul A., Maheshwary S., Pudi V. AutolearnвЂ”automated feature generation and selection //2017 IEEE International Conference on data mining (ICDM). - IEEE, 2017. - P. 217-226.

[119] Khurana U. et al. Cognito: Automated feature engineering for supervised learning //2016 IEEE 16th international conference on data mining workshops (ICDMW). - IEEE, 2016. - P. 1304-1307.

[120] Lam H. T. et al. One button machine for automating feature engineering in relational databases //arXiv preprint: 1706.00327. - 2017.

[121] Cerqueira V., Moniz N., Soares C. Vest: Automatic feature engineering for forecasting //Machine Learning. - 2021. - P. 1-23.

[122] Li L. et al. Research on feature engineering for time series data mining //2018 International Conference on Network Infrastructure and Digital Content (IC-NIDC). - IEEE, 2018. - P. 431-435.

[123] Zdravevski E. et al. Robust histogram-based feature engineering of time series data //2015 Federated Conference on Computer Science and Information Systems (FedCSIS). - IEEE, 2015. - P. 381-388.

[124] Selvam S. K., Rajendran C. tofee-tree: au to matic fe ature e ngineering framework for modeling tre nd-cycl e in time series forecasting //Neural Computing and Applications. - 2023. - Vol. 35, No. 16. - P. 11563-11582.

[125] Punmiya R., Choe S. Energy theft detection using gradient boosting theft detector with feature engineering-based preprocessing //IEEE Transactions on Smart Grid. - 2019. - Vol. 10, No. 2. - P. 2326-2329.

[126] Hu Y. et al. Faster clinical time series classification with filter based feature engineering tree boosting methods //Explainable AI in Healthcare and Medicine: Building a Culture of Transparency and Accountability. - 2021. - P. 247-260.

[127] Shannon C. E. A mathematical theory of communication //The Bell system technical journal. - 1948. - Vol. 27, No. 3. - P. 379-423.

[128] Letham B. et al. Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. - 2015. - P. 1350-1371.

[129] Caruana R. et al. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission //Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining. - 2015. - P. 1721-1730.

[130] Agarwal R. et al. Neural additive models: Interpretable machine learning with neural nets //Advances in neural information processing systems. - 2021. - Vol. 34. - P. 4699-4711.

[131] Bayram F., Ahmed B. S., Kassler A. From concept drift to model degradation: An overview on performance-aware drift detectors //Knowledge-Based Systems. - 2022. - Vol. 245. - P. 108632.

[132] Agrahari S., Singh A. K. Concept drift detection in data stream mining: A literature review //Journal of King Saud University-Computer and Information Sciences. - 2022. - Vol. 34. - No. 10. - P. 9523-9540.