Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

Md Amjad Hossain

Explain errors and improve time-series forecasting models using XAI

Master's thesis in Informatics
Supervisor: Odd Erik Gundersen
Co-supervisor: Gleb Sizov, Liyuan Xing
July 2023

NTNU
Norwegian University of
Science and Technology

Md Amjad Hossain

Explain errors and improve time-series forecasting models using XAI

Master's thesis in Informatics
Supervisor: Odd Erik Gundersen
Co-supervisor: Gleb Sizov, Liyuan Xing
July 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Explain errors and improve time-series forecasting models using XAI

Md Amjad Hossain

July 2023

Abstract

Time series forecasting is crucial for financial institutions and industries, requiring diverse models for different seasons and timelines to achieve accurate predictions. However, unexpected errors often occur due to shifts in underlying patterns or external factors. In such cases, explanations of these errors can be valuable to reduce forecasting inaccuracies.

This thesis looks at both the theory and practical sides of Explainable AI (XAI) methods for time series and outlier detection. The theoretical part explores how XAI concepts can be used in time series analysis and improvement. In contrast, the experimental part focuses on generating explanations for both successful and erroneous forecasts and using those to improve model performance. Comparative evaluations of all experimental techniques are also conducted.

The investigation reveals that while various techniques can enhance model performance, most are unsuitable for time series data structures, necessitating modifications. A key hypothesis proposes that modifying training data is more effective in improving model performance than altering internal model structures or introducing value/policy functions. Indeed, the results demonstrate significant performance improvements through data modifications. However, further experiments with other available methods are pending.

The findings of this thesis show the potential of XAI in enhancing time series forecasting accuracy. Moreover, the work suggests the importance of continued research into different XAI techniques and their impact on model performance.

Sammendrag

Tidsserieprognoser er avgjørende for finansinstitusjoner og bransjer, og krever forskjellige modeller for forskjellige årstider og tidslinjer for å oppnå nøyaktige spådommer. Uventede feil oppstår imidlertid ofte på grunn av endringer i underliggende mønstre eller eksterne faktorer. I slike tilfeller kan forklaringer av disse feilene være verdifulle for å redusere prognoseunøyaktigheter.

Denne oppgaven ser på både de teoretiske og praktiske sidene av Explainable AI (XAI) metoder for tidsserier og avvikdeteksjon. Den teoretiske delen utforsker hvordan XAI-konsepter kan brukes i tidsserieanalyse og forbedring. Derimot fokuserer den eksperimentelle delen på å generere forklaringer for både vellykkede og feilaktige prognoser og bruke dem til å forbedre modellytelsen. Sammenlignende evalueringer av alle eksperimentelle teknikker er også utført.

Undersøkelsen avslører at selv om ulike teknikker kan forbedre modellytelsen, er de fleste uegnet for tidsseriedatastrukturer, noe som krever modifikasjoner. En nøkkelhypotese foreslår at modifisering av treningsdata er mer effektivt for å forbedre modellytelse enn å endre interne modellstrukturer eller introdusere verdi-/policyfunksjoner. Faktisk viser resultatene betydelige ytelsesforbedringer gjennom datamodifikasjoner. Imidlertid venter ytterligere eksperimenter med andre tilgjengelige metoder.

Funnene i denne oppgaven viser potensialet til XAI i å forbedre nøyaktigheten av tidsserieprognoser. Dessuten antyder arbeidet viktigheten av fortsatt forskning på forskjellige XAI-teknikker og deres innvirkning på modellens ytelse.

Preface

This thesis is the final work of the master's program in Informatics at the Department of Computer Science at NTNU.

I would like to express my sincere gratitude to my supervisor from NTNU, Associate Professor Odd Erik Gundersen, whose invaluable guidance has been instrumental in shaping and enriching this entire project. Furthermore, I extend my heartfelt appreciation to Aneo AS for granting me the opportunity to delve into this fascinating topic and work with their remarkable dataset. I am deeply grateful for their continuous support, insightful feedback, and unwavering guidance. I would like to extend special thanks to Liyuan Xing for her exceptional guidance throughout the project and to Gleb Sizov for his invaluable feedback, domain expertise, and guidance. Their contributions have played a pivotal role in defining the project's ultimate direction and evaluating the outcomes.

I extend my heartfelt appreciation to Reinhard Leperlier from *Technische Universität Kaiserslautern* for his invaluable contributions to this project. I am immensely grateful for his exceptional insight, innovative ideas, and unwavering motivation regarding the topic. His guidance and expertise have been instrumental in shaping the trajectory of this work, and I am truly thankful for his support.

I am also immensely grateful to my dearest friend, Masruk Ahmed Rudro, whose unwavering support and assistance have been a constant source of strength during numerous late nights and anxiety-ridden days as the thesis deadline drew near.

Contents

List of Figures

List of Tables

Abbreviations

A-ClArC Additive Class Artifact Compensation. 40

ABN Attention Branch Network. 39

AI Artificial Intelligence. 1, 2, 5, 7, 8, 33

ALE Accumulated Local Effect. 25–27

ANN Artificial Neural Networks. 12

ARIMA Autoregressive Integrated Moving Average. 11, 59

ARMA Autoregressive Moving Average. 11

BiLSTM Bidirectional LSTMs. 68

CAGR Compound Annual Growth Rate. 17

CAM Class Activation Mapping. 18, 22, 39

CH Conceptual Hazard. 40

ClArC Class Artifact Compensation. 40

CNN Convolutional Neural Network. 18, 19, 22

CNNs Convolutional Neural Networks. 18, 19, 22, 24, 45, 68

DeepLIFT Deep Learning Important Features. 18, 23, 24

DL Deep Learning. 18

DNNs Deep Neural Networks. 36

FAT Fairness, Accountability, and Transparency. 2

FNNs Feedforward Neural Networks. 24

GAP Global Average Pooling. 22

GPU Graphics Processing Unit. 51

GRAD-CAM Gradient-weighted Class Activation Mapping. 18, 22, 44, 45, 55, 67, 68

GRUs Gated Recurrent Units. 12

ICE Individual Conditional Expectation. 27–29

KernelSHAP Kernel SHapley Additive exPlanations. 31

KNN K-Nearest Neighbors. 69

LIME Local Interpretable Model-agnostic Explanations. 29, 30, 33

LoRE Local Rule-based Explanations. 33

LRP Layer-wise Relevance Propagation. 38

LSTM Long Short-term Memory. 12–14, 45, 51, 54, 59, 67, 68

MAE Mean Absolute Error. 15, 46, 55, 61

MAPE Mean Absolute Percentage Error. 15

ML Machine Learning. 2, 7, 8, 10, 12, 18, 35

MWh Megawatt-hours. 49

NN Neural Network. 68

NNs Neural Networks. 12, 43

P-ClArC Projective Class Artifact Compensation. 40

PAA Piecewise Aggregate Approximation. 33

PDP Partial Dependence Plot. 25, 26, 28, 29, 53, 56, 60, 68

PDS partial least squares. 27

PFI Permutation Feature Importance. 27

PRP Prototypical Relevance Propagation. 38

R2 R squared. 15, 61

ReLU Rectified Linear Unit. 19, 21, 22

RMSE Root Mean Squared Error. 15, 61

RNN Recurrent Neural Network. 12, 13

RNNs Recurrent Neural Networks. 12, 13, 24, 45

RRR Right for the Right Reasons. 40

SAX Symbolic Aggregate Approximation. 33

SHAP SHapley Additive ExPlanations. 8, 31, 67

SpRAy Spectral Relevance Analysis. 40

TimeSHAP Time Series SHapley Additive exPlanations. 31, 32

TreeSHAP Tree SHapley Additive exPlanations. 31

TSF Time Series Forecasting. 11

VGG Very Deep Convolutional Networks. 22

VNN Variational Neural Networks. 68

XAI Explainable Artificial Intelligence. 1–5, 7, 17, 18, 32–35, 37–43, 52, 61, 67–69

XIL Explanatory Interactive Learning. 38

Chapter 1

Introduction

This chapter offers an overview of Explainable Artificial Intelligence (XAI) and time series, along with the motivation behind and a description of the project's challenges. Additionally, the document outlines the project's specific aims and objectives, which are further divided into a series of research inquiries. Furthermore, various research methodologies are employed in order to address these research questions. Subsequently, the contributions of the thesis are deliberated upon, followed by the presentation of the remaining framework of the thesis.

1.1 Background

The utilization of Artificial Intelligence (AI) and machine learning models experience significant annual growth. While the utilization of AI by firms has remained relatively stable at a range of 50 to 60 percent in recent years, and its adoption has experienced a significant increase of over 100 percent since 2017 [1]. According to recent research [2], companies are currently experiencing significant financial gains as a result of their utilization of AI technology. The specific domains in which businesses derive value from AI have evolved over the course of time [3]. In 2018, the sectors of manufacturing and risk were identified by the majority of respondents as the two functions that exhibited the highest levels of perceived value from the implementation of AI. The domains of product and service development, strategy formulation, and corporate finance have emerged as the primary areas exhibiting notable revenue impacts resulting from the implementation of AI. Conversely, supply chain management has been identified as the domain yielding the most significant cost benefits as reported in various studies.[1]. There is a growing need for academics and practitioners to place greater emphasis on the process of constructing models and the subsequent interpretation of their outcomes. This is particularly important as these models are increasingly integrated into organizational practices and everyday work [4, 5]. The significance of this issue lies in the fact that due to the problem of explaining the AI "black box" critical decisions are progressively being automated by different algorithms that are not fully comprehended by individuals [6].

XAI plays a significant role in this context. Explainable AI (XAI) refers to a branch of AI that has been designed to provide understandable explanations regarding its objectives, rationale, and the manner in which it arrives at decisions, using language that is accessible to individuals with average comprehension abilities [7]. In order to enhance trust, XAI assists human users in understanding the underlying logic of Machine Learning (ML) and AI systems [8]. The concept of XAI is often discussed in relation to deep learning and plays a vital role in the Fairness, Accountability, and Transparency (FAT) ML framework. Organizations that aim to cultivate trust prior to implementing AI can derive advantages from explainable AI (XAI). Possible problems such as AI biases can be discerned through a more comprehensive comprehension of an AI model's behavior, facilitated by the utilization of XAI [9]. Although significant progress has been made, particularly in the domain of image recognition, there have also been efforts directed toward text, audio, and tabular data. However, a limited amount of research has been conducted on time-series data [10]. Time series data possess distinct characteristics that distinguish them from other data formats. These data exhibit various patterns, including trends, seasonal fluctuations, irregular cycles, and occasional shifts in level or variability [11]. These patterns are not easily discernible in alternative data formats.

1.2 Problem and Motivation

Forecasting is a widely employed practice across various industries, including but not limited to weather forecasting, climate forecasting, economic forecasting, healthcare forecasting, engineering forecasting, financial forecasting, retail forecasting, business forecasting, environmental studies forecasting, social studies forecasting, and other practical domains. Individuals who possess precise historical data can utilize time series analysis methodologies to examine the data before undertaking modeling, forecasting, and predictive tasks [11]. The importance of predicting in financial institutions comes from the fact that even small mistakes can have big financial effects [12].

This thesis uses electricity demand (consumption) data from Aneo As, which is a large Nordic renewable group with investment power, innovation power, and implementation power. Being a renewable energy company Aneo As has a couple of wind power station that helps the company to produce and contribute energy. 'Statnett' is the company that is responsible for deciding the electricity price. Now, based on the demand, Aneo As decided how much electricity they needed to produce. As they also need to report that to 'Statnett', how much energy they can contribute to the national grid. The national price is established by the competent authorities, taking into consideration the aggregate quantity of power generated and consumed. Aneo As will bear responsibility for mitigating losses in the event of substantial deviations from the projected power production, either by increasing the power supply or providing financial compensation.

Aneo As utilizes a considerable array of machine learning models to effectively forecast and anticipate production outcomes. To assess the efficacy of these models, Aneo As implements monitoring protocols to carry out manual examinations of the forecast. If substantial errors are detected, attempts are undertaken to clarify and manually determine the required adjustments to correct those errors. The aim of this study is to automate the process and generate a succinct explanation for the possible reasons behind inadequate performance, thereby improving the model's performance using these explanations.

The experiment outlined in this thesis was initially devised with the intention of integrating it directly into the system of Aneo As, thereby enhancing its predictive capabilities. However, as the passage of time ensues, it becomes evident that the task at hand is not a straightforward one due to the multitude of external factors that must be considered. These factors encompass the existing system, seasonal models, feedback on predictions, human involvement, and various other elements. Nevertheless, this thesis has the potential to serve as a foundational component for the development of a comprehensive automated system. Such a system would possess the capability to identify substantial errors, generate explanatory information, and provide reports on the potential causes of these errors, all without requiring human intervention. This stands in contrast to the conventional approach of relying on periodic human observers.

1.3 Goals and Research Questions

This section provides an overview of the primary goal of the project, along with the three research questions that needed to be addressed in order to achieve the objective. The research questions delineate the trajectory of the project and establish a systematic approach to attaining the objective. The primary objectives of the study are outlined below.

Goal *During the process of monitoring, come up with the most optimal explanation for the forecast error, and if possible use that information to enhance the model's performance.*

The phrasing of this objective contains certain underlying assumptions that require consideration. There are multiple methodologies available for the generation of explanations; however, only a particular approach allows for the generation of an explanation specifically for the error. Moreover, there are certain methodologies that are currently in the hypothesis stage. Moreover, the application of XAI data to improve the performance of models is currently constrained to its extent. Prior research has predominantly concentrated on improving the models for image and tabular data while allocating limited consideration to the time series model.

This objective can be divided into a number of research questions, each of which must be answered in order to fulfill the objective.

Research question 1 *What is the most recent state-of-the-art in XAI in general and for time series forecasting jobs in particular?*

An analysis of the time series forecasting literature in the context of XAI is required for this research subject. The results of this literature review include a field overview and a discussion of significant techniques that are currently available in this field. The review must describe methods with various approaches so that it is feasible to choose which method or ways are acceptable for the particular use case in order to serve as a foundation for further inquiry. With a discussion of the paths within XAI and in-depth descriptions of significant methodologies, Chapter 3 provides a solution to this research topic.

Research question 2 *What are the best possible ways to generate an explanation for forecasting error?*

In order to evaluate the best possible ways that could possibly also explain the forecasting error, either evaluation criteria or a matrix should be provided in order to take the decision. It could be either qualitative or quantitative. The project's fundamental premise is that could the information provided, be used in the third research question. Moreover, Is the information provided helps the model to improve its performance or not, that's the evaluation criteria. However, there can be other evaluation criteria that can help in this regard.

Research question 3 *What ways we could use the explanation information to improve the model performance?*

One of the primary objectives of this project is to identify strategies that can enhance the performance of the time-series model through the utilization of explanatory information. The crucial considerations encompass the nature of the information that can be derived from the explanation, as well as the qualitative or quantitative nature of said information. What might be a significant limitation if the data is qualitative in nature? The primary objective in this context is to enhance performance while minimizing human intervention.

1.4 Research Methods

The research conducted in this thesis can be divided into two distinct sections. The initial phase entails conducting a comprehensive literature review on the current state-of-the-art XAI methodologies, with a specific focus on those applicable to time series data. Additionally, an examination of the existing methodologies for leveraging XAI data to enhance the efficacy of models. The final component entails conducting experiments on selected methodologies to determine the most optimal solution for fulfilling the requirements.

The literature review for the initial section can encompass either a comprehensive examination of XAI techniques in general or a focused analysis of XAI techniques specifically tailored for time-series data. Both the XAI technique and performance improvements have been explored using the snowballing technique. This technique involves examining the reference lists of selected papers to identify additional relevant articles on the topic. To ascertain the efficacy of the methods and evaluate their outcomes, an analysis of the citations pertaining to the methods was conducted. The initial search was performed using popular search engines such as Google and Google Scholar, as well as the NTNU journal. Different combinations of keywords were used, including "XAI", "Explainable AI", "XAI for TS", "Outlier detection in TS", and "Performance improvement through XAI". In addition, we have utilized various synonym generator applications to generate alternative terms for these words, thereby providing substantial assistance to our research.

The justification for employing the literature review process lies in its ability to facilitate access to a wide range of relevant scholarly articles on the topic, achieved through comprehensive searches, survey articles, references, and citations. Additionally, the process of conducting a comprehensive and well-organized literature review requires a significant amount of time, which would inevitably reduce the available time for the implementation of the project's second phase.

The subsequent phase of this study involves the implementation of experiments in order to assess the findings and validate the hypothesis. In order to assess the efficacy of the methods employed, our primary approach involved the generation of graphical representations and plots, which were subsequently subjected to quantitative verification.

1.5 Contributions

The contribution of this research is divided into multiple components. The initial section of the discussion provides an overview of XAI and its application in the context of time series analysis. Furthermore, this paper provides a comprehensive examination of the methodologies that, when integrated with interpretable data, have the potential to improve the efficacy of the model. In our perspective, the most effective approach involves utilizing the grad-cam technique to track variations in gradients, evaluate the importance of the input dataset, and subsequently modify the training data to improve performance. Potential future research directions have been discussed in Section 8.

1.6 Thesis Structure

The subsequent sections of this thesis are structured in the following manner. The background theory presented in Chapter 2 encompasses various aspects, such as definitions, an exploration of the significance of explanations to human beings, guidelines for evaluation, and a comparison of time series forecasting problems with other data formats. The comprehension of the remaining components of the thesis is contingent upon a comprehensive understanding of the underlying theoretical framework. Chapter 3 will provide an overview of explainable AI, focusing specifically on its application in time series analysis. This chapter will delve into the topic of outlier detection in time series, as well as the provision of explanations for inaccurate forecasts. Chapter 4 provides an overview of the various XAI techniques that can be employed to enhance the model's performance. It also discusses the categorization and theoretical formulation of these methods. Chapter 5 delineates the chosen methods for XAI and enhancements, which have been determined through a rigorous selection process based on specific criteria. Furthermore, this chapter outlines the evaluation procedures that will be employed to assess the effectiveness of these methods. Chapter 6 provides an overview of the forecasting problem and the dataset that serves as the focal point of this thesis. Additionally, it outlines the experimental design that will be employed to assess and evaluate the proposed methods. Chapter 7 presents the findings obtained from the conducted experiments, accompanied by a comprehensive assessment of each individual experiment. Ultimately, the findings of this study are assessed and deliberated upon in Chapter 8, leading to the formulation of a comprehensive conclusion. Additionally, potential avenues for future research are proposed.

Chapter 2

Background Theory

The primary emphasis of this project lies in the domain of XAI. The chapter commences by presenting definitions of terminologies employed in the field of XAI. This section draws heavily from the research conducted for the course *IT3915 Computer Science, preparatory Project* Given the assumption that the intended audience of this thesis has not had access to the final report, certain details have been reiterated. This chapter seeks to elucidate the concept of explainability as it pertains to human understanding and explores the necessity of tailoring its presentation to various demographic groups. This paper examines the various XAI techniques that are currently available for both statistical models and neural networks. Could you please elaborate on the functioning of XAI methods in the context of time series analysis?

2.1 Terminology

The discipline of XAI endeavors to tackle the problem of algorithmic opacity. According to van Lent et al [13], XAI refers to the capability of presenting users with a coherent and comprehensible sequence of reasoning that connects the user's input, the AI's knowledge and inference processes, and the subsequent behavior exhibited by the AI system. However, the overarching inquiry persists: If a model demonstrates strong performance, what rationale exists for not placing trust in it?

2.2 Interpretability vs Explainability

Interpretability plays an important role in research areas like bias and fairness in ML models [14]. A deep dive into the inner workings of the AI/ML technique needs to be done if it is required to know precisely why and how the model is producing predictions. As a result, the offered output is determined by interpreting the model's weights and features. Interpretability refers to the ability to understand and explain the reasoning or decision-making process of a particular system or model.

Example 1 A multi-variate regression model may be constructed by an economist to forecast the inflation rate, Now the economist can view the estimated parameters of the model's variables to determine the expected result given various data examples. Given complete transparency in this instance, the economist can explain the precise why and how of the model's behavior.

Example 2 As a physician, you want to predict how effective will some drug be for a patient, and after getting the prediction, you want to know why such a decision has been made.

Therefore, in data mining and machine learning, according to Finale Doshi-Velez[15], interpretability is defined as the ability to explain or to provide meaning in understandable terms to a human.

Because people are curious about the reasons behind a decision in a social situation, explanations play a significant role in human relationships[16]. According to Miller [17], an explanation describes the process of abductive inference as well as the final product, i.e., the answer to a why question. When machine learning models are used more frequently, explanations are needed for a variety of reasons, including system verification, system improvement, system learning, and legal compliance [7].

In general, Explainability is the ability to translate the behavior of an ML model into understandable human language. You can't fully comprehend how and why the internal workings of complicated models (black boxes) affect the forecast. However, you can find significance between input data attributions and model outputs using model-agnostic techniques (such as partial dependence plots, SHapley Additive ExPlanations (SHAP) dependence plots, or surrogate models), which enables you to explain the nature and behavior of the AI/ML model.

Example 1 A neural network is used by a news organization to categorize various articles. The news organization cannot fully interpret the model, but they can compare the input article data to the model predictions using a model-neutral technique. Using this method, they discover that the model places business articles that mention sporting groups in the Sports category. The news source was able to come up with an explicable response to show the behavior of the model even though they did not use model interpretability.

Example 2 A convolutional neural network has been used in an educational institute to grade students' assignments, now the advisory board wants to know the inner working of the model considering that the board members are laypersons. That's explainability.

2.3 Importance of Interpretability

In Section 2.1, we have raised the question "If a machine learning model works great, why not just trust it ?", now, is the time to answer that question. The simple answer is "The problem is that a single metric, such as classification accuracy, is an incomplete description of most real-world tasks" - According to Doshi-Velez and Kim[15].

The need for interpretability arises from incompleteness in problem formalization [15]. This implies that getting the prediction alone may not be sufficient for some issues or jobs. Since a successful prediction only addresses a portion of your initial issue, the model must also justify how it concluded in terms of WHY.

In most cases, When something unexpected occurs, the mental model of the environment that humans have is updated. To perform this update, an explanation for the unexpected event must be found.

Example 1 If I am sick, then why I am sick?

Example 2 If I did poorly on my last exam, then why?

Example 3 If I failed my driving test, then why?

People like us enjoy memorizing contradictions or inconsistencies between elements in the knowledge model. For example, if the driving instructor can explain why you have failed this time and what has improved compared to last time, that makes much more sense, rather than saying why. The computer must explain its behavior the more a decision it makes affects a person's life. For instance, the social credit system [18] is heavily utilized in China for critical human decisions such as education, employment, housing, and others. Now, for this instance, the model needs to explain why someone didn't approve a loan request. The main idea is that the model must provide an explanation when a human being is making a decision that is of significant importance.

Also, Machine learning models by default incorporate biases from the training set. Your machine learning models might then start to bias against underrepresented groups on account of this. To find bias in machine learning models, interpretability is a helpful debugging tool. The artificial intelligence model you trained to automatically approve or deny credit applications may discriminate against a minority group that has traditionally been denied rights.

To promote social acceptance, the process of integrating algorithms and machines into our daily lives needs to be interpretable.

2.4 Human friendly explanation

It might be useful to investigate how humans use explanations to obtain a feel of what explanations can contribute to and what should be taken into account when developing a system for producing explanations. Now, the goal is to generate explanations mostly for developers and use the information further for debugging but before doing that, it's important to understand which criteria we should maintain to make the explanation friendly.

Miller [17] outlined how people define, produce, select, assess, and provide explanations. To concisely explain what explanation truly means, the article analyses disciplines like cognitive psychology, human-computer interaction, and philosophy. Let's dive into,

Contrastive explanations Humans enjoy counterfactual justifications a lot. Where "How would the forecast have been if input X had been different ?" is the meaning of the counterfactual. More likely examples are: for instance getting bad grades, people wanting to know which parameters they should change, or which categories they need to improve to get a good grade. In the case of loan approval, the applicant might ask, which documents need to be strengthened to get the load approved.

The concept of contrastive explanations is appealing to humans, as it involves constructing an example to illustrate how a desired outcome can be attained. However, the construction of examples is contingent upon the specific domain and problem at hand. For instance, it may be relatively straightforward to generate instances wherein the task at hand involves predicting residential property values. However, the analysis of time series data presents significantly greater complexity [19].

Selective explanations This means producing a short explanation. People usually don't love to see all the possible reasons for a cause, but rather the most related ones to the situation.

For example: why the loan didn't approve, because the last mortgage has not been paid yet. Why the team X failed to beet team Y, because, they had a very weak defense.

Social explanations This means that you should pay attention to the target group's social environment. Different classes of individuals should receive different explanations. For instance, a technical person's definition of ML would be - "The study of algorithms and statistical models that computer systems employ to carry out a particular task without being explicitly programmed is known as Machine Learning (ML). Learning algorithms for a variety of daily-use applications "by Mahesh, Batta [20]. Yet, it may be understood by laypeople as “It's a way to train machines to behave like humans”.

Emphasize the unusual This indicates that the focus of the explanation should be more on abnormality. Whenever an input feature differs from what is expected for a prediction, the explanation should concentrate more on it. Technically speaking, even if other "normal" features have the same influence on the prediction as the abnormal one if one of the input features for a prediction was abnormal in any way (such as a rare category of a categorical feature) and the feature affected the prediction, it should be included in an explanation.

Accurate justifications The justifications must be truthful and realistic. But, this is not the most crucial aspect of the justifications. Truthfulness is not as vital as being selective. Nonetheless, the justifications must apply to all of the inputs. For instance, a prediction model for grading students should produce the same justification for all students in the same category.

Excellent justifications are general and likely Humans frequently struggle to believe certain situations that have never happened before because they exhibit confirmation bias. For a model that forecasts power usage, for instance, the forecast should be low in the summer and during hot weather. So, if the temperature rises, consumption ought to go down. This should hold in any situation. Consistency is extremely difficult to maintain for ML models. In certain circumstances, monotonicity restrictions or linear models are required.

The aforementioned considerations suggest that certain criteria should be used when building explanations. One factor that is quite significant and essential to our undertaking is the social one. Because the developers are the primary focus of this thesis, it explains how the explanation information might be used to enhance model performance.

2.5 Time series forecasting

One of the most often used data science techniques in business, finance, supply chain management, production, and inventory planning is time series forecasting. In the simplest terms, time-series forecasting is a technique that utilizes historical and current data to predict future values over some time or a specific point in the future [21]. For time series forecasting, the fact that the future result is wholly unknown at the time of the task and can only be anticipated through analysis and evidence-based priors is an essential distinction in forecasting [11].

The problem discussed in this thesis is a multivariate time series problem. Consider n time series variables $\{y_{1t}, \ldots, y_{nt}\}$. A multivariate time series is the $(n \times 1)$ vector time series $\{Y_t\}$ where the ith row of Y_t is $\{y_{it}\}$. That is, for any time t, $Y_t = (y_{1t}, \ldots, y_{nt})^T$.

Multivariate time series, to put it simply, are situations where several variables change over time. a tri-axial accelerometer, as an illustration. Each of the three accelerations (x, y, and z) varies simultaneously over time.

Several industries use forecasting in a variety of ways. Weather forecasting, climate forecasting, economic forecasting, healthcare forecasting, engineering forecasting, financial forecasting, retail forecasting, business forecasting, environmental studies forecasting, social studies forecasting, and many more practical applications are among them. Companies with consistent historical data can create models and forecasts.

Time series models can be categorized into three main groups: traditional models, machine learning models, and deep learning models. The categorization of traditional models can be delineated into two main types: linear models and nonlinear models [22]. The Autoregressive Moving Average (ARMA) [23, 24] and Autoregressive Integrated Moving Average (ARIMA) models are widely recognized linear models that are capable of addressing time series data with stationary and nonstationary characteristics, respectively. A time series is considered to be stationary when its mean and variance remain constant over time, without any observable trend or drift. The primary constraints associated with the conventional Time Series Forecasting (TSF) models pertain to their utilization of regression techniques on a predetermined set of factors derived solely from the most recent historical data in order to generate predictions. Furthermore, conventional approaches exhibit an iterative nature and are frequently influenced by the initial conditions of the process. Furthermore, it should be noted that achieving stationarity in volatile time series is a challenging task, as it requires addressing not only drift, seasonality, autocorrelation, and heteroskedasticity but also adhering to strict conditions. Therefore, the utilization of machine learning models becomes necessary. Artificial Neural Networks (ANN) [25, 26] and deep learning Neural Networks (NNs) [27] have demonstrated superior performance compared to conventional methodologies. The most suitable machine learning techniques for time series forecasting are Recurrent Neural Network (RNN) [28] and Long Short-term Memory (LSTM) [29].

2.5.1 Forecasting methods

Time series models are employed to make predictions by leveraging historical data and established information. In the realm of time series prediction, one has the option to employ either statistical models, neural network models, or a combination of both methodologies.

Extrapolation of time series data is one of the most important components of time series forecasting [30]. It can be subjected to ML techniques including regression, neural networks, support vector machines, random forests, and XGBoost. Using models created from historical data to anticipate future observations is known as forecasting [31]. Recurrent Neural Networks (RNNs), LSTM networks, Gated Recurrent Units (GRUs), and the Transformer model are widely recognized and utilized neural network-based models for time series [32]. Not all models will yield the same results for the same dataset, so it's critical to determine which one works best based on the individual time series.

RNNs are particularly well-suited for the task of modeling time series data [33]. RNNs employ neural networks to represent the functional association between input characteristics in the immediate past and a target variable in subsequent time steps.

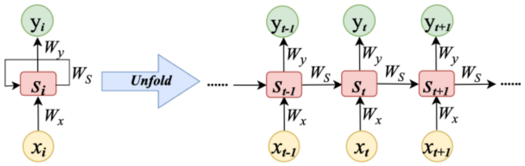


Figure 2.5.1: Structure of the RNN. Figure from [32]

As depicted in Figure 2.5.1, the RNN acquires knowledge iteratively from a training dataset comprising past observations. This learning process primarily emphasizes the evolution of an internal state, referred to as the hidden state, as it progresses from time $t - 1$ to time t. The model's outcome is determined by three parameter matrices, namely W_x, W_y, and W_s, along with two bias vectors, b_s and b_y, which collectively contribute to the definition of the model. The value of the output variable, denoted as y_t, is contingent upon the internal state variable, denoted as S_t. This internal state variable is influenced by both the current input variable, denoted as x_t, as well as the preceding state variable. The computational process of each hidden state, which refers to either a hidden unit or a hidden cell, is depicted in Figure 2.5.2. In a mathematical context, the given information can be expressed as follows:

$$\begin{aligned} S_t &= \tanh\left(W_{xs}.(x_t \oplus S_{t-1}) + b_s\right. \\ and \quad y_t &= \sigma(W_y.S_t + b_y) \end{aligned} \tag{2.1}$$

where $x_t \in \mathbb{R}^m$ represents the input vector consisting of m input features at time t, and $W_{xs} \in \mathbb{R}^{n*(m+n)}$, the parameter matrices $W_y \in \mathbb{R}^{n*n}$ are used in the context of a RNN layer. Here, n represents the number of neurons in the RNN layer. The bias vectors $b_s \in \mathbb{R}^N$ are associated with the internal state and output. The sigmoid activation function σ is utilized in the RNN. The internal state is denoted as S_t, while $x_t \oplus S_{t-1}$ represents the concatenation of vectors x_t and S_{t-1}.

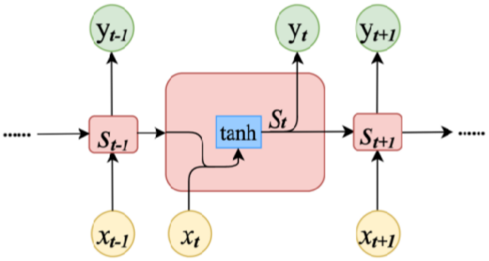


Figure 2.5.2: RNN computational process. Figure from [32]

One significant limitation of RNNs is the occurrence of the gradient vanishing problem during the repeated multiplication of the recurrent weight matrix [34]. This issue leads to a gradual decrease in the gradient magnitude over time, resulting in the RNNs ability to retain information for only short periods.

LSTM networks, a type of RNNs, have been developed as a solution to the vanishing gradient problem [35] and to effectively capture long-term dependencies in time series data. Further information regarding LSTM models can be accessed in references [36].

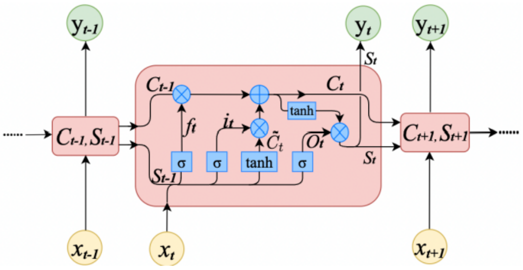


Figure 2.5.3: Structure of the LSTM. Figure from [32]

The entities are characterized at a specific moment t with respect to an internal (concealed) state denoted as S_t, as well as a cell state referred to as C_t. As depicted in Figure 2.5.3, the LSTM cell $(C - t)$ exhibits three distinct dependencies [37]: (1) the preceding cell state, denoted as C_{t-1}; (2) the preceding internal state, denoted as S_{t-1}; and (3) the input at the current time point, represented as x_t. The depicted process in Figure 2.5.3 showcases the utilization of various gates, namely the forget gate, input gate, addition gate, and output gate, to facilitate the removal/filtering, multiplication/combining, and addition of information. These gates correspond to the functions t_t, i_t, $\widetilde{C}_t$, and O_t, respectively. This mechanism enables more precise regulation of the learning of longer-term dependencies.

$$\begin{aligned}
f_t &= \sigma W_f.(x_t \oplus S_{t-1}) + b_f;\\
i_t &= \sigma W_i.(x_t \oplus S_{t-1}) + b_i;\\
\widetilde{C} &= \tanh W_c.(x_t \oplus S_{t-1}) + b_c;\\
C_t &= f_t.C_{t-1} + i_t.\widetilde{C}_t;\\
O_t &= \sigma W_o.(x_t \oplus S_{t-1}) + b_o;\\
S_t &= \tanh C_t.O_t; and\\
Y_t &= \sigma W_t.S_t + b_y;
\end{aligned} \tag{2.2}$$

where $x_t \in \mathbb{R}^m$ represents the input vector consisting of m input features at time t, and W_f, W_i, W_c, and $W_o \in \mathbb{R}^{n\times(m+n)}$ are matrices.The parameter matrices $W_y \in \mathbb{R}^{m*n}$ are used in the context of the LSTM layer, where n represents the number of neurons. Additionally, the bias vectors b_f, b_i, b_c, b_o, and $b_y \in \mathbb{R}^n$ are employed. The sigmoid activation function σ and the internal state S_t are also relevant components in this context. The forget gate, input gate, addition gate, and output gate are responsible for implementing the functions f_t, i_t, $\widetilde{C}_t$, and O_t, respectively.

2.5.2 Forecasting model selection

The key is to choose the best forecasting technique based on the properties of the time series data. Univariate or multivariate, autocorrelation, stationarity, differencing, and one-step or multi-step time series are some of the properties one could consider when deciding which model to use [38].

In this project, recurrent neural networks will be utilized to predict/forecast and generate explanations named LSTM, which has already been discussed in Section 2.5.1. Also, an encoder and decoder layer will be added on top of LSTM to generate an explanation, which will be discussed in Chapter 5. It is widely used for many things, including time series analysis and language recognition [39]. Though there are ways like adding a delay to the input could make time series problems into supervised machine learning problems [40]. Also, models like the random forest or gradient boosting regressor could be used, to solve the problem.

2.5.3 Accuracy metrics

Any machine learning project must carefully consider accuracy measures. A bad accuracy metric could taint your evaluation of models as well as the optimization (loss function) of your model.

The loss function is based on the error, which is the difference between the forecasted value and the true value for each time step [41]. The prediction error is the difference between an observed value and a prediction based on all previous observations [42]. For instance, If the error is shown as e(t) In this case, the prediction error can be written as e(t) = y(t) - $\hat{y}(t|t-1)$ where, y(t) = observation $\hat{y}(t|t-1)$ = indicates the prediction of y(t) Based on all observations so far Forecast error can be evaluated using various methods such as mean percent error, root mean square error, mean absolute percent error and mean squared error.

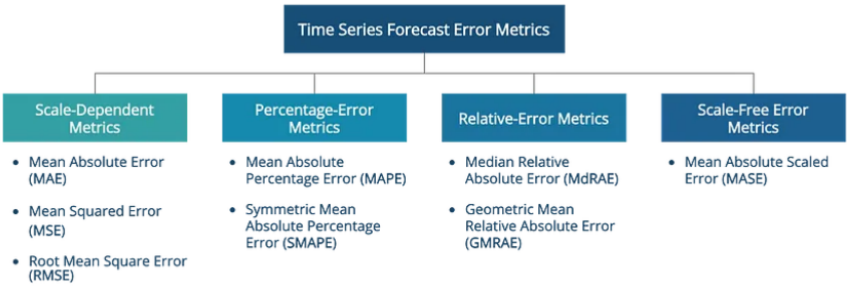

Time Series Forecast Error Metrics
Scale-Dependent Metrics
• Mean Absolute Error (MAE)
• Mean Squared Error (MSE)
• Root Mean Square Error (RMSE)
Percentage-Error Metrics
• Mean Absolute Percentage Error (MAPE)
• Symmetric Mean Absolute Percentage Error (SMAPE)
Relative-Error Metrics
• Median Relative Absolute Error (MdRAE)
• Geometric Mean Relative Absolute Error (GMRAE)
Scale-Free Error Metrics
• Mean Absolute Scaled Error (MASE)

Figure 2.5.4: Overview Time Series Forecast Error Metrics. Figure from [43]

Point forecast accuracy measures are categorized by Hyndman and Athanasopoulos [31] as scale-dependent, percentage errors, or scaled mistakes. As a result of being expressed in the same unit as the original values, scale-dependent metrics, like the Root Mean Squared Error (RMSE) specified in Equation 2.3 also Mean Absolute Error (MAE) specified in Equation 2.6, depending on the scale of the original data to determine how much the error is worth. As percentage mistakes are unit-free, it could be simpler to compare accuracy between different data sets. When the target variable's true value at a given time step is zero or the unit of measurement lacks a meaningful zero, percentage errors do not perform well. The Mean Absolute Percentage Error (MAPE), which is provided in Equation 2.4, is the most popular statistic that uses percentage error. Another percentage error, the R squared (R2) score, commonly represented as R2, quantifies the extent to which the variability in the dependent variable (y) can be accounted for by the independent variables (x) included in the model. The calculation is performed using the subsequent equation 2.5. Scaled errors, which are comparison measures for projections made for a variety of different units, are not pertinent to this theory.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2.3}$$

$$MAPE = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{2.4}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{2.5}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{2.6}$$

Most of the experiments in this project used either MAE, RMSE, Or R2

Chapter 3

State of the art

By outlining key work in the field of XAI, both in terms of general methodologies and for time series forecasting specifically, this chapter offers a solution to research questions 1 and 2 from Section 1.3. Also, the methods for describing forecasting errors will be covered. This chapter's goals are to provide an overview of the state-of-the-art in XAI and to explore in-depth a number of the strategies for explaining predictions in a variety of ways. The pros and cons of each strategy for achieving explainability will be highlighted in this chapter, along with which will be crucial to explain forecasting inaccuracy or whether we should look into any more specific area. This chapter will focus on a few selected XAI approaches rather than all of them.

With a classification of the methods used in the field and a description of the level of explainability each class can offer, Section 3.1 of the chapter provides an overview of the field. The description of intriguing XAI methods is then given, broken down into sections3.2, 3.3, and 3.3, which are devoted to model-agnostic methods, model-specific methods, time series data explanation methods. And finally in section 3.4 XAI for explaining the poor performance of a model will be discussed.

3.1 XAI outline

In the past few years, the XAI market has seen tremendous growth worldwide. With a Compound Annual Growth Rate (CAGR) of 20.1 percent from 2020 to 2030, it is anticipated to grow from a size of USD 3.55 billion in 2019 to USD 21.78 billion by 2030 [44]. There are two major techniques to generate explanations: either you create a black-box model and add a surrogate model to explain it, or you create an intrinsically interpretable prediction model, such as one using rule-based algorithms [45]. Applying methods that analyze the model after training is basically called the post hoc method [46]. Local and Global are two major categories, that the post hoc method could be divided into. Local models are to explain specific predictions and global models are to describe the typical behavior of your black-box models. Short decision trees and sparse linear models are examples of machine learning models that are thought to be intrinsically interpretable due to their straightforward form.

Model-specific and model-agnostic explanations are the two basic categories when it comes to an explanation on a per-model basis. Tools for model-specific interpretation are restricted to particular model classes. For instance, how to interpret a linear model's regression weights. Moreover, techniques that are model-specific, such as neural networks, only work with that sort of model. In order to function, model-agnostic algorithms often examine feature input and output pairs. These methods, by definition, are unable to access model internals like weights or structural data.

3.2 Model-specific or model-agnostic Methods

The debate between model-specific and model-agnostic explainability approaches is that they differ in that the XAI technique either makes use of the peculiarities of the structure of the ML model being utilized or is independent of it. Similar to how testing software applications is done using white box versus black box techniques. White box approaches are model-specific, whereas black box approaches are model-agnostic.

3.2.1 Model specific methods

The benefits of utilizing model-specific models are that they enabled the development of a more individualized explainable model, hence gaining a deeper knowledge of the decision [47]. On the other hand, because the ML or DL model has to be recreated, the entire model's structures have to be revisited, which will affect the model's performance [48]. The deconvolution-based methods for deep learning models, which follow Convolutional Neural Networks (CNNs) (which go from image input to the final class) journey in reverse order, are some of the popular model-specific approaches to these models (from final class to original image pointing out specific regions in the image which contribute to the decision). Guided backpropagation, Deep Learning Important Features (DeepLIFT), Gradient-weighted Class Activation Mapping (GRAD-CAM), Score Class Activation Mapping (CAM), and Grad-CAM++ are extensions of the deconvolution-based methods.

3.2.1.1 Deconvolution Networks

It is important to understand Deconvolution before jumping into guided back-propagation. The research done by Zeiler et al [49] on deconvolutional networks (deconvnets) is where the concept of deconvolution originated. Deconvnets [50] can be trained using an unsupervised method and are designed to function similarly to convolutional networks but in reverse (reversing pooling, reversing filter, etc.). In the context of model analysis, a deconvolutional approach involves utilizing a deconvolutional network not for training purposes, but rather as a means to investigate the inner workings of a Convolutional Neural Network (CNN).

The CNN, also known as ConvNet, is a type of deep neural network primarily employed for tasks such as image recognition, image classification, and object detection [51]. In the context of CNNs, an image is utilized as input, wherein the network is capable of attributing significance to the diverse aspects or features present within the image, thereby enabling the network to discern and distinguish

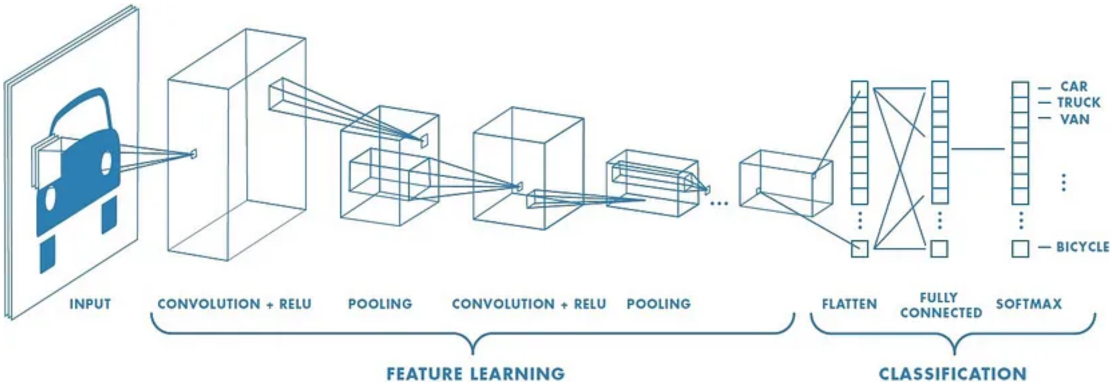


Figure 3.2.1: CNN Architecture. Figure from [51]

between them. The amount of pre-processing needed in CNNs is significantly lower in comparison to alternative classification algorithms.

A typical CNN architecture generally comprises three fundamental layers: a convolutional layer, a pooling layer, and a fully connected layer. The primary aim of convolution is to extract various features, including edges, colors, and corners, from the given input. As the network delves further into its layers, it progressively discerns more intricate characteristics, including shapes, numerical figures, and facial components. The primary objective of the pooling layer is to reduce the computational workload necessary for data processing. The final layer, known as the fully connected layer, is tasked with transforming the image into a single-column vector. This flattened output is then passed through a feed-forward neural network, and the process of backpropagation is applied during each iteration of the training process.

Starting from the desired layer, the activation signal is passed down through the layers (similar to back-propagation), through the max pooling layer, Rectified Linear Unit (ReLU), and weight multiplication [52].

Initialize: Start with the desired layer to project down and set the initial value of the reconstructed signal to correspond to its activations. back-propagate the signal that was rebuilt downward.

MaxPool: Look for indices from where the inputs were pooled and passed up in the forward pass when the MaxPooling layer is encountered. Pass the values of the reconstructed signal to these indices during the backward pass while zeroing out the other places.

ReLU: Pass the rebuilt signal only if it is positive when the ReLU layer is encountered; otherwise, zero it out.

Weights: Transpose the weights to the reconstructed signal and multiply it before passing it down when the CNN layer is encountered or any other weight multiplication.

The CNN is not perfectly inverted with this technique. Just the pixels that encourage the activation of a hidden layer are projected. The majority of deconvolution processes resemble gradient backpropagation.

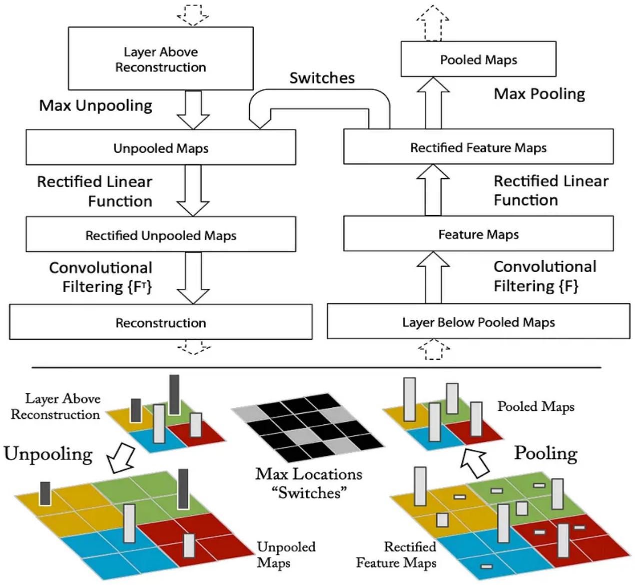


Figure 3.2.2: Forward Pass Vs Backward Pass. Figure from [49]

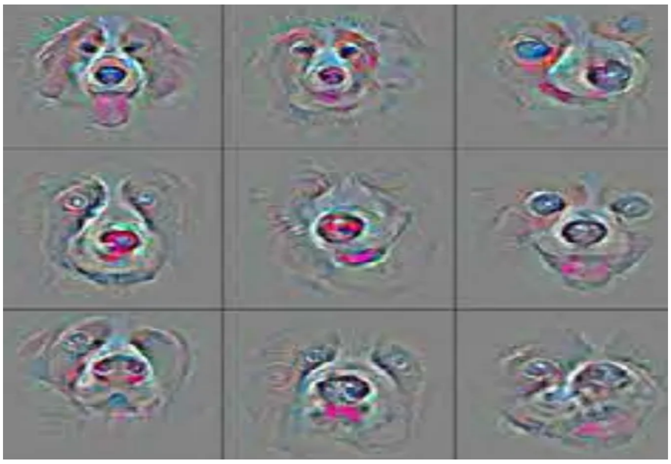


Figure 3.2.3: DeConv of various dog images projected from layer 4 to input image layer. Figure from [49]

3.2.1.2 Guided backpropagation

With a few exceptions, gradient backpropagation and deconvolution are very similar. How different is the deconvolution approach from gradient backpropagation, given that gradients can be utilized as a saliency map for comprehending the choices made by a neural network?

In deconvolution, the (gradient-like) reconstructed signal is only transmitted when it is positive, or in other words, we only transmit signals that contribute to the activation is increased. In contrast, whenever the ReLU passed the inputs up in the forward pass, the gradient was passed down through the ReLU. The differences between the two are subtle.

Initialize $g(x) = x^N$ for the desired layer of the project. Starting from the desired layer, propagate g down the layers till the input image.

According to Jost Tobias Springenberg [53] guided backpropagation combines gradient and deconvolution backpropagation techniques. At the ReLU stages, the gradient only back propagates for guided backpropagation if the gradient is positive. The formulas are as follows:

Gradient backpropagation except at ReLU:

$$Z_n = max(Y_n, 0) \tag{3.1}$$

The backward pass:

$$g_{y_n} = \begin{cases} g_{z_n} & g_{z_n} > 0 \quad \text{and} \quad y_n > 0 \\ 0 & Otherwise \end{cases} \tag{3.2}$$

Only at the ReLU stage does Guided BackPropagation diverge from "vanilla" gradient backpropagation.

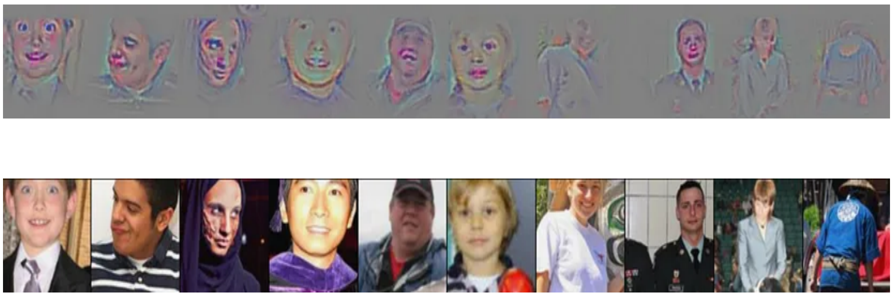


Figure 3.2.4: Guided BackProp Results on sample images. Figure from [53]

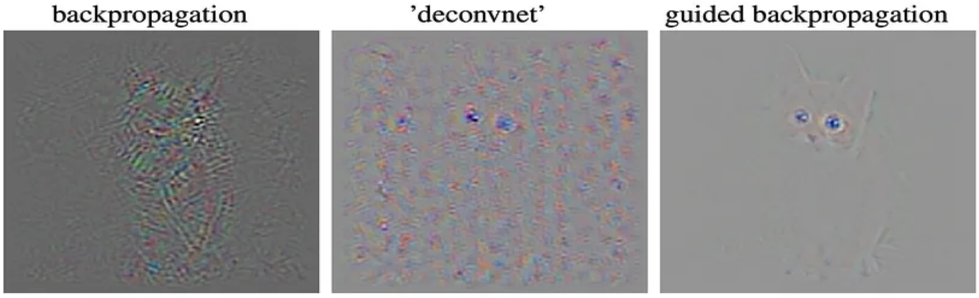


Figure 3.2.5: BackProp Vs DeConv Vs Guided BackProp. Figure from [53]

3.2.1.3 GRAD-CAM

In this research project, we have used the grad-cam extensively to identify and improve prediction.

CAM is an explanation method for CNNs, introduced by [54]. The authors evaluate networks with Global Average Pooling (GAP) architecture, which averages feature map activations, concatenates them, and outputs a vector. This architecture highlights important regions by projecting back the output weights on convolutional feature maps.

According to the GRAD-CAM paper [55] GRAD-CAM, a more flexible variation of CAM, can create visual explanations for any CNN, even if the network also contains a stack of fully linked layers (e.g. the Very Deep Convolutional Networks (VGG) networks). A saliency map and importance score based on the gradients, respectively, were constructed in order to get the GRAD-CAM of a given image and a class of interest. This coarse localization map highlighted the key areas in the image for predicting that notion.

More formally, at first, the gradient of the target class, and the activations maps of the final convolutional layer are computed and then the gradients are averaged across each feature map to produce an importance score.

$$\alpha_k^c(GlobalAveragePooling) = \frac{1}{z}\sum_i\sum_j \tag{3.3}$$

$$Gradients \quad via \quad backprop = \frac{\partial y^c}{\partial A_{ij}^k} \tag{3.4}$$

Where c is the class of interest and k is the index of the activation map in the final convolutional layer. The above-calculated alpha indicates the significance of feature map k for the intended class c. Lastly, after multiplying each activation map by its alpha importance score, the results are produced. A ReLU nonlinearity is also used in the summation in order to only take into account the pixels that have a favorable impact on the score of the class of interest. The final equation:

$$L_{Grad-CAM}^c = ReLU(\sum_k \alpha_k^c A^k) \tag{3.5}$$

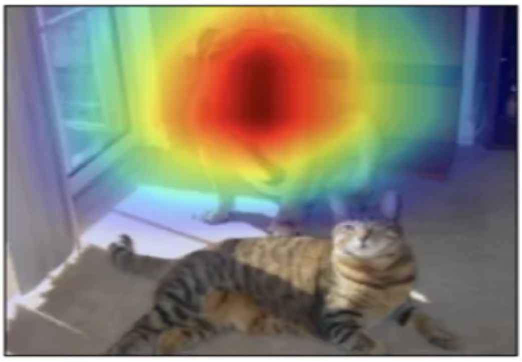


Figure 3.2.6: GRAD-CAM visualization of an example image for the class of ‘Dog’. Figure from [55]

The authors suggest Guided Grad-CAM after incorporating guided backpropagation into their methodology. It merely accomplishes this by elementally multiplying Grad-visualization CAMs and guided-backpropagation visualization:

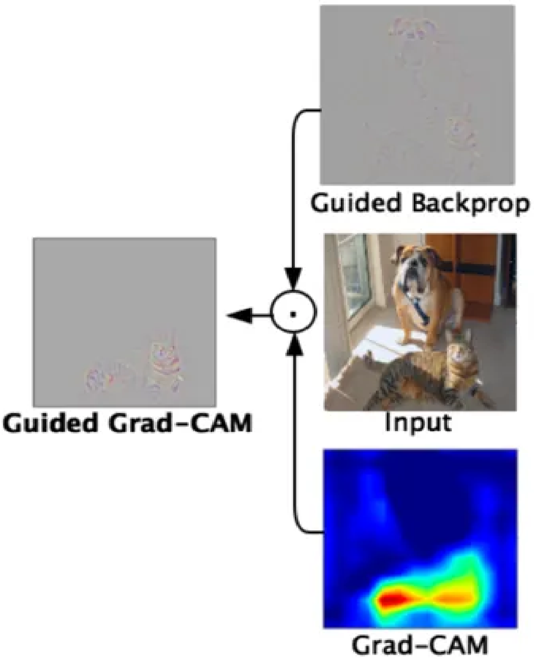


Figure 3.2.7: Guided Grad-CAM as a combination of Grad-CAM and Guided Backprop. Figure from [55]

3.2.1.4 DeepLIFT

DeepLIFT [52] is a technique for breaking down a neural network's output prediction from a particular input by backpropagating each neuron's contribution to the input's many features.

Each neuron's activation is compared to its "reference activation" by DeepLIFT, which then calculates contribution scores based on the difference. DeepLIFT may choose to separately take into account positive and negative contributions, which can help it identify dependencies that other methods might have overlooked [56].

Let t be a target neuron, and t^0 denote the target neuron's reference activation, which represents the target neuron's activation for the reference input. Using the formula $\Delta t = t - t^0$ as a reference, define the difference amount as Δt. Let $x_1, x_2, ..., x_n$ represent the necessary and sufficient neurons in one or more intermediate layers for computing t. Then DeepLIFT assigns $C_{\Delta x_i \Delta t}$ contribution scores to Δx_i such that the sum of the contribution scores for all x_i equals the deviation from the reference, Δt shown below,

$$\sum_{i=1}^{n} C_{\Delta x_i \Delta t} = \Delta t \tag{3.6}$$

A neuron's activation on the reference input is the reference output of the neuron. The user must choose the reference input, which typically involves subject expertise in order to select an appropriate reference that produces useful results.

CNNs, RNNs, and Feedforward Neural Networks (FNNs) are just some of the deep learning designs that can benefit from using DeepLIFT. The model's output reveals which properties or neurons are crucial to its decision-making. The resulting importance scores can be shown graphically as heatmaps or feature attributions, respectively, to draw attention to the most influential regions or features in producing the model's predictions [57].

3.2.2 Model agnostic methods

In simple terms, a model-agnostic method can be applied to any model [58]. It is recommended by Ribeiro, Singh, and Guestrin[59] to Separate the explanations from the machine learning model as it has some advantages.

Model-agnostic interpretation methods offer flexibility for machine learning developers, allowing them to use any model they like. This approach is independent of the underlying model, making it easier to compare models in terms of interpretability. This approach is particularly useful for evaluating multiple types of machine learning models.

3.2.2.1 Global methods

Global methods reflect typical behavior. The interpretation techniques are ideal for debugging and understanding the underlying mechanisms of a model's data.

3.2.2.1.1 Partial dependence plot

If a machine learning model has features $x_1, x_2, ..., x_n$, the partial dependency plot can determine whether the relationship between the target and a feature is linear, monotonic, or more complex. Partial dependence function for regression, shown in Equation, 3.7.

$$\hat{f}_S(x_S) = E_{X_C}\left[\hat{f}(x_S, X_C)\right] = \int \hat{f}(x_S, X_C)\, d\mathbb{P}(X_C) \tag{3.7}$$

Often, S contains simply a single or a small number of features. We are interested in the impact on the prediction of the feature(s) in S. To implement partial dependence, we marginalize the output of the machine learning model across the distribution of the features in set C. This allows us to see how features in set S relate to the anticipated outcome.

The Monte Carlo function, which is used to calculate the partial function, is given in Equation, 3.8.

$$\hat{f}_S(x_S) = \frac{1}{n}\sum_{i=1}^{n} \hat{f}(x_S, x_C^{(i)}) \tag{3.8}$$

With a set of feature S values, the partial function will reveal the average marginal effect on the prediction. The formula looks like this, where n is the total number of instances in the dataset and $x_C^{(i)}$ are the actual feature values from the dataset for the features in which we are not interested.

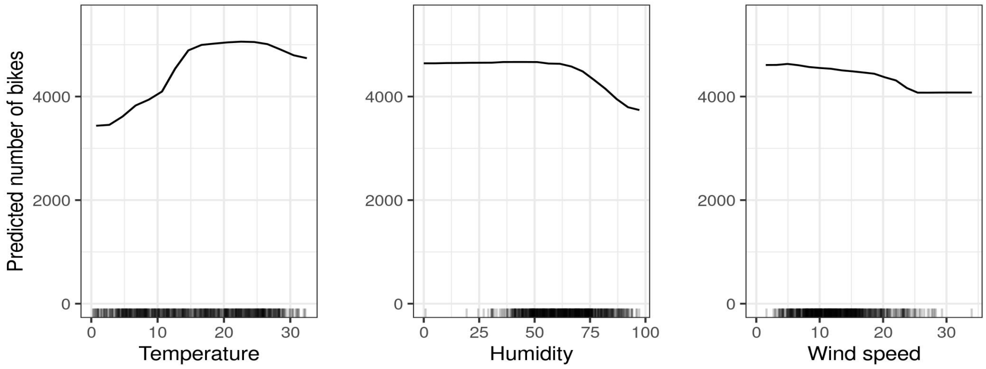


Figure 3.2.8: PDPs for the bicycle count prediction. Example and Figure from [60]

3.2.2.1.2 Accumulated local effect plots

Accumulated Local Effect (ALE) is a faster and unbiased alternative to a Partial Dependence Plot (PDP). Both ALE and PDP share lots of common characteristics and similar goals.

The PDP approach shares a commonality in that it simplifies the complex prediction function f to a function that depends on only one (or two) features. All three approaches involve averaging over the impacts of the other features to bring down the function, but the details of how this is done and whether or not the effects are averaged over the marginal or conditional distribution are what set these two approaches apart. Partial dependency plots with marginal distribution, are shown in equation 3.9.

$$\hat{f}^{(}_{S,PDP}x) = E_{X_C}\left[\hat{f}(x_S, X_C)\right] = \int_{X_c} \hat{f}(x_S, X_C)\, d\mathbb{P}(X_C) \tag{3.9}$$

For PDP, the effect of a particular feature has been calculated by replacing the other elements with the same value and averaging them, Which in terms generates some unrealistic relation sometimes. ALE plots solve this problem by focusing on the variance between predictions rather than averaging them. We can prevent the influence of linked traits by focusing on differences rather than averages [60].

In general, ALE means that we can make a reasonably accurate estimate of the change across a relatively short time range. Next, we can get a complete picture of how our input impacts our output by adding together all of the regional results. To calculate the impact of temperature on our runners at 20 degrees Celsius, we would measure the impact at 21 degrees and then deduct the difference at 19 degrees. The impact of the feature during that window can be determined by averaging the changes in prediction. After that, we aggregate the results across all of the data [61].

3.2.2.1.3 Feature interaction

Most of the time, prediction can not be expressed as the sum of the feature effects. As the effect of one feature depends on the other feature as well.

The feature interaction method basically deals with two scenarios. First, it establishes the relationships between features, and second, it provides a global measure of interaction that indicates if and to what extent a given feature interacts with all other features in the model.

If two features do not interact, we can use PDP to determine the dependency or ALE for another case. The main attraction of the Feature interaction method is the H statistics. Given below in equation 3.10 and 3.11.

$$H_j^2 k = \frac{\sum_{i=1}^{n}[PD_{jk}(x_j^{(i)}, x_k^{(i)}) - PD_j(x_j^{(i)}) - PD_k(x_k^{(i)})]^2}{\sum_{i=1}^{n} PD_{jk}^2(x_j^{(i)}, x_k^{(i)})} \tag{3.10}$$

$$H_j^2 = \frac{\sum_{i=1}^{n}[\hat{f}(x^{(i)}) - PD_j(x_j^{(i)}) - PD_{-j}(x_{-j}^{(i)})]^2}{\sum_{i=1}^{2} \hat{f}^2(x^{(i)})} \tag{3.11}$$

The 3.10 equation helps to determine the relationship between the feature j and k. And the other equation 3.11 helps to determine the relationship between feature j and others.

The H-statistic is time-consuming to calculate since iterating over all n data points is required, and at each point, the partial dependence must be calculated. To calculate the two-way H-statistic (j versus k), we need at most $2n2$ invocations of the predicted function of the machine learning models, and $3n2$ in total (j vs. all). We can use sampling to quickly evaluate n data points.

If no interaction exists, the statistic is zero, and if all of the variances have the same PD_jk or if the total variance $\hat{f}$ can be described by the partial dependence functions, it is one. Having an interaction value of 1 between two features indicates that the effect on the prediction comes solely from the interaction, as both features PD functions are held constant.

3.2.2.1.4 Functional decomposition

When given a high-dimensional feature vector, a supervised machine learning model can be thought of as a function that returns a prediction or classification score. An interpretation method called functional decomposition takes a high-dimensional function and expresses it as the combined effects of its features and interactions.

Let's consider a prediction function $\hat{f}$, that takes p features as input, where $\hat{f} : \mathbb{R}^p \to \mathbb{R}$ that generates the output. It's possible for this to be a regression function, but it can also be a classification probability, a cluster score, or a cluster score (unsupervised machine learning). When broken down into its constituent parts, the prediction function looks like this:

$$\begin{aligned} f(x) = & \hat{f}_0 + \hat{f}_1(x_1) + \cdots + \hat{f}_p(x_p) \\ & + \hat{f}_{1,2}(x_1, x_2) + \cdots + \hat{f}_{1,p}(x_1, x_p) + \cdots \\ & + \hat{f}_{p-1,p}(x_{p-1}, x_p) + \cdots + \hat{f}_{1,\ldots,p}(x_1, \ldots, x_p) \end{aligned} \tag{3.12}$$

Let's shorten the equation, Let's consider all features from $1, ..., p$ as S, Which in terms $S \subseteq \{1, \ldots, p\}$. Within S, the set contains the intercept ($S = \varnothing$) and main effects ($|S| = 1$) and all interactions ($|S| \geq 1$). The final equation would be like 3.13.

$$\hat{f}(x) = \sum_{S \subseteq \{1,\ldots,p\}} \hat{f}s(xS) \tag{3.13}$$

In the formula, x_S is the vector of features in the index set S. And each subset S represents a functional component, for example, a main effect if S contains only one feature or interaction if $|S| > 1$.

The functional decomposition is the core concept of machine learning interpretability [60]. Decomposing high-dimensional and complicated machine learning models into individual effects and interactions is a vital step toward interpreting individual effects, and this is where functional decomposition comes in. Statistical regression models, ALE, (generalized) functional analysis of variance, partial least squares (PDS), the H-statistic, and ICE curves all have their roots in the concept of functional decomposition.

3.2.2.1.5 Permutation feature importance

The significance of permutation features largely follows the prediction error, modifies the feature value, and follows the forecast shortfall.

Permutation feature importance was first introduced by Breiman [62] in his paper about random forests. Based on this idea, Fisher, Rudin, and Dominici [63] proposed a model-agnostic version of the feature importance and called it model reliance. The concept is really straightforward: We measure the importance of a feature by calculating the increase in the model's prediction error after permuting the feature. A feature is "important" if shuffling its values increases the model error because in this case, the model relied on the feature for the prediction. A feature is "unimportant" if shuffling its values leaves the model error unchanged because in this case, the model ignored the feature for the prediction.

The algorithm to generate PFI is based on Fisher, Rudin, and Dominici. Let's consider the model equation as $\hat{f}$, feature matrix X, target vector y, and error measure $L(y, \hat{f})$.

1. Estimate the original model error

   $$e_{\text{orig}} = L(y, \hat{f}(X))$$

   (e.g., mean squared error)

2. For each feature $j \in \{1, \ldots, p\}$, do:

   - Generate feature matrix X_{perm} by permuting feature j in the data X. This breaks the association between feature j and true outcome y.
   - Estimate error $e_{\text{perm}} = L(Y, \hat{f}(X_{\text{perm}}))$ based on the predictions of the permuted data.

- Calculate permutation feature importance as the quotient $FI_j = \frac{e_{\text{perm}}}{e_{\text{orig}}}$ or difference $FI_j = e_{\text{perm}} - e_{\text{orig}}$.

3. Sort features by descending FI.

It was recommended by Fisher, Rudin, and Dominici that, instate of permutating the feature, split the dataset in half and swap the values of feature j of the two halves.

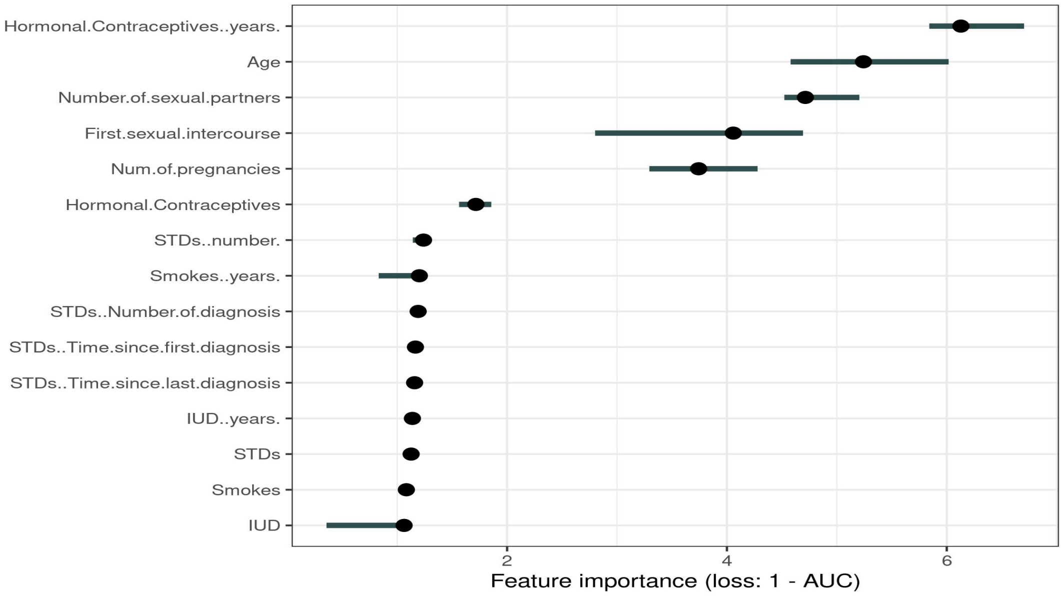


Figure 3.2.9: The importance of each of the features for predicting cervical cancer with a random forest. Example and Figure from [60]

3.2.2.2 Local methods

Local methods only explain certain inputs of a model. Over global methods, local methods could help to understand the data patterns.

3.2.2.2.1 Individual conditional expectation curves

Individual conditional expectation curves are quite similar to PDP. PDP generates the overall effect of a particular feature. According to [64] The equivalent to a PDP for individual data instances is called an Individual Conditional Expectation (ICE) plot.

In contrast to partial dependency plots, which only show the dependence of the prediction on a feature as a single line, ICE plots show the dependence of the forecast on a feature for each instance as a separate line. The mean of the ICE plot lines is the PDP. By maintaining the status quo for all other features, generating variants of this instance by substituting values from a grid in place of the feature's value, and then making predictions with the black box model, the values for a line (and one instance) can be determined. The end result is a set of points representing a single instance, each of which contains the grid's feature value and the related predictions.

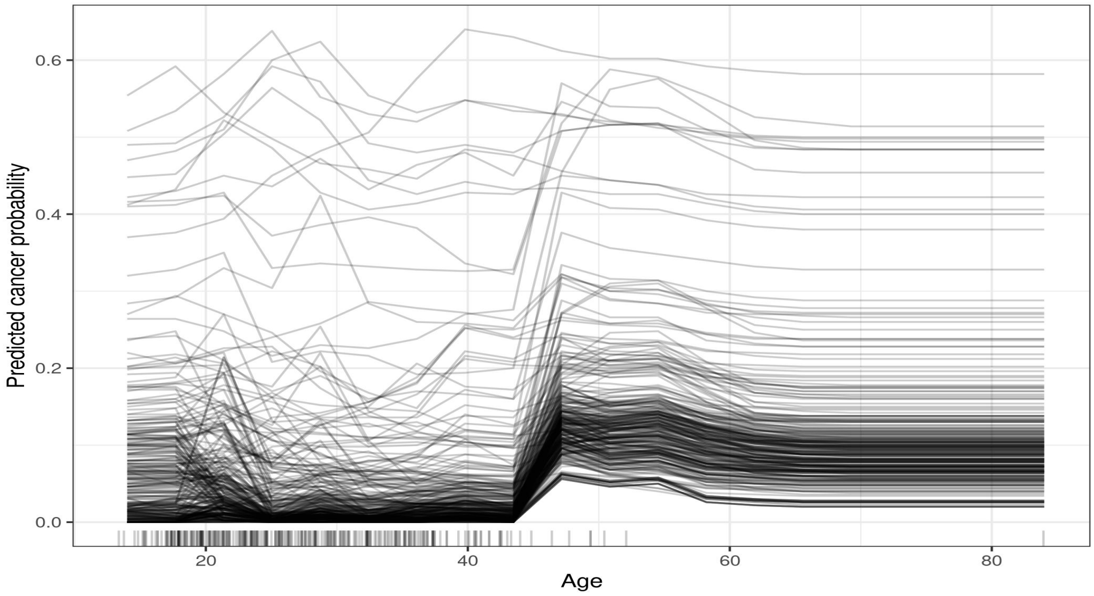


Figure 3.2.10: ICE plot of cervical cancer probability by age. Example and Figure from [60]

There are a couple of variants of ICE plots available, ex: Centered ICE Plot and Derivative ICE Plot. Those are there to solve the centering and heterogeneity problem of the PDP/ICE plot.

3.2.2.2.2 Local surrogate models (LIME)

Local interpretable model-agnostic explanations were first introduced in a paper by Ribeiro, Marco Tuli, and others [65]. LIME is mostly used to explain black box machine learning individual prediction. LIME is a surrogate model, which in terms could be either Lasso or a decision tree, or any other explainable model.

The concept is simple to grasp. To begin, preclude the training data and treat the model as a black box into which you can feed data and obtain predictions. As often as you like, probe the container. Your mission is to figure out how the machine-learning model arrived at its conclusion. LIME investigates how machine-learning model predictions change when inputted with different data. Using the black box model's predictions and the original data, LIME creates a new dataset with perturbed samples. LIME then trains an interpretable model on this new dataset, giving more weight to instances that are closer to the instance of interest.

Mathematically, Local surrogate models with interpretability constraints can be expressed as equation 3.14.

$$explanation(x) = \arg\min_{g \in G} L(f, g, \pi_x) + \Omega(g) \tag{3.14}$$

Model g (e.g. a linear regression model) is chosen as the explanation model x if and only if it minimizes loss L (e.g. a mean squared error) as a measure of how closely the explanation matches the prediction of the original model f (e.g. an xgboost model) while keeping the model complexity $\Omega(g)$ to a minimum (e.g. prefer fewer features). For example, all feasible linear regression models belong to the family G of plausible explanations. The size of the region around Instance x that is taken into account for the explanation is specified by the proximity measure πx. In actuality, LIME merely works to reduce losses. The complexity is set by the user, who may, for instance, limit the number of features available to the linear regression model.

To generate a local surrogate model, The below steps could be followed.

- Choose the case you're interested in understanding better by selecting it from the drop-down menu.
- Make changes to your data and see what the black box predicts will happen.
- The additional samples should be weighted based on how close they are to the instance of interest.
- Use the dataset with the variants to train a weighted, interpretable model.
- Justify the foresight by detailing how you understood the regional model.

3.2.2.2.3 Counterfactual explanations

Counterfactual explanations fall into the example-based explanation category. It's mostly used to describe a particular instance or prediction behavior.

If "X" hadn't happened, then "Y" wouldn't have happened; this could be an example of a counterfactual explanation. A more realistic example would be - "you didn't get the house loan because of your low household income". Or the example could explain a world, where you could get your desired result. That's why it's called counterfactual.

“A counterfactual explanation of a prediction describes the smallest change to the feature values that changes the prediction to a predefined output” - by Molnar, Christoph [60]. During generating a counterfactual explanation, it is recommended to take into consideration of the human aspect of the explanation, referring to section 2.4.

There are a lot of optimization algorithms to generate counterfactual explanations, but the simple approach would be a search algorithm, trial, and error, maybe with some guidance. As mentioned, some algorithms work with loss function and optimization methods. One by Wachter et al [66] and Dandl et al [67]. Both of the algorithms work by measuring the distance between x and x'. Where x' is the target instance. Different distance measuring technique has been used by both of the methods. Also, another Nobel technique related to genetic algorithms has been discussed in the paper [68] to generate a counterfactual explanation and gradually improve it.

3.2.2.2.4 SHAP

Shapley Additive exPlanations was first introduced by Lundberg and Lee [69] in one of their research papers. It's a method to explain individual predictions based on the game's theoretically optimal Shapley values.

SHAP aims to explain a data instance's prediction by computing the contribution of each feature to the prediction. It uses Shapley values from coalitional game theory, where feature values act as players in a coalition. Shapley values help distribute the "payout" among features, and SHAP is an additive feature attribution method, a linear model. SHAP explanation equation 3.15, given below.

$$g(Z') = \phi_0 + \sum_{j=1}^{m} \phi_j z'_j \tag{3.15}$$

Here, g is the explanation model $z' \in \{0,1\}^M$ is the coalition vector, M is the maximum coalition size, and $\phi_j \in \mathbb{R}$ is the feature attribution for a feature j, the Shapley values.

Time Series SHapley Additive exPlanations (TimeSHAP), Tree SHapley Additive exPlanations (TreeSHAP), and Kernel SHapley Additive exPlanations (KernelSHAP) are some of the versions of SHAP, which have been most eligible for different types of Models and Data formats.

3.3 XAI for time-series

The majority of cutting-edge techniques used in time series are deep learning techniques, which are too complicated to be understood. In contrast to the domains of computer vision or natural language processing, the explainability of models applied to time series has not attracted much attention.

3.3.1 TimeSHAP

The TimeSHAP recurrent explanation is a novel model-neutral extension of the KernelSHAP architecture that operates in the recurrent domain [70]. Since our method uses input perturbations to get several forms of explanation, TimeSHAP can be used to explain any tabular recurrent or sequential model.

TimeSHAP offers three local explanations: event-, feature-, and cell-level, enabling users to understand the relevance of past events and features in current predictions.

TimeSHAP is a sequential domain adaptation of KernelSHAP, working with input sequences as matrices representing features throughout time and contiguous events. It aims to attribute importance to both rows and columns, obtaining cell-level attributions for specific features.

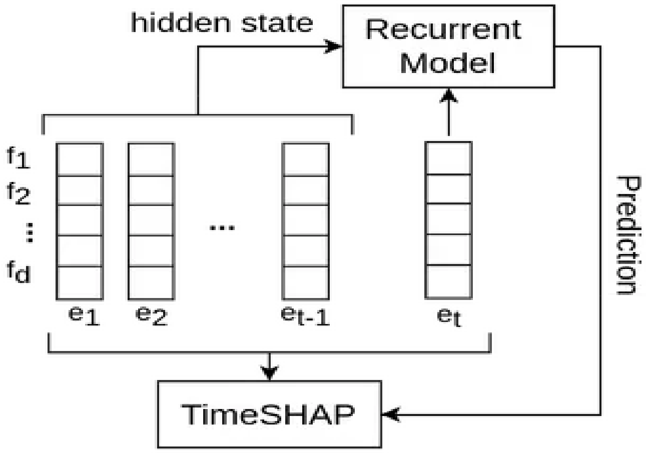


Figure 3.3.1: TimeSHAP takes into consideration the recurrence of RNNs when explaining them. Example and Figure from [71]

TimeSHAP generates event-level, feature-level, and cell-level explanations, the latter of which indicates which characteristic of which historical event was most essential for the present prediction.

Feature level: The rows of our input matrix represent time, and TimeSHAP perturbs characteristics during this time period to provide explanations for them.

$$f_X^f(z) = D_z X + (1 - D_z)B \quad \text{where} \quad D_z = diag(z) \tag{3.16}$$

Event level: TimeSHAP executes perturbations on events by toggling entire columns on and off in our input matrix in order to gain explanations at the event level.

$$h_X^e(z) = X D_z + B(1 - D_z) \tag{3.17}$$

Cell level: Cell-level explanations require turning individual cells on and off, but this approach has a fast scaling problem. For example, with 40 features and 20 events, 800 cells can lead to two possible coalitions. To obtain relevant explanations, the total number of cells considered needs to be reduced.

3.4 Explain bad forecast

Explaining bad forecasting is an important branch of this research. According to the section 1.3 the final goal of this research is to improve model performance, and the way to do so, is to explain the reason for poor performance. It is important for the end users to understand if a particular forecast instance can be relied upon, in terms says detect the anomaly.

Anomaly detection algorithms and the XAI method are two practical approaches to explaining poor forecasts, as described in the paper [72] explaining by Joe Roanec, Elena Trajkova, and others. This abstract is derived from that paper.

3.4.1 Time Series Anomaly Detection

Available anomaly detection algorithms can be classified into three categories: statistical, distance-based approaches, and model-based approaches. Non-parametric techniques, such as histogram-based approaches and bitmap time series anomaly detectors, allow fast computations and are adopted when speed is of primary importance. Parametric methods, such as Gaussian methods and least squares regression, can be used to estimate outliers based on kernel density estimation.

Statistical anomaly detection methods cannot be applied to datasets with unknown distribution, so different approaches have been developed to overcome this issue. Model-based techniques can be divided into learning and predicting whether a value is anomalous and comparing the potential outlier with expected values drawn from a generative model or data distribution. Active learning can be utilized to minimize labeling efforts.

Isolation Forest models, random forests, gradient-boosted machines, artificial neural networks, and voting ensembles are also used for anomaly detection. Models from this group have multiple configurations, varying generative methods, and outlier detection criteria.

Anomaly detection algorithms can identify anomalous forecasts in the context of a particular time series. Further insights can be obtained through XAI to understand which features were most influential to such forecasts and provide counterfactual examples to highlight value changes that would produce a better outcome.

Z-Score, Modified Z-Score, Percentile-based, Histogram-based, and Quantile-based techniques available within histogram-based approaches.

Piecewise Aggregate Approximation (PAA), Symbolic Aggregate Approximation (SAX), and Shapelet Transform are the techniques available within the Bitmap-based approach.

3.4.2 XAI for forecasting error

Researchers have developed a number of methods, ex: LIME, Anchors, Local Foil Trees, and Local Rule-based Explanations (LoRE) to provide black-box explanations of forecasting models, some of them have already been discussed in section 3.2. Using an approximation of Shapley values, explanations based on cooperative game theory calculate the relevance of features.

Research on XAI for time series has mainly focused on explainability for deep learning models. Methods like Gradient*Input, Deep Learning Important Features, integrated gradients, and Smooth-Grad have been developed to provide insights into which points in time are relevant to the forecast. Attention mechanisms have also been introduced to explain detected anomalies.

Comparative explanations, such as counterfactual and directive explanations, are used to explain the underlying reasoning of AI models. Good explanations should convey meaningful information, target a specific user profile, focus on actionability, and provide counterfactual examples. They should consider relevant context, such as the target user profile, explanation goals, and focus.

Joe Roanec, Elena Trajkova, and others also proposed a novel architecture that combines anomaly detection and explainability methods to improve forecast accuracy and give users access to more relevant background data. A bad forecast can be found with the help of the anomaly detection module. When this occurs, the forecast is either reverted to a local statistical model or the user is warned that they should not place faith in the prediction because of how the underlying time series has behaved in the past. Details can be found in their paper [72].

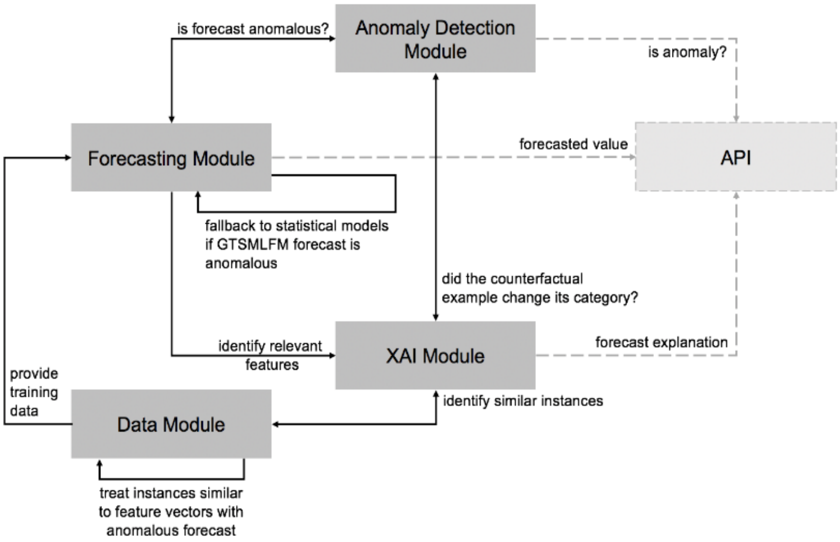

is forecast anomalous?
Anomaly Detection Module
is anomaly?
Forecasting Module
forecasted value
API
fallback to statistical models if GTSMLFM forecast is anomalous
did the counterfactual example change its category?
identify relevant features
XAI Module
forecast explanation
provide training data
Data Module
identify similar instances
treat instances similar to feature vectors with anomalous forecast

Figure 3.4.1: Modular architecture for Explaining bad forecasting.Figure from [72]

The methodology and dashboard presented in this study are designed to be applicable to a wide range of global time series models. These models are limited to machine learning algorithms that utilize input features that can be easily understood by humans. Additionally, a significant number of the explanations presuppose that the features communicate a specific significance to the user. This requirement can only be fulfilled by handcrafted features. As a consequence of this constraint, the efficacy of this methodology is not infallible for deep learning models. However, we contend that it is highly motivating to establish a framework for enhancing time series through XAI.

Chapter 4

XAI-based model improvement

Despite the recent advancement in the field of XAI, both for statistical models and neural networks, these tools have only been used for visualization purposes [73]. But, there have been some works recently, that use the explanation information to improve model performance.

Incorporating human knowledge into machine learning models for reasoning correction is an old idea. Expert Systems, Rule-Based Systems, and Case-Based Reasoning are some of them. However, recent years have seen a rise in efforts to incorporate explanations similarly, all with the end goal of enhancing the present ML models' many good qualities.

This chapter offers a solution for research question 3 from section 1.3. The goal of this chapter is to review the available methods to leverage XAI to obtain better models.

4.1 Enhance model properties with XAI

Talking about ML models or algorithms, just being accurate is not enough, it also has to be trustworthy and reliable. Though the traditional optimization matrix is not enough to judge an ML model, there are some other properties that need to be considered. Not only performance, but explanation could also help to improve several other desirable properties.

4.1.1 Performance

In the realm of machine learning, models that exhibit a strong capacity for generalization and high levels of accuracy are generally regarded as desirable. However, in certain cases, the phenomenon of overfitting and the utilization of domain-specific input features can lead to an escalation in complexity, consequently affecting the overall performance of the model. The accuracy of a test is primarily determined by the test data set. However, it is important to note that this characteristic may not always accurately reflect the model's true generalization ability.

4.1.2 Convergence

The desirability of faster convergence is often hindered by the challenge of striking a balance between convergence speed and achieving an optimum that attains state-of-the-art performance.

4.1.3 Robustness

The concept of domain dependency refers to the extent to which a particular phenomenon or concept is influenced by the specific domain Neural networks exhibit a degree of sensitivity to the quantity of data available. Minor alterations can lead to significantly divergent forecasts. The justifications for their decisions can be subject to manipulation through the arbitrary alteration of inputs, while still maintaining the same prediction outcome. Both effects can be alleviated by enhancing the model's resilience to minor modifications of the input.

4.1.4 Efficiency

Deep Neural Networks (DNNs) necessitate a substantial volume of data. At times, it can be challenging to collect a substantial volume of data. Frequently, a substantial amount of data can be acquired through the utilization of expert opinion or crowd-sourcing methods. Training this particular model and attaining the desired level of accuracy poses a considerable challenge. It is prudent to take into account the reduction of data complexity as a means to enhance model performance.

4.1.5 Reasoning

Enhancing reasoning abilities can often pose a challenge. However, it can be argued that reasoning is closely linked to performance. Modern machine learning models exhibit a high degree of reliance on data, making them susceptible to capturing patterns present in the training data, regardless of their applicability to real-world scenarios. It is imperative that specific attributes are present in both the training and test datasets.

4.1.6 Equality

Ensuring that the training data exhibits a balanced distribution of examples across all classes is of utmost significance. In order to attain optimal performance across various scenarios, it is imperative to ensure equitable treatment of all populations and data sources. It should be noted that this particular characteristic is distinct from the concept of fairness, which is a subject of extensive research in its own right but is not addressed in this study.

Furthermore, it is imperative to consider additional factors such as data quality, outliers, data drift, and concept drift. The topic of Outliers has been previously addressed in Chapter 3. The efficacy of the models employed may diminish over time as a result of a phenomenon referred to as "model drift". The deployed model is continuously being updated with incoming data to generate predictions. Nevertheless, it is plausible that this data could exhibit a distinct probability distribution in comparison to the one utilized for training the model. ML model drift can be classified into two overarching categories: concept drift and data drift. Concept drift occurs when there is a change in the posterior probabilities

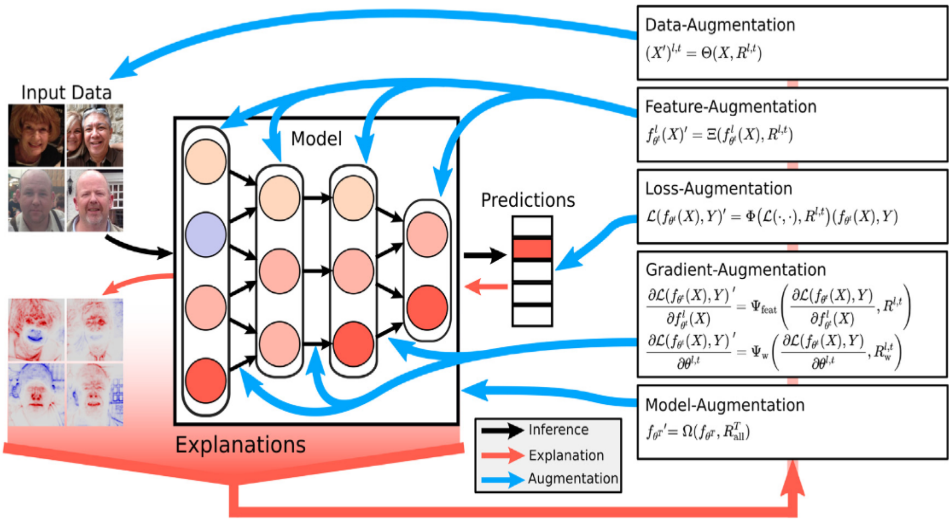

Input Data
Model
Predictions
Explanations
Inference
Explanation
Augmentation
Data-Augmentation
$(X')^{l,t} = \Theta(X, R^{l,t})$
Feature-Augmentation
$f^l_{\theta^t}(X)' = \Xi(f^l_{\theta^t}(X), R^{l,t})$
Loss-Augmentation
$\mathcal{L}(f_{\theta^t}(X), Y)' = \Phi\big(\mathcal{L}(\cdot,\cdot), R^{l,t}\big)(f_{\theta^t}(X), Y)$
Gradient-Augmentation
$\frac{\partial \mathcal{L}(f_{\theta^t}(X), Y)'}{\partial f^l_{\theta^t}(X)} = \Psi_{\text{feat}}\left(\frac{\partial \mathcal{L}(f_{\theta^t}(X), Y)}{\partial f^l_{\theta^t}(X)}, R^{l,t}\right)$
$\frac{\partial \mathcal{L}(f_{\theta^t}(X), Y)'}{\partial \theta^{l,t}} = \Psi_{\text{w}}\left(\frac{\partial \mathcal{L}(f_{\theta^t}(X), Y)}{\partial \theta^{l,t}}, R^{l,t}_{\text{w}}\right)$
Model-Augmentation
$f_{\theta^T}' = \Omega(f_{\theta^T}, R^T_{\text{all}})$

Figure 4.2.1: Model improvement with XAI. Explanations offer information about the model's decision-making and behavior, which may in turn be leveraged to improve models by augmenting different components of the training process or by adapting the trained model. Figure from [73]

of X (input) and Y (output), specifically referring to the probability of Y being the output given X as the input [74]. On the contrary, data drift refers to the scenario in which there is a change in the input distribution of the model. Data drift can occur when there is an uneven distribution of training data, leading to an imbalance in certain terms.

4.2 Theoretical formalization for improvement

In the given scenario, we have a model denoted as f^t_θ that is parameterized by θ^t after undergoing training iteration t ranging from 1 to T. The model consists of L layers, denoted as l, with each layer's parameters represented as $\theta^{l,t}$ after t training iterations. The input features to layer l are denoted as $f^l_{\theta^t}$, where X represents the input data, and $f_{\theta^t}(X)$ represents the model's output obtained from the last layer L, i.e., $f_{\theta^t}(X) = f^L_{\theta^t}(X)$.

Additionally, we assume the availability of a (local) XAI technique that provides explanations $R^{l,t}$ for the model's decisions at each intermediate layer l and iteration t. These explanations correspond to the intermediate features $f^l_{\theta^t}(X)$. According to [73] utilizing these explanations, we can enhance each component separately, including Data, Feature Representations, Loss Function, Gradient, and the Trained Model.

4.2.1 Data augmentation

XAI is a field of study that focuses on interpretability and transparency The process of augmentation utilizes explanations to modify the arrangement of the data.

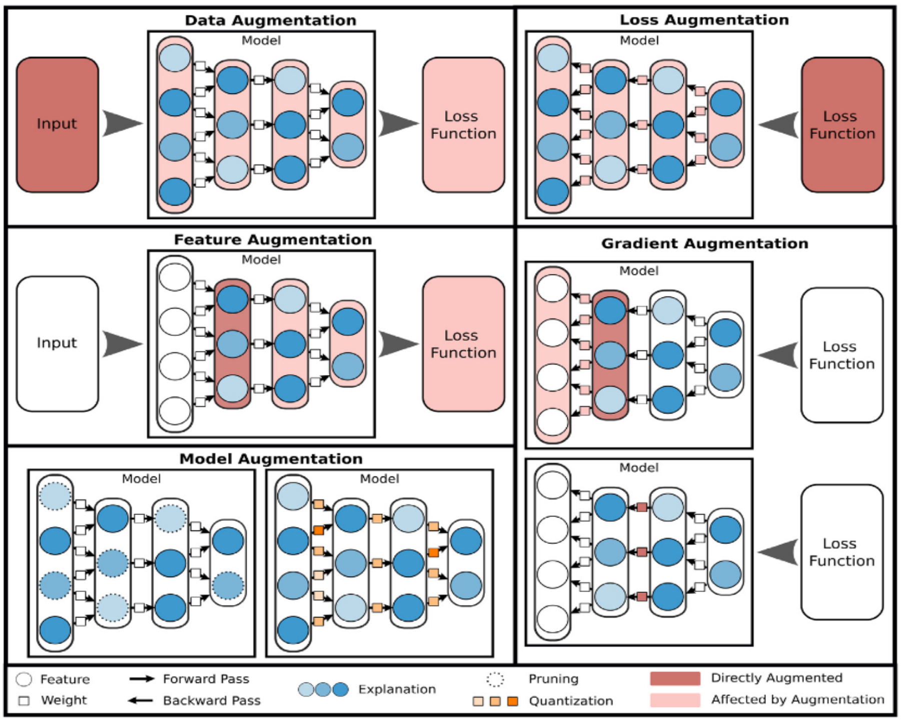


Figure 4.2.2: Types of XAI-based augmentation. Figure from [73]

As shown in Fig. 4.2.2 (top left), this form of augmentation is implemented during the initial phase of the forward-backward training loop.

The aforementioned methodologies seek to mitigate the presence of biased or erroneous decision-making in models through the manipulation of the sample distribution. Explanations are employed for the purpose of generating synthetic samples that serve as countermeasures against undesirable behavior [75, 76]. Explanatory Interactive Learning (XIL) allows users to intermittently provide input in order to rectify the decision-making process of a model. The utilization of local explanations and heatmaps has been found to improve model reasoning and foster human trust in decision-making processes [75, 76].

The researchers in reference [77] direct their attention towards enhancing model reasoning capabilities through the identification and elimination of artifacts present in the dataset. They achieve this by employing the ProtoPNet architecture and Prototypical Relevance Propagation (PRP) technique, which is based on Layer-wise Relevance Propagation (LRP) as described in [78]. In the context of Medical Image Analysis, the utilization of local explanations is observed in [79]. This approach involves the selection of informative samples by considering their explanation scores.

The issue of imbalanced data is addressed in a study by [80], where XAI-guided imbalance mitigation techniques are proposed. Scalar metrics that are derived from attribution maps, such as entropy and pairwise distances, aid in the estimation of a model's generalization performance and convergence. The utilization of these metrics serves the purpose of achieving equilibrium in class-specific performances during the training process, resulting in accelerated convergence and enhanced accuracy.

In brief, these methodologies utilize explanations as a means to mitigate bias, enhance model reasoning, optimize sample selection, and tackle challenges related to imbalanced data.

4.2.2 Intermediate features augmentation

The utilization of explanations can be advantageous in determining the significance of different features. This knowledge can then be applied to adjust, conceal, or modify intermediate features. As depicted in Figure 4.2.2 (located in the center-left position), the augmentation of intermediate features has no impact on the inputs of the model, but it does affect all subsequent components of the training process. Mainly two types of methods exist in this category.

4.2.2.1 Attention and Intermediate Feature Masking

The objective of this XAI-guided feature augmentation technique is to enhance the performance of a model by leveraging explanations to differentiate between relevant intermediate features and irrelevant ones. In order to fulfill this objective, it is necessary to employ XAI techniques that are capable of offering intermediate explanations. The intermediate explanations exhibit a similar structure to the features they elucidate, enabling their direct utilization for generating a mask that represents the importance of these features and assigns weights to them during the forward pass, akin to an attention mechanism.

In the domain of image recognition, the Attention Branch Network (ABN) [81] interprets local explanations as an attention map provided by extending CAM [82]. ABN consists of a feature extractor, an attention branch, and a perception branch. The attention branch computes the attention map based on the feature extractor's output, while the perception branch acts as a standard classifier. The attention map masks the perception branch input, enabling the model to focus on the most important parts of a given sample. The ABN is trained using a loss function that combines the losses from the attention branch ($\mathcal{L}$att) and the perception branch ($\mathcal{L}$per):

$$\mathcal{L}\text{abn}(f\theta^t(x_i), y_i) = \mathcal{L}\text{att}(f\theta^t(x_i), y_i) + \mathcal{L}\text{per}(f\theta^t(x_i), y_i) \tag{4.1}$$

To further improve the ABN, a reason loss term ($\mathcal{L}$reason) can be added, which measures the discrepancy between the original attention maps ($\mathbf{a}l, t^i$) and the edited attention maps ($\mathbf{r}^i_{l,t}$) generated by a human expert:

$$\mathcal{L}\text{abn}(f\theta^t(x_i), y_i) = \mathcal{L}\text{att}(f\theta^t(x_i), y_i) + \mathcal{L}\text{per}(f\theta^t(x_i), y_i) + \gamma\mathcal{L}_{\text{reason}}(r_i, a_i) \tag{4.2}$$

where $\mathcal{L}\text{reason}(r_i, a_i) = |\mathbf{a}l, t^i - \mathbf{r}^i_{l,t}|_2$. This additional term encourages alignment between the model's attention and the human expert's corrections.

Another technique, dropout, is employed to prevent overfitting by randomly setting a subset of features to zero during training. In [83], an improved dropout method based on XAI, specifically Excitation Backpropagation, is proposed. This method identifies and drops out more important neurons with a higher probability, resulting in enhanced generalization ability and reduced degradation compared to random dropout.

4.2.2.2 Intermediate Feature Transformation

The aforementioned attention and feature masking techniques make use of the feature-wise information provided by XAI to assess the significance of intermediate features and adjust their weights accordingly. In contrast, the methods discussed in this section take a more indirect approach by leveraging explanations and relying on intricate feature transformations like translation and projection to rectify a model's reasoning.

With this goal in mind, the Class Artifact Compensation (ClArC) framework [84] aims to identify and remove biases, artifacts, and Conceptual Hazard (CH) behavior. The framework involves three steps: artifact identification, artifact model estimation, and updating the predictor model. The Spectral Relevance Analysis (SpRAy) algorithm [84] is extended for artifact identification using local explanations to identify behavioral patterns. Once an artifact is found, two variants of ClArC are used for removal: Augmentative Additive Class Artifact Compensation (A-ClArC) and Projective Class Artifact Compensation (P-ClArC). A-ClArC adds the artifact to all samples, desensitizing the model, while P-ClArC suppresses the artifact during inference. Both approaches can be applied in feature space or input space. The effectiveness of artifact mitigation depends on the complexity of the artifact and the layer in which it occurs.

4.2.3 Loss function augmentation

The loss function determines the behavior of a model. Thus, augmenting the loss function based on explanations can help specify which behavior is desired, using explanations as feedback. Augmenting the loss function in this manner only affects the backward pass (see Fig. 4.2.2 (top right)).

The introduction of a regularisation term in the loss function allows for the adjustment of a model's learning behavior to achieve a range of desired outcomes. An approach was to ensure that reasoning aligns with expert knowledge, by evaluating explanations against a ground truth that incorporates human expectations [85, 86, 87]. In contrast, the implementation of a human-independent constraint on explanations may yield enhanced reasoning abilities, increased robustness, and improved performance [88].

Right for the Right Reasons (RRR) is one such framework, that is designed to enhance the optimization of a model's reasoning process. The researchers make the assumption of a dataset X that includes, alongside the ground truth class labels, a binary annotation mask a_i^l for each sample $X_i \in X$. This mask indicates, for each input dimension $\delta \in (1 \ldots D)$, whether it should be considered irrelevant $(a_i^l[\delta] = 1)$ to the decision-making process of the model.

It should be noted that in the initial approach, the value of l is set to 0, resulting in the consideration of only annotation masks a_i^0 within the input space. The loss function can be enhanced by incorporating an additional regularisation term, which seeks to align the explanation of each prediction with the corresponding annotation mask.

$$\mathcal{L}_{rrr}(f_\theta(x_i), y_i) = \mathcal{L}_{pred}(f_\theta(x_i), y_i) + \lambda \mathcal{L}_{reason}(r_i^{l,t}, a_i^l) \quad (4.3)$$

In the given context, the symbol λ represents a regularisation parameter. The term $\mathcal{L}_{pred}$ refers to the standard prediction loss, which measures the discrepancy between the actual and predicted class probabilities. This term also encompasses any regularisation terms that are not based on explainable artificial intelligence (XAI). In conjunction with acquiring the ability to make accurate predictions using $\mathcal{L}_{pred}$, the inclusion of the reasoning loss term $\mathcal{L}_{reason}$ serves to ensure the correctness of reasoning.

4.2.4 Gradients augmentation

In a manner akin to the utilization of explanations for enhancing feature representations in the forward pass, the insights provided by explanations regarding feature importance can also be applied in the backward pass. There exist two distinct forms of feature augmentation that can be employed. To begin with, as illustrated in the upper section of the lower right panel of Figure 4.2.2, it is possible to manipulate the intermediate feature gradients at layer l through scaling, masking, or transforming. This mirrors the feature augmentations discussed earlier during the forward pass. In an alternative approach, the gradients of the parameters can be directly augmented by calculating importance scores R_w^l for each parameter.

The gradient is responsible for determining both the direction and magnitude of the updates made to a model's parameters during the backward pass. The convergence behavior and performance can be enhanced by controlling the backward flow of weight updates through the modification of either the intermediate feature gradients or the parameter gradients directly. In contrast to the intermediate feature masking methods discussed in Section 4.2.2.1, which involve generating a mask from intermediate explanations to assign weights to features during the forward pass, a comparable mask can be computed using XAI techniques to indicate the significance of gradients during the backward pass. All gradient transformation approaches aim to modify the proportion in which model parameters are updated.

4.2.5 Model augmentation

Even after a model has been trained, the XAI's provision of intermediate feature importance information can still be utilized to enhance the overall model. This can involve modifying the model's structure or reducing the storage space needed for its parameters.

In practical applications, XAI techniques are commonly utilized for two purposes: pruning the model (as depicted in Figure 4.2.2, on the left side of the bottom left panel) or quantizing the model (as depicted in Figure 4.2.2, on the right side of the bottom left panel). It is important to acknowledge that in the field of literature, the aforementioned categories of XAI-based augmentation are generally implemented individually rather than simultaneously. Nevertheless, as a result of each category modifying distinct elements of the training process, it is theoretically possible to apply multiple augmentations simultaneously, such as targeting the same model property, thereby altering different components of the training process concurrently.

Most of the above types of augmentation are used during training to improve the model's performance. But even if you get a good model, it may still have some bad things about it, like too many factors that take up a lot of space on your hard drive and require a lot of processing power. The above properties depend on how the model is defined, so the best way to improve these qualities would be to improve the model. How well algorithms work for trimming or quantization depends on how well they can estimate how important each parameter is to how the model makes decisions and how well it works. So, the information given by XAI is essentially a criterion that can be used to improve the efficiency of models in terms of how much computing power is needed for inference or how much space they can take up. To reach this goal, you need to use XAI methods that can explain each layer. Based on what was found, a model's connections and features are changed. Previous methods [89, 57] calculated intermediate attributions to help with model trimming. These attributions are calculated for a small number of reference samples and then averaged to come up with a pruning measure shown as importance scores. So, the first thing that happens is that the neurons or filters with the least value are cut. This makes the model more space-efficient. In the same way, the importance scores can be used to separate the model's weights, which makes them easier to remember.

Chapter 5

Method

This chapter describes the approach used for the practical part of this thesis. The original plan was to build an explanation prototype for our wind production forecasting errors, which is used in monitoring after the wind production forecast is done, and also improve model performance using XAI. Which in terms revived in a form that unified model that will be integrated within the production forecast and can generate explanations and fix itself based on the explanations.

But the plan changed later due to technical issues and limited time and other external factors. Instead, the plan becomes to explore and find the best possible way to explain forecasting errors and improve the model based on XAI. This chapter examines the methods that have been chosen based on specific selection criteria outlined in Chapters 3 and 4. It also explores the process by which the explanations generated by these methods will be assessed. Moreover, explanation data could be analyzed to enhance the model's performance.

As outlined in the existing literature, there exist two primary methodologies to explain how forecast predictions are made. The first approach involves employing interpretable forecasting models that generate internal information, which can subsequently be utilized as explanations or serve as the foundation for explanations. The second approach entails the application of post-hoc explanation models on non-interpretable forecasting models.

One benefit of employing an interpretable model lies in its ability to provide explicit details regarding its internal mechanisms, thereby enabling a comprehensive understanding of how these components utilize the available data. Conversely, a post-hoc explanatory model has the capability to be employed across various forecasting methods, thereby allowing for flexibility in selecting a forecasting method and subsequently altering it without compromising the ability to provide explanations. Nevertheless, post-hoc techniques are inherently limited in their ability to comprehensively comprehend the internal workings of a forecasting model. Though it's extremely difficult to understand the internal working of NNs model due to the model complexity and millions of gradients.

5.1 Selection criteria

Chapter 3 provides an overview of prominent techniques in XAI, which encompass diverse approaches for generating explanations. Additionally, it explores time series forecasting methods that exhibit varying levels of interpretability.

And chapter 4 provides an overview of the available techniques for XAI-based model improvements. Not all the available methods are going to be suitable for all types of models, Bet methods related to Data modification and augmentation could be suitable for any kind of ML problem.

To determine the suitability and relevance of various approaches for the purpose of this thesis, a set of selection criteria is established. These criteria aim to characterize the problem under investigation and outline the expectations for an explanation method that is deemed suitable for the task.

First, the explanation target group has to be identified as described in Section 2.4. For this research project, the target group would be mostly developers. Or more specifically, the explanation should be qualitatively interpretable. As for our research goal, we will use interpretable information to improve the prediction model. Though qualitative information could be converted to mathematical terms, the other way around is preferable.

Second, why the target group wants explanations are critical to understanding what type of information the models should output. Which we have already discussed in the first criterion.

Third, It's important to determine how useful the explanation information is, Which in terms directly affects the model improvement ratio for this thesis.

Finally, As discussed in Section 1.2, the prediction problem related to this thesis is financially sensitive. So, It's very important to keep a balance between the explanation and accuracy.

Overall, the implementations of the methods must be taken into account, as the implementation affects how closely the methods achieve their goal. In this thesis, rather than focusing on the popular explainable method, we take into consideration the numerical explainability of methods Due to the final goal and chose accordingly.

5.2 Choice of methods

Based on the selection criteria outlined in section 5.1, the chosen methods or approaches should possess the capability to offer developers pertinent and comprehensible explanations. The temporal dynamics or explanation holds paramount significance in the context of this research endeavor. The results should be displayed in both graphical and numerical formats. An XAI-based model enhancement approach should possess the capability to leverage this information in order to enhance the accuracy of predictions. The methods discussed in both Chapter 3 and Chapter 4 are presented in this section, as they were emphasized and aligned with the research questions and objectives of the thesis. There may exist alternative methodologies that could potentially yield superior results within each respective approach. However, it is imperative that these methodologies remain indicative of the various approaches to explanations.

As stated in Chapter 3, GRAD-CAM has been extensively utilised in this research endeavour. As stated in Section 3.2.1.3 The GRAD-CAM technique demonstrates a high level of effectiveness when applied to fully connected layers, particularly in the context of various neural network architectures such as CNNs, LSTM networks, and others. In accordance with the objective of this research endeavor, we have opted to employ an architecture based on LSTM, which has been augmented with an attention mechanism and guided backpropagation. This proposed approach aims to facilitate prediction while also providing local temporal interpretability for each input, as outlined in the work by Schockaert et al. (2020) [90]. The utilization of temporal attention will afford us a significant opportunity to leverage the available information in order to enhance the model. This approach primarily employs RNNs, while sharing the same fundamental mechanism as GRAD-CAM.

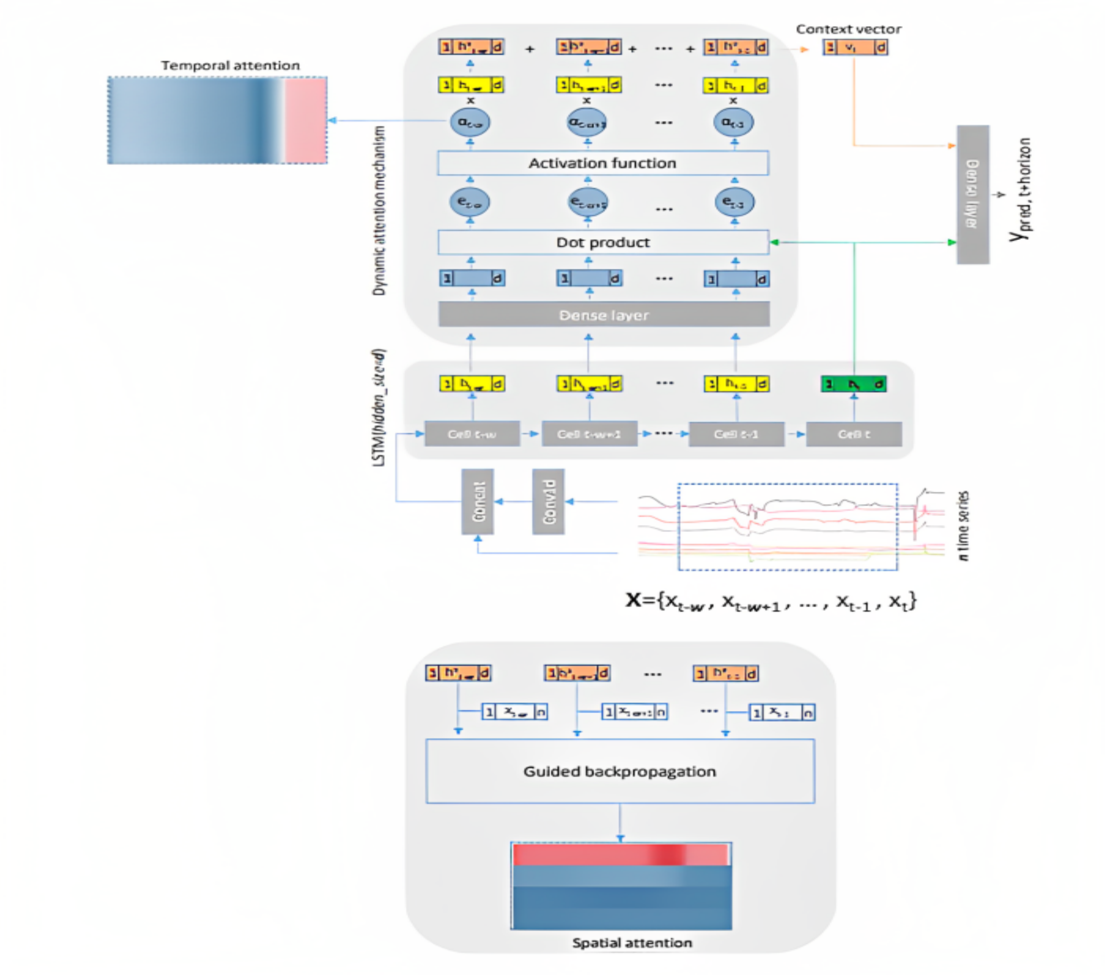


Figure 5.2.1: time series local spatial/temporal attention mechanism architecture. Figure from [90]

The architecture depicted in Figure 3.2.6 was initially introduced by Cedric Schockaert, Reinhard Leperlier, and Assaad Moawad in their scholarly publication [90]. The input time series, denoted as n, is augmented by incorporating time series generated through the utilization of a one-dimensional convolutional layer (conv1d). This augmentation process facilitates the learning of significant transformations of the original n time series. The architecture incorporates an LSTM layer that produces a hidden state h_i for each time step i within the time range $[t-w,\dots,t]$ of the multivariate time series X. This LSTM layer is concatenated with the output of *conv1d* within a time window of size w. The application of dynamic temporal attention involves considering the $w - l$ preceding hidden states, while the calculation of the context vector v_t follows the procedure outlined in Figure 5.2.1. The dense layer within the attention mechanism block acquires knowledge of the context, which is specifically defined within the n time series denoted as X. By acquiring knowledge of the context, we are able to produce dynamic attention weights $\alpha = \{\alpha_{t-w}, \alpha_{t-w+1}, \ldots, \alpha_{t-1}\}$, which offer a means of local temporal interpretability for the prediction $Y_{pred,t+horizon}$.

For each time step t, a guided backpropagation-based approach is employed to compute the relationship between the modified hidden state h_i^a and the original input vector x_i, where i ranges from $t - w$ to t. The aim is to identify the specific time series within the input vector x_i that is causing a change in the hidden state h_i^a.

In order to enhance the XAI-based model, we have chosen to employ the data augmentation technique, as discussed in Section 4.2.1, taking into account the constraints imposed by the available information and time limitations. The heatmap produced by the aforementioned model has the potential to serve as a diagnostic tool for pertinent time series data, which we utilized to enhance the predictability of the model.

The partial dependency plot, as discussed in Chapter 3, has also been utilized for enhancing the model. A partial dependency plot enables the assessment of the impact of individual features on the target variable. Based on the aforementioned information, it is possible to make modifications to the training data in order to enhance the model.

5.3 Evaluation

The comparative evaluation of the forecasting method's accuracy will be conducted by comparing it to a baseline model, as discussed in [91]. The MAE, as defined in Section 2.5.3, serves as an accuracy metric. It is widely recognized as a dependable measure of accuracy, particularly in the context of time series prediction. Aneo As, in its extensive machine learning models, also employs MAE as an accuracy metric.

The thesis incorporates multiple accurate matrices, not solely relying on MAE. Additionally, the accuracy metric known as "skill score" [92] has been extensively employed in this project. As presented in equation 5.1.

$$\text{Skill Score} = \frac{\text{score for the forecast} - \text{score for the standard forecast}}{\text{perfect score} - \text{score for the standard forecast}} \tag{5.1}$$

The skill score quantifies the precision of a prediction by comparing it to the precision of a standard forecast or baseline model, as elaborated in Section 6.4.2. The baseline model typically refers to a forecast that is readily accessible to a forecaster, although it does not necessarily demand any exertion or expertise on their part for its preparation. For instance, in the context of forecasting the required quantity of goods in a grocery store, it is possible that the amount needed on a particular day could be equivalent to the quantity required during the previous week on the same day.

A skill score refers to the evaluation of a forecast's score in relation to the score achieved by a standard forecast, both of which are based on the same set of verification data. The accuracy score utilized can be any of the prevalent metrics employed in the field of verification. Skill scores for continuous variables are typically derived from either the mean absolute error or the mean squared error. In general, the skill score is a numerical metric that evaluates the efficacy of a model by comparing its performance to a predetermined reference value. This metric typically falls within the range of 0 to 1.

Chapter 6

Experiments

The following section provides a detailed account of the experimental procedures carried out in the course of this study. The text commences by providing an overview of the dataset and the preprocessing techniques employed on the dataset. Subsequently, it delineates the time series forecasting predicament that this study aims to elucidate. Subsequently, a comprehensive account of the experimental configuration and the experimental methodology is provided.

6.1 Dataset and data preprocessing

The dataset employed in this study is derived from Aneo AS, a Norwegian power producer. The dataset, referred to as the "electricity demand (consumption) dataset", offers a comprehensive representation of the hourly electrical energy demand in various cities within the Nordic countries. The data is employed to predict the projected demand in various cities for the forthcoming 24-hour period, with particular emphasis on the temperature factor. Aneo AS is the operator of multiple wind power plants located in various cities within Norway. Aneo As relies on the observed demand in various cities to provide information to Statnett, the Norwegian state-owned enterprise, regarding the potential energy contribution to the Nordic power line.

The dataset comprises hourly data, temperature data, and location (city) information. Each row in the dataset contains a timestamp indicating the date and hour, the temperature recorded at that specific time, and details regarding the corresponding location. The dependent variable within the dataset pertains to the electricity demand for a specific hour and city, measured in Megawatt-hours (MWh). The temperature is expressed in the Celsius scale. Subsequently, there will be further supplementary features that will be elaborated upon in subsequent sections.

6.1.1 Data preprocessing

The dataset consists of 49,494 rows that correspond to six cities: Bergen, Helsingfors (Helsinki), Oslo, Stavanger, TromsÃ, and Trondheim. With the exception of Helsinki, all the aforementioned locations are situated in Norway. However, it is important to note that the data pertains to a specific group of clients within these cities, rather than representing the overall demand for the entire city. For security

City	Data Count
Bergen	8641
Helsingfors	6289
Oslo	8641
Stavanger	8641
TromsÃ	8641
Trondheim	8641

Table 6.1.1: Data distribution by Cities

Date	Place	Count
2022-04-07	bergen	3
2022-04-07	oslo	3
2022-04-07	stavanger	3
2022-04-07	tromsø	3
2022-04-07	trondheim	3
2022-07-14	helsingfors	3
2023-04-02	bergen	22
2023-04-02	helsingfors	22
2023-04-02	oslo	22
2023-04-02	stavanger	22
2023-04-02	tromsø	22
2023-04-02	trondheim	22

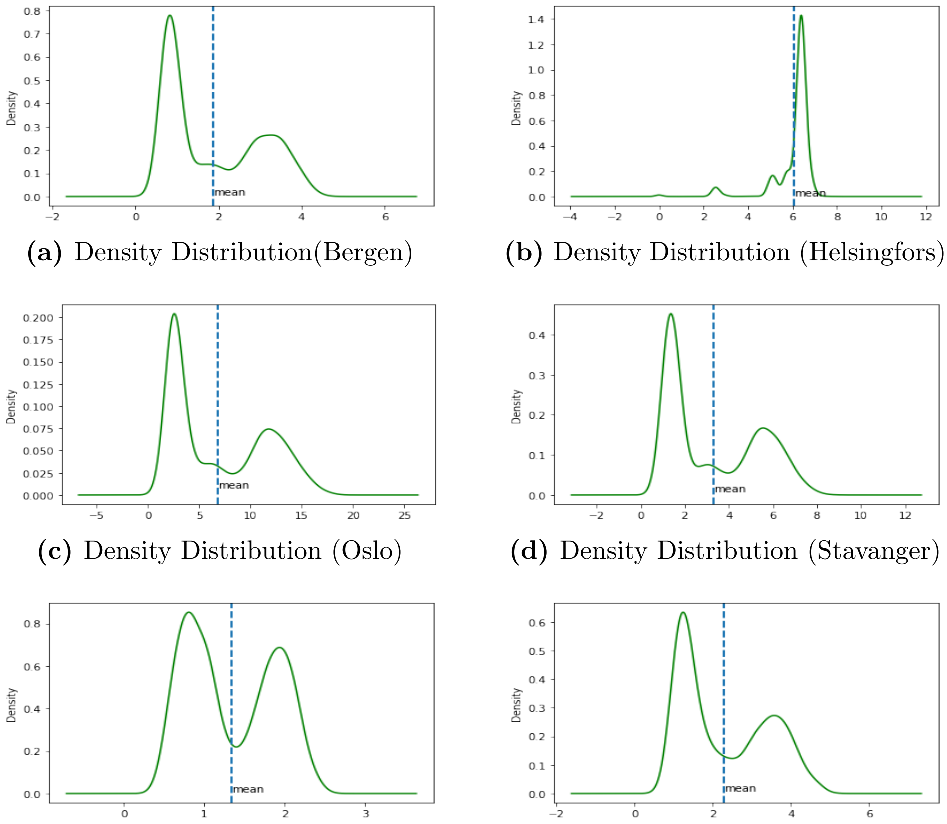
Table 6.1.2: Invalid Data distribution by Cities

purposes, the company name has been substituted with the names of the cities in which they are situated. The dataset encompasses a time period commencing on April 7, 2022, at 21:00 and concluding on April 2, 2023, at 21:00 for all cities, with the exception of Helsingfors. For Helsingfors, the data collection begins on July 14, 2022, at 21:00 and concludes on April 2, 2023, at 21:00. In order to enhance simplicity and optimize training effectiveness, the dataset is partitioned according to the distinct cities, each of which possesses a varying number of records, as outlined in Table 6.1.1. For training and testing split city-based dataset has been split into 80% and 20% margins.

Given the distribution of data based on hours, it is expected that each hour would consist of 24 records. However, Table 6.1.2 reveals discrepancies (Days that have less than 24 records) in this pattern for certain cities. The removal of these records has been undertaken with the aim of enhancing training efficiency.

Various features have been incorporated into the process of feature engineering, including Month, Day, Hour, Business Hour, Season, Weekend, Daylight, and Holiday. Additionally, the Lag feature and baseline target have been incorporated as special features.

The variability in data distribution across different cities is depicted in figure 6.1.1. The findings presented in Chapter 7 exhibit a significant correlation with the distribution of the data, primarily impacting the performance of the models.



(a) Density Distribution(Bergen)

(b) Density Distribution (Helsingfors)

(c) Density Distribution (Oslo)

(d) Density Distribution (Stavanger)

(e) Density Distribution (TromsÃ)

(f) Density Distribution (Trondheim)

Figure 6.1.1: Density distribution of various cities

6.2 Forecasting problem

The present thesis addresses the time series forecasting issue pertaining to the demand forecasting problem encountered by Aneo As. Specifically, the task involves predicting the demand for each hour of the following day, based on data collected at 24-hour intervals. Although the historical data is not accessible within a 24-hour timeframe, this limitation does not affect the interpretation of either the heatmap or the assessment of performance improvement. If t is the current time series then the prediction could be $[t, (t + 24)]$.

6.3 Experimental setup

The programming language utilized for all experimental code is Python. The implementation of the Time series forecasting method and heatmap generation method is based on the architecture depicted in Figure 3.2.7. The code was implemented and validated utilizing Jupyter Notebooks. The versions of each package are indicated in Section 6.4, where the explanation of each method is provided. The LSTM model utilized for prediction has been trained on a Graphics Processing Unit (GPU) specifically the NVIDIA GeForce RTX 3070. The methods were trained and tested using a laptop equipped with an Intel Core i7-10750H CPU.

6.4 Experimental Plan

This section provides a description of the experiments carried out in the course of this study. The process commences with the utilization of a prediction model to forecast the desired outcome and produce a Heatmap, as outlined in Chapter 5. Additionally, a baseline model is employed as a point of comparison for subsequent experiments. Subsequently, the insights gained from XAI are leveraged to enhance the precision of the predictions, as elucidated in Chapter 4. The objective of the experiments is to provide empirical evidence in addressing research question 3 as outlined in Section 1.3, pertaining to the efficacy of the chosen methodology in enhancing the predictive model.

6.4.1 Experimental setup

Elaborating on the experimental pipeline is of paramount importance in order to facilitate the replication of the obtained results.

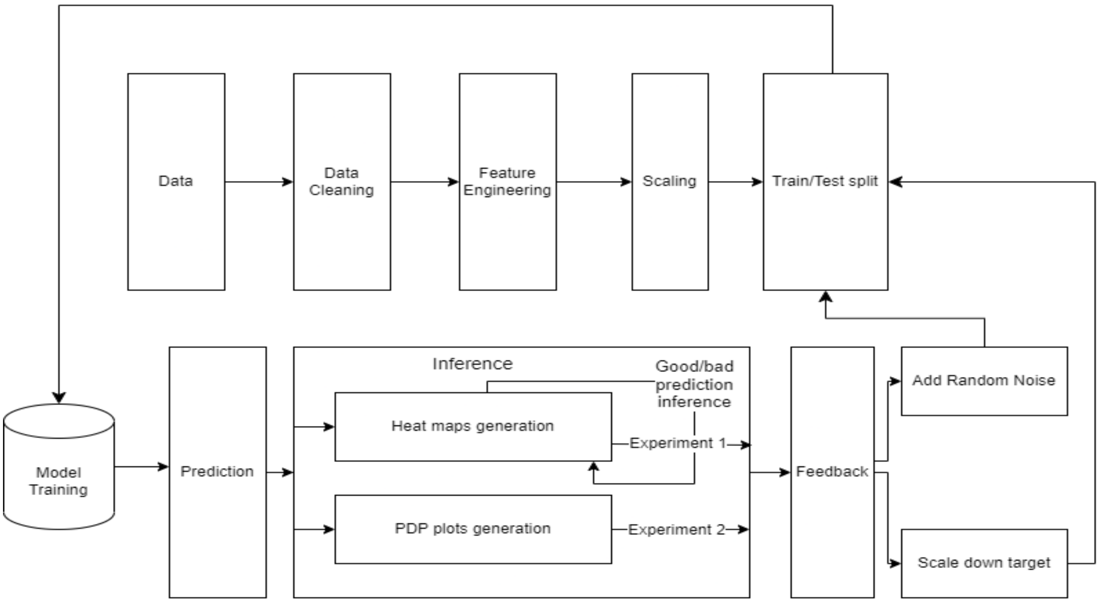


Figure 6.4.1: Experimental pipeline.

As depicted in Figure 6.4.1, the pipeline has been partitioned into two distinct layers. The initial layer denotes the preprocessing steps involved in any machine learning project. Upon receiving a dataset, it is necessary to conduct data cleaning and feature engineering, as elaborated in Section 6.1. In the context of the neural network-based model, the process of scaling plays a crucial role in ensuring effective training and optimal performance [93]. While various scaling algorithms can be employed, we suggest utilizing the Min-Max scaler, which has also been utilized in the present study. In this project, a train/test split of 80-20 has been employed, although other train test sizes could also be considered.

The subsequent layer initiates with the process of model training, where the initial model is referred to as experiment 0. The training data for this model is supplied by the preceding layer. This project has provided a concise overview of three experiments, labeled as Experiment 1, Experiment 2, and Experiment 3 in Section 6.4.5, 6.4.6, and 6.4.7, correspondingly. The experimental model has been addressed in Section 5.2 within the context of this project. The central focus of Experiments 1 and 2 centered on improvements in data augmentation, particularly in the context of heat maps and PDP plots. After the completion of training the model, it becomes capable of making predictions. At this stage, we can utilize the model to generate heatmaps and PDP plots for the purpose of conducting inference. Additionally, the data has been divided into two subsets based on performance to further investigate the limitations of prediction inference. PDP plots have been exclusively employed for the purposes of enhancing models and providing explanations on a feature-by-feature basis. The heat-map technique is commonly employed to generate graphical representations that illustrate the importance of features and data. In the present study, the noise was deliberately incorporated into the training data, and the target variable was subsequently modified to facilitate the analysis of PDP. Following this, the data has been transferred to the train/test pipeline.

6.4.2 Baseline model

The utilization of a baseline model for the purpose of comparing the performance of predictive models is a widely prevalent practice. In more straightforward language, a baseline in forecast performance serves as a reference point for comparison.

The algorithm is designed to make predictions based on the majority class in classification scenarios or the average outcome in regression scenarios. This approach may be applicable for analyzing time series data, but it does not adequately account for such datasets' inherent serial correlation structure. The persistence algorithm is the corresponding technique employed for time series datasets. The persistence algorithm employs the value observed at the preceding time step (t-1) in order to forecast the anticipated outcome at the subsequent time step (t+1). In the context of this particular problem, the expression can be represented as (t-(24 * 7)), where t represents the variable of interest. Exactly one week ago, at the same hour.

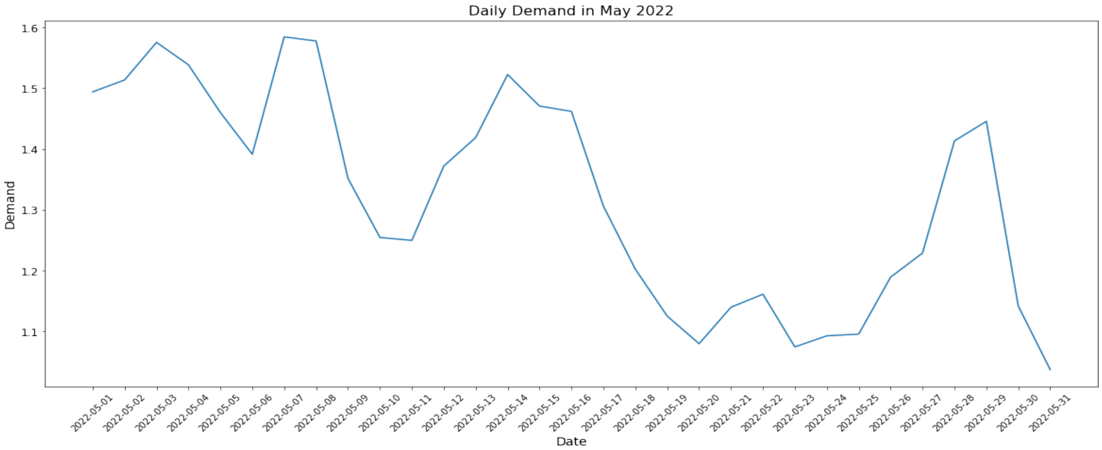


Figure 6.4.2: Daily demand plot; Trondheim city; year 2022; month May.

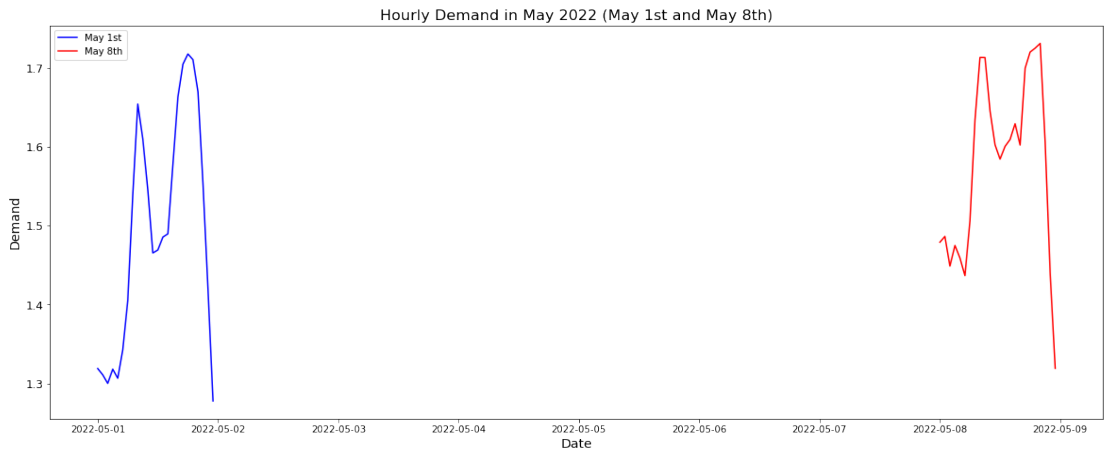


Figure 6.4.3: Weak apart comparison; Trondheim city; year 2022; month May.

The rationale for selecting a reference point from a previous week as a baseline can be confirmed by examining Figures 6.4.2 and 6.4.3. The similarity in trends between the dates of May 1st and May 8th is readily apparent.

6.4.3 Forecasting method with Heat-map

As previously mentioned in Section 5.2, it is imperative that the explanation be capable of being interpreted numerically. Based on the LSTM architecture depicted in Figure 5.2.1, it is plausible to express the generated heatmap in numerical form.

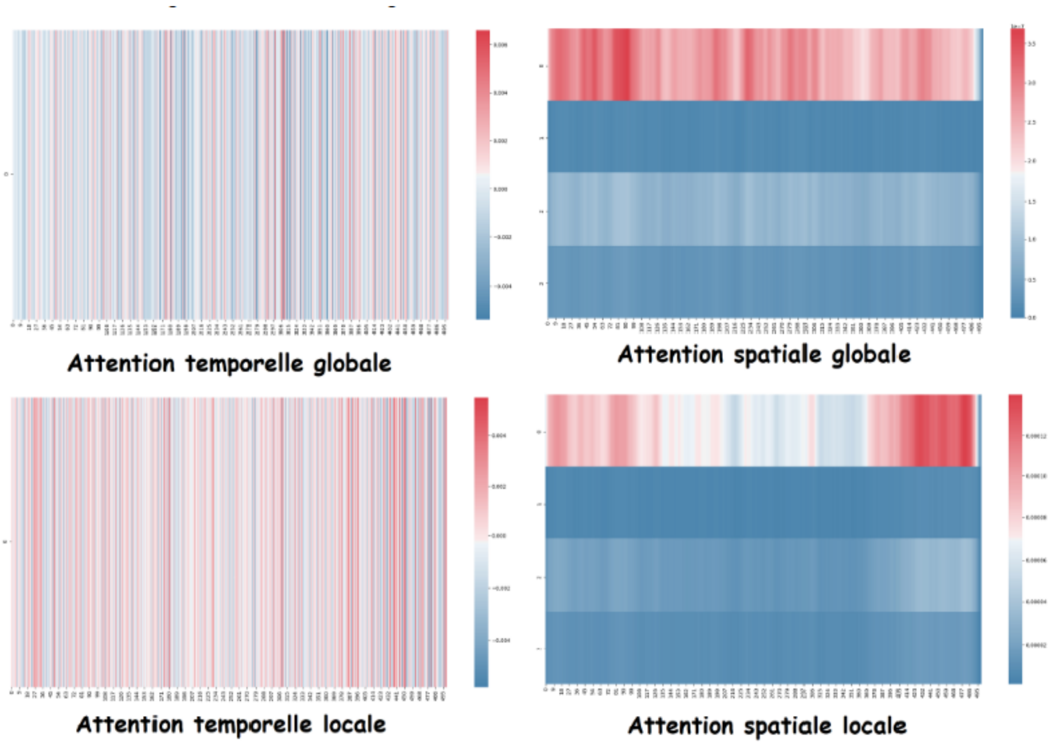


Figure 6.4.4: Heat map generated by LSTM Model. Fig from [90]

Based on the provided image (see Figure 6.4.4), it is evident that certain timelines have been assigned greater significance, both at a local and global level. According to Selvaraju et al [55], the GRAD-CAM technique suggests that regions in the heatmap exhibiting *light blue or dark red coloration may indicate their relevance*. According to this assertion, the model can be enhanced by employing the Data-Augmentation technique discussed in Chapter 4.

6.4.4 XAI based improvement

Each of the experiments will be evaluated using a consistent time window and prediction horizon. However, it is important to note that for every experiment conducted, a novel model will be trained using a modified dataset. As outlined in Section 5.2, our primary emphasis will be on Data augmentation as one of the model improvement techniques examined in Chapter 4.

6.4.5 Experiment 1

In this experimental study, the model undergoes training using the complete dataset, considering the input window as the data from the previous seven days, which corresponds to a total of 24 hours multiplied by 7 days. Furthermore, the model accurately forecasted the anticipated demand for the upcoming 24-hour period. This process is iterated for each day in the test set, ensuring that any potential information leakage is carefully considered. The Mean Absolute Error (MAE) is computed and subsequently compared against both the test and actual data. This experiment was regarded as a foundational study, as we have not yet undertaken any form of model enhancement.

The heatmap derived from this experiment will be utilized in Experiment 2. The model in question will be referred to as the Base Model.

6.4.6 Experiment 2

The prediction window, test matrix, and all other matrices utilized in this experiment are consistent with those employed in Experiment 1. In this experiment, we employed both the MAE metric and the Skill score, as discussed in section 5.3, to assess and compare the performance of various models.

The primary objective is to substitute the information contained within these temporary, less valuable data points with arbitrary noise throughout the training process. Furthermore, it is imperative to preserve the model by incorporating the adjusted data and subsequently evaluating its performance.

6.4.7 Experiment 3

The experiment is based on the principles of Parallel Distributed Processing, as depicted in Figure 7.4.1. In the initial experiment, it was noted from the prediction plot that the model demonstrated a deficiency in accurately capturing the trend for a particular component, as depicted in Figure 7.4.2. Nevertheless, it was successful in accurately capturing the variability linked to this particular component. The majority of observations did not accurately document the declining pattern. A decision was taken to reduce the demand for particular influential feature target variables by employing a data augmentation technique. The objective is to identify the influential feature with a higher demand using PDP and determine its corresponding timeline, such as observing a significant increase in demand during the summer season. The initial idea was to reduce the demand for data augmentation to a specific level, either the minimum or average, in order to enhance the performance of the model.

In this experiment, we employed the same evaluation matrices and configurations that were utilized in previous studies.

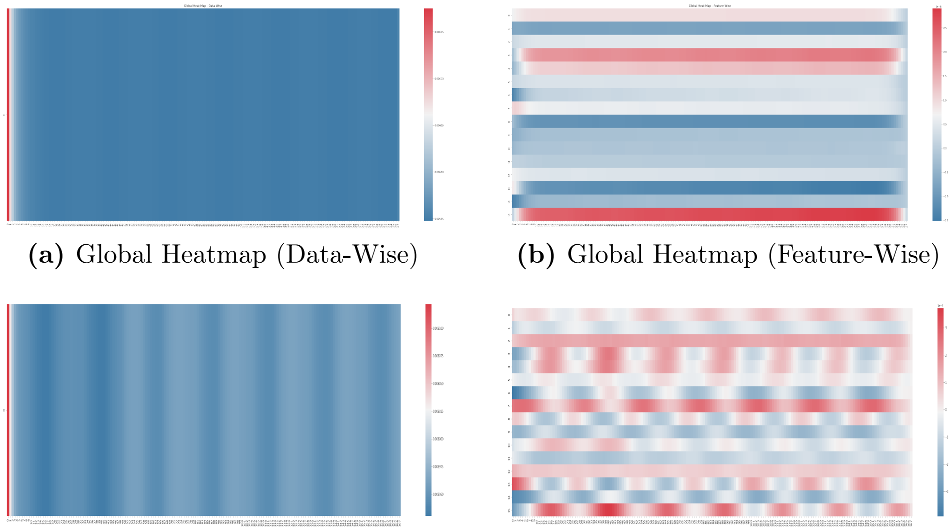
Chapter 7

Results

This chapter will provide an exposition of the findings derived from the analysis that was undertaken. Initially, a concise examination is conducted to assess the predictive performance of the models.

This section primarily presents the findings of the experiments described in Chapter 6. Additionally, statistical test results will be presented in order to evaluate their differences.

7.1 Heat maps



(a) Global Heatmap (Data-Wise)

(b) Global Heatmap (Feature-Wise)

(c) Individual Heatmap (Data-Wise)

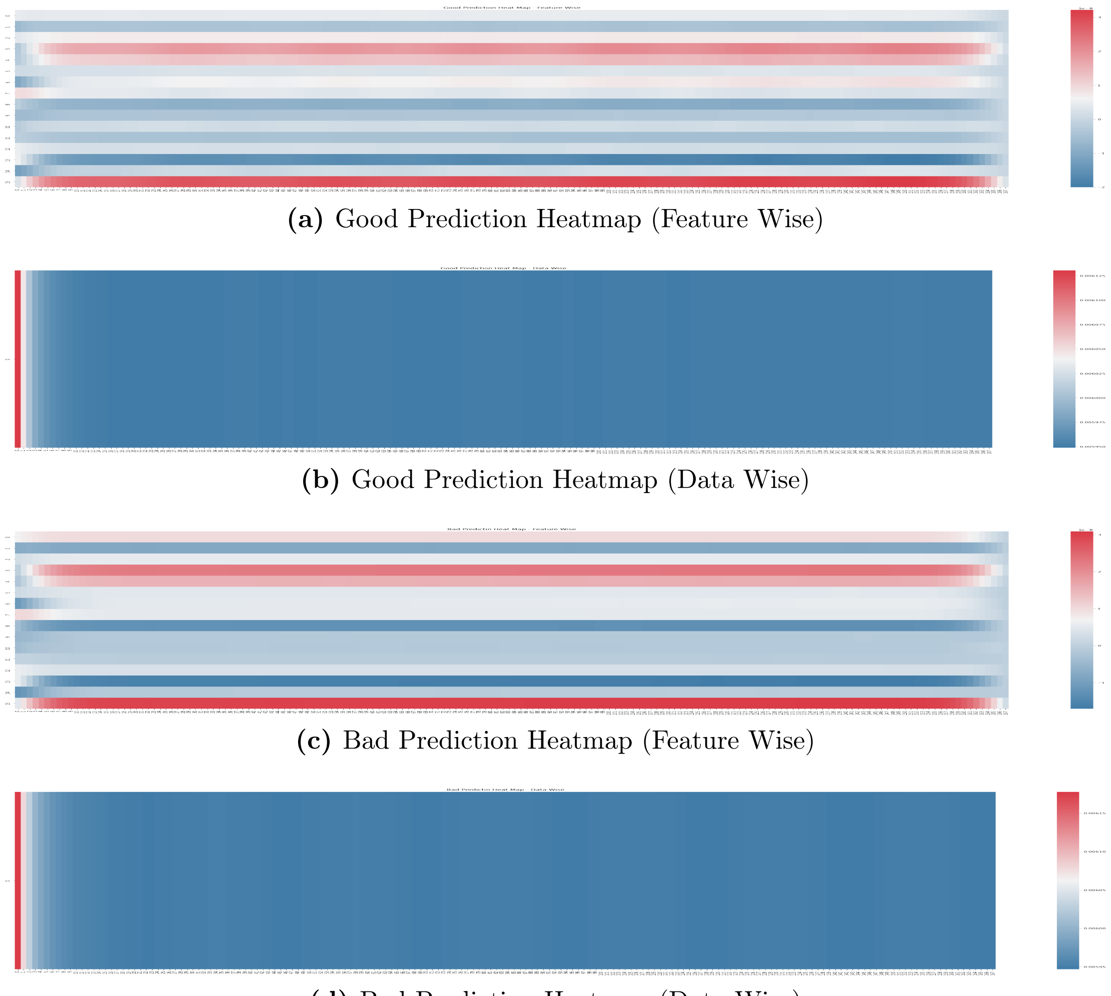
(d) Individual Heatmap (Feature-Wise)

Figure 7.1.1: Base Model Heatmap (Trondheim city)

The heat map generated by the model described in Experiment 1. Heatmaps are produced for both global(The average heat map of every instance) and local(for a single instance) datasets. Furthermore, a heatmap has been generated to visually depict the features and data. The production of the heatmap is limited to the city of Trondheim, as described in Section 6.1, where a city-specific model has been constructed. With respect to the remaining cities, the Heatmap demonstrates a comparable visual depiction, albeit with noticeable numerical discrepancies.

The heat map displayed in this study is the outcome of data generation using a window size of seven days and integrating 16 unique features. According to the findings presented in Section 6.4.3, it can be observed that the colors light blue and dark red exhibit the greatest level of significance. The heatmap facilitates the identification of the features and data points that have been assigned the highest degree of significance. The provided information possesses the potential to be employed for the purpose of data augmentation in subsequent endeavors. To facilitate the execution of experiment 2, it is feasible to introduce extraneous variables into the less consequential data. Based on the depicted diagram, it can be inferred that features 14, 15, 12, 10, 7, 5, 2, and 0 exhibit the greatest significance. Furthermore, the analysis of the 168 input data enables the differentiation between data points of importance and those of lesser significance. However, when comparing the global heatmap to the individual heatmap, it is evident that the former is generally more effective. The frequent and dynamic changes in individual circumstances make it challenging to effectively monitor and interpret them.

7.2 Forecasting error Heat maps



(a) Good Prediction Heatmap (Feature Wise)

(b) Good Prediction Heatmap (Data Wise)

(c) Bad Prediction Heatmap (Feature Wise)

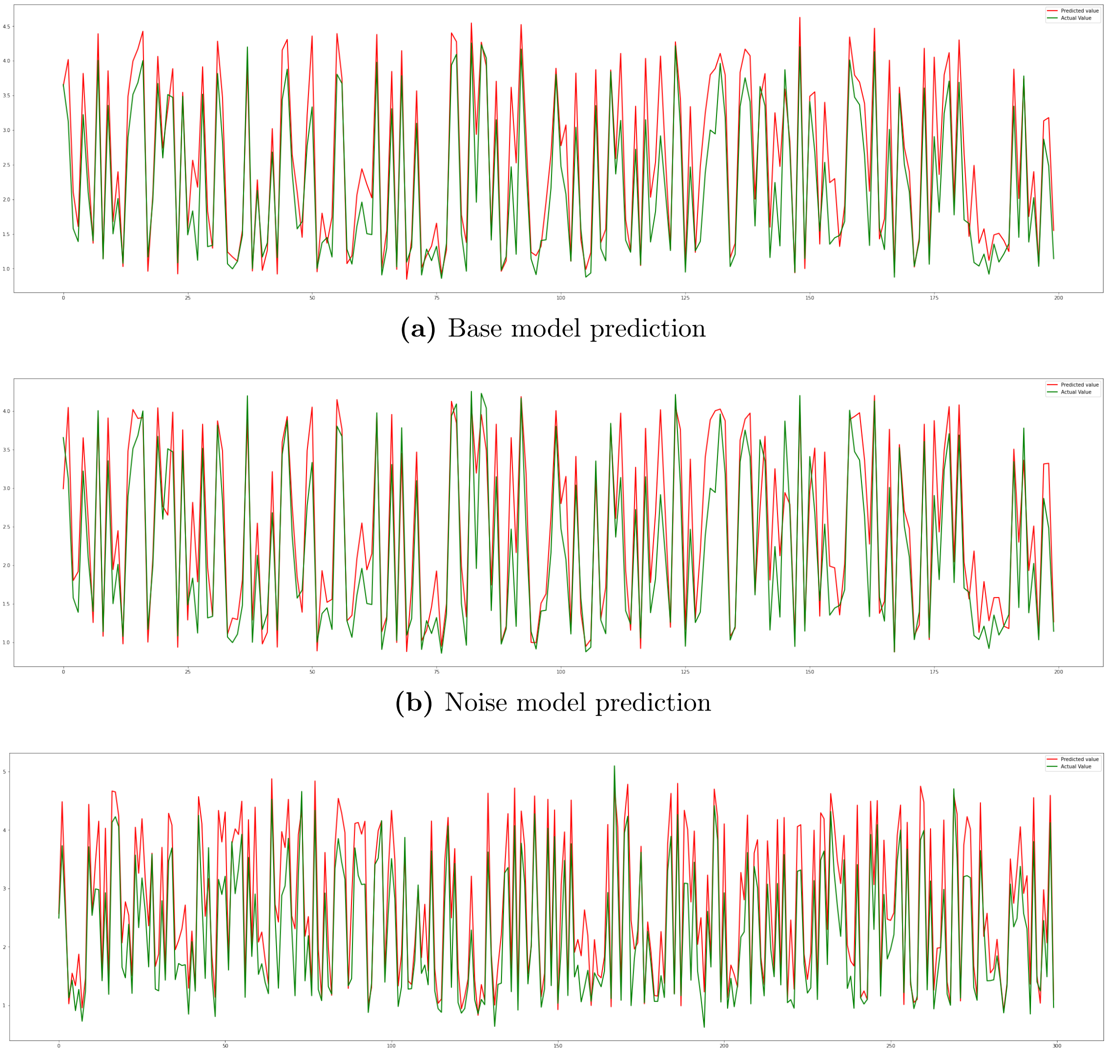
(d) Bad Prediction Heatmap (Data Wise)

Figure 7.2.1: Forecasting error Heatmaps

Figure 7.2.1 visualize the heatmap for both good predictions and bad predictions. The test set has been split into two on the basis of the MAE average and considers the first half as good and the second half as bad.

7.3 Prediction

However, it has been observed that LSTM models exhibit less favorable performance when applied to time series data compared to their performance in linguistic models. However, in order to generate an explanation, we have employed LSTM as a predictive model instead of ARIMA [94] or NeuralProphet [95], among other alternatives.



(a) Base model prediction

(b) Noise model prediction

(c) Low demand model prediction

Figure 7.3.1: Different experimental model prediction

In this context, the models referred to as the Base Model, Noise Model, and Low-demand Model correspond to the models that were trained for Experiments one, two, and three, respectively. While the visual representation is limited to Trondheim City, a comprehensive representation of the data is provided in Table

7.5.1. It is evident that the Noise model has emerged as the superior performer in terms of performance.

7.4 PDP

Experiment 3 utilized the PDP analysis method. Despite its limitations and the lack of comprehensive numerical analysis, the PDP remains a valuable tool for augmenting training data and improving model performance, as discussed in Chapter 3. This discourse aims to examine the data concerning Trondheim City, taking into consideration the potential applicability of the findings to other urban regions.

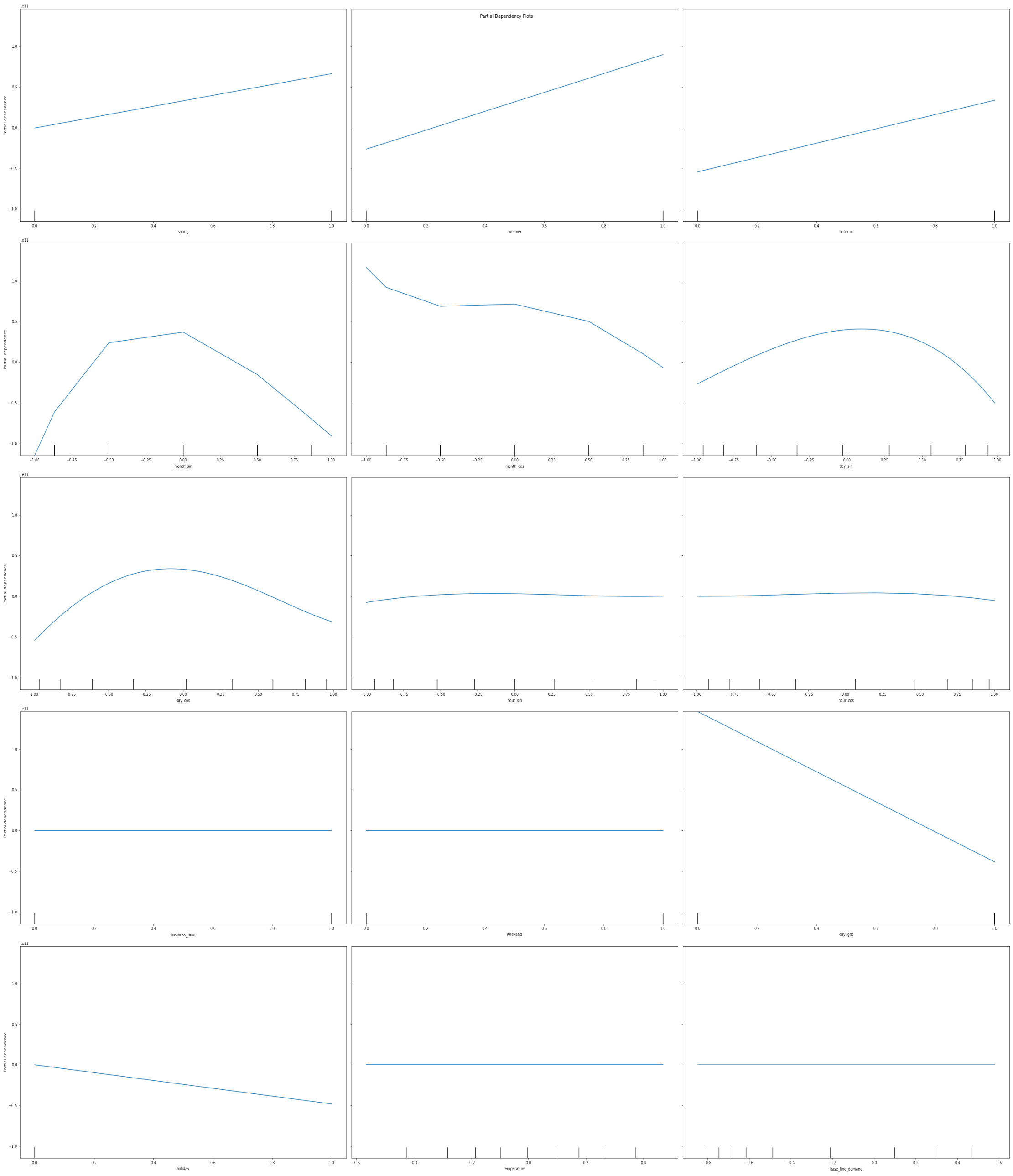


Figure 7.4.1: Partial dependency plot (Trondheim)

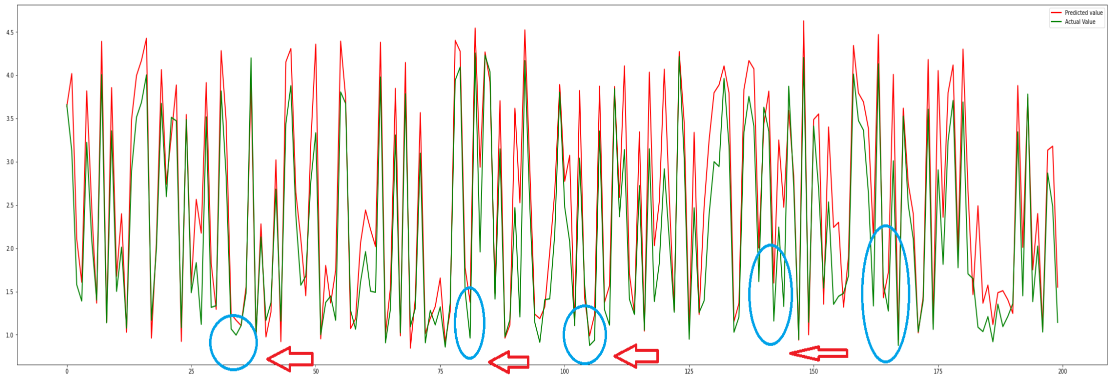


Figure 7.4.2: Base model Prediction (Some missing trend but captures variation)

A total of fifteen plots have been generated, with each plot corresponding to a distinct set of fifteen features. The issue of missing trends is clearly illustrated in Figure 7.4.2. In order to tackle this matter, it is possible to explore alternative methodologies such as feature engineering, utilization of diverse models, or modification of data representation. However, in order to improve the performance of our model using XAI, we made the decision to utilize data augmentation, as shown in Figure 7.4.1. The comprehensive performance is displayed in Table 7.5.1.

7.5 Model Performance

Table 7.5.1 presents the accuracy matrix outcomes for three distinct experiments as outlined in Chapter 6. RMSE, MAE, and R2 scores are provided for six distinct split datasets. During the training process, the RMSE matrix is utilized for both forward and backward propagation. The MAE is employed to assess the performance, while the R2 score is employed to quantify the percentage performance. Figure 7.5.1 illustrates the comparative prediction performance of the three distinct models in relation to the baseline model.

The skill score in comparison to the baseline model has been presented in Table 7.5.2. In order to gain a comprehensive understanding and facilitate visualization of the models that have exhibited strong performance. Figure 7.5.2 illustrates the comparative evaluation of the performance of different experimental models across multiple cities, as measured by their respective R2 scores.

City	Model	RMSE	MAE	R2 Score	K-Fold R2 Score (Average)
Bergen	Base	1.100	0.296	88.0	59.7
Bergen	Noise	1.109	0.295	89.0	69.6
Bergen	Low-Demand	1.103	0.478	75.0	73.9
Helsingfors	Base	0.630	0.287	54.0	42.0
Helsingfors	Noise	0.619	0.258	60.0	33.3
Helsingfors	Low-Demand	0.626	0.273	61.0	65.0
Oslo	Base	4.525	1.742	77.0	46.0
Oslo	Noise	4.474	1.034	92.0	65.6
Oslo	Low-Demand	4.531	1.296	86.0	47.5
Stavanger	Base	2.012	0.592	89.0	72.9
Stavanger	Noise	2.012	0.518	90.0	71.0
Stavanger	Low-Demand	2.026	0.565	88.0	60.7
Tromsø	Base	0.540	0.141	90.0	55.0
Tromsø	Noise	0.534	0.130	91.0	83.4
Tromsø	Low-Demand	0.542	0.149	89.0	67.5
Trondheim	Base	1.081	0.404	79.0	50.9
Trondheim	Noise	1.091	0.377	81.0	66.9
Trondheim	Low-Demand	1.084	0.469	72.0	47.1

Table 7.5.1: Performance of the experiments in various cities.

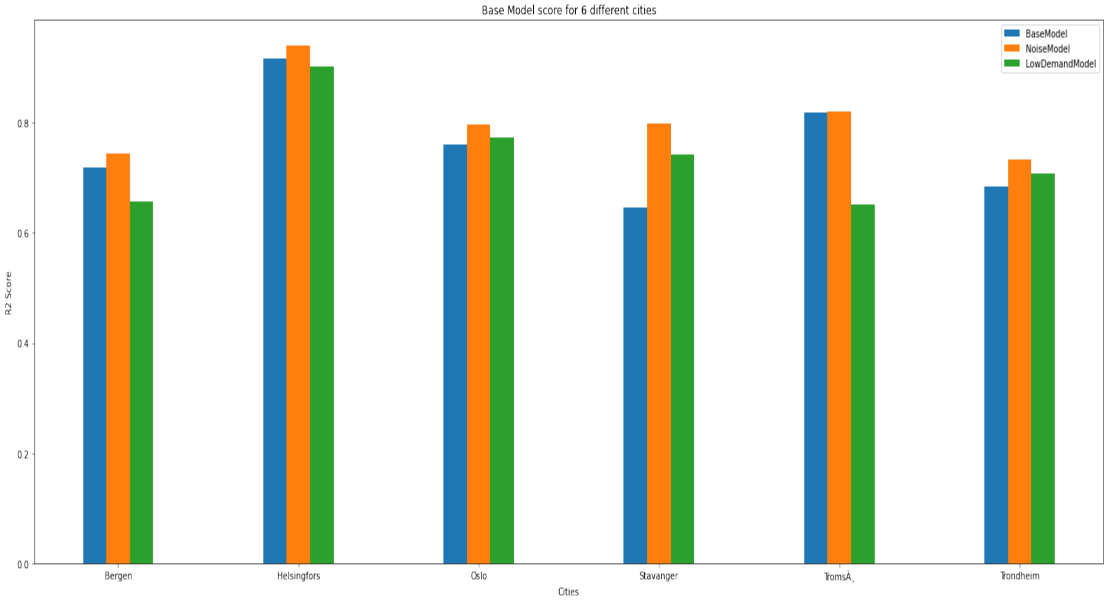
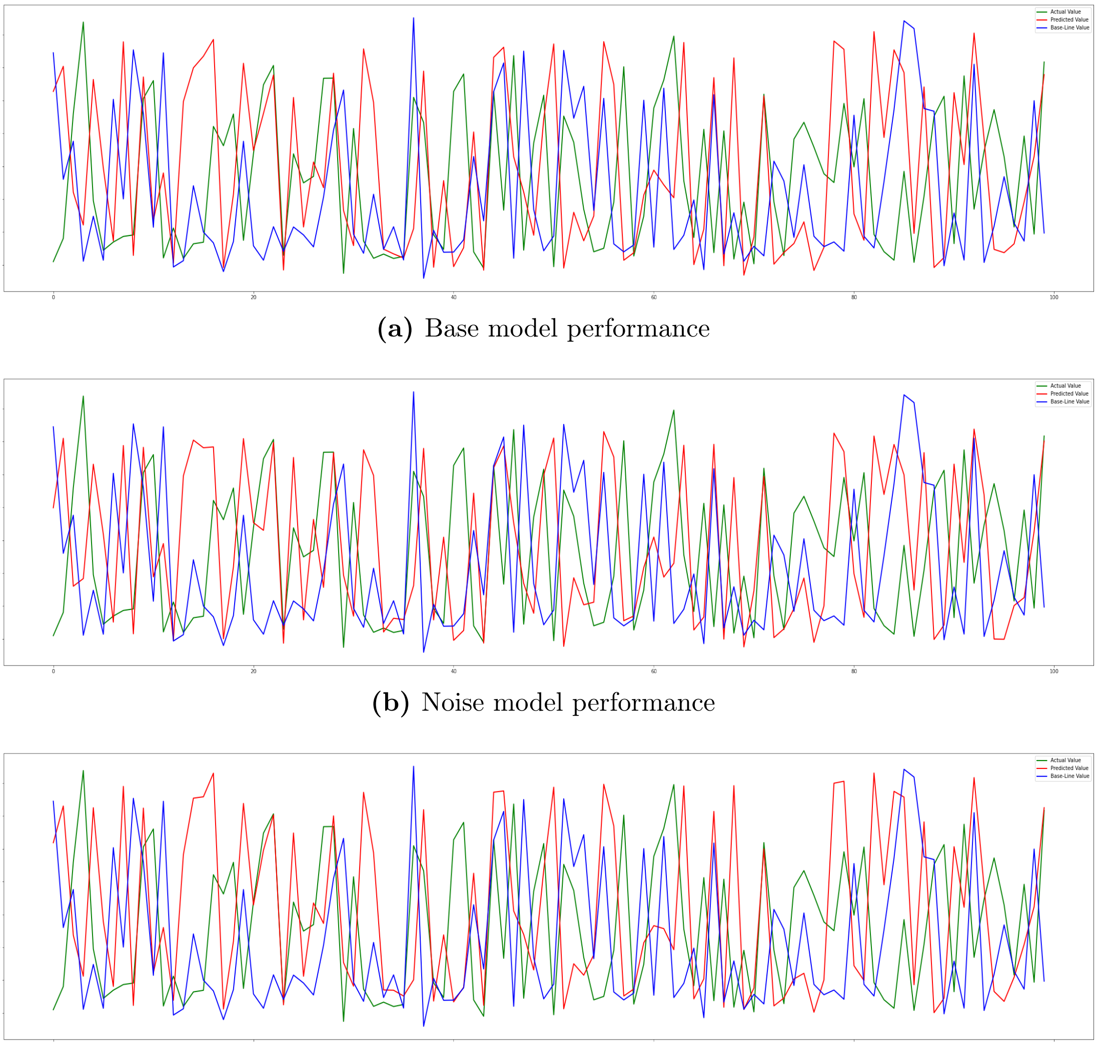


Figure 7.5.2: R2 score for various cities Based on different models



(a) Base model performance

(b) Noise model performance

(c) Low demand model performance

Figure 7.5.1: Varying performance compare to Baseline Model 6.4.2 of different experimental models

Based on the aforementioned data, it is evident that the "Noise" model exhibits the highest R2 scores among the Base, Noise, and Low-Demand models across various cities. This finding highlights the superior predictive accuracy of the "Noise" model, as it achieves an average R2 score of 81.3%. The "Low-Demand" model demonstrates varied outcomes, with an average R2 score of 75.2%, indicating strong performance in certain cities but relatively lower efficacy in others. The performance of the "Base" model exhibits variability, as evidenced by an average R2 score of 80.5%. This suggests that while the model yields satisfactory outcomes in certain cities, it tends to produce higher errors in others. In general, the "Noise" model demonstrates the highest potential for precise prediction across a range of urban datasets, surpassing alternative models in terms of R2 scores and overall accuracy in forecasting.

City	Base	Noise	Low-Demand
Bergen	0.818	0.819	0.650
Helsingfors	0.683	0.732	0.706
Oslo	0.645	0.798	0.742
Stavanger	0.759	0.796	0.773
TromsÃ	0.916	0.938	0.900
Trondheim	0.718	0.743	0.657

Table 7.5.2: Skill scores obtained from the experiments of various cities

The presented table displays skill scores at the city level for the Base, Noise, and Low-Demand models. The models developed by Bergen achieved accuracy scores of 0.818, 0.819, and 0.650. Helsingfors achieved scores of 0.683, 0.732, and 0.706. The Oslo models achieved scores of 0.645, 0.798, and 0.742. The scores obtained by Stavanger were 0.759, 0.796, and 0.773. Trondheim achieved scores of 0.718, 0.743, and 0.657, whereas Tromsø attained scores of 0.916, 0.938, and 0.900. The skill scores provide an assessment of the performance of each model across various cities. In the majority of cities, Noise models exhibit superior skill scores compared to both Base and Low-Demand models. The Noise model exhibits superior performance in the cities of Bergen, Oslo, Stavanger, and Trondheim, thereby showcasing its robustness. The performance of the Base model is superior to that of the Noise model specifically in the Tromsø region, but this discrepancy is not observed in other locations. The models characterized by low demand exhibit the lowest skill scores, yet demonstrate superior performance in the context of Helsingfors. Based on the findings, it can be concluded that the Noise model exhibits the highest level of reliability and adaptability in urban settings. Additional research and careful examination of domain-specific factors may be necessary in order to determine the most suitable model for each individual city.

7.6 statistical significance

Based on the results of the paired t-test given in Table 7.6.1, Model A(Base model) is statistically different from Model B(Noise model) in Bergen, Stavanger, Tromsø, and Trondheim, as shown by the T-values of -1.484, 0.403, -5.522, and -2.293 and the P-values of 0.155, 0.691, 3.041, and 0.033. Model A is also statistically different from Model C(Low-Demand model) in the cities of Helsingfors, Tromsø, and Trondheim, with T-values of -4.360, -2.083, and 0.468 and P-values of 0.000, 0.051, and 0.645, respectively. The T-values of -1.098 and -5.109 and the P-values of 0.286 and 7.333 show that there is no statistically significant difference between Model A and Model C in Bergen and Oslo. Also, there are statistically significant differences between Model B and Model C in all areas, with T-values ranging from 0 to 3.694 and P-values from 0.001 to 0.082. Based on these data, it seems that the performance of Model A is different in different cities, while Models B and C are always different.

City	T Value (A Vs B)	T Value (A Vs C)	T value (B Vs C)	P Value (A Vs B)	P Value (A Vs C)	P value (B Vs C)
Bergen	-1.484	-1.098	0.0	0.155	0.286	1.0
Helsingfors	-4.785	-4.360	1.838	0.000	0.000	0.082
Oslo	-6.314	-5.109	2.200	5.954	7.333	0.041
Stavanger	0.403	2.944	2.094	0.691	0.008	0.050
TromsÃ	-5.522	-2.083	3.694	3.041	0.051	0.001
Trondheim	-2.293	0.468	2.713	0.033	0.645	0.014

Table 7.6.1: Paired t-Test Results

City	**Model**	**Fold 1(R2)**	**Fold 2(R2)**	**Fold 3(R2)**	**Fold 4(R2)**	**Fold 5(R2)**	**Fold 6(R2)**	**Fold 7(R2)**	**Fold 8(R2)**	**Fold 9(R2)**	**Fold 10(R2)**
Bergen	Base	61.0	59.0	59.0	70.0	61.0	55.0	68.0	70.0	77.0	40.0
Bergen	Noise	75.0	63.0	84.0	63.0	68.0	57.9	56.9	61.0	87.0	73.0
Bergen	Low-Demand	72.0	83.0	81.0	79.0	83.0	75.0	27.0	56.9	65.0	67.0
Helsingfors	Base	81.0	44.0	12.0	28.0	86.0	73.0	92.0	24.0	92.0	25.0
Helsingfors	Noise	0.0	72.0	47.0	42.0	24.0	22.0	76.0	58.0	99.0	91.0
Helsingfors	Low-Demand	72.0	74.0	43.0	45.0	73.0	97.0	67.0	20.0	36.0	96.0
Oslo	Base	54.0	16.0	60.0	18.0	81.0	12.0	34.0	37.0	67.0	55.0
Oslo	Noise	47.0	79.0	54.0	69.0	83.0	57.9	86.0	78.0	52.0	50.0
Oslo	Low-Demand	63.0	30.0	49.0	28.9	46.0	62.0	68.0	56.9	61.0	43.0
Stavanger	Base	89.0	68.0	65.0	70.0	80.0	60.0	68.0	75.0	82.0	72.0
Stavanger	Noise	79.0	73.0	75.0	44.0	79.0	61.0	61.0	75.0	81.0	82.0
Stavanger	Low-Demand	72.0	57.9	75.0	63.0	55.0	64.0	67.0	45.0	47.0	61.0
TromsÃ	Base	60.0	43.0	34.0	39.0	52.0	55.0	71.0	56.9	84.0	55.0
TromsÃ	Noise	88.0	86.0	78.0	68.0	85.0	84.0	91.0	83.0	90.0	81.0
TromsÃ	Low-Demand	77.0	60.0	67.0	56.9	60.0	66.0	84.0	81.0	76.0	47.0
Trondheim	Base	39.0	54.0	51.0	27.0	41.0	78.0	74.0	37.0	39.0	69.0
Trondheim	Noise	49.0	74.0	79.0	63.0	73.0	83.0	76.0	51.0	47.0	74.0
Trondheim	Low-Demand	48.0	63.0	14.0	47.0	67.0	56.9	52.0	16.0	42.0	65.0

Table 7.6.2: Cross validation score

Chapter 8

Evaluation and Conclusion

This study investigates five distinct methodologies for augmenting models and diverse strategies for effectively leveraging models. It is crucial to begin by developing a comprehensive comprehension of the notion of model explanations and the need for variations in explanations depending on different target classes. In the following section, we will explore different methodologies that can be utilized to effectively leverage the information provided by XAI in order to improve the model's performance. The advantages and disadvantages of these noble techniques have also been taken into consideration.

In order to perform a comparative analysis of the chosen methodologies, we proceeded with the training of a LSTM model using the dataset provided by Aneo As. Furthermore, statistical significance tests were performed in order to ascertain whether there was any discernible enhancement in the model. The practical implementation of the measures provided empirical evidence for their theoretical characteristics. The achievements of this thesis are concisely summarised through a reexamination of the research inquiries:

8.1 Research questions 1 and 2

This thesis explores the application of both model-agnostic and model-dependent XAI methods. Additionally, there are methodologies that closely align with models based on time series. Time series data poses challenges for methods that incorporate features. Due to the observed correlation patterns exhibited by the data. Methods that are designed to handle feature interaction tend to have longer processing times for data and are often considered to be less reliable. Although the time series version of SHAP demonstrates satisfactory performance, it does not align with the primary objective of this thesis. The utilization of GRAD-CAM effectively fulfills our objectives and is in line with the overarching goal of our thesis. This has the potential to generate importance in terms of both features and data.

While the detection of anomalies and the improvement of model performance using XAI can provide explanations for poor model performance, it's still not clear how well these methods fit with the goals of this thesis. Using the GRAD-CAM heat map, on the other hand, makes it clear how to tell the difference between correct and incorrect estimates.

8.2 Research question 3

Data augmentation is a highly promising technique that has been discussed in this thesis as a means to enhance model performance. This approach is in line with the objectives of the thesis and is also aligned with the time constraints that have been imposed. By integrating data argumentation techniques with GRAD-CAM explanation, we have achieved significant enhancements in the performance of the model. We have seen 0.122%, 7.19%, 23.72%, 4.88%, 2.40%, and 3.48% improvement across Bergen, Helsingfors, Oslo, Stavanger, Tromsø, and Trondheim respectively. The overall noise model is 6.90% better compare to the base model. The utilization of the PDP plot technique has proven to be effective in enhancing the performance of our model. However, it should be noted that this method does not align with the objectives outlined in our second research question.

Based on the conducted statistical analysis and subsequent cross-validation, it can be concluded that the p-values for all cities are found to be less than the significance level of 0.05. This indicates that there exists sufficient evidence to reject the null hypothesis. Therefore, it can be concluded that there is a significant distinction between Model B and Model C across all cities. However, it is important to acknowledge that the use of different data subsets for cross-validation could potentially influence the results observed in null hypothesis analysis. Nevertheless, it is crucial to undertake comprehensive research or surveys to conclusively establish this assertion.

8.3 Limitation

There exists variation in the quantity of data points across different cities. Despite the existence of various trained models, it has been noted that the data distribution deviates from normality. The factor mentioned above has implications for both the statistical analysis and the performance of cross-validation.

This thesis employed a LSTM based model in conjunction with GRAD-CAM. However, alternative models such as Variational Neural Networks (VNN), CNNs, Bidirectional LSTMs (BiLSTM), and various other variations of Neural Network (NN) could have been considered and investigated. Furthermore, potential improvements in performance could have been achieved by modifying and fine-tuning the hidden layers of the LSTM, as well as optimizing and adjusting the loss function.

The model's performance is not optimal; however, it is widely acknowledged within the scientific community that the inclusion of LSTM and XAI techniques can lead to a decrease in model performance. However, there has been a lack of emphasis on parameter optimization as a means to enhance the performance of the model. Additionally, it should be noted that the dataset provided lacks sufficient features to capture the entirety of the variance and enable comprehensive generalization.

Merely relying on date-time and weather variables is insufficient for accurate prediction.

The objective of this research was to employ XAI techniques in order to enhance the performance of the model from a time-series perspective. In the theoretical framework, several techniques for enhancing the model have been examined. However, due to constraints on time, not all of these methods were implemented.

8.4 Future Work

The extension of this project could involve an exploration of various improvement methods that have been discussed in the research, such as loss function-based and feature augmentation-based methods. One of the primary objectives at the outset of this thesis was the development of a robust methodology for analyzing time series data.

One proposed approach involved the utilization of K-Nearest Neighbors (KNN) clustering algorithms to identify clusters based on given input and assess the distance of their target values by employing any suitable distance-measuring algorithm. The hypothesis is that data points within the same cluster will exhibit similar behavior/target values. If not, the data could be changed based on real-world findings and advice from experts in the field to make the model work better. However, it should be noted that the presence of the same cluster does not necessarily imply similar targets, particularly in the context of time-series data, which introduces additional complexities. It is necessary to take into account the historical value effect, window effect, and other properties associated with time series data.

Bibliography

[1] *The state of AI in 2022—and a half decade in review | McKinsey*. URL: https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai-in-2022-and-a-half-decade-in-review (visited on 06/23/2023).

[2] *Generative AI Could Raise Global GDP by 7%*. en-US. June 2023. URL: https://www.goldmansachs.com/intelligence/pages/generative-ai-could-raise-global-gdp-by-7-percent.html (visited on 06/25/2023).

[3] Michael Chui. "Artificial intelligence the next digital frontier". In: *McKinsey and Company Global Institute* 47.3.6 (2017).

[4] Rob Kitchin. "Thinking critically about and researching algorithms". en. In: *Information, Communication & Society* 20.1 (Jan. 2017), pp. 14–29. ISSN: 1369-118X, 1468-4462. DOI: 10.1080/1369118X.2016.1154087. URL: https://www.tandfonline.com/doi/full/10.1080/1369118X.2016.1154087 (visited on 06/23/2023).

[5] Min Kyung Lee. "Understanding perception of algorithmic decisions: Fairness, trust, and emotion in response to algorithmic management". In: *Big Data & Society* 5 (June 2018), p. 205395171875668. DOI: 10.1177/2053951718756684.

[6] *Implementing Ethics in AI: Initial Results of an Industrial Multiple Case Study*. en. URL: https://www.springerprofessional.de/en/implementing-ethics-in-ai-initial-results-of-an-industrial-multi/17398510 (visited on 06/23/2023).

[7] Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller. *Explainable Artificial Intelligence: Understanding, Visualizing and Interpreting Deep Learning Models*. arXiv:1708.08296 [cs, stat]. Aug. 2017. DOI: 10.48550/arXiv.1708.08296. URL: http://arxiv.org/abs/1708.08296 (visited on 06/25/2023).

[8] Krishna Gade et al. "Explainable AI in industry: Practical challenges and lessons learned". In: *Companion Proceedings of the Web Conference 2020*. 2020, pp. 303–304.

[9] *What is Explainable AI (XAI)? | Definition from WhatIs*. en. URL: https://www.techtarget.com/whatis/definition/explainable-AI-XAI (visited on 06/23/2023).

[10] Thomas Rojat et al. "Explainable artificial intelligence (xai) on timeseries data: A survey". In: *arXiv preprint arXiv:2104.00950* (2021).

[11] *Time Series Data - an overview | ScienceDirect Topics.* URL: https://www.sciencedirect.com/topics/computer-science/time-series-data (visited on 06/23/2023).

[12] Seyed Amirhossein Hosseini and Omar Smadi. "How prediction accuracy can affect the decision-making process in pavement management system". In: *Infrastructures* 6.2 (2021), p. 28.

[13] Michael Lent, William Fisher, and Michael Mancuso. "An Explainable Artificial Intelligence System for Small-unit Tactical Behavior." In: Jan. 2004, pp. 900–907.

[14] Moritz Hardt, Eric Price, and Nati Srebro. "Equality of opportunity in supervised learning". In: *Advances in neural information processing systems* 29 (2016).

[15] Finale Doshi-Velez and Been Kim. "Towards a rigorous science of interpretable machine learning". In: *arXiv preprint arXiv:1702.08608* (2017).

[16] Zachary C Lipton. "The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery." In: *Queue* 16.3 (2018), pp. 31–57.

[17] Tim Miller. "Explanation in artificial intelligence: Insights from the social sciences". In: *Artificial intelligence* 267 (2019), pp. 1–38.

[18] *China Social Credit System Explained - How It Works [2023].* en-US. Running Time: 487 Section: Local Knowledge. June 2022. URL: https://nhglobalpartners.com/china-social-credit-system-explained/ (visited on 06/26/2023).

[19] Emre Ates et al. "Counterfactual explanations for multivariate time series". In: *2021 International Conference on Applied Artificial Intelligence (ICAPAI).* IEEE. 2021, pp. 1–8.

[20] Batta Mahesh. *Machine Learning Algorithms -A Review.* Jan. 2019. DOI: 10.21275/ART20203995.

[21] *What Is Time-Series Forecasting?* en. Sept. 2022. URL: https://www.timescale.com/blog/what-is-time-series-forecasting/ (visited on 06/27/2023).

[22] Ratnadip Adhikari and Ramesh K Agrawal. "An introductory study on time series modeling and forecasting". In: *arXiv preprint arXiv:1302.6613* (2013).

[23] George EP Box et al. *Time series analysis: forecasting and control.* John Wiley & Sons, 2015.

[24] Keith W Hipel and A Ian McLeod. *Time series modelling of water resources and environmental systems.* Elsevier, 1994.

[25] Maria Elena Nor et al. "Neural network versus classical time series forecasting models". In: *AIP Conference Proceedings.* Vol. 1842. 1. AIP Publishing. 2017.

[26] Werner Kristjanpoller and Marcel C Minutolo. "Gold price volatility: A forecasting approach using the Artificial Neural Network–GARCH model". In: *Expert systems with applications* 42.20 (2015), pp. 7245–7251.

[27] Shamsul Masum, Ying Liu, and John Chiverton. “Multi-step time series forecasting of electric load using machine learning models”. In: *Artificial Intelligence and Soft Computing: 17th International Conference, ICAISC 2018, Zakopane, Poland, June 3-7, 2018, Proceedings, Part I 17*. Springer. 2018, pp. 148–159.

[28] Paulin Coulibaly and Connely K Baldwin. “Nonstationary hydrological time series forecasting using nonlinear dynamic methods”. In: *Journal of Hydrology* 307.1-4 (2005), pp. 164–174.

[29] Bibhuti Bhusan Sahoo et al. “Long short-term memory (LSTM) recurrent neural network for low-flow hydrological time series forecasting”. In: *Acta Geophysica* 67.5 (2019), pp. 1471–1481.

[30] Jon Scott Armstrong. *Principles of forecasting: a handbook for researchers and practitioners*. Vol. 30. Springer, 2001.

[31] Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

[32] Jimeng Shi, Mahek Jain, and Giri Narasimhan. “Time Series Forecasting (TSF) Using Various Deep Learning Models”. en. In: ().

[33] Yoshua Bengio, Yann Lecun, and Geoffrey Hinton. “Deep learning for AI”. In: *Communications of the ACM* 64.7 (2021), pp. 58–65.

[34] Phong Le and Willem Zuidema. “Quantifying the vanishing gradient and long distance dependency problem in recursive neural networks and recursive LSTMs”. In: *arXiv preprint arXiv:1603.00423* (2016).

[35] Sepp Hochreiter and Jürgen Schmidhuber. “Long short-term memory”. In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[36] *LSTM | Introduction to LSTM | Long Short Term Memory Algorithms*. URL: `https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/` (visited on 07/18/2023).

[37] Pranj52 Srivastava. *Essentials of Deep Learning : Introduction to Long Short Term Memory*. en. Dec. 2017. URL: `https://www.analyticsvidhya.com/blog/2017/12/fundamentals-of-deep-learning-introduction-to-lstm/` (visited on 07/18/2023).

[38] Joos Korstanje. *How to Select a Model For Your Time Series Prediction Task [Guide]*. en-US. Sept. 2022. URL: `https://neptune.ai/blog/select-model-for-time-series-prediction-task` (visited on 07/18/2023).

[39] *Time Series Forecasting And Machine Learning*. en-US. Section: Machine Learning. Nov. 2022. URL: `https://dataconomy.com/2022/11/25/time-series-forecasting-machine-learning/` (visited on 06/27/2023).

[40] Robert H. Shumway and David S. Stoffer. *Time series analysis and its applications: with R examples*. eng. Fourth edition. Springer texts in statistics. OCLC: 966563984. Cham, Switzerland: Springer, 2017. ISBN: 978-3-319-52451-1.

[41] Academic Accelerator. *Forecast Error: The Most Up-to-Date Encyclopedia, News, Review & Research*. en. URL: `https://academic-accelerator.com/encyclopedia/forecast-error` (visited on 06/27/2023).

[42] Douglas C Montgomery, Cheryl L Jennings, and Murat Kulahci. *Introduction to time series analysis and forecasting.* John Wiley & Sons, 2015.

[43] Konstantin Rink. *Time Series Forecast Error Metrics you should know.* en. Nov. 2021. URL: `https://towardsdatascience.com/time-series-forecast-error-metrics-you-should-know-cc88b8c67f27` (visited on 06/27/2023).

[44] *(12) Explainable AI (XAI) Market by Global Demand, Latest Technology and Precise Outlook 2021 to 2030 | LinkedIn.* URL: `https://www.linkedin.com/pulse/explainable-ai-xai-market-global-demand-/` (visited on 06/29/2023).

[45] *Explainable Artificial Intelligence - an overview | ScienceDirect Topics.* URL: `https://www.sciencedirect.com/topics/computer-science/explainable-artificial-intelligence` (visited on 06/29/2023).

[46] Nickil Maveli. *Demystifying Post-hoc Explainability for ML models.* en-US. URL: `https://spectra.mathpix.com/article/2021.09.00007/demystify-post-hoc-explainability` (visited on 06/29/2023).

[47] Waddah Saeed and Christian Omlin. "Explainable AI (XAI): A systematic meta-survey of current challenges and future opportunities". en. In: *Knowledge-Based Systems* 263 (Mar. 2023), p. 110273. ISSN: 0950-7051. DOI: `10.1016/j.knosys.2023.110273`. URL: `https://www.sciencedirect.com/science/article/pii/S0950705123000230` (visited on 06/29/2023).

[48] Benedikt Leichtmann et al. "Effects of Explainable Artificial Intelligence on trust and human behavior in a high-risk decision task". en. In: *Computers in Human Behavior* 139 (Feb. 2023), p. 107539. ISSN: 0747-5632. DOI: `10.1016/j.chb.2022.107539`. URL: `https://www.sciencedirect.com/science/article/pii/S0747563222003594` (visited on 06/29/2023).

[49] Matthew D. Zeiler and Rob Fergus. *Visualizing and Understanding Convolutional Networks.* arXiv:1311.2901 [cs]. Nov. 2013. DOI: `10.48550/arXiv.1311.2901`. URL: `http://arxiv.org/abs/1311.2901` (visited on 06/29/2023).

[50] Matthew D. Zeiler, Graham W. Taylor, and Rob Fergus. "Adaptive deconvolutional networks for mid and high level feature learning". In: *2011 International Conference on Computer Vision.* ISSN: 2380-7504. Nov. 2011, pp. 2018–2025. DOI: `10.1109/ICCV.2011.6126474`.

[51] *Introduction to Deep Learning: What Are Convolutional Neural Networks? Video.* en. URL: `https://se.mathworks.com/videos/introduction-to-deep-learning-what-are-convolutional-neural-networks--1489512765771.html` (visited on 07/19/2023).

[52] G. Roshan Lal. *Explainable Neural Networks: Recent Advancements, Part 2.* en. Feb. 2021. URL: `https://towardsdatascience.com/explainable-neural-networks-recent-advancements-part-2-8cce67833ba` (visited on 06/29/2023).

[53] Jost Tobias Springenberg et al. *Striving for Simplicity: The All Convolutional Net.* arXiv:1412.6806 [cs]. Apr. 2015. DOI: `10.48550/arXiv.1412.6806`. URL: `http://arxiv.org/abs/1412.6806` (visited on 06/30/2023).

[54] Bolei Zhou et al. *Learning Deep Features for Discriminative Localization.* arXiv:1512.04150 [cs]. Dec. 2015. DOI: 10.48550/arXiv.1512.04150. URL: http://arxiv.org/abs/1512.04150 (visited on 06/30/2023).

[55] Ramprasaath R. Selvaraju et al. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization". In: *International Journal of Computer Vision* 128.2 (Feb. 2020). arXiv:1610.02391 [cs], pp. 336–359. ISSN: 0920-5691, 1573-1405. DOI: 10.1007/s11263-019-01228-7. URL: http://arxiv.org/abs/1610.02391 (visited on 06/30/2023).

[56] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. *Learning Important Features Through Propagating Activation Differences.* arXiv:1704.02685 [cs]. Oct. 2019. DOI: 10.48550/arXiv.1704.02685. URL: http://arxiv.org/abs/1704.02685 (visited on 06/30/2023).

[57] Muhammad Sabih, Frank Hannig, and Juergen Teich. "Utilizing explainable AI for quantization and pruning of deep neural networks". In: *arXiv preprint arXiv:2008.09072* (2020).

[58] Conor O'Sullivan. *What are Model Agnostic Methods?* en. Mar. 2023. URL: https://towardsdatascience.com/what-are-model-agnostic-methods-387b0e8441ef (visited on 06/29/2023).

[59] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Model-agnostic interpretability of machine learning". In: *arXiv preprint arXiv:1606.05386* (2016).

[60] Christoph Molnar. *Interpretable Machine Learning.* URL: https://christophm.github.io/interpretable-ml-book/ (visited on 06/30/2023).

[61] *Interpreting Machine Learning Models Part 1: Accumulated Local Effects - ENJINE.* URL: https://www.enjine.com/blog/interpreting-machine-learning-models-accumulated-local-effects/ (visited on 07/01/2023).

[62] Leo Breiman. "Random forests". In: *Machine learning* 45 (2001), pp. 5–32.

[63] Aaron Fisher, Cynthia Rudin, and Francesca Dominici. *All Models are Wrong, but Many are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously.* arXiv:1801.01489 [stat]. Dec. 2019. DOI: 10.48550/arXiv.1801.01489. URL: http://arxiv.org/abs/1801.01489 (visited on 07/02/2023).

[64] Alex Goldstein et al. "Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation". In: *journal of Computational and Graphical Statistics* 24.1 (2015), pp. 44–65.

[65] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "" Why should i trust you?" Explaining the predictions of any classifier". In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining.* 2016, pp. 1135–1144.

[66] Sandra Wachter, Brent Mittelstadt, and Chris Russell. "Counterfactual explanations without opening the black box: Automated decisions and the GDPR". In: *Harv. JL & Tech.* 31 (2017), p. 841.

[67] Susanne Dandl et al. “Multi-objective counterfactual explanations”. In: *International Conference on Parallel Problem Solving from Nature*. Springer. 2020, pp. 448–469.

[68] Kalyanmoy Deb et al. “A fast and elitist multiobjective genetic algorithm: NSGA-II”. In: *IEEE transactions on evolutionary computation* 6.2 (2002), pp. 182–197.

[69] Scott M Lundberg and Su-In Lee. “A unified approach to interpreting model predictions”. In: *Advances in neural information processing systems* 30 (2017).

[70] João Bento et al. “TimeSHAP: Explaining Recurrent Models through Sequence Perturbations”. In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. arXiv:2012.00073 [cs]. Aug. 2021, pp. 2565–2573. DOI: `10.1145/3447548.3467166`. URL: `http://arxiv.org/abs/2012.00073` (visited on 07/02/2023).

[71] Joao Bento. *TimeSHAP: Explaining recurrent models through sequence perturbations*. en. Jan. 2023. URL: `https://medium.com/feedzaitech/timeshap-explaining-recurrent-models-through-sequence-perturbations-41f2324bfe5f` (visited on 07/02/2023).

[72] Jože Rožanec et al. “Explaining bad forecasts in global time series models”. In: *Applied Sciences* 11.19 (2021), p. 9243.

[73] Leander Weber et al. “Beyond explaining: Opportunities and challenges of XAI-based model improvement”. en. In: *Information Fusion* 92 (Apr. 2023), pp. 154–176. ISSN: 1566-2535. DOI: `10.1016/j.inffus.2022.11.013`. URL: `https://www.sciencedirect.com/science/article/pii/S1566253522002238` (visited on 02/11/2023).

[74] Philip Tannor. *Data Drift vs. Concept Drift*. en. Oct. 2021. URL: `https://deepchecks.com/data-drift-vs-concept-drift-what-are-the-main-differences/` (visited on 07/19/2023).

[75] Stefano Teso and Kristian Kersting. “Explanatory interactive machine learning”. In: *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. 2019, pp. 239–245.

[76] Patrick Schramowski et al. “Making deep neural networks right for the right scientific reasons by interacting with their explanations”. In: *Nature Machine Intelligence* 2.8 (2020), pp. 476–486.

[77] Srishti Gautam et al. “This looks more like that: Enhancing self-explaining models by prototypical relevance propagation”. In: *Pattern Recognition* 136 (2023), p. 109172.

[78] Sebastian Bach et al. “On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation”. In: *PloS one* 10.7 (2015), e0130140.

[79] Dwarikanath Mahapatra et al. “Interpretability-driven sample selection using self supervised learning for disease classification and segmentation”. In: *IEEE transactions on medical imaging* 40.10 (2021), pp. 2548–2562.

[80] Leander Weber. "Towards a more refined training process for neural networks: Applying layer-wise relevance propagation to understand and improve classification performance on imbalanced datasets". In: *Technische Universität Berlin* (2020).

[81] Hiroshi Fukui et al. "Attention branch network: Learning of attention mechanism for visual explanation". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.* 2019, pp. 10705–10714.

[82] Bolei Zhou et al. "Learning deep features for discriminative localization". In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2016, pp. 2921–2929.

[83] Andrea Zunino et al. "Excitation dropout: Encouraging plasticity in deep neural networks". In: *International Journal of Computer Vision* 129 (2021), pp. 1139–1152.

[84] Christopher J Anders et al. "Finding and removing Clever Hans: using explanation methods to debug and improve deep models". In: *Information Fusion* 77 (2022), pp. 261–295.

[85] Andrew Slavin Ross, Michael C Hughes, and Finale Doshi-Velez. "Right for the right reasons: Training differentiable models by constraining their explanations". In: *arXiv preprint arXiv:1703.03717* (2017).

[86] Frederick Liu and Besim Avci. "Incorporating priors with feature attribution on text classification". In: *arXiv preprint arXiv:1906.08286* (2019).

[87] Gabriel Erion et al. "Improving performance of deep learning models with axiomatic attribution priors and expected gradients". In: *Nature machine intelligence* 3.7 (2021), pp. 620–631.

[88] Mengnan Du et al. "Learning credible deep neural networks with rationale regularization". In: *2019 IEEE International Conference on Data Mining (ICDM).* IEEE. 2019, pp. 150–159.

[89] Seul-Ki Yeom et al. "Pruning by explaining: A novel criterion for deep neural network pruning". In: *Pattern Recognition* 115 (2021), p. 107899.

[90] Cedric Schockaert, Reinhard Leperlier, and Assaad Moawad. *Attention Mechanism for Multivariate Time Series Recurrent Model Interpretability Applied to the Ironmaking Industry.* arXiv:2007.12617 [cs]. July 2020. DOI: 10.48550/arXiv.2007.12617. URL: http://arxiv.org/abs/2007.12617 (visited on 04/12/2023).

[91] Jason Brownlee. *How to Make Baseline Predictions for Time Series Forecasting with Python.* en-US. Dec. 2016. URL: https://machinelearningmastery.com/persistence-time-series-forecasting-with-python/ (visited on 07/06/2023).

[92] *Skill Scores.* URL: https://resources.eumetrain.org/data/4/451/english/msg/ver_cont_var/uos5/uos5_ko1.htm (visited on 07/06/2023).

[93] Yann LeCun et al. "Efficient backprop". In: *Neural networks: Tricks of the trade.* Springer, 2002, pp. 9–50.

[94] Hussan Al-Chalabi, Yamur K Al-Douri, and Jan Lundberg. "Time Series Forecasting using ARIMA Model". en. In: (2018).

[95] Oskar Triebe et al. *NeuralProphet: Explainable Forecasting at Scale*. arXiv:2111.15397 [cs, stat]. Nov. 2021. DOI: `10.48550/arXiv.2111.15397`. URL: `http://arxiv.org/abs/2111.15397` (visited on 07/08/2023).

Appendices

A - Github repository

The Github repository linked below contains all the codes utilized in this project. Additional elucidations are provided within the readme file.

Github repository link

- `https://github.com/nayan2/XAI-based-Model-Improvement`

B - Sidenote statistics

B1 - Cross validation Results

City	Model	Matrix	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10
Stavanger	Base	MAE	0.604	0.939	0.988	0.996	0.776	1.129	0.890	0.907	0.708	0.949
Stavanger	Base	RSME	2.020	1.985	2.072	2.142	2.101	2.172	2.055	2.076	2.106	2.199
Stavanger	Noise	MAE	0.739	0.856	0.826	1.296	0.791	1.094	1.013	0.827	0.706	0.756
Stavanger	Noise	RSME	2.010	1.980	2.069	2.156	2.099	2.162	2.048	2.088	2.098	2.209
Stavanger	Low-Demand	MAE	0.867	0.862	0.863	1.018	0.967	1.077	0.980	1.308	1.325	1.030
Stavanger	Low-Demand	RSME	1.995	1.982	2.082	2.139	2.112	2.142	2.039	2.055	2.049	2.176
TromsÃ	Base	MAE	0.274	0.372	0.423	0.380	0.372	0.349	0.264	0.287	0.181	0.308
TromsÃ	Base	RSME	0.515	0.537	0.563	0.558	0.577	0.600	0.577	0.522	0.564	0.584
TromsÃ	Noise	MAE	0.142	0.165	0.213	0.264	0.189	0.199	0.149	0.177	0.151	0.209
TromsÃ	Noise	RSME	0.521	0.538	0.565	0.560	0.580	0.599	0.575	0.523	0.560	0.584
TromsÃ	Low-Demand	MAE	0.200	0.307	0.289	0.323	0.334	0.303	0.196	0.192	0.234	0.341
TromsÃ	Low-Demand	RSME	0.516	0.545	0.584	0.560	0.578	0.600	0.578	0.524	0.555	0.581
Trondheim	Base	MAE	0.734	0.623	0.659	0.803	0.714	0.505	0.488	0.723	0.725	0.554
Trondheim	Base	RSME	1.084	1.082	1.126	1.119	1.137	1.225	1.157	1.109	1.099	1.201
Trondheim	Noise	MAE	0.639	0.461	0.419	0.537	0.477	0.437	0.437	0.614	0.622	0.503
Trondheim	Noise	RSME	1.090	1.088	1.108	1.116	1.137	1.215	1.158	1.113	1.090	1.203
Trondheim	Low-Demand	MAE	0.682	0.567	0.929	0.684	0.512	0.691	0.685	0.934	0.721	0.610
Trondheim	Low-Demand	RSME	1.080	1.082	1.143	1.120	1.141	1.212	1.154	1.111	1.074	1.198

Table B.1: Cross validation score

City	Model	Matrix	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10
Bergen	Base	MAE	0.604	0.625	0.594	0.563	0.621	0.672	0.530	0.520	0.467	0.807
Bergen	Base	RSME	1.103	1.094	1.113	1.136	1.149	1.196	1.123	1.120	1.120	1.209
Bergen	Noise	MAE	0.448	0.549	0.362	0.566	0.517	0.573	0.578	0.586	0.318	0.515
Bergen	Noise	RSME	1.103	1.092	1.105	1.134	1.147	1.206	1.123	1.123	1.127	1.193
Bergen	Low-Demand	MAE	0.500	0.363	0.414	0.410	0.383	0.455	0.879	0.602	0.542	0.572
Bergen	Low-Demand	RSME	1.026	0.877	1.143	1.318	0.670	1.556	0.845	0.848	2.010	0.914
Helsingfors	Base	MAE	1.842	2.197	4.339	3.802	3.593	2.545	3.138	3.589	2.835	2.859
Helsingfors	Base	RSME	0.793	0.434	0.690	0.733	0.436	0.610	0.368	0.732	0.512	0.386
Helsingfors	Noise	MAE	0.653	0.979	1.315	0.964	0.518	0.549	1.054	0.907	1.135	0.688
Helsingfors	Noise	RSME	0.792	0.438	0.690	0.744	0.423	0.614	0.373	0.740	0.525	0.379
Helsingfors	Low-Demand	MAE	1.027	0.877	1.143	1.318	0.670	1.556	0.845	0.848	2.010	0.914
Helsingfors	Low-Demand	RSME	4.708	3.986	4.884	4.435	5.325	4.059	4.198	3.143	6.388	5.152
Oslo	Base	MAE	0.792	0.433	0.687	0.751	0.416	0.620	0.349	0.748	0.505	0.385
Oslo	Base	RSME	4.412	4.403	4.586	4.731	4.709	4.787	4.469	4.671	4.657	4.952
Oslo	Noise	MAE	1.921	3.926	2.634	3.381	2.261	2.355	1.742	2.493	2.176	2.812
Oslo	Noise	RSME	4.401	4.377	4.624	4.707	4.696	4.801	4.485	4.665	4.567	4.906
Oslo	Low-Demand	MAE	1.027	0.877	1.143	1.318	0.670	1.556	0.845	0.848	2.010	0.914
Oslo	Low-Demand	RSME	2.449	1.573	2.254	1.978	1.281	2.360	1.285	1.804	2.414	2.644

Table B.2: Cross validation score

B3 - Data distribution(Across all cities)

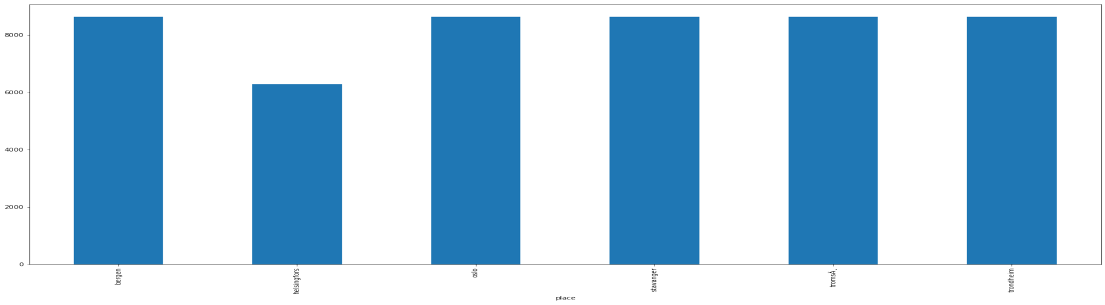


Figure B.1: Data distribution; city wise

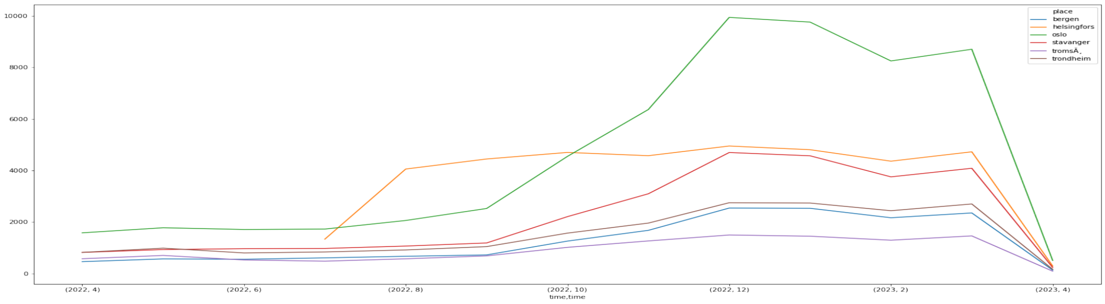
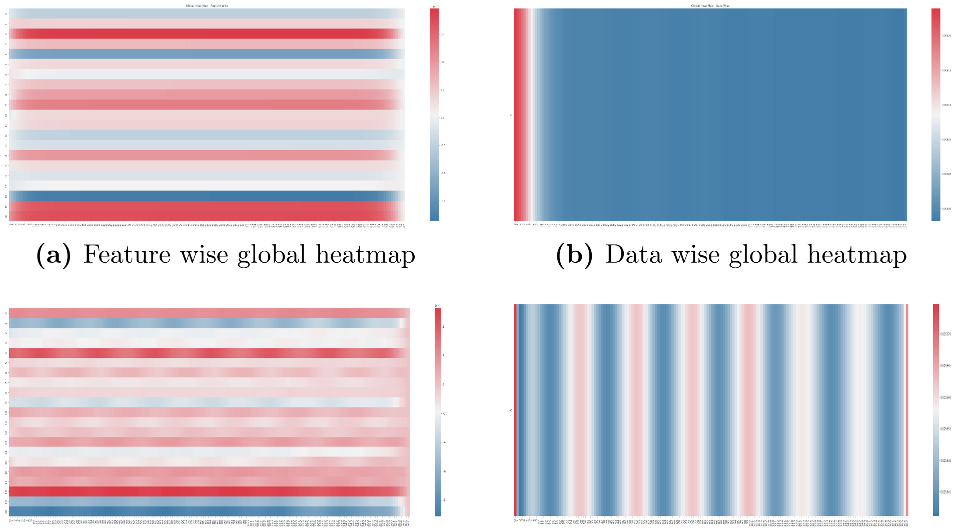


Figure B.2: Demand plot; City wise



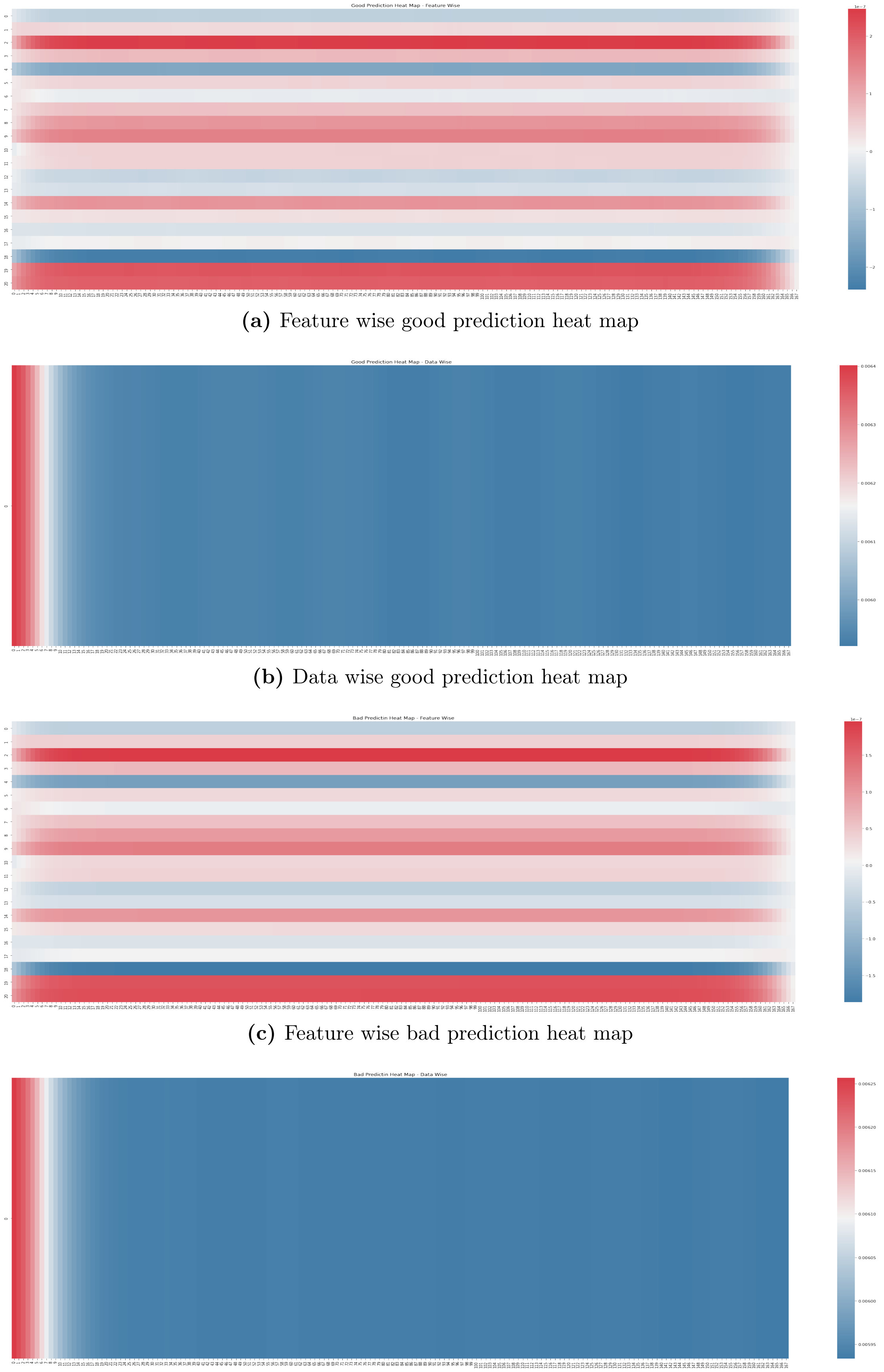
(a) Feature wise global heatmap

(b) Data wise global heatmap

(c) Feature wise local heatmap

(d) Data wise local heatmap

Figure B.3: Heat maps; Across cities



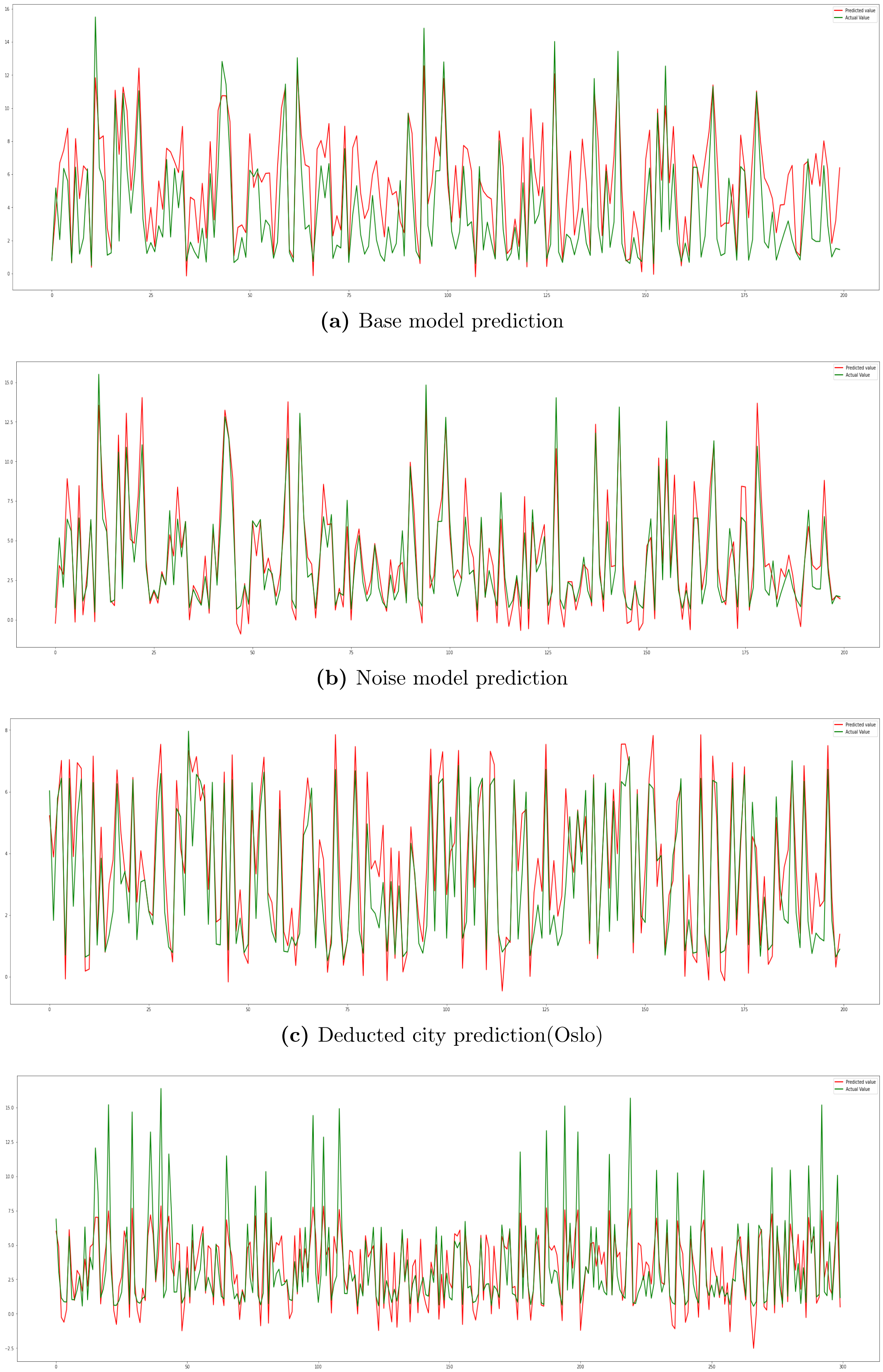
(a) Feature wise good prediction heat map

(b) Data wise good prediction heat map

(c) Feature wise bad prediction heat map

(d) Data wise good prediction heat map

Figure B.4: Good and bad predictions heat maps; Across cities



(a) Base model prediction

(b) Noise model prediction

(c) Deducted city prediction(Oslo)

(d) Low demand model prediction

Figure B.5: Models prediction; Among all cities

NTNU

Norwegian University of
Science and Technology