Pontifícia Universidade Católica
do Rio de Janeiro

Matheus Alves Pereira dos Santos

Advancements in Time Series Modeling: Using Modern Optimization and Robustness Techniques with Score-Driven Models

Dissertação de Mestrado

Dissertation presented to the Programa de Pós–graduação em Engenharia de Produção of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Engenharia de Produção.

Advisor: Prof. Davi Michel Valladão

Rio de Janeiro
Fevereiro 2024

Pontifícia Universidade Católica
do Rio de Janeiro

Matheus Alves Pereira dos Santos

Advancements in Time Series Modeling: Using Modern Optimization and Robustness Techniques with Score-Driven Models

Dissertation presented to the Programa de Pós–graduação em Engenharia de Produção of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Engenharia de Produção. Approved by the Examination Committee:

Prof. Davi Michel Valladão
Advisor
Departamento de Engenharia Industrial – PUC-Rio

Prof. Cristiano Augusto Coelho Fernandes
Departamento de Engenharia Elétrica – PUC-Rio

Prof. Fernando Luiz Cyrino Oliveira
Departamento de Engenharia Industrial – PUC-Rio

Dr. Joaquim Masset Lacombe Dias Garcia
PSR Power Systems Research

Rio de Janeiro, Fevereiro 26th, 2024

Matheus Alves Pereira dos Santos

Majored in Statistics from the Fluminense Federal University (UFF).

Bibliographic data

Santos, Matheus Alves Pereira dos

Advancements in time series modeling: using modern optimization and robustness techniques with Score-Driven Models / Matheus Alves Pereira dos Santos ; advisor: Davi Michel Valladão. – 2024.

97 f. : il. color. ; 30 cm

Dissertação (mestrado)–Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Industrial, 2024.

Inclui bibliografia

1. Engenharia Industrial – Teses. 2. Séries temporais. 3. Modelos Score-Driven. 4. Otimização robusta. I. Valladão, Davi Michel. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Industrial. III. Título.

CDD: 658.5

To my family, my girlfriend and my friends.
All of you were extremely important for this achievement, whether with guidance or support in moments of discouragement. None of this would have been possible without each of you.

Acknowledgments

Firstly, I would like to thank my family, my parents, my sisters and my grandmother, for all the support they gave me, not only during the preparation of this work, but throughout my life. I would not even be able to start if it were not for you. I would also like to thank my girlfriend, as she is always willing to listen to my outbursts and help me stay calm in difficult times. Part of this achievement is also yours. Thank you very much to each of you!

I would also like to thank all my colleagues at LAMPS who contributed in some way to the preparation of this work. Finally, I would like to thank my advisor Davi, for all the guidance not only in the development of this work but also throughout the master's degree.

This study was financed in part by the Coordenação de Aperfeiçoaento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

Abstract

Alves Pereira dos Santos, Matheus; Valladão, Davi Michel (Advisor). **Advancements in Time Series Modeling: Using Modern Optimization and Robustness Techniques with Score-Driven Models**. Rio de Janeiro, 2024. 97p. Dissertação de Mestrado – Departamento de Engenharia Industrial, Pontifícia Universidade Católica do Rio de Janeiro.

The study of time series plays a pivotal role in the decision-making process, giving rise to numerous methodologies over time. Within this context, score-driven models emerge as a flexible and interpretable approach. However, due to the significant number of parameters involved, the estimation process for these models tends to be complex. To address this complexity, this study aims to evaluate how the adoption of modern optimization techniques impacts the final performance of the model. Beyond simplifying the parameter estimation process, this shift in paradigm allows for the integration of new techniques, such as robust optimization, into the model formulation, thereby potentially enhancing its performance. The *SDUC.jl* package, which facilitates the adjustment and prediction of score-driven models based on unobservable components using modern optimization techniques, represents one of the main contributions of this study. By utilizing well-known time series to illustrate its functionality and monthly electrical load data from the Brazilian system, the study was able to demonstrate the flexibility of the package and its robust performance, even during periods of regime change in the data, thanks to the application of robustness techniques.

Keywords

Time Series; Score-Driven Models; Robust Optimization.

Resumo

Alves Pereira dos Santos, Matheus; Valladão, Davi Michel. **Avanços na modelagem de séries temporais: utilizando otimização moderna de técnicas de robustez com modelos score-driven**. Rio de Janeiro, 2024. 97p. Dissertação de Mestrado – Departamento de Engenharia Industrial, Pontifícia Universidade Católica do Rio de Janeiro.

O estudo de séries temporais desempenha um papel fundamental no processo de tomada de decisão, dando origem a inúmeras metodologias ao longo do tempo. Dentro desse contexto, os modelos score-driven surgem como uma abordagem flexível e interpretável. No entanto, devido ao número significativo de parâmetros envolvidos, o processo de estimação desses modelos tende a ser complexo. Para lidar com essa complexidade, este estudo tem como objetivo avaliar como a adoção de técnicas modernas de otimização impacta o desempenho final do modelo. Além de simplificar o processo de estimação de parâmetros, essa mudança de paradigma permite a integração de novas técnicas, como a otimização robusta, na formulação do modelo, potencialmente aprimorando seu desempenho. O pacote *SDUC.jl*, que facilita o ajuste e a previsão de modelos impulsionados por escores com base em componentes não observáveis usando técnicas modernas de otimização, representa uma das principais contribuições deste estudo. Ao utilizar séries temporais conhecidas para ilustrar sua funcionalidade e dados mensais de carga elétrica do sistema brasileiro, o estudo foi capaz de demonstrar a flexibilidade do pacote e seu desempenho robusto, mesmo durante períodos de mudança de regime nos dados, graças à aplicação de técnicas de robustez.

Palavras-chave

Séries Temporais; Modelos Score-Driven; Otimização Robusta.

Table of contents

List of figures

List of tables

List of algorithms

List of codes

List of Abreviations

ACF – Autocorrelation Function

AIC – Akaike Information Criterion

AICc – Akaike Information Criterion corrected

AR – Autoregressive

ARIMA – Autoregressive Integrated Moving Average

ARMA – Autoregressive Moving Average

BAGGING – Bootstrap Aggregating

BIC – Bayesian Information Criterion

CDF – Cumulative Distribution Function

CPA – Conditional Predictive Ability

DCS – Dynamic Conditional Score

GARCH – Generalized Autoregressive Conditional Heteroskedasticity

GAS – Generalized Autoregressive Score

MAE – Mean Absolute Error

MAPE – Mean Absolute Percentage Error

MASE – Mean Absolute Scaled Error

OLS – Ordinary Least Squares

PIT – Probability Integral Transform

RMSE – Root Mean Square Error

RO – Robust Optimization

R-SDM – Robust Score-Driven Model

SDM – Score-Driven Model

SDUC – Score-Driven Unobserved Components

TSL – Time Series Lab

UC – Unobserved Components

VaR – Value at Risk

1
Introduction

Given the broad applicability of time series analysis, it plays a crucial role in decision-making processes across various fields of knowledge, including finance, energy, epidemiology, among others. Time series analysis can serve as the final output of an analysis or be utilized to generate future scenarios for optimization problems, thereby contributing significantly to the decision-making process. Consequently, numerous methodologies for time series forecasting have been developed over time, such as the Autoregressive Integrated Moving Average (ARIMA) model [1] and the State Space model [2]. While these models represent two of the most commonly employed approaches, they also exhibit some significant limitations.

ARIMA assumes that the data follows a Gaussian distribution, which constitutes a significant limitation of this model type. Additionally, it necessitates the time series to be stationary, implying that its mean and variance should remain constant over time. This requirement can be addressed by employing transformations to render the data stationary. In essence, as implied by the model's name, stationarity is achieved through the integration of the time series.

On the other hand, State Space models are capable of directly handling non-stationary time series by representing their dynamics through unobserved components. While these models overcome one of the problems of ARIMA, they also assume that the data follows a Normal distribution, thus resulting in the same limitation as described previously.

The generalized autoregressive score (GAS) or score-driven models (SDM) framework, independently developed by Creal, Koopman & Lucas[3] and Harvey[4], emerged as a paradigm shift concerning the assumption of data normality inherent in various models. This class of models provides a

flexible alternative to time series modeling by enabling the derivation of models with time-varying parameters, thereby accommodating various discrete or continuous distributions beyond the Gaussian assumption. This flexibility addresses the limitations of ARIMA and State Space models, thus allowing for the application of these models in a wide array of contexts.

The version proposed in [3], referred to as GAS(p, q), assumes that the dynamics of the time-varying parameters follow an ARMA-like process. While it is generalizable to different distributions, this model still requires the data to be stationary. However, Harvey[4] proposed the utilization of unobserved components (UC) to characterize the dynamics of the parameters, akin to the approach taken by State Space models. This latest version creates models capable of overcoming both aforementioned limitations, rendering them applicable to various problem types. Therefore, this work will primarily focus on this class of models, which will be referred to as UC score-driven models.

In general, the estimation of score-driven models is conducted using the maximum likelihood method. However, analytically evaluating the required derivatives to obtain maximum likelihood is a demanding and challenging task. Therefore, the most common solution is to numerically evaluate these derivatives using a global optimization method such as Nelder-Mead [5] and L-BFGS [6].

Despite the widespread use of these methods, it is noteworthy that their methodologies originated from a time when optimization techniques were much more limited compared to today. With advancements in computational capacity and optimization, exact methods are no longer a predominant concern. Therefore, framing time series models such as the score-driven framework as mathematical programming problems enables the utilization of state-of-the-art optimization techniques.

Moreover, representing these models as optimization problems opens doors to the utilization of various techniques from the field of optimization

that have the potential to enhance the performance of the models. In this context, Bertsimas & Dunn[7] stand out as primary authors. In their book, they demonstrate how different optimization techniques can be incorporated into the formulation of machine learning models to enhance their performance across various aspects.

1.1 Literature Review

The flexibility offered by score-driven models, regarding both data distribution and dynamic representation, makes it a versatile framework that has been successfully employed in various domains. In risk analysis, they are utilized for forecasting extreme financial risk [8], estimating the probability of zero returns [9], and modeling risk metrics such as expected shortfall and value at risk (VaR) [10]. In the energy sector, they are applied to futures hedging of crude oil and natural gas [11], investigating different VaR forecasts for daily energy commodities returns [12], and simulating long-term joint scenarios for multivariate wind power generation [13]. In finance, score-driven models have been used to include random shifts in the process of modeling stock returns volatility [14], analyze global equity market co-movement [15], and measure the association between asset returns [16]. In the retail sector, these models have been applied to predict both lumpy and intermittent demand [17] and short-term demands [18]. Score-driven models can be applied to many other fields, which can be found in the online repository of GAS models (http://www.gasmodel.com).

An interesting observation about all the papers cited above is that [17] is the only paper that considers a score-driven model based on unobserved components (UC). This highlights that, despite its limitations, GAS(p, q) emerges as the most commonly used version of the score-driven framework. The preference for ARMA-type dynamics may be associated with the characteristics of the fields where these models are applied, where the assumption of data

stationarity is reasonable.

One possible consequence of this phenomenon is the availability of implementations of these models. Open-source packages like GAS [19] and ScoreDrivenModels.jl [20], available in R [21] and Julia [22] languages respectively, allow for the use of score-driven models, but they mainly focus on the GAS(p, q) version or some special cases of it, as these are the most commonly used versions of the score-driven framework. The Time Series Lab [23], in its score edition, is the only implementation found that allows for the definition of UC models. However, the fact that the latter is a commercial software may limit its accessibility and use.

As previously mentioned, traditionally, including the implementations mentioned earlier, score-driven models are estimated using methods that no longer represent the state of the art in optimization. By representing models as mathematical programming problems, Bertsimas & Dunn[7] not only enables the adoption of more modern optimization techniques for model estimation but also facilitates the consideration of various methods in this field in the model formulation.

As an example of methods discussed in [7], one can mention the use of integer optimization to perform explanatory variable selection, resulting in sparse models. Additionally, this method can serve as an automatic means to define model orders. Another intriguing possibility is the utilization of robustness techniques such as regularization and others, which have the potential to enhance accuracy.

Focusing on the latter, robust optimization techniques have garnered significant interest across various applications due to the possibility of making optimal decisions under the worst-case scenario within a user-defined uncertainty set. Although this field has gained greater notoriety in the last 15 years, its earliest publication dates back to the 1970s [24], wherein a linear optimization problem was proposed, with its optimal value being feasible in all

potential scenarios within a convex set. However, in exchange for providing solutions accommodating all potential realizations of uncertainty parameters, this model tends to produce overly conservative optimal solutions from a practical standpoint.

In an effort to mitigate the excessive conservatism of the initial method, several alternative approaches were proposed in the subsequent years. This includes the work of Ben-Tal & Nemirovski[25], who introduced a set of ellipsoidal uncertainty sets, and Bertsimas & Sim[26], who pioneered the concept of an uncertainty "budget," enabling the control of the solution's conservatism level. More recently, Fernandes et al.[27] proposed a robust adaptive data-driven solution to the portfolio problem.

Bertsimas & Paskov[28] successfully applied the principles of robust optimization to the realm of time series modeling. By employing the concept of optimization grounded on the worst-case scenario within an uncertainty set, they transformed an AR(p) model into one resilient to regime changes in the data.

Given these considerations, the primary objective of this study is to explore the benefits derived from representing UC score-driven models as optimization problems. This exploration encompasses not only the conventional formulation of these models but also their integration with features from optimization methodologies. While numerous features could be incorporated into the model's formulation, the focus here lies on a robustness technique that aims to mitigate the impact of regime changes in the data. As a result of this endeavor, an open-source package will be developed in the Julia language. This package will enable users to estimate various UC score-driven models using modern optimization techniques. This initiative has the potential to address a gap in the literature and promote the utilization of UC score-driven models in a wider range of applications.

1.2 Illustrative Example

To illustrate a key aspect and anticipate some results of this work, consider the example described below. One of the robustness techniques that will be included in the model's formulation, and will be discussed in detail in Chapter 2, aims to make the model robust against regime changes in the data [28]. With that in mind, monthly data of electric load from Brazil's North system was selected. The choice of this data is based on the regime change induced by the COVID-19 pandemic starting in 2020. The purpose of this example is to compare the accuracy of predictions generated by the robust score-driven model against those of its non-robust counterpart.

To model this data, a Gaussian UC score-driven model was proposed, where only the mean parameter varies over time. The selected dynamics for this parameter included a local linear trend model along with a stochastic seasonal component.

To estimate this model, its formulation as an optimization problem is done as in Equation 1-2. It is important to highlight that all time-varying parameters are conditioned on their own realizations in the directly preceding period. The notation used to indicate this fact is given by the sub index $_{t|t-1}$. Trying to promote a deeper understanding about this model's formulation, each of its decision variables and constraints will be described bellow.

- $\boldsymbol{\mu}$: A vector of length T (the number of observations in the time series) representing the time-varying mean parameter of the Normal distribution.
- σ^2: The fixed variance parameter of the Normal distribution.
- $\mathbf{m}$: A vector of length T representing the level component.
- $\mathbf{b}$: A vector of length T representing the slope component.
- $\mathbf{S}$: A vector of length T representing the seasonal component.

- $\boldsymbol{\gamma}$, $\boldsymbol{\gamma}^*$: Two matrices with dimensions equal to the number of harmonics per series size, representing the coefficients of the harmonics in the seasonal component dynamics.
- κ_m, κ_b, κ_S: Represent the variability of the level, slope, and seasonal components, respectively.

At this juncture, it is important to emphasize that, in the formulation presented in Equation 1-2, alongside the decision variables mentioned earlier, there exists a term s_t which is not a decision variable of the model. In actuality, s_t, representing the scaled score (further elaboration on this will be provided in Chapter 2), is a function of the observed values (y_t) and the parameters of the Normal distribution. Hence, the most comprehensive and accurate notation for this term would be $s_t(y_{t-1}, \mu_{t-1|t-2}, \sigma^2)$. However, for the sake of simplicity in notation, the scaled score will consistently be denoted as s_t.

To further comprehend the formulation of this model, it is imperative to discuss each of its constraints, which elucidate the dynamics of the time-varying parameter. The initial constraint pertains to the conditional mean, linking it in each time period to the sum of the level and seasonal components. The subsequent two constraints delineate the dynamics of the level component, which in this instance follows a local linear trend dynamic. Lastly, the final two constraints govern the seasonal dynamics of the model, dictated by the sum of stochastic trigonometric terms.

$$l(y_t; \mu_{t|t-1}, \sigma^2) = -\frac{1}{2}\log(2\pi\sigma^2) - \frac{(y_t - \mu_{t|t-1})^2}{2\sigma^2} \tag{1-1}$$

Finally, the expression shown in Equation 1-1 represents the log-likelihood function for the Normal distribution. Remarkably, the model depicted in Equation 1-2, when optimized, provides a maximum likelihood estimate of the distribution parameters. It is noteworthy to emphasize that this approach allows for the incorporation of regularization or other robustness techniques with certain adjustments in the objective function and/or constraints.

$$
\begin{aligned}
&\underset{\substack{\boldsymbol{\mu},\mathbf{m},\mathbf{b},\mathbf{S},\boldsymbol{\gamma},\boldsymbol{\gamma}^{*} \\ \kappa_m \geq 0, \kappa_b \geq 0, \\ \kappa_S \geq 0, \sigma^2 > 0}}{\text{Min}} \quad -\sum_{t=1}^{T} l(y_t; \mu_{t|t-1}, \sigma^2) \\
&\text{s. t.} \quad \mu_{t|t-1} = m_{t|t-1} + S_{t|t-1}, && \forall t = 1, \cdots, T \\
&\qquad m_{t|t-1} = m_{t-1|t-2} + b_{t-1|t-2} + \kappa_m s_{t-1}, && \forall = 1, \cdots, T \\
&\qquad b_{t|t-1} = b_{t-1|t-2} + \kappa_b s_{t-1}, && \forall t = 1, \cdots, T \\
&\qquad S_{t|t-1} = \sum_{j=1}^{6} \gamma_{j,t|t-1}, && \forall t = 1, \cdots, T \\
&\qquad \begin{bmatrix} \gamma_{j,t|t-1} \\ \gamma^{*}_{j,t|t-1} \end{bmatrix} = \begin{bmatrix} \cos\left(\frac{2\pi j t}{12}\right) & \sin\left(\frac{2\pi j t}{12}\right) \\ -\sin\left(\frac{2\pi j t}{12}\right) & \cos\left(\frac{2\pi j t}{12}\right) \end{bmatrix} \begin{bmatrix} \gamma_{j,t-1|t-2} \\ \gamma^{*}_{j,t-1|t-2} \end{bmatrix} + \begin{bmatrix} \kappa_S s_{t-1} \\ \kappa_S s_{t-1} \end{bmatrix}, && \forall t = 1, \cdots, T
\end{aligned}
\tag{1-2}
$$

To illustrate the effectiveness of the robustness technique, both the UC score-driven model formulated in Equation 1-2 and its robust counterpart (to be described later) were estimated using the aforementioned data. The training set comprised observations from the beginning of 2014 until April 2020, while the test set included the 12 subsequent observations. Figure 1.1 displays out-of-sample results.

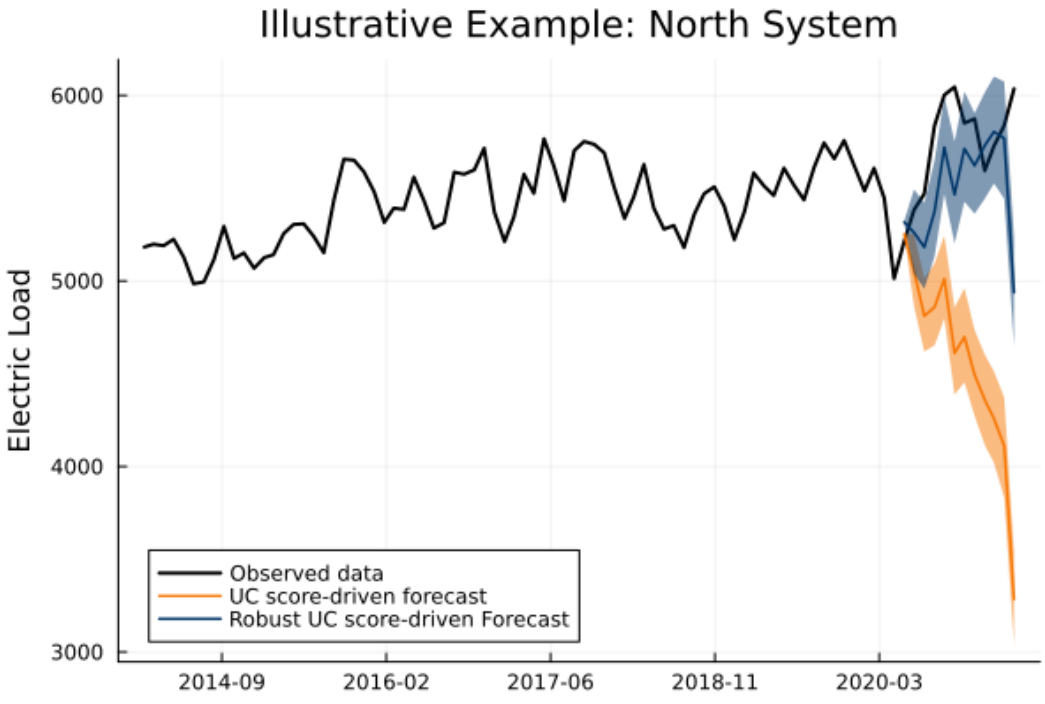

Figure 1.1: Illustrative example based on the model depicted in Equation 1-2.

It is evident that the two models exhibit markedly different out-of-sample performance. While the robust model produced predictions that closely aligned with the actual observations, its non-robust counterpart generated predictions

with a decreasing trend that was not observed. This straightforward experiment demonstrates the potential benefits, in terms of model accuracy, that the incorporation of robust optimization techniques in the model formulation can yield, underscoring the potential relevance of this work to the field of time series analysis.

This document is structured as follows. In Chapter 2, the theoretical background considered in this work will be presented. Section 2.1 focuses on the main aspects associated with the UC score-driven formulations, while Section 2.2 discusses the robustness techniques considered in this work. In Chapter 3, the discussion will be focused on the proposed methodology, beginning with how a UC score-driven model can be formulated as an optimization problem. Section 3.1 provides a similar discussion about the robustness feature included in the model's formulation. Subsequently, Section 3.2 delves into the SDUC.jl (score-driven unobserved components) package itself, while considering some examples to illustrate its operation. In Chapter 4, an experiment will be performed to evaluate the package's performance during regime changes periods. Finally, in Chapter 5, the conclusions and future works are discussed.

2
Theoretical Background

This chapter covers two separate subjects that will be reviewed individually. Firstly, the formulation of score-driven models (SDM) will be discussed in detail. Subsequently, the optimization robustnes methodology and its potential contributions to time series modeling will be explored.

2.1
Unobserved components score-driven models

This section provides a thorough review of the UC score-driven framework, starting with its formulation and rationale. Additionally, a brief overview of the primary unobserved components used in the model will be presented. The section concludes with a discussion on the analysis of residuals and the prediction procedure.

2.1.1
Model Formulation

Score-driven models belong to the class of observation-driven time-varying parameter models, as classified by Cox et al.[29]. It is worth noting that well-established observation-driven models, such as GARCH [30], can be regarded as particular cases of SDM.

Like other observation-driven models, the SDM has the advantage of considering likelihood functions whose evaluation is straightforward. Moreover, these models allow for the formulation of non-Gaussian distributions, which can be continuous or discrete. This flexibility enables the incorporation of complex dynamics, including asymmetry, long memory, and heavy tails, through appropriate distribution specifications [3].

The selection of the predictive or conditional probability distribution is crucial in the model formulation. Let y_t denote a univariate time series of interest, $f_{t|t-1}$ represent the vector of time-varying parameters for the pre-

dictive distribution, and $\boldsymbol{\theta}$ be a vector of fixed parameters. The available information at time t is denoted by $\mathbf{Y}_{t-1}$ and $\mathbf{F}_{t-1}$, where $\mathbf{Y}'_t = \{y_1, \cdots, y_t\}$ and $\mathbf{F}'_t = \{f_0, f_{1|0}, \cdots, f_{t|t-1}\}$. In the score-driven framework, a satisfactory modeling of y_t is achieved by considering the conditional or predictive density/probability function given by:

$$y_t \sim p(y_t | f_{t|t-1}, \mathbf{Y}_{t-1}, \mathbf{F}_{t-1}; \theta). \tag{2-1}$$

Once the predictive distribution is determined, the dynamics of the time-varying parameters must be defined. Score-driven models offer flexibility in this regard, allowing for the specification of ARMA-like processes (GAS(p, q) models) or the inclusion of stochastic unobserved components, such as level and seasonality, known as unobserved components score-driven models. This versatility makes these models applicable to a wide range of problems. In this work, the focus will be on the latter.

For the model formulation, a local level model will be considered:

$$\begin{aligned} h(f_{t|t-1}) &= m_{t|t-1} \\ m_{t|t-1} &= m_{t-1|t-2} + \kappa \tilde{s}_{t-1}, \quad \kappa \geq 0 \end{aligned} \tag{2-2}$$

Here, $h(\cdot)$ represents an appropriate link function that ensures the estimates of the parameter $f_{t|t-1}$ adhere to their respective domains. In other words, if ones consider that the time-varying parameter f_t only assumes positive values, simply relying on the random walk dynamic of the m_t component, as shown in Equation 2-2, is not sufficient to guarantee the domain of the parameter. The most common solution to this issue is to apply an appropriate link function $h(\cdot)$. For this example, a suitable solution would be to set $h(\cdot) = \log(\cdot)$, which would result in

$$\begin{aligned} f_{t|t-1} &= e^{m_{t|t-1}} \\ m_{t|t-1} &= m_{t-1|t-2} + \kappa \tilde{s}_{t-1}, \quad \kappa \geq 0. \end{aligned} \tag{2-3}$$

With the formulation presented in Equation 2-3, $f_{t|t-1}$ will assume

positive values regardless of the value of the $m_{t|t-1}$ component. It is important to note that $\tilde{s}_t$ represents the scaled score obtained when considering $h(\cdot)$.

Irrespective of the link function utilized, Equation 2-2 underscores that the time-varying parameter is contingent upon an unobserved component characterized by a random walk dynamic. Furthermore, the stochasticity of the m_t component is influenced by $\tilde{s}_t$, given that κ is a fixed parameter. It is noteworthy that this formulation permits the estimation of a deterministic component by setting $\kappa = 0$. The scaled score function, under the assumption of the identity link function, s_t, is defined as:

$$s_t = R_t \cdot \nabla_t, \tag{2-4}$$

where ∇_t is the score function of the predictive distribution given by

$$\nabla_t = \frac{\partial \ln \left(p(y_t | f_{t|t-1}, \mathbf{F}_{t-1}, \mathbf{Y}_{t-1}, \boldsymbol{\theta}) \right)}{\partial f_t}. \tag{2-5}$$

It is important to emphasize that when the link function $h(\cdot)$ is not the identity, the scaled score function is denoted as $\tilde{s}$ to accommodate the influence of the link function.

The utilization of the score function to update the time-varying parameters is rooted in its definition as the steepest ascent direction for enhancing the local fit of the model in terms of the likelihood at time t, given the current parameter [3]. Unlike other observation-driven models in the literature, the fact that the score function depends on the entire distribution and not just the first and second moments provides the score-driven model with a superior representation of the distribution's uncertainty in the update mechanism.

The term R_t in Equation 2-4 represents a scaling matrix. Various specifications of this matrix can affect the properties of the model. In many instances, a prevalent approach to defining this scaling matrix relies on the variance of the score [3]. An intriguing aspect of this scaling choice is that the variance of the score function corresponds to the Fisher information matrix

(I) [31]. Consequently, the scaled score can be reformulated as:

$$s_t = I_{t|t-1}^{-d} \cdot \nabla_t, \quad d = 0, \frac{1}{2}, 1 \tag{2-6}$$

The scale parameter d associated with the Fisher Information matrix is typically utilized to introduce a bit more flexibility in defining the scaled score, taking values from the set $\left\{0, \frac{1}{2}, 1\right\}$. When $d = 1$ or $d = 0$, $I_{t|t-1}^{-d}$ is well-defined. For $d = \frac{1}{2}$, $I_{t|t-1}^{-\frac{1}{2}}$ results from the Cholesky decomposition of $I_{t|t-1}^{-1}$ [3].

Understanding the interpretation of each term associated with s_t clarifies the effect of the scale parameter d. For $d = 1$, the scaled score has a constant unit variance. On the other hand, setting $d = 0$ assigns the entire variance of the score function to s_t. Finally, $d = \frac{1}{2}$ represents an intermediate scenario.

Finally, the estimation of the vector of fixed parameters ($\boldsymbol{\theta}$) in these models is carried out via maximum likelihood. Obviously, this framework allows for the specification of many different models, considering dynamics other than the one presented here. Therefore, in addition to understanding the rationale behind the model formulation, it is also necessary to comprehend the formulation of the components. The next subsection will provide a brief review of the main unobserved components used.

2.1.2 Unobserved Components

As mentioned previously, the entire dynamics of time-varying parameters is driven by unobserved components. The choice of these components offers a wide range of possibilities for formulating the model, making knowledge about their structures and how to identify them in the data essential. This subsection aims to briefly review some components that can be included in the score-driven formulation and the main ways in which they can be represented. The discussion will also cover the intuition behind when each component can be used to better represent data behavior. It is important to highlight that, as the score-driven framework considers components to always be stochastic, this

review will focus more on this type of component. More detailed descriptions of many of the models discussed can be found in [2].

2.1.2.1 Trend

The trend component in a time series signifies the long-term systematic movement or direction of the data. It is instrumental in capturing patterns such as growth or decline that occur over extended periods, largely unaffected by short-term fluctuations. This persistence renders the trend a smoother representation of the data compared to other components.

Various models have been proposed to address different levels of complexity for a stochastic trend. One of the simplest models is the local level model, which posits that the trend component follows a random walk process. It comprises a solitary latent state variable, commonly denoted as the level, which represents the present value of the trend.

A straightforward extension of the local level model, the random walk with drift model, posits that the trend undergoes a steady change over time. This adjustment enables the representation of time series with a trend that maintains a constant, non-zero slope. This pattern is attained by incorporating a deterministic component, referred to as the drift, into the random walk process. Regarding the predictive distribution in Equation 2-1, this model is mathematically delineated in the score-driven framework as:

$$\begin{aligned} h(f_{t|t-1}) &= m_{t|t-1} \\ m_{t|t-1} &= m_{t-1|t-2} + b + \kappa \tilde{s}_{t-1}, \quad \kappa \geq 0. \end{aligned} \tag{2-7}$$

While this model offers a notable extension compared to the local level model, it still does not account for variations in the rate of change of the trend.

An intuitive solution to this limitation is to allow the drift to vary over time. The local linear trend model achieves this by assuming that the trend follows a random walk with a drift for the level, and simply a random walk for the slope component. This modification enables capturing changes in both the

level and the rate of change of the trend (slope), making it the most versatile model discussed here. Mathematically, this model can be represented as:

$$\begin{aligned} h(f_{t|t-1}) &= m_{t|t-1} \\ m_{t|t-1} &= m_{t-1|t-2} + b_{t-1|t-2} + \kappa_m \tilde{s}_{t-1}, \qquad \kappa_m \geq 0 \\ b_{t|t-1} &= b_{t-1|t-2} + \kappa_b \tilde{s}_{t-1}, \qquad \kappa_b \geq 0. \end{aligned} \tag{2-8}$$

It is important to note that since this model has two stochastic components, it is necessary to consider two distinct fixed parameters, denoted as κ, one for each component.

Although these three models capture series with various types of trends, they all assume that this component exhibits non-stationary behavior. Therefore, to model temporal data with stationary characteristics, it is necessary to turn to another model. The most common option for this scenario is to consider that the trend component follows an $AR(1)$ process and constrain the coefficient of this process to enforce stationarity. Mathematically, this model can be formulated as:

$$\begin{aligned} h(f_{t|t-1}) &= m_{t|t-1} \\ m_{t|t-1} &= \omega + \phi m_{t-1|t-2} + \kappa \tilde{s}_{t-1}, \quad -1 < \phi < 1, \kappa \geq 0. \end{aligned} \tag{2-9}$$

It is important to highlight that in this model, the inclusion of the constant ω is necessary to enable component m to have a non-zero conditional mean. Another significant aspect of using autoregressive processes in UC models is that they can be included in a local level model, for example, to help the model deal with short-term dependencies that would not be well captured by other components. Therefore, it is notable that the AR process can be used to model the trend or as a component itself.

2.1.2.2 Seasonality

The seasonality component represents periodic fluctuations that occur within a year or less. These patterns may be associated with climatic factors

like seasons, socio-cultural events such as holidays, or administrative cycles. Understanding and incorporating seasonal patterns can offer insights into the underlying dynamics of a time series, help identify anomalies or irregularities, and enhance forecast accuracy by leveraging historical patterns within specific time intervals.

Given the diverse nature of seasonality, several approaches have emerged over time to model stochastic seasonality. One of the simplest methods involves using seasonal dummies. This approach utilizes binary variables to denote each seasonal period, capturing the discrete shifts or changes in the time series associated with each period. The primary advantages of this methodology lie in its simplicity of implementation and straightforward interpretation, as the coefficients linked to the dummy variables directly reflect the seasonal effects.

To illustrate, let's take a monthly time series with a seasonal period of 12 (i.e., one year). In the context of a score-driven model, concentrating solely on the seasonal component and employing the predictive distribution from Equation 2-1, the seasonal dummies can be mathematically expressed as:

$$\begin{aligned} h(f_{t|t-1}) &= S_{t|t-1} \\ S_{t|t-1} &= -\sum_{j=1}^{11} S_{t-j|t-j-1} + \kappa_S \tilde{s}_{t-1}, \quad \kappa_S \geq 0 \end{aligned} \tag{2-10}$$

One limitation of this approach is that representing all seasons with dummy variables introduces a perfect multicollinearity problem. Various techniques can be employed to address this issue. In Equation 2-10, the seasonal effects are constrained to sum to zero. Therefore, only eleven coefficients need to be estimated, with the last one being the negative summation of the others. This effectively resolves the perfect multicollinearity problem.

However, this approach may increase the dimensionality of the model, particularly when dealing with high-frequency time series that exhibit multiple seasonal periods. Furthermore, assuming a fixed seasonal pattern that repeats over time may not be reasonable in certain cases. To address this concern, a

variation of representing seasonality using seasonal dummies was proposed by Harvey et al. (2013), wherein the seasonal pattern is allowed to vary over time through a random walk dynamic. In this variation, for each period of time t, the seasonal effects are forced to sum to one. For detailed information about this variation, refer to the provided reference.

An alternative method for representing time series seasonality is using harmonic seasonality based on the Fourier Theorem. According to this theorem, any reasonably well-behaved periodic function can be represented as an infinite sum of sine and cosine functions with different frequencies and amplitudes [32].

The key advantage of this approach lies in its flexibility. It allows for the representation of seasonal patterns with varying frequencies and amplitudes, enabling the modeling of seasonalities with different levels of complexity. Unlike seasonal dummies, harmonic seasonality represents the seasonal component as a continuous function, resulting in a smoother representation of seasonal patterns and facilitating more precise modeling and forecasting.

To incorporate this approach into the previously discussed score-driven model and introduce stochasticity into the seasonal component, it can be mathematically defined as follows:

$$
\begin{aligned}
h(f_{t|t-1}) &= S_{t|t-1} \\
S_{t|t-1} &= \sum_{j=1}^{6} \gamma_{j,t|t-1} \\
\begin{bmatrix} \gamma_{j,t|t-1} \\ \gamma^*_{j,t|t-1} \end{bmatrix} &= \begin{bmatrix} cos(\frac{2\pi jt}{12}) & sin(\frac{2\pi jt}{12}) \\ -sin(\frac{2\pi jt}{12}) & cos(\frac{2\pi jt}{12}) \end{bmatrix} \begin{bmatrix} \gamma_{j,t-1|t-2} \\ \gamma^*_{j,t-1|t-2} \end{bmatrix} + \begin{bmatrix} \kappa_S \tilde{s}_{t-1} \\ \kappa_S \tilde{s}_{t-1} \end{bmatrix}, \quad \kappa_S \geq 0
\end{aligned}
\tag{2-11}
$$

It is worth noting that the maximum number of harmonics that need to be considered is equal to half of the seasonal period. If the seasonal period is an odd number, one less than the seasonal period is divided by two. This is because including more harmonics would result in repeated frequencies, providing no additional information to the model. Sometimes, using fewer than

the maximum number of harmonics is sufficient to capture the seasonal pattern. Therefore, when dealing with a high-frequency time series, it may be necessary to calibrate the number of harmonics using cross-validation, as considering all harmonics can lead to overfitting problems.

While this methodology offers a flexible approach to representing the seasonal component, it also exhibits a limitation in terms of interpretability. Unlike the dummy approach, interpreting harmonic seasonality is not straightforward. To compute the seasonal effect in this case, it is necessary to sum up the contributions of each harmonic.

2.1.3 Explanatory Variables

While unobserved components are valuable for capturing the dynamics of a time series, there are instances where it is necessary to examine the relationship between the data of interest and external factors. This can be achieved by incorporating explanatory variables into the model formulation within the UC score-driven framework. Let's consider the local level model presented in Equation 2-2 to illustrate how explanatory variables can be included.

$$\begin{aligned} h(f_{t|t-1}) &= m_{t|t-1} + \boldsymbol{\beta}'\mathbf{X}_t \\ m_{t|t-1} &= m_{t-1|t-2} + \kappa\tilde{s}_{t-1}, \quad \kappa \geq 0. \end{aligned} \tag{2-12}$$

In this formulation, $\boldsymbol{\beta}$ represents the vector of coefficients for the explanatory variables, and $\mathbf{X}_t$ denotes the exogenous variables at time t. Equation 2-12 demonstrates how straightforward it is to incorporate exogenous variables into the model formulation. Further theoretical details on this inclusion can be found in [4].

Sometimes, assuming that the effect of the explanatory variable remains constant over time may not be reasonable. To account for such cases, the UC score-driven framework also allows for the possibility of the $\boldsymbol{\beta}$ coefficient to vary over time following an $AR(1)$ process. While alternative dynamics are

possible, this is the most common choice. Mathematically, this extension can be represented as:

$$\begin{aligned} h(f_{t|t-1}) &= m_{t|t-1} + \boldsymbol{\beta}'_{t|t-1}\mathbf{X}_t \\ m_{t|t-1} &= m_{t-1|t-2} + \kappa_m \tilde{s}_{t-1} \qquad \kappa_m \geq 0. \\ \boldsymbol{\beta}_{t|t-1} &= \boldsymbol{\phi}\boldsymbol{\beta}_{t-1|t-2} + \kappa_\beta \tilde{s}_{t-1}, \qquad \kappa_\beta \geq 0. \end{aligned} \tag{2-13}$$

The inclusion of explanatory variables is essential as it offers a deeper understanding of the relationship between the time series and relevant external factors, such as economic indicators, demographic information, weather conditions, or other predictors of interest. Integrating these variables can improve forecast accuracy, streamline data interpretation, and enable the integration of domain-specific knowledge directly into the model.

However, it is important to note that the inclusion of explanatory variables should be based on sound theoretical or empirical foundations. The relevance and significance of these variables should be assessed using appropriate statistical and diagnostic methods. The selection of explanatory variables should be guided by subject matter expertise, a comprehensive understanding of the context and domain of the time series being modeled, and statistical techniques that aid in variable selection may be considered.

Having explored various possibilities for model formulation within the score-driven framework, it is evident that it offers substantial flexibility. Once a model is defined and estimated through maximum likelihood, it becomes crucial to assess the validity of the assumptions made regarding the predictive distribution and dynamics of the data. This assessment is typically performed through residual analysis, which will be discussed in more detail in the next subsection.

2.1.4 Residuals Analysis

After fitting a model, it is crucial to evaluate its goodness of fit to the data and verify whether the assumptions made during its formulation hold true for the dataset of interest. Residual analysis is a standard approach to detect potential model misspecifications [33]. This is because any underlying dependency structure not adequately captured by the model can often be detected through the analysis of residuals.

In general, the residuals of a time series model are defined mathematically as:

$$r_t = y_t - \hat{y}_{t|t-1}. \tag{2-14}$$

In this equation, y_t represents the observed series, and $\hat{y}_{t|t-1}$ denotes the model's point estimate for period t using the information up to time $t-1$. Typically, $\hat{y}_{t|t-1} = E(y_{t|t-1})$. It's worth noting that variations of residuals, such as Pearson residuals, can be derived from the equation to better suit specific situations. Residual analysis involves graphical examinations and statistical tests to assess their distribution and autocorrelation function (ACF). For a more detailed understanding of residual analysis, refer to [34].

In the context of UC score-driven models, residual analysis primarily focuses on evaluating assumptions related to the choice of the conditional distribution and the dynamics of model parameters. The following paragraphs present several definitions of residuals that are particularly useful in diagnosing score-driven models, along with the key insights that can be derived from them.

Pearson residuals, also known as standardized residuals, are obtained by dividing the difference between observed and predicted values by the estimated standard deviation of the fitted values. They provide a standardized measure of discrepancies between observed and fitted values, offering insights into model adequacy and goodness of fit. Mathematically, Pearson residuals are defined as:

$$r_t^p = \frac{y_t - \hat{y}_{t|t-1}}{\sqrt{Var(\hat{y}_{t|t-1})}} \tag{2-15}$$

The analysis of Pearson residuals is particularly useful for identifying outliers or extreme observations within the time series. Outliers refer to data points that significantly deviate from the overall pattern and can disproportionately impact model estimation. By examining the magnitudes of Pearson residuals, one can identify observations with unusually large residuals, signaling potential outliers that require further investigation or consideration for model refinement.

Pearson residuals can also be utilized to assess the adequacy of the chosen predictive distribution and the dynamics of time-varying parameters. However, alternative analyses may potentially yield more informative results in such diagnostic procedures. For instance, Harvey[4] proposed two approaches to evaluate the goodness of fit of the selected predictive distribution.

The first approach involves employing a QQ-plot, where the T equally spaced quantiles of the predictive distribution are plotted on the vertical axis, while the order statistics of the sample, denoted as $y_{(1)} \leq y_{(2)} \leq \cdots \leq y_{(T)}$, are plotted on the horizontal axis. If the sample is drawn from the comparison distribution, the resulting plot will closely resemble a straight 45-degree line. It is worth noting that closed-form expressions for quantile functions are generally unavailable for certain distributions, necessitating the use of numerical methods for their evaluation.

The second graphical approach, aimed at providing the same information while circumventing potential issues associated with the availability of closed-form quantile functions, involves utilizing the Probability Integral Transform (PIT). The PIT of a variable is based on its Cumulative Distribution Function (CDF). Assuming $y_t \sim p(y_t|f_{t|t-1}, \mathbf{Y}t-1, \mathbf{F}t-1; \theta)$, the PIT is defined as:

$$PIT(y_t) = F(y_t|f_{t|t-1}, \mathbf{Y}_{t-1}, \mathbf{F}_{t-1}; \theta). \tag{2-16}$$

In a correctly specified scenario, the PIT(y_t) will follow a standard uniform distribution. Therefore, assessing the distribution of PIT(y_t) through techniques such as histograms or statistical tests like the Kolmogorov-Smirnov test is equivalent to evaluating the accuracy of the predictive distribution.

An advantageous transformation for this approach involves using the inverse Gaussian Cumulative Distribution Function (Φ^{-1}) to transform the PIT into a standard Gaussian distributed variable [35]. The rationale behind this transformation is the greater availability of methodologies for testing the normality of data compared to those for testing the uniformity of a distribution, thereby increasing the flexibility of the analysis [35]. It is important to note that this proposed approach is equivalent to using randomized quantile residuals [36], which are defined as:

$$r_t^q = \Phi^{-1}(\widehat{PIT}(y_t)). \tag{2-17}$$

Where $\widehat{PIT}(y_t)$ represents the PIT defined in Equation 2-16, considering the model's estimates for both fixed and time-varying parameters.

Another important type of residual is the conditional score residual, which provides a general framework for diagnostic analysis of time series models and includes standard definitions of residuals such as the Pearson residual. Mathematically, the conditional score residual is defined as follows:

$$r_t^{cs} = \hat{I}_t^{-\frac{1}{2}} \cdot \hat{\nabla}_t. \tag{2-18}$$

where $\hat{I}t$ represents the model's estimate of the Fisher information matrix and $\hat{\nabla}t$ denotes the score function at period t. It is important to note that the conditional score residual is equivalent to the estimation of the scaled score of the score-driven model, considering the scale parameter $d = \frac{1}{2}$. Further information about the definition and asymptotic properties of this approach can be found in [33].

In the context of score-driven models, conditional score residuals are

particularly useful for evaluating the model's dynamic specification. Since they can be defined for each time-varying parameter, analyzing their autocorrelation function (ACF) helps assess whether any refinements are necessary in the dynamics of specific parameters to better capture their patterns.

To illustrate this, let's consider fitting a Gaussian score-driven model with a time-varying mean parameter, while being uncertain about whether the variance parameter also needs to vary over time. One approach would be to fit a model with only the mean parameter μ varying over time. After estimating the model, analyzing the ACF of the conditional score residuals associated with the variance parameter will indicate if the model formulation was correct or if σ^2 should also be considered as time-varying. Additionally, the ACF plot will reveal any remaining structures present in the residuals, suggesting which dynamics should be considered for the parameters.

In conclusion, the analysis of residuals plays a crucial role in evaluating the effectiveness of time series models, including those within the UC score-driven framework. By leveraging the three defined types of residuals outlined in this section, it becomes feasible to conduct thorough diagnostic analyses that efficiently tackle significant challenges in time series modeling and pinpoint potential misalignments in score-driven models.

After assessing the goodness of fit of the specified model, the next step is to leverage this model for making accurate predictions. The next subsection will focus on this critical aspect of time series modeling, exploring the techniques and methodologies used to produce reliable forecasts within the score-driven framework.

2.1.5 Forecast

Forecasting stands as a pivotal and concluding step in time series modeling, wielding significant influence over informed decision-making processes. Moreover, it holds utmost importance during the model selection phase, serv-

ing as a vital component of out-of-sample evaluation by facilitating predictions for validation data sets.

In the context of score-driven models, the one-step-ahead forecast is straightforward as its distribution is derived from the model itself, as shown in Equation 2-1. With an already estimated model, the point forecast can be computed by determining the conditional mean, median, or mode of the predictive distribution. A interesting practice with score-driven models, whenever possible, is to define one of the time-varying parameters to represent the mean (μ_t), enabling the expression:

$$E(y_t|\mathbf{Y}_{t-1}, \mathbf{F}_{t-1}; \theta) = \mu_{t|t-1}. \tag{2-19}$$

The primary challenge with this approach is that for most score-driven formulations, there is no analytical expression for the integral associated with Equation 2-19. To address this limitation, the commonly employed solution involves using Monte Carlo simulation [37]. In this approach, given the updated parameters, M observation scenarios are sampled from the predictive distribution, enabling both point and probabilistic forecasts by computing its mean and quantiles from this sample.

The forecasting process becomes more intricate when aiming to perform multi-step-ahead predictions. In contrast to the previous case, obtaining the predictive distribution for a generic k steps ahead is not directly achievable from the model. To illustrate how this distribution can be obtained, let's consider an already fitted score-driven model trained with T observations. For a two-steps-ahead forecast, the correct conditional distribution needs to be determined. For notation simplicity, let's assume that $p(y_{T+2}|\mathcal{F}T) = p(yT+2|\mathbf{Y}T, \mathbf{F}T; \theta)$. The conditional density/probability function two steps ahead is defined as:

$$\begin{aligned} p(y_{T+2}|\mathcal{F}_T) &= \int p(y_{T+2}, y_{T+1}|\mathcal{F}_T) d_{y_{T+1}} \\ &= \int p(y_{T+2}|y_{T+1}, \mathcal{F}_T) p(y_{T+1}|\mathcal{F}_T) d_{y_{T+1}} \quad [1] \\ &= \int p(y_{T+2}|\mathcal{F}_{T+1}) p(y_{T+1}|\mathcal{F}_T) d_{y_{T+1}}. \end{aligned} \tag{2-20}$$

Note that both density/probability functions $p(y_{T+2}|\mathcal{F}T+1)$ and $p(yT+1|\mathcal{F}_T)$ can be obtained from the predictive distribution by setting $t = T+1$ and $t = T+2$, respectively.

Following the same rationale as in Equation 2-20, it is possible to generalize this result and find that the k-steps-ahead density/probability function is given by:

$$p(y_{T+k}|\mathcal{F}_T) = \int \int \cdots \int \prod_{j=1}^{k} p(y_{t+j}|\mathcal{F}_{t+j-1}) d_{y_{t+1}} d_{y_{t+2}} \cdots d_{y_{t+j-1}}, \quad k = 2, 3, \cdots \tag{2-21}$$

However, as with the one-step-ahead case, obtaining an analytical expression for the conditional mean of this distribution is generally infeasible. Hence, the Monte Carlo simulation approach is employed again. For the first step ahead, the simulation procedure has already been discussed. For subsequent steps, the time-varying parameter must be updated considering the model's dynamics for each of the M sampled scenarios. After updating the parameters, a unique observation is sampled from the predictive distribution for each scenario. Repeating this process for all k steps ahead results in sampled distributions for each step, each with M observations. Utilizing these distributions allows the computation of point forecasts using measures like mean, median, or mode, and probabilistic forecasts through the computation of their quantiles.

A more detailed illustration of this simulation process can be found in [37]. In this work, the authors also propose alternative approaches to incorporate parameter uncertainty when constructing predictive intervals.

Up to this point, the main aspects of each step in time series modeling

[1] Derived using the relation $P(A, B|C) = P(A|B, C)P(B|C)$, where $A = y_{T+2}$, $B = y_{T+1}$ and $C = \mathcal{F}_T$

within the score-driven framework have been covered, beginning with model formulation and extending to the forecasting process. The subsequent section will delve into the second topic crucial for this chapter: optimization robustness techniques and their application in time series modeling.

2.2 Robust Optimization Techniques

Before delving into this section, it's crucial to acknowledge that various modeling frameworks, including linear regressions or time series models, are specific instances of optimization problems. In these frameworks, the objective is to find a set of parameters that minimize a given error measure. Effectively managing the uncertainty associated with the process of interest within the model framework is a significant challenge. Understanding more about these uncertainties can be highly beneficial. According to Bertsimas & Dunn[38], the uncertainty in optimization problems can be classified into four types:

(a) Measurement error: Arises from inaccuracies in the measuring process used to obtain data or from errors intentionally introduced for privacy reasons.

(b) Estimation error: Occurs when parameters are estimated based on expert opinions and/or historical data, which contain uncertainty regarding their future values.

(c) Implementation error: In some cases, it may not be possible to implement the exact optimal solution, necessitating the use of approximations that introduce errors to the model.

(d) Errors due to inexact data: Directly linked to the quality of data. Using data with poor quality will result in models with imprecise results.

Probability theory has long been the predominant approach to modeling uncertainties over time. However, many methodologies based on probability

theory encounter scalability issues when dealing with high-dimensional problems. Notably, even important works, including Nobel prize-winning ones, solve only small-scale problems while struggling with their high-dimensional counterparts [38]. The authors speculate that this limitation may be due to the fact that the principles of probability theory were not developed with a strong emphasis on computational efficiency.

In contrast, Optimization Theory has been developed with a focus on computational efficiency, allowing for solutions to multi-dimensional problems. Consequently, Robust Optimization (RO) offers a natural approach to addressing uncertainties in problems. RO diverges from adopting the axioms of probability theory as its foundations, choosing instead to consider the conclusions derived from probability theory as its basis. Illustrative examples of this alternative approach can be found in [38]. While this approach provides robustness, it often comes with a trade-off in optimality.

The trade-off between robustness and optimality is a crucial concept in RO. While RO techniques may sacrifice a small degree of optimality, they substantially enhance the model's robustness, as exemplified in [38]. Further information about robust optimization and its different classes is available in [7].

This work concentrates on one specific robustness technique that introduces a method to enhance robustness against variations in sample data, especially pertinent for time series applications. The subsequent subsections will delve into this methodology in detail.

2.2.1 Sample Robustness

An essential aspect of modern machine learning practice revolves around leveraging available data for training and assessing model performance. The conventional procedure typically entails randomly selecting a subset of the data as the test set, while the remainder is partitioned into training and validation

sets. Model parameters are estimated using the training data, and performance evaluation is conducted on the validation set. After several iterations of this process, the model's accuracy is assessed on the test set.

Despite its widespread use, this approach introduces inherent randomness, leading to uncertainties in the model's parameters. Each realization of the training and validation sets can yield different models. The sensitivity of the model to these realizations can potentially alter the interpretation of the model's parameters, thereby raising questions about the validity of the conclusions derived from it [39].

Bootstrap Aggregating (Bagging) presents itself as an ensemble method that offers a solution to the aforementioned problem [40]. To enhance model accuracy and robustness, Bagging generates different training sets using bootstrapping and estimates the model for each set. The final predictions are aggregated from the results of each model, typically through voting or averaging, depending on the model type considered.

While Bagging has shown satisfactory results in numerous applications, it has the potential to become computationally expensive due to the repeated estimation of the model. As an alternative, Bertsimas & Dunn[39] proposed a method employing RO to enhance model robustness against variations in the training set. The core idea behind this new approach is to allow optimization to select the most challenging subset of the training data, following some criteria, and employ this worst-case scenario to train the model. The rationale is to estimate a model that performs adequately even in the worst-case scenario. For more detailed information and illustrative examples of this approach, please refer to [39].

Translating the concept of model robustness to variations in data to the time series domain, one can associate this with models capable of handling regime changes. Such issues, commonly encountered during time series analysis, involve changes in the series' dynamics and may be triggered by various

factors like economic crises, natural disasters, pandemics, and more. Traditional time series models were not designed to address regime changes. Consequently, proposed methodologies for modeling such data typically involve estimating multiple models or employing explicit models and other techniques to infer these regime changes [41].

Attempting to propose a new approach to handle regime changes, Bertsimas & Paskov[28] demonstrated that the methodology proposed in [39] can be utilized to make time series models robust against regime changes in the data. It is important to mention that in their paper, Bertsimas & Paskov[28] only considered an AR(p) model, but here, we will consider a generic version of this approach. Before delving into discussing how this model works, it is necessary to understand how a generic time series model can be expressed as an optimization problem. Equation 2-22 illustrates this formulation.

$$\underset{\boldsymbol{\theta}\in\Theta}{\text{Min}} \quad \sum_{t=1}^{T} g(y_t, \boldsymbol{\theta}) \tag{2-22}$$

where $\boldsymbol{\theta}$ represents the vector of the model's parameters that may vary over time, and $g(y_t, \boldsymbol{\theta})$ denotes a general loss function dependent on the observed time series (y_t) and $\boldsymbol{\theta}$. It is crucial to note that depending on the model, the set Θ may either characterize constraints or describe the time-varying nature of the parameters.

Expanding on the concept proposed by Bertsimas & Dunn[39], which entrusts optimization with selecting the worst subset of training data, it is feasible to extend Equation 2-22 to:

$$\underset{\boldsymbol{\theta}\in\Theta}{\text{Min}} \quad \underset{\mathbf{z}\in Z}{\text{Max}} \quad \sum_{t=1}^{T} z_t g(y_t, \boldsymbol{\theta}), \tag{2-23}$$

The rationale behind Equation 2-23 is to train the model using the observed values that maximize the error. In other words, for a given model with optimized parameters, the inner maximization problem selects the sub-sample that leads to the highest error for this model. This selection is determined by

the binary variable z_t, which belongs to the set Z and governs the criteria for selection. Note that when $z_t = 1$, it implies that y_t will be included in the training set. Solving this problem yields a model trained on the most challenging training set under certain criteria, resulting in improved robustness and stability during regime changes.

Bertsimas & Paskov[28] presented two different methods for defining the set Z. This work will concentrate on the definition that yields the most challenging sub-sample of length K. This is accomplished by incorporating two constraints into the inner maximization problem, as depicted in Equation 2-26. The rationale behind this selection is supported by the observation that this definition resulted in more robust models compared to the alternative as showed in [28].

Note that since the inner maximization problem is linear in $\mathbf{z}$, solving the problem in Equation 2-23 is equivalent to optimizing over the convex hull of Z [39], which is defined as:

$$conv(Z) = \left\{ z : \sum_{t=1}^{T} z_t = K, 0 \leq z_i \leq 1, t \in [1, T] \right\}. \tag{2-24}$$

Reformulating Equation 2-23 to consider the convex hull of Z, it is possible to find the following definition:

$$\underset{\boldsymbol{\theta} \in \Theta}{\text{Min}} \quad \underset{\mathbf{z} \in conv(Z)}{\text{Max}} \quad \sum_{t=1}^{T} z_t g(y_t, \boldsymbol{\theta}), \tag{2-25}$$

Indeed, this modification implies that the $\mathbf{z}$ variable can be defined in the range $[0, 1]$. This change simplifies the model since there is no longer a need to deal with integer variables.

Regardless of how Z and its convex hull are defined, Equation 2-25 cannot be solved in its current formulation due to the presence of a maximization inside a minimization problem. To address this issue, it is important to notice that this formulation belongs to the class of Robust Optimization (RO) problems. A review of this theme can be found in [42]. The

two most used methods for solving this class of problems are reformulation to a deterministic optimization problem, called the robust counterpart, or a cutting-plane method [28]. For this specific problem, reformulating using duality principles is possible. The idea behind this is to transform the min-max problem into a computationally tractable minimization problem.

To effect this reformulation, duality theory is employed to convert the maximization problem into a minimization one. This process involves introducing dual variables:

$$\begin{aligned} \underset{z_t}{\text{Max}} \quad & \sum_{t=1}^{T} z_t g(y_t, \boldsymbol{\theta}) \\ \text{s.t.} \quad & \sum_{t=1}^{T} z_t = K, && \times(\delta) \\ & 0 \leq z_t \leq 1, \quad \forall t = 1 \cdots T && \times(u_t) \end{aligned} \tag{2-26}$$

By adding the product of each constraint and its corresponding dual variable as presented in Equation 2-26, it is possible to obtain:

$$\sum_{t=1}^{T} z_t g(y_t, \boldsymbol{\theta}) \leq \delta \sum_{t=1}^{T} z_t + \sum_{t=1}^{T} z_t u_t \leq K\delta + \sum_{t=1}^{T} u_t \tag{2-27}$$

Therefore, this inequality holds if and only if $(\delta + u_t) \geq g(y_t, \boldsymbol{\theta})$ for all $t = 1, \cdots, T$. Here, it is important to remember that the vector of parameters $\boldsymbol{\theta}$, which is considered to be given here, may vary over time. Consequently, the robust counterpart of the problem stated in Equation 2-25 is given by:

$$\begin{aligned} \underset{\boldsymbol{\theta} \in \Theta, \delta \geq 0, u_t \geq 0}{\text{Min}} \quad & K\delta + \sum_{t=1}^{T} u_t \\ \text{s.t.} \quad & \delta + u_t \geq g(y_t, \boldsymbol{\theta}) \qquad \forall t = 1, \cdots, T \end{aligned} \tag{2-28}$$

In the robust model presented in Equation 2-28, additional variables and constraints were introduced to account for robustness considerations. The parameter δ represents the worst-case deviation from the considered loss function, and u_t are non-negative slack variables. The objective function includes a penalty term $K\delta$ and a sum of slack variables $\sum_{t=1}^{T} u_t$ to ensure that the worst-case deviations and slacks are minimized.

The constraints enforce that the sum of δ and u_t is greater than or equal to the loss function considered, $g(y_t, \boldsymbol{\theta})$, at each time point. This ensures that the robust model provides a good fit to the observed data while accounting for potential deviations. By solving the optimization problem in Equation 2-28, it is possible to estimate the parameters $\boldsymbol{\theta}$ of the robust time series model while considering robustness aspects. Furthermore, it is relevant to highlight that the inclusion of the robustness mechanisms does not change the class of the optimization problem that represents the time series model. In other words, this methodology improves the model's robustness without adding complexity to the problem.

By following the manipulations demonstrated previously, the robust time series model presented in Equation 2-25 has been reformulated into a computationally tractable form, which can be readily solved using commercial solvers. It is important to note that the development outlined in this subsection considers a general formulation for time series models, making it applicable to various types of methodologies, including the score-driven framework.

3
Proposed methodology

This chapter introduces a distinctive approach to modeling time series using UC score-driven models, representing them as non-linear optimization problems. Leveraging the inherent characteristics of such problems, including the formulation of constraints, opens avenues to explore diverse possibilities in specifying this type of models.

As previously demonstrated by Equation 1-2, the formulation of a score-driven model as an optimization problem is elegantly straightforward. Its fundamental concept involves delineating both fixed and time-varying parameters, as well as the components, as decision variables within the optimization framework. The temporal dynamics and possible restrictions regarding the fixed parameters' domain are ensured by constraints embedded in the problem's formulation. Finally, an objective function is introduced to minimize the negative logarithm of the likelihood of the predictive distribution, a pivotal step considering that score-driven models are typically estimated through maximum likelihood estimation.

For further clarity, let's consider $l(y_t; f_{t|t-1}, \theta)$ as the logarithm of the likelihood of the generic predictive distribution described in Equation 2-1, with the fixed parameter θ assumed to be positive. The local linear trend score-driven model can be expressed as the following optimization problem:

$$
\begin{aligned}
\underset{\mathbf{f},\mathbf{m},\mathbf{b},\theta,\kappa_m,\kappa_b}{\text{Min}} \quad & -\sum_{t=1}^{T} l(y_t; f_{t|t-1}, \theta) \\
\text{s. t.} \quad & f_{t|t-1} = m_{t|t-1}, && \forall t = 1, \cdots, T \\
& m_{t|t-1} = m_{t-1|t-2} + b_{t-1|t-2} + \kappa_m s_{t-1}, && \forall t = 1, \cdots, T \\
& b_{t|t-1} = b_{t-1|t-2} + \kappa_b s_{t-1}, && \forall t = 1, \cdots, T \\
& \theta > 0 \\
& \kappa_m, \kappa_b \geq 0.
\end{aligned}
\tag{3-1}
$$

This illustration underscores that different dynamics can be captured by simply adjusting the model's constraints.

It is important to highlight that in Equation 3-1, the scaled score was not considered as a decision variable because it is a function of the parameters of the predictive distribution and of the observations, which are decision variables. Thus, the most correct notation for this term would be $s_t(y_{t-1}, f_{t-1|t-2}, \theta)$. However, for simplicity in notation, it was represented as s_t. Further discussions on implementation aspects will shed light on why the scaled score does not need to be a decision variable.

Another interesting aspect of this new approach is that the problem illustrated in Equation 3-1 is an optimization problem in which the number of decision variables and constraints depends on the size of the time series of interest (T). This has the potential to become a considerably large problem. On the other hand, this formulation allows for the initialization of all decision variables of the model. This means that this new methodology enables the use of initial values for both time-varying parameters and components for every period of time, rather than just for the first one, as is typically done.

This new way of initializing the model has a considerable impact on the computational efficiency of the model. It also helps to prevent the model from becoming stuck at a local minimum, since there is no guarantee of global optimality when dealing with nonconvex problems. More details on how this initialization is performed will be given in the following sections.

3.1 Robustness Features

As previously discussed, various optimization techniques can be incorporated into the definition of time series models to potentially enhance their performance. Examples include the use of integer optimization for exogenous variable selection and robust optimization techniques. Given the objectives of this work, which involve evaluating the impact of the robustness techniques

discussed in Section 2.2 on model performance, this section aims to describe how these features are included in the context of UC score-driven models.

3.1.1 Sample Robustness

In pursuit of robustness against regime changes in time series, the methodology proposed by Bertsimas & Paskov[28] is integrated directly into the model's formulation. In practice, to transform a score-driven model specified as a optimization problem into its robust counterpart, the following modifications are applied: a change in the objective function and the introduction of a robustness constraint, both defined in terms of dual variables. To illustrate, consider the exemplary model presented in Equation 3-1. Its robust counterpart takes the following form:

$$
\begin{aligned}
\underset{\substack{\mathbf{f},\mathbf{m},\mathbf{b},\mathbf{u},\\ \theta,\kappa_m,\kappa_b,\delta}}{\text{Min}} \quad & K\delta + \sum_{t=1}^{T} u_t \\
\text{s. t.} \quad & \delta + u_t \geq -l(y_t; f_{t|t-1}, \theta), && \forall t = 1, \cdots, T \\
& f_{t|t-1} = m_{t|t-1}, && \forall t = 1, \cdots, T \\
& m_{t|t-1} = m_{t-1|t-2} + b_{t-1|t-2} + \kappa_m s_{t-1}, && \forall t = 1, \cdots, T \\
& b_{t|t-1} = b_{t-1|t-2} + \kappa_b s_{t-1}, && \forall t = 1, \cdots, T \\
& \theta > 0 \\
& \kappa_m, \kappa_b \geq 0.
\end{aligned}
\tag{3-2}
$$

3.2 The SDUC.jl package

This section introduces SDUC.jl (Score-Driven Unobserved Components), an innovative open-source package currently in development. Built entirely in the Julia language [22], SDUC.jl arises directly from this research, aiming to empower users to specify a diverse range of models. With SDUC.jl, users enjoy the flexibility to select the predictive distribution, scaling parame-

ter, time-varying parameters, and latent components that govern the model's dynamics. Moreover, the package conceptualizes the model as mathematical programming problems, facilitating the integration of optimization techniques into its formulation.

The decision to utilize Julia as the development language for this package is grounded in its distinct advantages. Julia offers a high-level programming syntax conducive to efficient prototyping and development, while still maintaining computational performance. Additionally, Julia provides a robust set of tools for tackling optimization challenges. JuMP.jl [43], an open-source modeling language, streamlines the formulation of various optimization problems, guiding the model to a designated solver and presenting results in a user-friendly format. It's crucial to emphasize that JuMP.jl serves as a conduit between user specifications and solvers. By appropriately selecting a solver, JuMP.jl empowers users to address diverse classes of optimization problems, including linear, mixed-integer, semidefinite, non-linear, and more.

The proposed package utilizes JuMP's modeling language to formulate various score-driven models as optimization problems. Figure 3.1 illustrates a simplified version of how the model presented in Equation 3-1 can be implemented using JuMP.jl. As discussed earlier, SDUC.jl provides flexibility in model formulation by enabling the customization of various components. Concerning the trend component, the package supports different modeling options, including a random walk, a random walk with slope, or an AR(1) process. These diverse approaches to representing the trend component allow for modeling time series with various trends, such as growth or decline. This can be achieved through the utilization of a local linear trend model for capturing local trends, stationarity via AR(1), or even capturing more random patterns via a local level model.

```
function SDM_local_linear_trend(y::Vector{Float64}, d::Float64)

    T = length(y)

    model = Model(Ipopt.Optimizer)
    set_optimizer_attribute(model, "print_level", 0)

    @variable(model, μ[1:T])
    @variable(model, m[1:T])
    @variable(model, b[1:T])
    @variable(model, σ_2 >= 1e-6)
    @variable(model, κ_m >= 1e-6)
    @variable(model, κ_b >= 1e-6)

    set_start_value.(μ, y)

    @NLexpression(model, s[t = 2:T], fisher_information(σ_2, d) * score(σ_2, μ[t - 1], y[t - 1]))

    @constraint(model, [t = 1:T], μ[t] == α[t])
    @NLconstraint(model, [t = 2:T], m[t] == m[t - 1] + b[t - 1] + κ_m * s[t])
    @NLconstraint(model, [t = 2:T], b[t] ==  b[t-1] + κ_b * s[t])

    @NLobjective(model, Max, sum(-0.5 * log(2 * π * σ_2) - ((y[t] - μ[t])^2)/(2 * (σ_2)) for t = 1:T))
    optimize!(model)
    @info termination_status(model)

    return value.(μ), value(σ_2), value.(m), value.(b), value(κ_m), value(κ_b)
end
```

Figure 3.1: llustrative code demonstrating how a Gaussian local linear trend SDM can be implemented using JuMP.

The seasonal component is characterized by the harmonic specification detailed in Equation 2-11. By enabling users to specify the seasonal period, the package effectively accommodates time series with various temporal granularities. For a seasonal period S, the number of considered harmonics equals $\frac{S}{2}$ for even S, or $\frac{S-1}{2}$ otherwise. Including all harmonics ensures that the seasonal factor aligns with outcomes that would be obtained if seasonal dummies were used. Importantly, although users can model series with different seasonal periods (monthly, weekly, daily, etc.), the package only allows one seasonal component to be defined, thus preventing the modeling of time series with multiple seasonal patterns.

While seasonality is crucial for modeling many time series, treating it as stochastic may increase model complexity due to the numerous decision variables and constraints associated with its specifications. With this consideration, the proposed package also permits the inclusion of a deterministic version of harmonic seasonality, as defined in Equation 3-3. This alternative approach allows for the consideration of the seasonality component without af-

fecting the model's complexity. In cases where this deterministic version proves inadequate for modeling the seasonal pattern of the data, users can resort to the stochastic formulation.

$$\begin{aligned} h(f_{t|t-1}) &= S_{t|t-1} \\ S_{t|t-1} &= \sum_{j=1}^{6} \left\{ \gamma_j cos\left(\frac{2\pi jt}{12}\right) + \gamma_j^* sin\left(\frac{2\pi jt}{12}\right) \right\} \end{aligned} \tag{3-3}$$

As discussed in Subsection 2.1.2.1, an autoregressive process can serve both as the trend or as a standalone component, typically aiding the model in handling short-term dependencies that were not adequately modeled by other components. With this in mind, an autoregressive component can be defined by explicitly specifying the orders to be considered, without necessitating the inclusion of all orders up to p. This feature allows for the utilization of the AR(p) component to effectively represent seasonality, with p set equal to the seasonal period. This aspect enables the accurate representation of multiple seasonalities by either merging the seasonal component with the autoregressive one or exclusively utilizing the autoregressive component with appropriate orders.

Furthermore, the proposed package facilitates the integration of explanatory variables into the dynamics of the mean parameter. This capability enables users to incorporate exogenous influences into the model specification, potentially improving predictive performance. It is essential to note that in this initial version of the package, the coefficients of the explanatory variables can only be specified as fixed parameters. However, the possibility of specifying dynamics for these parameters has already been discussed.

In addition to flexibility in defining parameter dynamics, the JuMP.jl package offers other noteworthy features. The first of these is the utilization of what JuMP.jl terms "expressions." These objects are essentially functions defined based on the model's decision variables and can be incorporated into its formulation. For score-driven models, these expressions are particularly

valuable for representing the scaled score without requiring them to be defined as decision variables, thereby preventing an increase in the model's complexity.

Another significant feature is the integration of JuMP.jl with the ForwardDiff.jl package [44], capable of symbolically calculating derivatives of the expression defined as the objective function of the optimization problem. In the context of score-driven models, this means that JuMP.jl has access to the exact derivatives of the considered log-likelihood function. Consequently, there is no longer a need to rely on numerical optimization methods such as Nelder-Mead [5] and L-BFGS [6]. Therefore, JuMP.jl enables the utilization of state-of-the-art nonlinear optimization methods based on interior points. These methodologies can be accessed through the Ipopt.jl package, which acts as a wrapper for the Ipopt algorithm [45].

The differentiation capability provided by the ForwardDiff.jl package can offer significant advantages by automatically deriving expressions for the score function and Fisher's information matrix for a given distribution. Although this functionality was not implemented in the initial version of the package, acknowledging the potential associated with its inclusion is crucial for expanding the range of available distributions, thereby enhancing the applicability of this package across various scenarios. Given its significance, the exploration of such an approach will be prioritized in future iterations of the package.

A significant challenge in nonlinear optimization is the absence of guarantees regarding global optimal points. Consequently, the initialization process becomes crucial to prevent the model from becoming trapped in local optima. In conventional implementations of score-driven models, initialization typically concentrates on fixed parameters and initial values of time-varying components. However, as discussed earlier, the optimization-based approach allows for an enhanced initialization process.

It is crucial to acknowledge that since both fixed and time-varying

parameters, as well as components, are specified as decision variables, the solver's algorithm does not inherently recognize their temporal dependencies. Furthermore, during an iteration of the interior point method, the entire vector of decision variables is updated simultaneously across all time periods. These intricacies enable the model to incorporate initial values for time-varying parameters and components throughout all time periods, significantly mitigating issues associated with local optima. Importantly, the SDUC.jl package not only provides its algorithm for accurate initial values but also allows users to employ their custom initialization procedures.

The initial phase of the package's initialization involves acquiring good initial values for the parameters of the predictive distribution. For a time-varying mean parameter, it is logical to utilize the observed time series itself. Conversely, for other parameters, heuristic methods can be valuable. For instance, in the case of a monthly time series, non-conditional estimates for the desired parameter can be calculated for each month, and these values can be utilized to construct a time series by appropriately repeating them within the respective months. Such non-conditional estimates may also function as initial values for fixed parameters if applicable.

Once the time series of initial values for time-varying parameters has been obtained, the estimation of the components driving their dynamics can be carried out using suitable time series models such as State Space models. In this context, the StateSpaceModels.jl package [46] has been utilized. This package facilitates modeling, forecasting, and simulating time series within a state-space framework.

By leveraging the similarity of available components, a structural model that mirrors the dynamics of the specified score-driven model is fitted to the initial values of time-varying parameters. The resulting estimates are then used to initialize the components of the UC score-driven model. It's worth noting that since StateSpaceModels.jl does not support an AR component,

the initialization of such a component must be conducted in two stages, if necessary. For score-driven models featuring both an autoregressive component and another component, an AR model written as an optimization problem with the same order as the score-driven AR component is estimated on the innovations of the structural model. Conversely, if the desired model solely includes the AR component, the optimization-based AR model is directly estimated on the initial parameter values. In either scenario, the estimated values are used for initialization. Finally, the fixed parameters, which multiply the scaled score, are initialized to 0.02, similar to the practice in the Time Series Lab Score Edition software [47].

This multi-step initialization procedure has proven effective in speeding up the optimization process and safeguarding the model from converging to suboptimal solutions. While the package features an efficient initialization mechanism, as previously mentioned, it remains open to user-defined initial values that can be tailored to meet the specific requirements of various applications.

After fully specifying the optimization problem and initializing it following the above process, the model is ready for optimization. The package enables users to customize this step by defining the maximum number of iterations for the interior points method and the time limit within which the algorithm must find the optimum. These simple adjustments allow users to tailor the estimation process more precisely to their applications.

All discussions regarding the model's formulation and initialization remain consistent irrespective of the model's dynamics and predictive distribution, except for the location-scale t distribution. Certain unique characteristics of this distribution required the development of a distinct approach to accurately estimate models based on it. Further details about this approach can be found in Appendix A.

After exploring how modern optimization techniques can enhance the

performance of time series models and how the proposed package enables the application of this new approach, it is crucial to understand how to use it. The following subsection will utilize illustrative examples to demonstrate how SDUC.jl can be employed to model a time series.

3.2.1 Illustrative examples of the package

As mentioned earlier, this subsection comprises small experiments aimed at illustrating the time series modeling process using the proposed package. Alongside presenting the obtained results, examples of the code used to define, estimate, and make predictions with a UC score-driven model will also be discussed. For these demonstrations, three well-known time series datasets were utilized. Further information about these time series can be found in [34].

The first example aims to demonstrate how one can utilize the proposed package to obtain the unobserved components of a time series. To achieve this, the "Airline Passengers" time series was utilized. This dataset showcases the monthly totals of international airline passengers from 1949 to 1960.

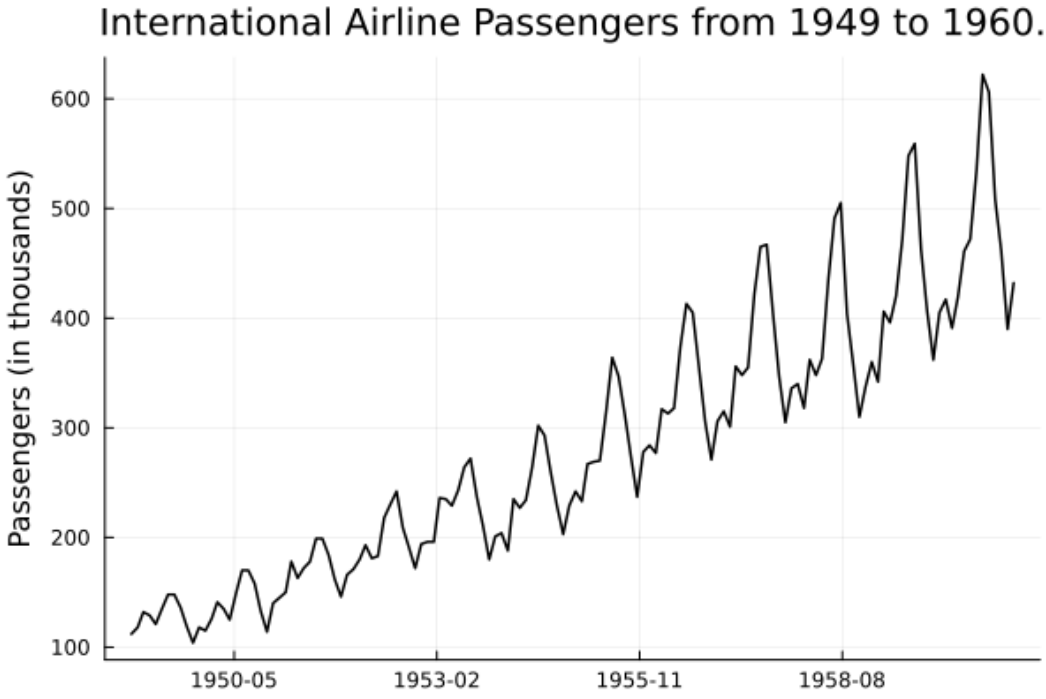

Figure 3.2: Observed data from the "Airline Passengers" time series.

As depicted in Figure 3.2, this series exhibits a prominent increasing trend and strong seasonal patterns. It's noteworthy that the variance of these data appears to increase over time, likely due to the escalation of seasonal peaks. This characteristic complicates the modeling process, often necessitating a logarithmic transformation of the data. By taking the logarithm of this series, it becomes possible to stabilize the variance without altering the presented trend characteristics, as illustrated in Figure 3.3.

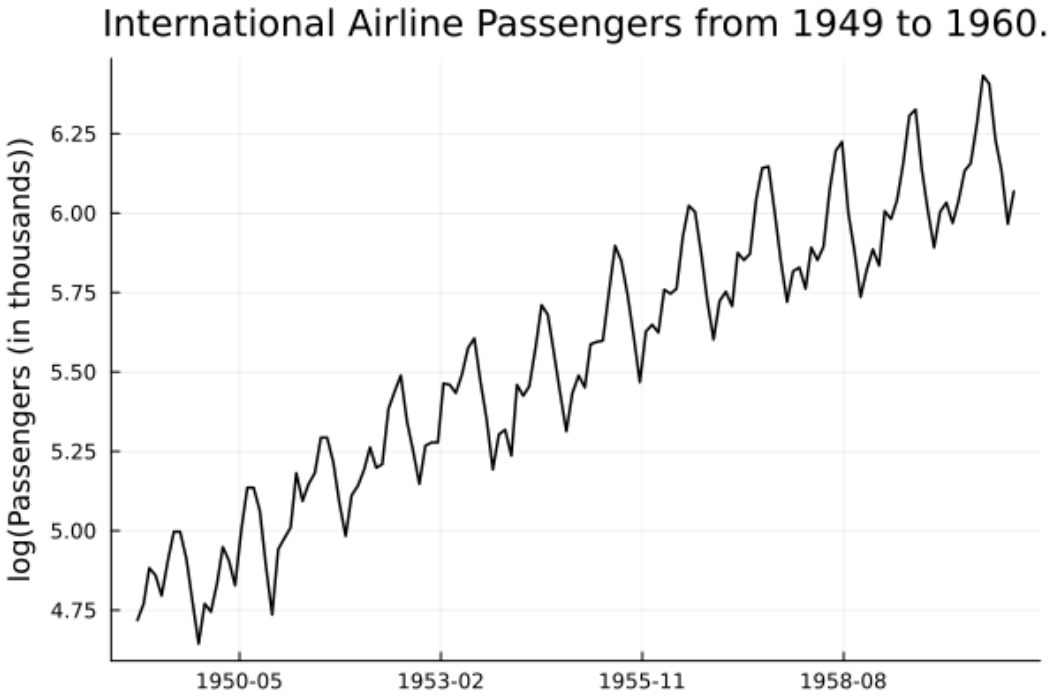

Figure 3.3: The logarithm of the observed data from the "Airline Passengers" time series.

To properly address the previously discussed characteristics of this data, an appropriate model must be specified. For this purpose, a model based on the Normal distribution, with the mean parameter being time-varying, was defined. The dynamics of this parameter are governed by a random walk with slope level and a stochastic monthly seasonality. Equation 3-4 presents the complete formulation of the model, where $l_1(y_t; \mu_{t|t-1}, \sigma^2)$ represents the logarithm of the likelihood of the Normal predictive distribution.

$$
\begin{aligned}
&\underset{\substack{\boldsymbol{\mu},\mathbf{m},\mathbf{b},\mathbf{S},\boldsymbol{\gamma},\boldsymbol{\gamma}^*,\\ \sigma^2,\kappa_m,\kappa_b,\kappa_S}}{\text{Min}} && -\sum_{t=1}^{T} l_1(y_t;\mu_{t|t-1},\sigma^2) && \\
&\text{s. t.} && \mu_{t|t-1} = m_{t|t-1} + S_{t|t-1}, && \forall t = 1,\cdots,T \\
&&& m_{t|t-1} = m_{t-1|t-2} + b_{t-1|t-2} + \kappa_m s_{t-1}, && \forall t = 1,\cdots,T \\
&&& b_{t|t-1} = b_{t-1|t-2} + \kappa_b s_{t-1}, && \forall t = 1,\cdots,T \\
&&& S_{t|t-1} = \sum_{j=1}^{6} \gamma_{j,t|t-1}, && \forall t = 1,\cdots,T \\
&&& \begin{bmatrix} \gamma_{j,t|t-1} \\ \gamma^*_{j,t|t-1} \end{bmatrix} = \begin{bmatrix} cos(\frac{2\pi jt}{12}) & sin(\frac{2\pi jt}{12}) \\ -sin(\frac{2\pi jt}{12}) & cos(\frac{2\pi jt}{12}) \end{bmatrix} \begin{bmatrix} \gamma_{j,t-1|t-2} \\ \gamma^*_{j,t-1|t-2} \end{bmatrix} + \begin{bmatrix} \kappa_S s_{t-1} \\ \kappa_S s_{t-1} \end{bmatrix}, && \forall t = 1,\cdots,T \\
&&& \sigma^2 > 0 && \\
&&& \kappa_m, \kappa_b, \kappa_S \geq 0 &&
\end{aligned}
\tag{3-4}
$$

where $s_t = y_t - \mu_{t|t-1}$, for all $t = 1,\cdots,T$, and y_t represents the observation of the series at time t.

Code 1 illustrates how to specify the model presented in Equation 3-4 using the proposed package.

Code 1: Model's Specification

```
using SDUC

#-------------------------------------------
# Defining model's dynamic
#-------------------------------------------

distribution        = SDUC.NormalDistribution()
time_varying_params = [true, false];
d                   = 1.0;
level               = ["random walk slope", ""];
seasonality         = ["stochastic 12", ""];
ar                  = missing;

#-------------------------------------------
# Creating GAS model object
#-------------------------------------------

model = SDUC.GASModel(distribution, time_varying_params, d, level,
    seasonality, ar)
```

It is worth noting that the `time_varying_params` object considers the parameter's order as it appears in the `distribution` object. Therefore, to specify only the Normal's variance as time-varying, this object would be defined as `[false, true]`. This order also corresponds to the keys of the dictionary indicating the components' dynamics. Lastly, there is no need to create key-value pairs for fixed parameters.

After properly defining and estimating this model, obtaining the fitted values for the components is straightforward. Figure 3.4 showcases the results of this example, where it's noticeable that the component estimates are well-behaved and aligned with the observed data. The trend component consists of a level with almost constant growth and a nearly deterministic slope, while the seasonality, although well-behaved, exhibits some level of stochasticity.

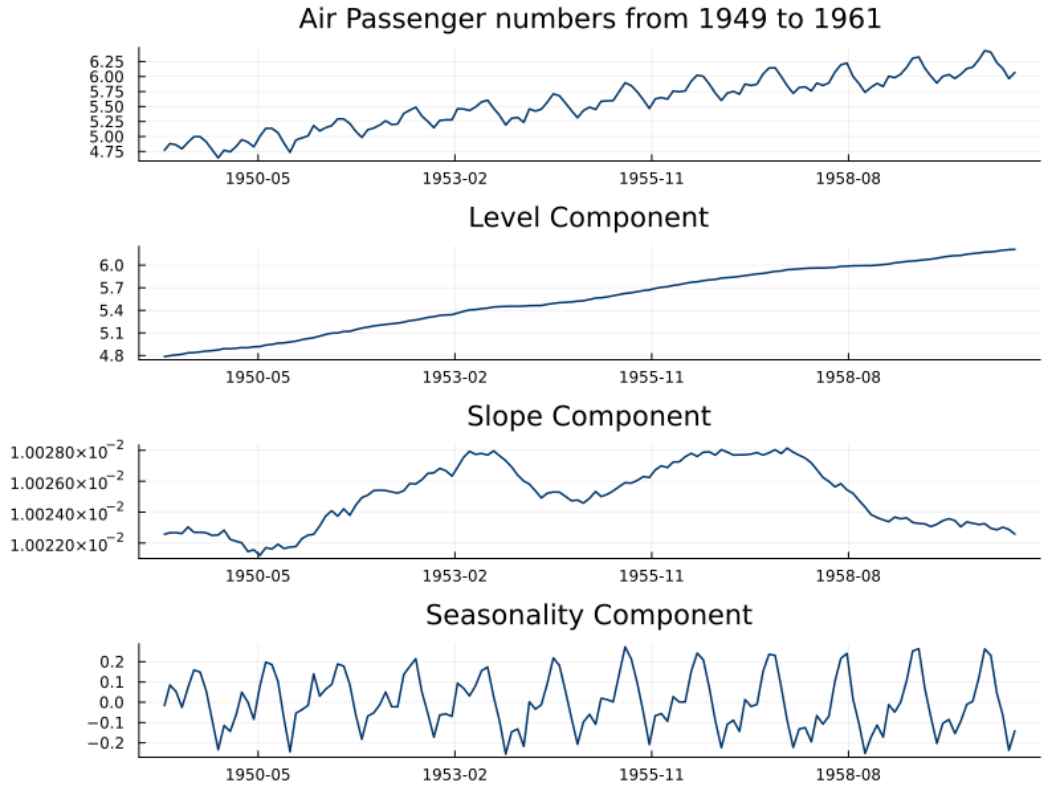

Figure 3.4: Fitted components for the logarithm of the "Airline Passengers" time series.

In addition to analyzing its components, another crucial step in the time series modeling process is examining the model's residuals. This step aims

to verify the validity of the model's assumptions, such as the dynamics and distribution adopted. To address this, a second example was conducted using another time series.

The "House Sales" time series represents monthly sales of new one-family houses sold in the USA since 1973. This series does not exhibit a discernible trend (as shown in Figure 3.5). However, it does display a notable seasonal pattern. To model this data, a model featuring a level component with a random walk dynamic and stochastic monthly seasonality was utilized.

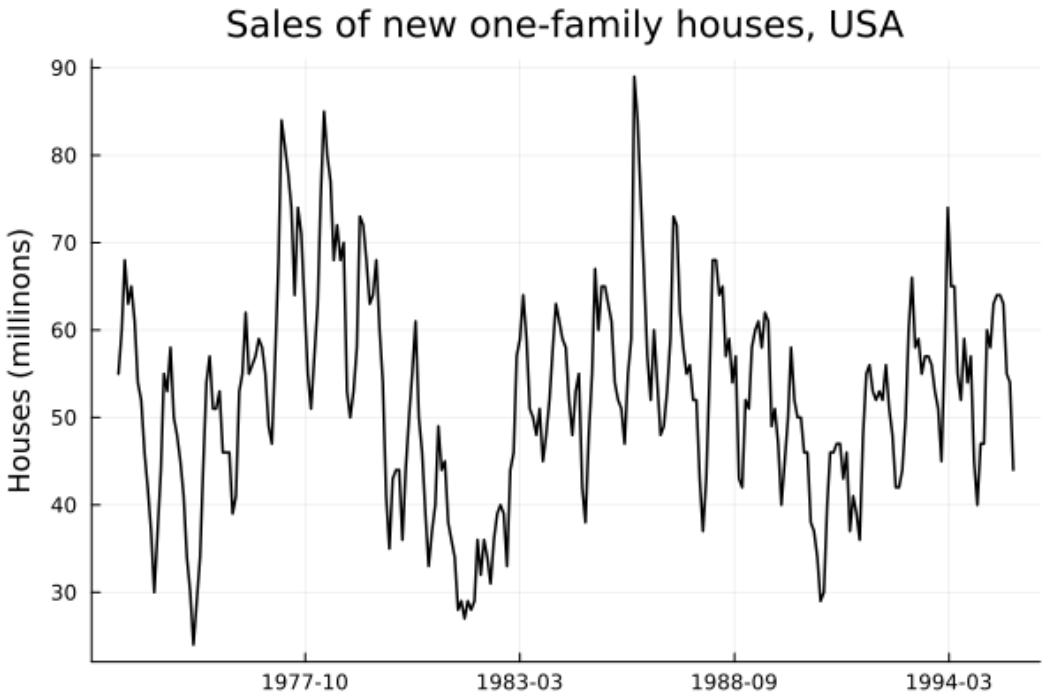

Figure 3.5: Observed data from the "House Sales" time series.

The considered model is based on a t-Student distribution with location and scale, with only the mean parameter exhibiting time-varying behavior, and a scale parameter $d = 1$. Equation 3-5 presents the complete formulation of the model, where $l_2(y_t; \mu_{t|t-1}, \sigma^2)$ represents the logarithm of the likelihood of the t-Student distribution with location and scale predictive distribution.

$$
\begin{aligned}
&\underset{\substack{\boldsymbol{\mu},\mathbf{m},\mathbf{S},\boldsymbol{\gamma},\boldsymbol{\gamma}^*,\\ \sigma^2,\kappa_m,\kappa_S}}{\text{Min}} && -\sum_{t=1}^{T} l_2(y_t;\mu_{t|t-1},\sigma^2) && \\
&\text{s. t.} && \mu_{t|t-1} = m_{t|t-1} + S_{t|t-1}, && \forall t = 1,\cdots,T \\
&&& m_{t|t-1} = m_{t-1|t-2} + \kappa_m s_{t-1}, && \forall t = 1,\cdots,T \\
&&& S_{t|t-1} = \sum_{j=1}^{6} \gamma_{j,t|t-1}, && \forall t = 1,\cdots,T \\
&&& \begin{bmatrix} \gamma_{j,t|t-1} \\ \gamma^*_{j,t|t-1} \end{bmatrix} = \begin{bmatrix} cos(\frac{2\pi jt}{12}) & sin(\frac{2\pi jt}{12}) \\ -sin(\frac{2\pi jt}{12}) & cos(\frac{2\pi jt}{12}) \end{bmatrix} \begin{bmatrix} \gamma_{j,t-1|t-2} \\ \gamma^*_{j,t-1|t-2} \end{bmatrix} + \begin{bmatrix} \kappa_S s_{t-1} \\ \kappa_S s_{t-1} \end{bmatrix}, && \forall t = 1,\cdots,T \\
&&& \sigma^2 > 0 && \\
&&& \kappa_m, \kappa_S \geq 0 &&
\end{aligned}
\tag{3-5}
$$

Once the desired model has been fully defined, the next step is to activate JuMP to carry out the optimization process and thus perform the model estimation. Code 2 illustrates how the fit function can be used. First, a `model` object, created similarly to what is showed in Code 1 and the data are passed as inputs. It is important to mention that if there are no explanatory variables, only the `model` and `y` objects need to be provided to the function. The next two inputs controls if the robust counterpart of the model will be used and the penalty the K hyperparameter for the robust model, if used, respectively. Regarding the latter, it is important to note that once the robust version of a model is specified, the K parameter will be set as $K = T \times$ `robust_prop`, where T is the length of the series. Setting the `initial_values` input as missing will make the package use the methodology described in Section 3.2 to calculate the values used to initialize the optimization problem. Users are also allowed to define their own initialization procedure and provide it to the fit function via the `initial_values` input. The remaining inputs control two aspects of the Ipopt solver. The first one indicates the maximum number of iterations that the optimization algorithm can perform, while the second one defines the time limit in seconds for the solver to find the optimal solution.

Code 2: Fitting the Specified Model

```
#-------------------------------------------
# Fitting the specified model
#-------------------------------------------

fitted_model = SDUC.fit(model, y, X; robust = false, robust_prop =
    0.7, initial_values = missing, number_max_iterations = 30000,
    max_optimization_time = 180.0)

#-------------------------------------------
# Accessing some results
#-------------------------------------------

fit        = fitted_model.fit_in_sample
components = fitted_model.components
residuals  = fitted_model.residuals
```

Once the specified model is estimated, it's possible to begin the analysis of its residuals. As previously discussed, the proposed package returns three different types of residuals that provide complementary insights about the model's adequacy. The analysis starts with Figure 3.6, which displays the behavior of the standardized or Pearson residuals over time. From this graph, two relevant results are evident: firstly, these residuals have an average around zero, and secondly, they do not exhibit any notable pattern.

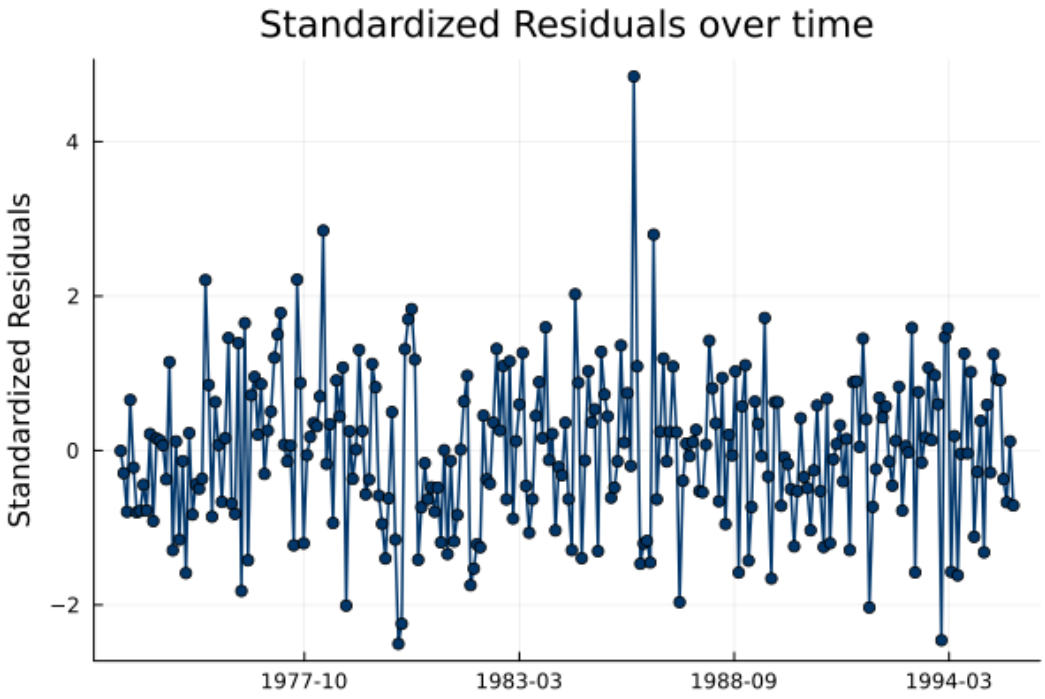

Figure 3.6: Standardize/Pearson Residuals over time.

The Pearson residuals are also useful for determining if the specified model's dynamics were capable of accurately capturing the data patterns. This can be achieved by analyzing its ACF plot, as shown in Figure 3.8. This figure confirms the model's ability to capture series dynamics, as the ACF plot only indicates significant autocorrelation at spurious lags.

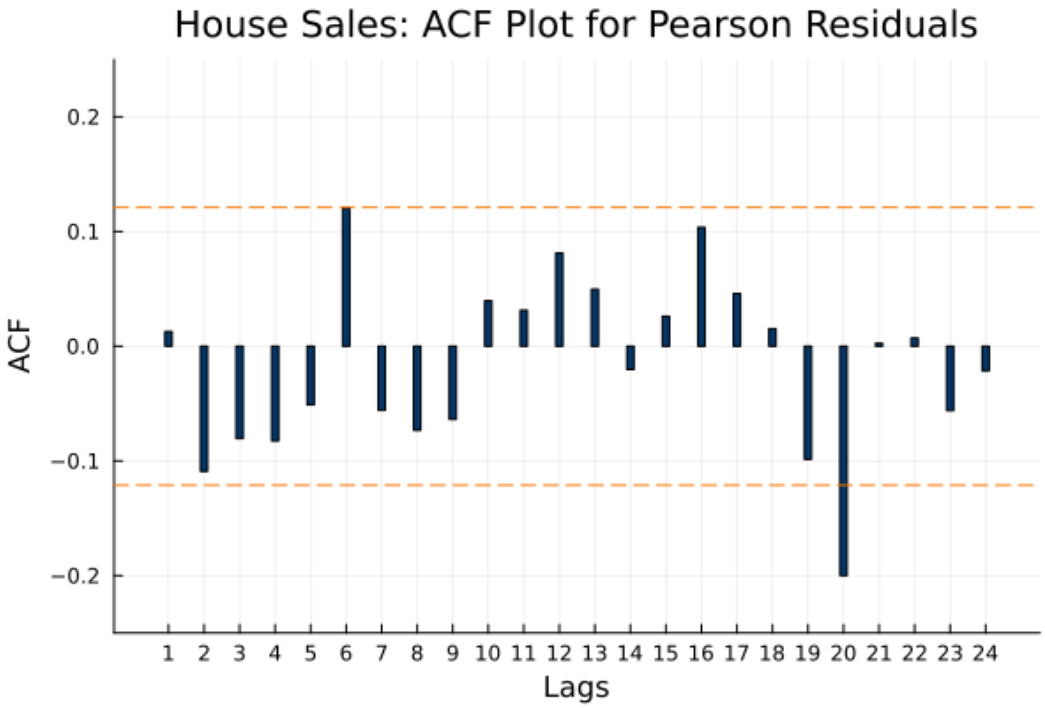

Figure 3.7: ACF plot of Standardize/Pearson Residuals.

As discussed earlier, conditional score residuals play a crucial role in score-driven models as they help verify if the parameter dynamics were correctly specified. From the top chart in Figure 3.7, it can be concluded that the dynamics used for the mean parameter were able to effectively capture the data patterns. It's important to note that the same results presented by this figure and Figure 3.8 were expected, since Pearson residuals are a special case of conditional score residuals when the models consider a distribution from the exponential family [33]. The absence of significant lags in the bottom chart of Figure 3.7, which shows the ACF plot of the conditional score residuals for the scale parameter, suggests that modeling the variance as a time-varying parameter is unnecessary for this dataset.

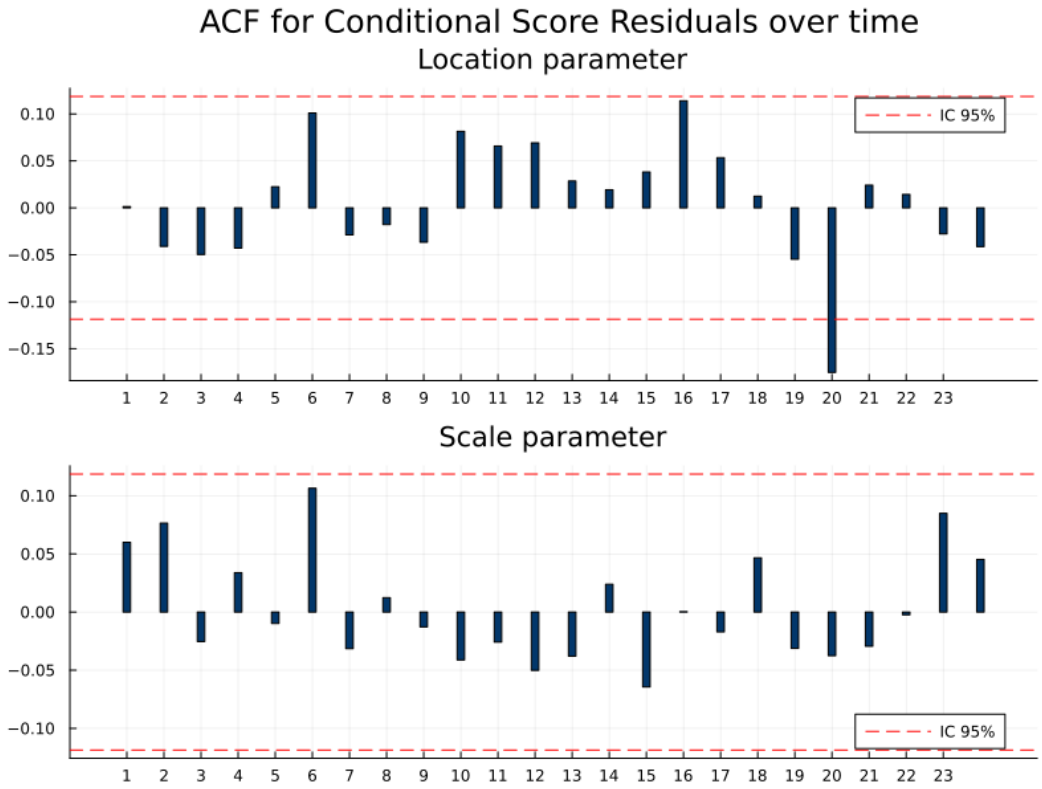

Figure 3.8: ACF plot of Conditional Score Residuals.

In addition to using residuals to evaluate the specified dynamics, it's crucial to assess the choice of the predictive distribution. To accomplish this, a normality test on the quantile residuals using the Jarque-Bera test was conducted, and QQ-Norm plots were generated.

The results of the test (Figure 3.9) indicate the non-Gaussianity of the quantile residuals, suggesting that the t with location and scale distribution is not suitable for modeling this data. However, upon closer analysis of Figure 3.9, it becomes evident that the quantile residuals mostly adhere to the identity line, with a few outliers. This observation suggests that with appropriate treatment of these outliers, the test might indicate that the used distribution is a good choice to model this series.

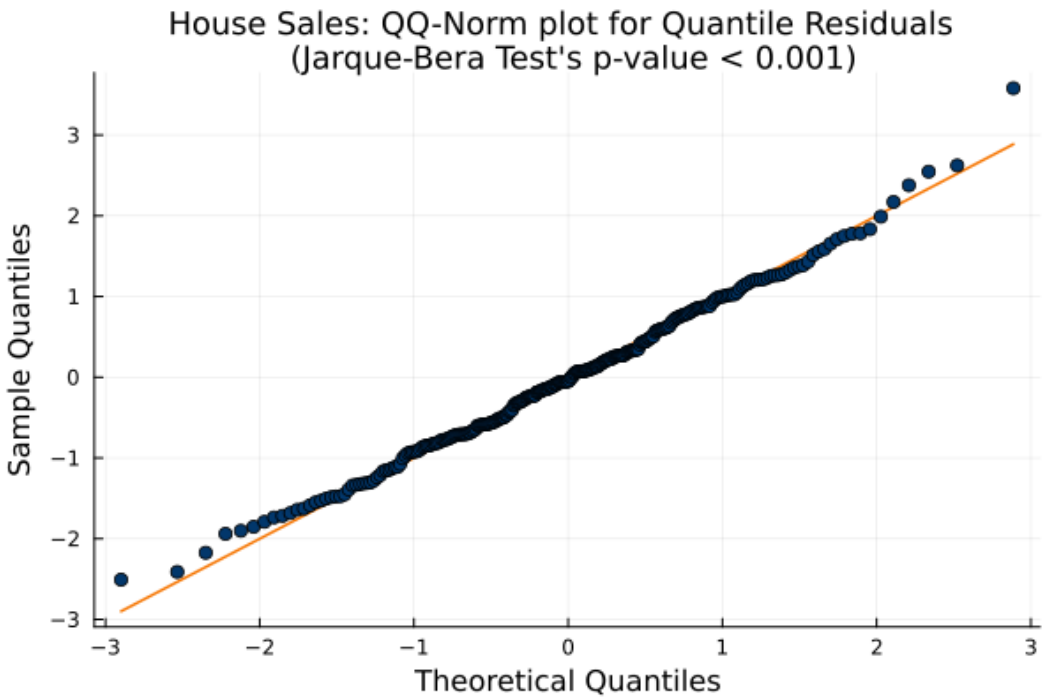

Figure 3.9: QQ-Norm plot of Quantile Residuals and Jarque-Bera Test.

Typically, after confirming the adequacy of the model using the analysis of residuals, the last step of the time series modeling process is making predictions. This third example aimed to demonstrate how predictions can be obtained using the proposed package. For this final example, data on total employment in the USA retail sector from January 1939 to June 2019 were utilized. Figure 3.10 depicts the characteristics of this series, which appears to exhibit a stochastic trend and a strong seasonal pattern. It is important to highlight that, since the objective of this example is to perform predictions, the last two years of observations were set aside as a validation set.

$$
\begin{aligned}
&\underset{\substack{\boldsymbol{\mu},\mathbf{m},\mathbf{b},\mathbf{S},\boldsymbol{\gamma},\boldsymbol{\gamma}^*,\\ \sigma^2,\kappa_m,\kappa_b}}{\text{Min}} \quad -\sum_{t=1}^{T} l_3(y_t;\mu_{t|t-1},\sigma^2) \\
&\text{s. t.} \quad \mu_{t|t-1} = m_{t|t-1} + S_{t|t-1}, \qquad \forall t = 1,\cdots,T \\
&\qquad\quad m_{t|t-1} = m_{t-1|t-2} + b_{t-1|t-2} + \kappa_m s_{t-1}, \qquad \forall t = 1,\cdots,T \\
&\qquad\quad b_{t|t-1} = b_{t-1|t-2} + \kappa_b s_{t-1}, \qquad \forall t = 1,\cdots,T \\
&\qquad\quad S_{t|t-1} = \sum_{j=1}^{6} \left\{ \gamma_j cos\left(\frac{2\pi jt}{12}\right) + \gamma_j^* sin\left(\frac{2\pi jt}{12}\right) \right\}, \qquad \forall t = 1,\cdots,T \\
&\qquad\quad \sigma^2 > 0 \\
&\qquad\quad \kappa_m, \kappa_b \geq 0
\end{aligned}
\tag{3-6}
$$

To correctly model this data, a UC score-driven model based on the Log-Normal distribution was defined, with only the mean parameter exhibiting time-varying behavior, following a random walk with slope plus a deterministic seasonality dynamic. Equation 3-6 presents the complete formulation of the model, where $l_3(y_t; \mu_{t|t-1}, \sigma^2)$ represents the logarithm of the likelihood of the Log-Normal predictive distribution.

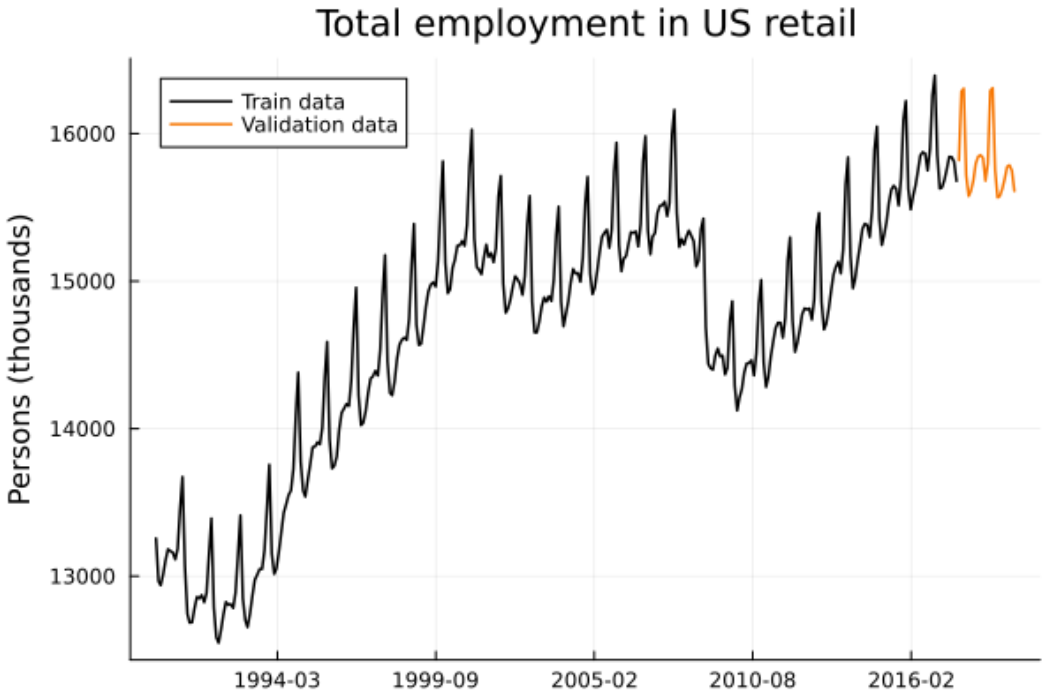

Figure 3.10: Observed data of total employment in the US retail sector.

After correctly estimate this model and check its adequacy, it is possible to use the model to make predictions. Code 3 demonstrates how this procedure can be executed using the proposed package. Once more, it's crucial to emphasize that the `X_forecast` input is only necessary if the model was estimated considering explanatory variables.

Code 3: Performing Predictions

```
#-------------------------------------------
# Making predictions
#-------------------------------------------

steps_ahead = 24
number_of_scenarious = 500

forecast = SDUC.predict(model, fitted_model, y, X_forecast,
    steps_ahead, number_of_scenarious; probabilistic_intervals =
    [0.8, 0.95])

#-------------------------------------------
# Accessing some results
#-------------------------------------------

point_forecast         = forecast["mean"]
probabilistic_forecast = forecast["intervals"]
```

As discussed in Sub-Section 2.1.5, the multi-step-ahead predictions for score-driven models are performed by a simulation procedure. Figure 3.11 compares the model's out-of-sample performance with validation data, considering two different ways of representing the uncertainty of these predictions. The top chart of Figure 3.11 displays this uncertainty as predictive intervals, while the bottom chart shows all the simulated scenarios. Additionally, Figure 3.11 presents the mean of the scenarios as the point predictions. Considering both the point forecast and its uncertainty, it is evident that the model was able to provide predictions very close to the validation data.

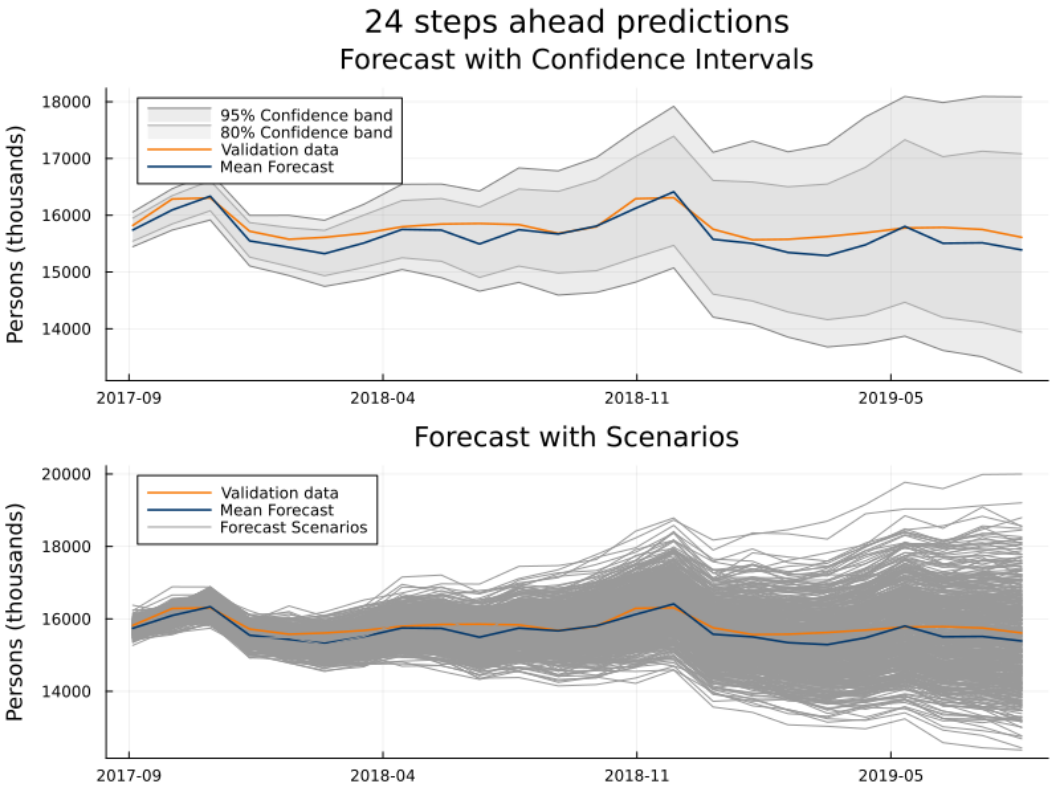

Figure 3.11: Twenty-four steps-ahead forecast with confidence intervals and scenarios.

With these simple examples, it was possible to illustrate how the proposed package works and how users can, with just a few lines of code, estimate various score-driven models. Furthermore, it became evident how the package contributes to all stages of temporal data modeling, ranging from model specification to predictions, facilitating inputs for residual analysis. Now, to truly test the package's capacity in terms of the benefits generated by robust techniques, the next chapter aims to conduct an experiment aimed at achieving this objective.

4
Experimental Results

In this chapter, the results of an experiment obtained through the application of SDUC.jl in scenarios with and without regime changes in the time series will be presented. These experiments are designed to demonstrate the package's performance and evaluate its robustness characteristics.

4.1
SDUC.jl during regime change periods

This experiment aims to assess the additional accuracy gained by implementing robustness in the sampling method within the framework of score-driven models. Conducting this test required considering time series exhibiting regime changes. This aspect of the data was crucial, as the aim was to evaluate the advantages of employing robustness techniques in enhancing the model's performance during these more challenging periods. While data from various contexts could have been considered, it was decided to utilize electrical load data from different Brazilian systems. This decision stemmed from the fact that their historical patterns were disrupted by events resulting from the COVID-19 pandemic, as depicted in Figures 4.1 to 4.4.

Figures 4.1 to 4.4 display data for each system from 2015 to 2021. These figures reveal seasonal patterns throughout the year, accompanied by trends characterized by subtle growth in some systems and more stationary behavior in others. However, starting from 2020, a noticeable departure from the usual patterns emerges, likely attributable to the COVID-19 pandemic. It is important to note that while this shift in data patterns following the pandemic appears to have persisted longer for some systems than others, it has adversely affected the quality of forecasts for all systems.

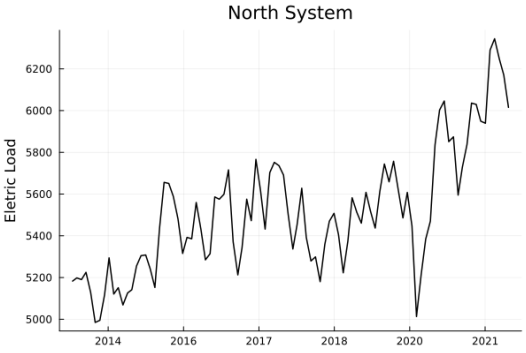

Figure 4.1: Monthly electric load data for the North Brazilian system.

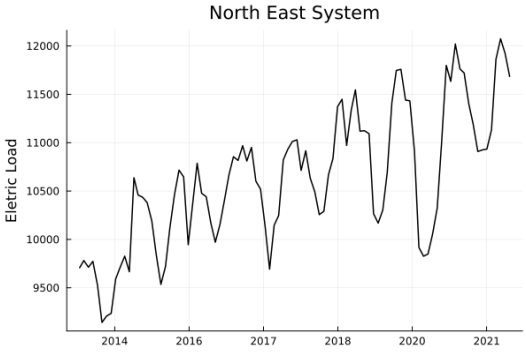

Figure 4.2: Monthly electric load data for the North East Brazilian system.

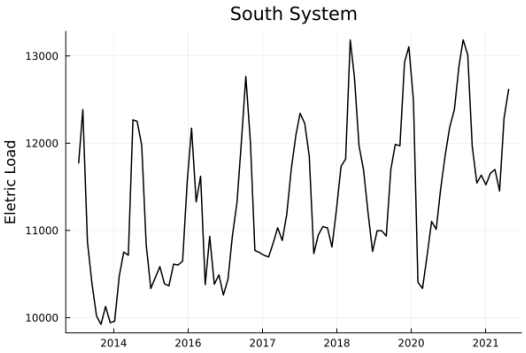

Figure 4.3: Monthly electric load data for the South Brazilian system.

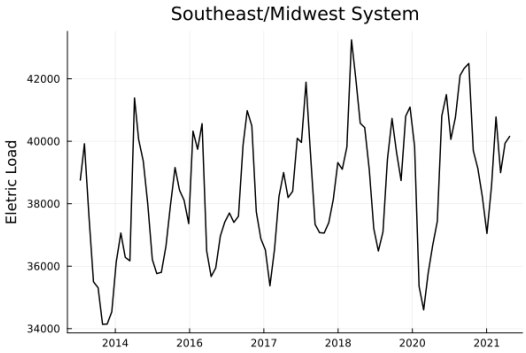

Figure 4.4: Monthly electric load data for the Southeast/Midwest Brazilian system.

To effectively model this data, score-driven models must be appropriately specified. Based on the data characteristics and preliminary test results, the proposed specification for the UC score-driven models includes stochastic seasonality with an AR(1) level for all systems, except for the South system, which was modeled with a random walk plus slope level. For all models, the Normal distribution is used with only the mean parameter varying over time, following the dynamics mentioned earlier. Table 4.1 summarizes the specified models used in this experiment. Notably, the only difference between the score-driven model (SDM) and its robust counterpart (R-SDM) is the inclusion of the robustness technique applied to make the model robust to 80% of the training data. In other words, the K parameter of robust models was set to be equal to round$(0.8 \times T)$, where T is the length of the training data.

Table 4.1: Model Specifications for the Robustness Experiment

System	Distribution	d	Level	Seasonality
North	Normal(μ_t, σ^2)	1.0	AR(1)	Stochastic
North East	Normal(μ_t, σ^2)	1.0	AR(1)	Stochastic
South	Normal(μ_t, σ^2)	1.0	Random Walk + Slope	Stochastic
Southeast/Midwest	Normal(μ_t, σ^2)	1.0	AR(1)	Stochastic

As mentioned earlier, the evaluation of the gain in prediction accuracy of the Robust score-driven model against its non-robust version was performed using monthly electric load data for the four Brazilian systems. Leveraging the regime change induced by the COVID-19 pandemic, a comparison was made between the forecast accuracy of the robust model and its non-robust version during typical and atypical periods. Here, the typical period refers to the pre-pandemic period from 2014 to 2018, while the atypical period encompasses the years from 2019 onwards. It is important to highlight that although the pandemic started in 2020, data from 2019 was considered as an atypical period to help evaluate how models would perform during the transition from the pre-pandemic to the pandemic period.

The evaluation of the model's forecasting accuracy is based on twelve-step ahead predictions using twenty-five rolling windows. In typical periods, the model is estimated using data from 2014 to 2016 for the first rolling window, and predictions are made for 2017. Subsequently, for each subsequent rolling window, the estimation period includes data from January 2016, and predictions are made for the subsequent twelve steps ahead, providing forecasts for 2018 and 2019. This process repeats for all twenty-five rolling windows, covering the years 2014 to 2019. In the atypical period, the first estimation window spans from 2014 to 2018, and predictions are made for 2019, 2020, and 2021. This process repeats for all systems, covering the years 2014 to 2021.

To compare the models' conditional predictive ability (CPA), the

Giacomini-White test [48] is utilized. Additionally, their accuracy is assessed using several error metrics, including MAPE (Mean Absolute Percentage Error), MASE (Mean Absolute Scaled Error), MAE (Mean Absolute Error), and RMSE (Root Mean Square Error).

While the experiment relies on cross-validation utilizing error metrics to gauge the model's performance, the assessment of model performance begins with a graphical analysis. In this analysis, the predictions generated in each rolling window are plotted simultaneously on a single graph. This approach enables the visualization of the model's out-of-sample performance across the entire validation period. The rationale for adopting this analysis lies in the belief that human judgment, supported by graphical analysis, can serve as a valuable model selection criterion alongside those based on cross-validation [49].

To clarify the content depicted in Figures 4.6 to 4.21, a condensed version is presented in Figure 4.5. This figure illustrates the observed data (depicted by the black curve) for the North East system throughout the entire typical period. Additionally, it simultaneously displays the twelve-step-ahead predictions for three distinct windows. From this simplified visualization, it is apparent that the model exhibited better predictive performance in the last window (indicated by the blue curve) compared to the others. It is worth noting that in this illustrative example, the windows displayed do not intersect. However, in the subsequent graphs, overlapping windows are presented as they were designed to generate intersecting periods.

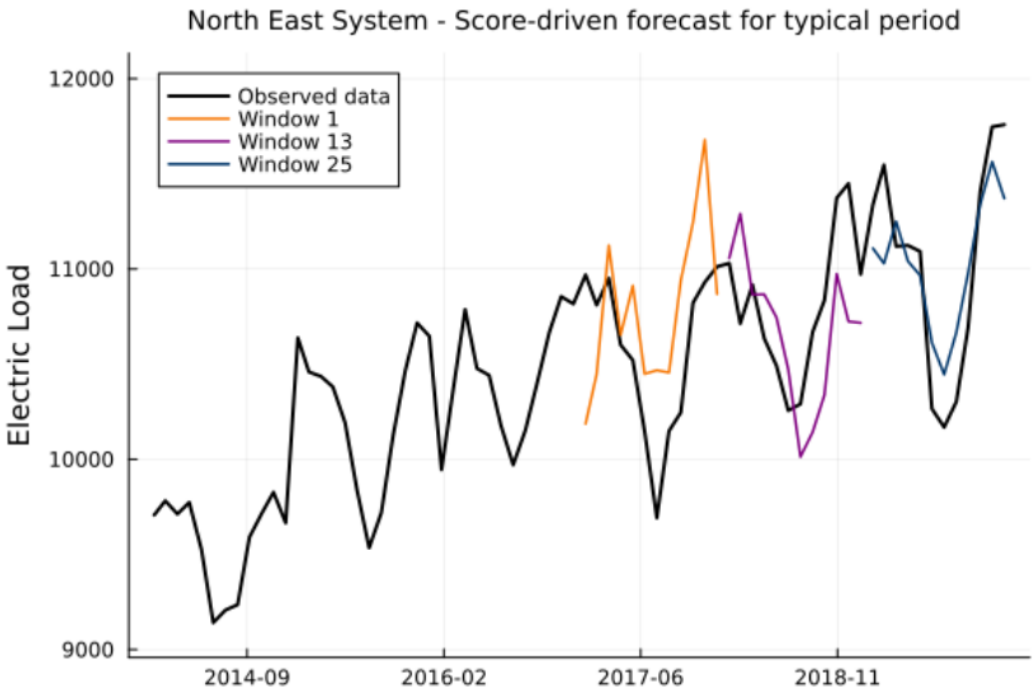

Figure 4.5: Exemplary predictions of score-driven model during the typical period for North East system.

With this in mind, let's delve into the graphical analysis by examining Figures 4.6 to 4.21. These figures depict the predictive performance during typical and atypical periods.

Figures 4.6 to 4.12 reveal that even in typical periods, the robust model yields more stable predictions compared to its non-robust counterpart, particularly for the North East and Southeast/Midwest systems. This observation stems from the fact that the SDM occasionally exhibits weaker performance in specific windows, despite its overall satisfactory performance—a characteristic almost not observed in the robust model, which demonstrates consistent performance across windows.

The increased stability provided by the sampling robustness technique becomes more evident during atypical periods (Figures 4.14 to 4.21), where the instability of the score-driven model's predictions becomes more pronounced, notably affecting its overall performance. Conversely, the robust model maintains a much more consistent performance throughout the rolling windows.

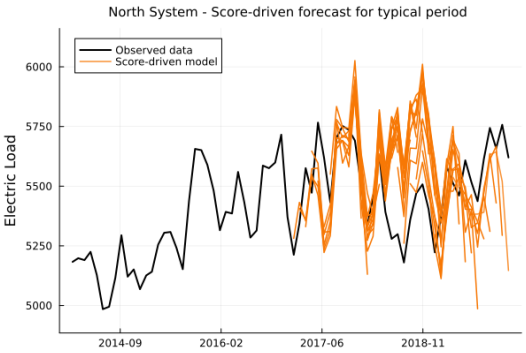

Figure 4.6: Predictions of SDM during typical period of North System.

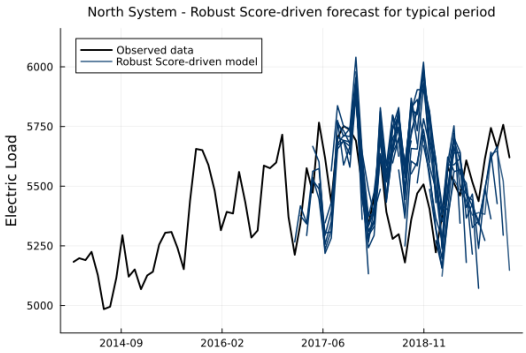

Figure 4.7: Predictions of R-SDM during typical period of North System.

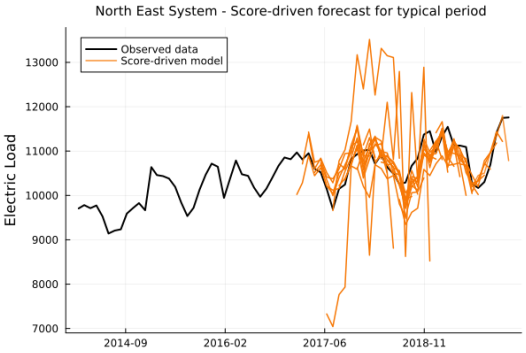

Figure 4.8: Predictions of SDM during typical period of North East System.

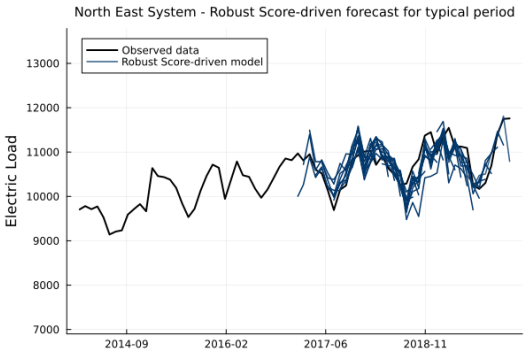

Figure 4.9: Predictions of R-SDM during typical period of North East System.

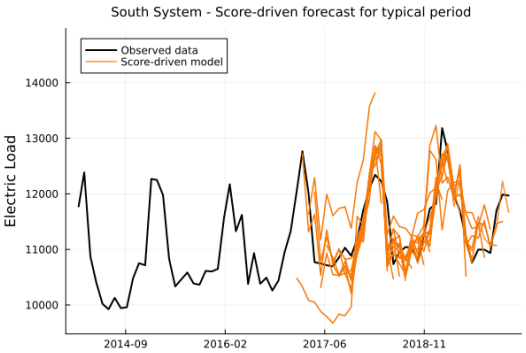

Figure 4.10: Predictions of SDM during typical period of South System.

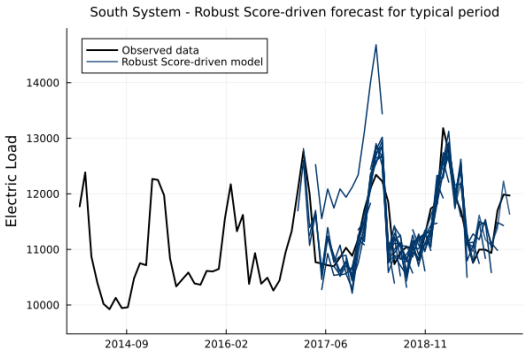

Figure 4.11: Predictions of R-SDM during typical period of South System.

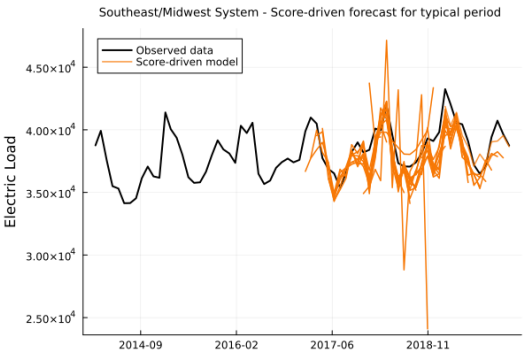

Figure 4.12: Predictions of SDM during typical period of Southeast/Midwest System.

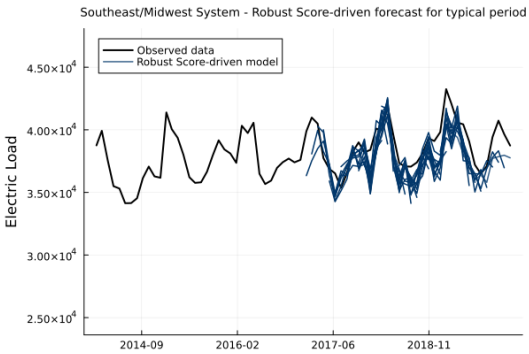

Figure 4.13: Predictions of R-SDM during typical period of Southeast/Midwest System.

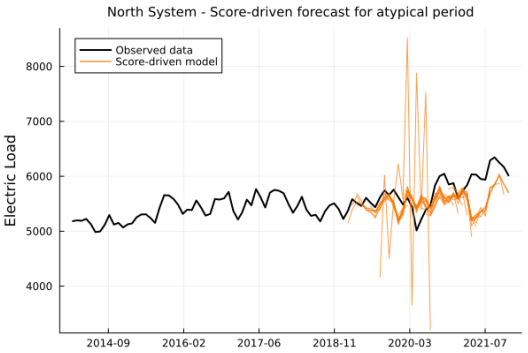

Figure 4.14: Predictions of SDM during atypical period of North System.

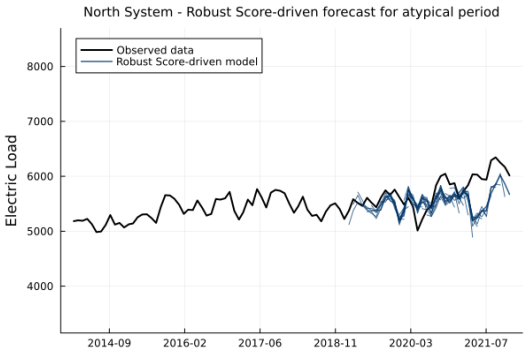

Figure 4.15: Predictions of R-SDM during atypical period of North System.

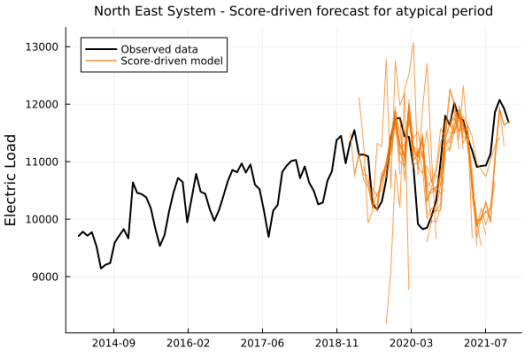

Figure 4.16: Predictions of SDM during atypical period of North East System.

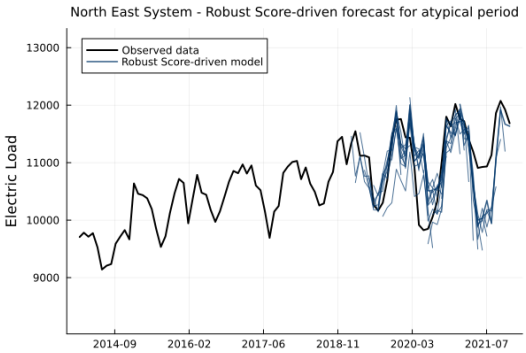

Figure 4.17: Predictions of R-SDM during atypical period of North East System.

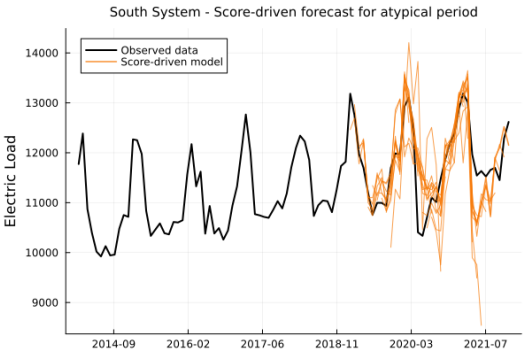

Figure 4.18: Predictions of SDM during atypical period of South System.

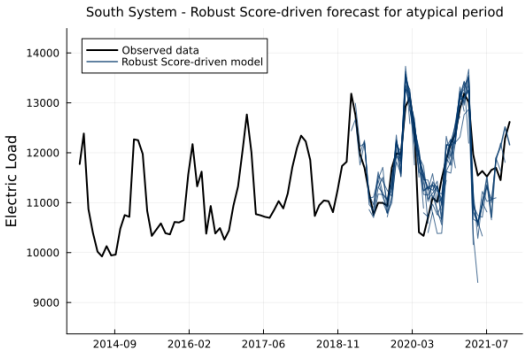

Figure 4.19: Predictions of R-SDM during atypical period of South System.

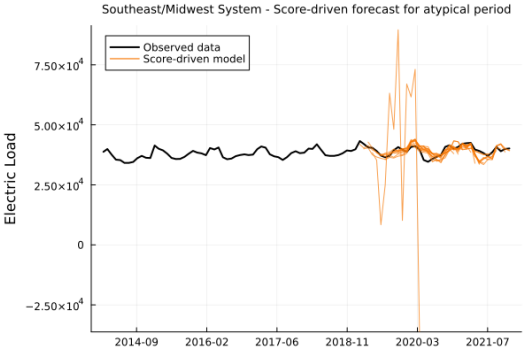

Figure 4.20: Predictions of SDM during atypical period of Southeast/Midwest System.

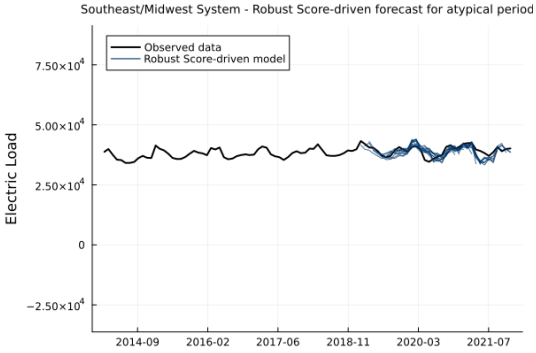

Figure 4.21: Predictions of R-SDM during atypical period of Southeast/Midwest System.

While human judgment has the potential to support model selection [49], it is also important to consider more traditional methods for comparing model performance based on cross-validation. The following analysis will compare the accuracy of each model during both periods by considering the four error metrics previously mentioned.

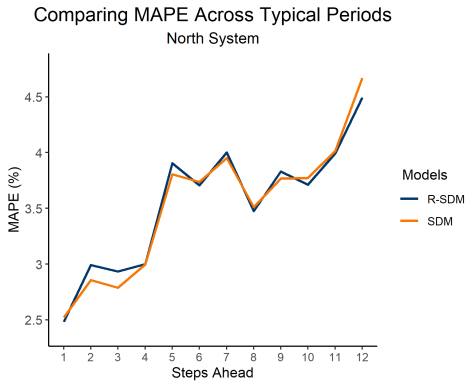

Figure 4.22: Comparison of MAPE for each step ahead during typical periods in the North system.

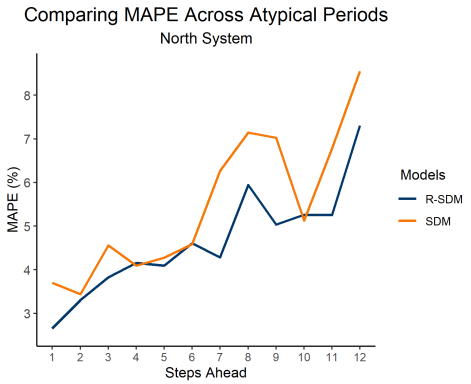

Figure 4.23: Comparison of MAPE for each step ahead during atypical periods in the North system.

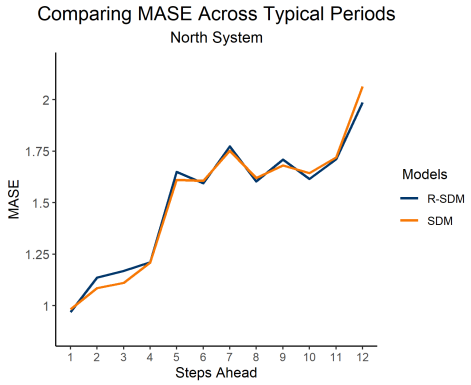

Figure 4.24: Comparison of MASE for each step ahead during typical periods in the North system.

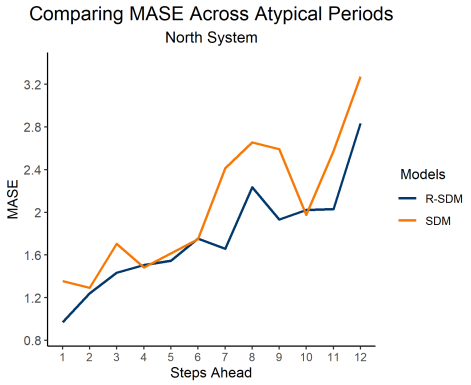

Figure 4.25: Comparison of MASE for each step ahead during atypical periods in the North system.

In Figures 4.22 to 4.29, it is evident that, for the North system, both models presented very similar performances during typical periods in all four metrics. On the other hand, during atypical periods, the robust model consistently outperformed its non-robust version, showing lower error metrics across virtually the entire forecast horizon.

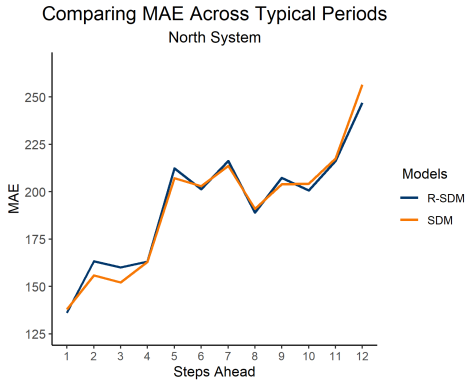

Figure 4.26: Comparison of MAE for each step ahead during typical periods in the North system.

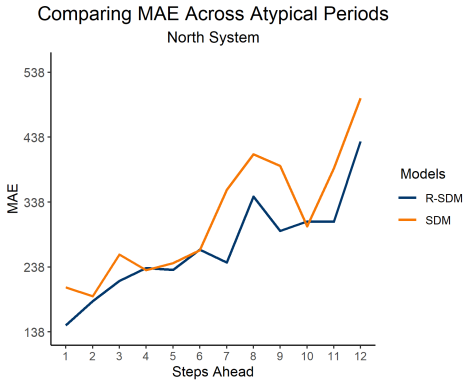

Figure 4.27: Comparison of MAE for each step ahead during atypical periods in the North system.

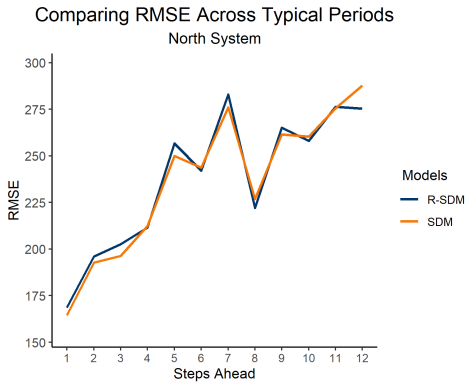

Figure 4.28: Comparison of RMSE for each step ahead during typical periods in the North system.

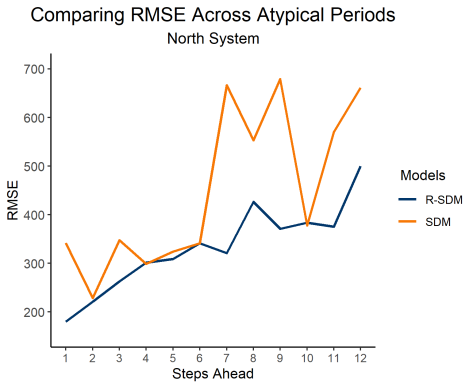

Figure 4.29: Comparison of RMSE for each step ahead during atypical periods in the North system.

Moving to the results for the North East system, a different behavior of the error metrics becomes apparent. Figures 4.30, 4.32, 4.34, and 4.36 reveal that R-SDM outperforms SDM in both periods for all the four metrics considered. An interesting observation in this system is that the difference between the error metrics of both models seems to be bigger during the typical period than the atypical one.

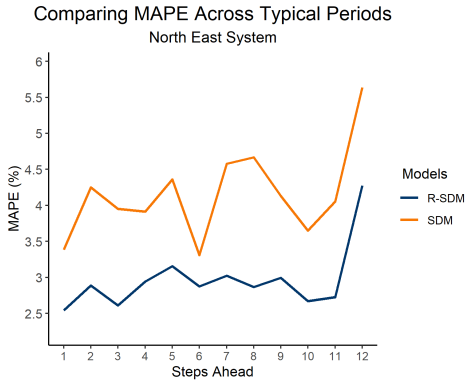

Figure 4.30: Comparison of MAPE for each step ahead during typical periods in the North East system.

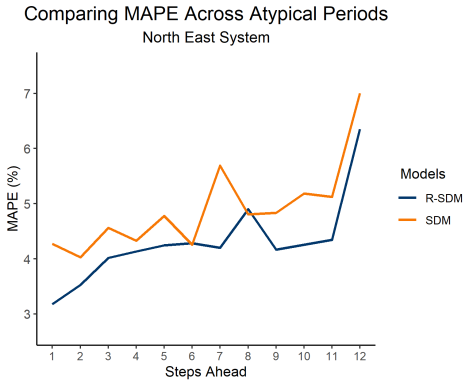

Figure 4.31: Comparison of MAPE for each step ahead during atypical periods in the North East system.

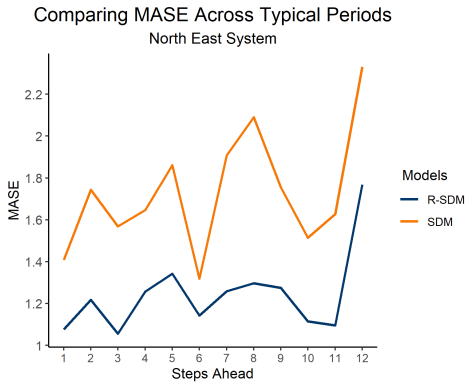

Figure 4.32: Comparison of MASE for each step ahead during typical periods in the North East system.

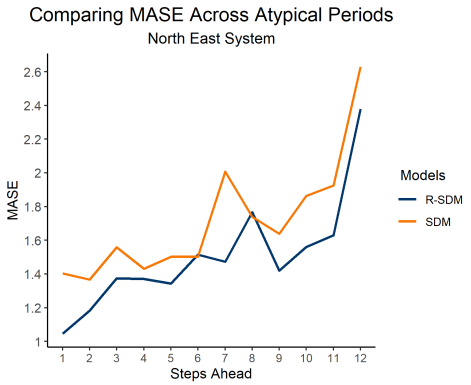

Figure 4.33: Comparison of MASE for each step ahead during atypical periods in the North East system.

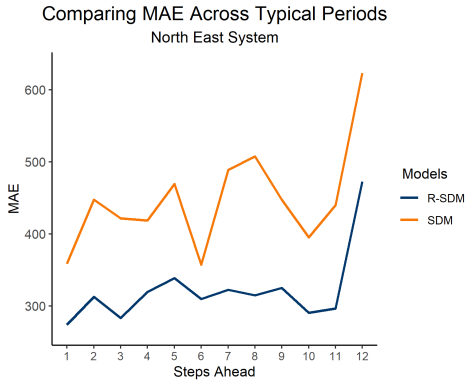

Figure 4.34: Comparison of MAE for each step ahead during typical periods in the North East system.

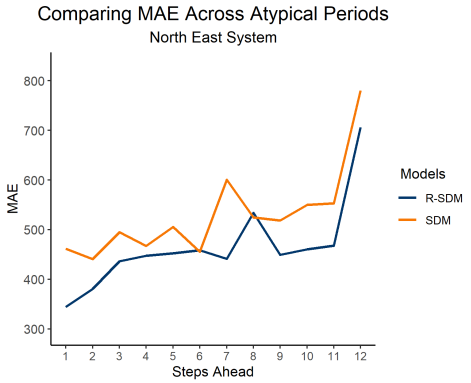

Figure 4.35: Comparison of MAE for each step ahead during atypical periods in the North East system.

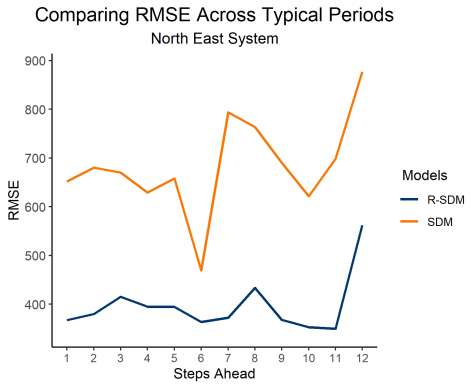

Figure 4.36: Comparison of RMSE for each step ahead during typical periods in the North East system.

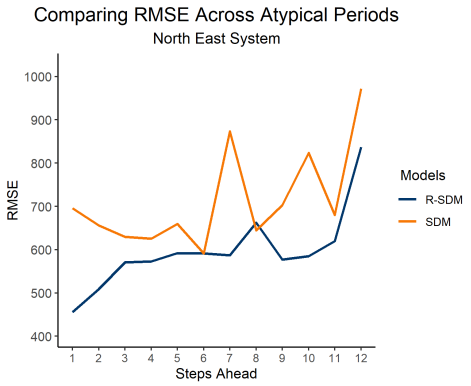

Figure 4.37: Comparison of RMSE for each step ahead during atypical periods in the North East system.

In relation to the South system, Figures 4.38 to 4.45 demonstrate that both models exhibit similar performance during typical periods. The model with the lowest metric alternated during the forecast horizon. However, it is worth highlighting that the robust model tended to perform slightly better for shorter-term predictions across all considered metrics.

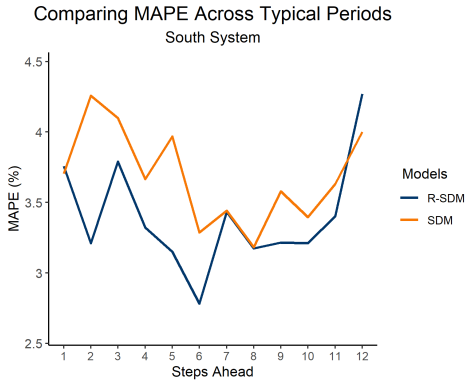

Figure 4.38: Comparison of MAPE for each step ahead during typical periods in the South system.

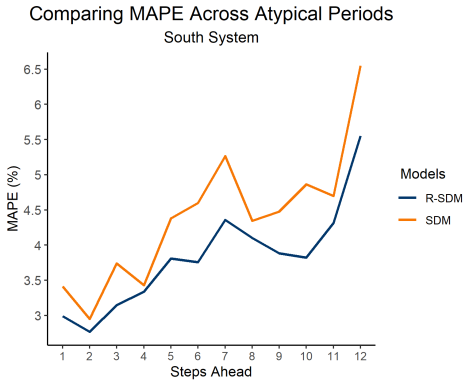

Figure 4.39: Comparison of MAPE for each step ahead during atypical periods in the South system.

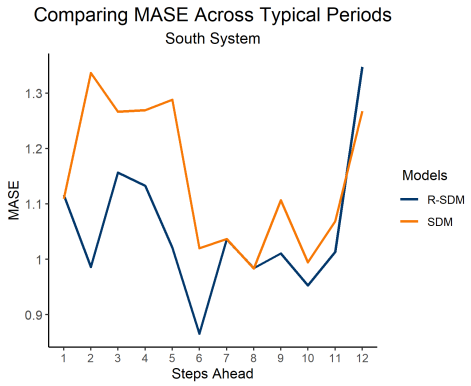

Figure 4.40: Comparison of MASE for each step ahead during typical periods in the South system.

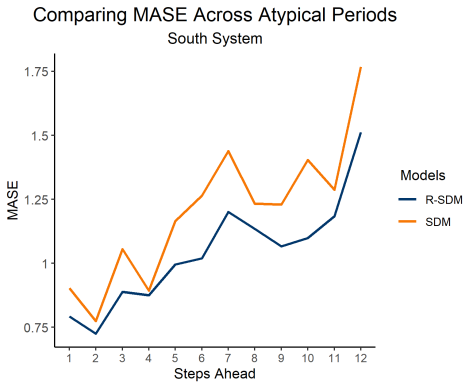

Figure 4.41: Comparison of MASE for each step ahead during atypical periods in the South system.

As observed in previous systems, the robust model demonstrated superior performance across the forecast horizon of the atypical period for all analyzed metrics.

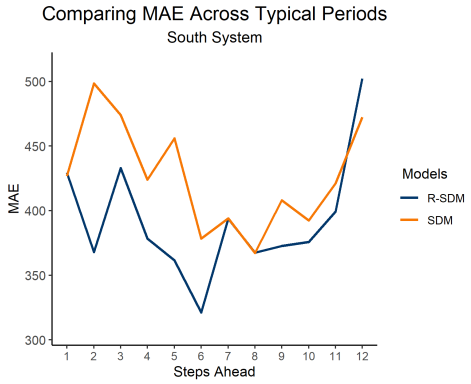

Figure 4.42: Comparison of MAE for each step ahead during typical periods in the South system.

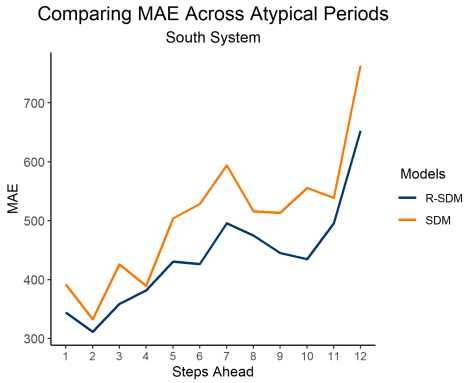

Figure 4.43: Comparison of MAE for each step ahead during atypical periods in the South system.

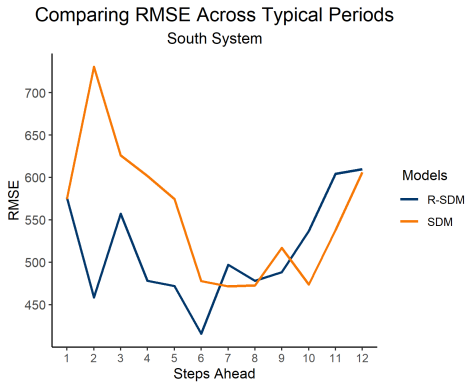

Figure 4.44: Comparison of RMSE for each step ahead during typical periods in the South system.

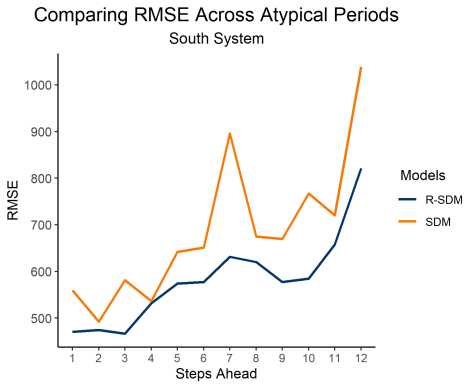

Figure 4.45: Comparison of RMSE for each step ahead during atypical periods in the South system.

Finally, Figures 4.46 to 4.53 depict the results for the Southeast/Midwest system. It is evident that during typical periods, both models exhibit much more comparable performance than observed in the South system. However, during atypical periods, once again, the robust model outperforms its non-robust version across practically the entire forecast horizon.

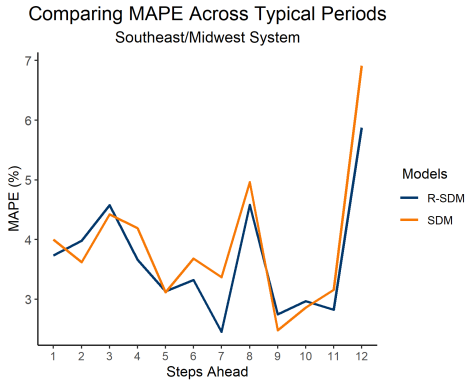

Figure 4.46: Comparison of MAPE for each step ahead during typical periods in the Southeast/Midwest system.

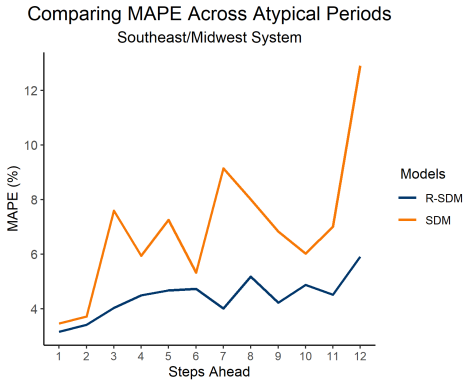

Figure 4.47: Comparison of MAPE for each step ahead during atypical periods in the Southeast/Midwest system.

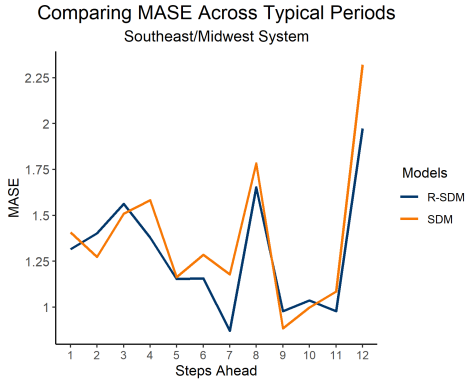

Figure 4.48: Comparison of MASE for each step ahead during typical periods in the Southeast/Midwest system.

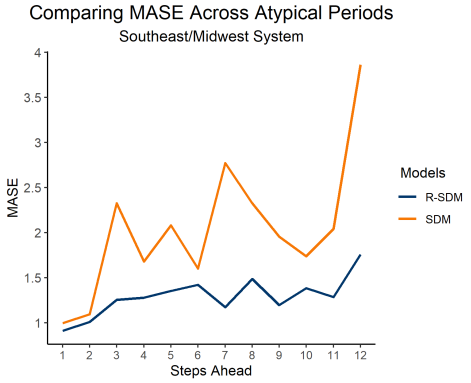

Figure 4.49: Comparison of MASE for each step ahead during atypical periods in the Southeast/Midwest system.

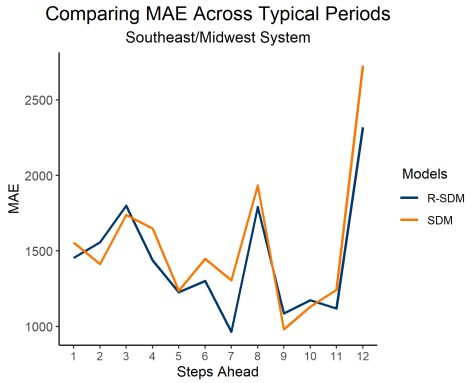

Figure 4.50: Comparison of MAE for each step ahead during typical periods in the Southeast/Midwest system.

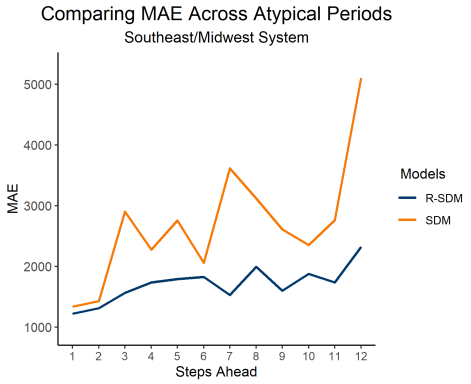

Figure 4.51: Comparison of MAE for each step ahead during atypical periods in the Southeast/Midwest system.

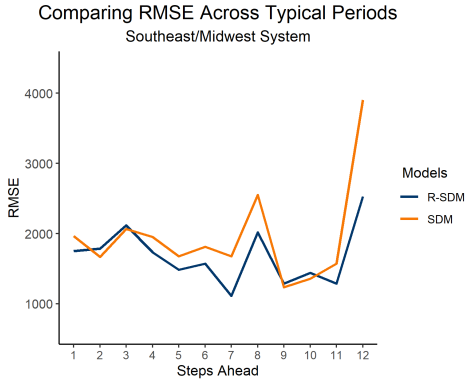

Figure 4.52: Comparison of RMSE for each step ahead during typical periods in the Southeast/Midwest system.

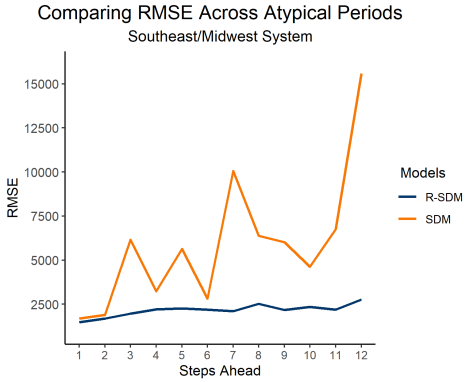

Figure 4.53: Comparison of RMSE for each step ahead during atypical periods in the Southeast/Midwest system.

Although all the graphical analyses indicate a better performance of the robust model, especially during atypical periods, it is also important to measure the statistical significance of this result. In this regard, the unilateral version of the Giacomini-White test was conducted using the `eptftoolbox` library [50]. The aim was to compare the Conditional Predictive Ability (CPA) of both models during both typical and atypical periods. In this analysis, the test evaluates the null hypothesis that the CPA of the SDM is higher or equal to that of the R-SDM. Therefore, rejecting the null hypothesis implies that the forecasts of the R-SDM are significantly more accurate than those of the SDM.

Table 4.2 presents the results of the test in terms of its p-value. Using

these results, it is possible to conclude that during typical periods, the test indicates a significantly better performance of the robust model for the North East and the Southeast/Midwest systems, considering a significance level (α) of 5%. With a slightly higher α (at least 7%), it is possible to conclude that the R-SDM also had a significantly better performance than SDM in the South system. This indicates that, although the difference between both models' CPA is smaller in the South system than in the North East and the Southeast/Midwest, it is still relevant. Only for the North system does the test indicate no significant difference in the models' CPA, which is in agreement with the error metrics analysis.

On the other hand, for atypical periods, the test indicated a significantly better performance of the robust model for all systems considered. It is important to highlight that, similarly to what happens with the South system in the typical period, a small adjustment of the α value must be made (at least 6%) to consider the difference between models' CPA in the North system significant.

Table 4.2: The p-value of the Giacomini-White test assesses the comparison of Conditional Predictive Ability (CPA) between both models across typical and atypical periods.

System	Typical Period	Atypical Period
North	1.0000	0.0574
North East	0.0055	0.0146
South	0.0644	0.0012
Southeast/Midwest	0.0352	0.0297

After conducting this comprehensive analysis, deeper insights emerge regarding the advantages of incorporating sample robustness techniques into score-based model formulations. It became evident during typical periods that the robust model exhibited accuracy levels at least comparable to those of its non-robust counterpart. However, for certain systems, significantly better

performance of the robust model was observed even during typical periods. It is essential to note that while the robust model did not consistently demonstrate significantly superior performance during typical periods, it showcased the ability to produce more stable predictions. This stability was evidenced by its resilience to fluctuations within specific windows, as depicted in Figure 4.6, for instance.

In addition to showcasing significantly more stable performance compared to its non-robust counterpart, R-SDM exhibited superior accuracy in terms of error metrics and predictive capacity during atypical periods. Consequently, R-SDM emerges as a valuable tool for professionals and policymakers entrusted with making predictions amid unforeseen disturbances. By providing more precise and stable forecasts during challenging periods, it can facilitate enhanced decision-making and resource allocation.

5
Conclusions and future works

Dealing with temporally structured data poses challenges across various applications. Extracting valuable insights and making precise predictions from time series data are critical for informed decision-making in numerous fields. With this premise in mind, this study aimed to investigate the potential benefits of utilizing modern optimization techniques to define and estimate time series models. Furthermore, the primary outcome of this study is the SDUC.jl package, an open-source implementation intended to empower users to define and estimate UC score-driven models.

The creation of this package represents a significant contribution to the literature, addressing the lack of open-source implementations for this model type. Furthermore, it introduces a series of reforms in the approach to time series modeling. The primary innovation lies in the formulation of the models themselves. By harnessing advancements in computational capabilities and optimization techniques, the SDUC.jl package formulates these models as nonlinear optimization problems. This shift in model representation has the potential to greatly enhance predictive accuracy [7]. Importantly, this approach introduces a novel initialization process, enabling the initialization of time-varying parameters across all time periods, rather than solely at the outset as conventionally practiced.

Additionally, by framing the model as an optimization problem, numerous additional features can be seamlessly incorporated into the model formulation to enhance its capabilities. However, this study has primarily focused on introducing a feature aimed at enhancing model robustness, which also represents a significant contribution.

This feature addresses sample robustness and is based on the method proposed in [28]. This technique trains the model using the worst sub-sample

of the training data, following specific criteria. In this case, the worst subsample of length K was considered. Implementing this approach yields models resilient to regime changes in time series data, a phenomenon observed across various domains that can potentially compromise model accuracy.

Through a few simple illustrative examples utilizing three well-known time series datasets, it was possible to demonstrate how the proposed package can be effectively utilized in all stages of time series modeling, from model definition to prediction application, alongside a detailed analysis of residuals. Furthermore, these examples indicate the package's flexibility in defining various score-driven models. Subsequently, another experiment was conducted to evaluate the package's performance during regime changes.

This experiment aimed to evaluate the enhancement in accuracy achieved by employing the robustness method during regime change periods. Although the experiment could have utilized data from various domains, real data on monthly electric load for the four Brazilian systems was selected, with the onset of the COVID-19 pandemic signaling the start of the regime change period.

The procedure involved conducting a rolling windows process to estimate and make predictions using both robust and non-robust score-driven models for all systems during pre- and post-pandemic periods. The results indicated that the robust model exhibited more stable performance throughout the rolling windows, particularly during atypical periods. Additionally, it was observed that the robust model generated more accurate predictions during atypical periods, as evidenced by the reduction in certain error metrics and the higher Conditional Predictive Ability (CPA) compared to its non-robust counterpart, as indicated by the Giacomini-White test. Importantly, these benefits during regime change periods were achieved without notable drawbacks during typical periods.

In conclusion, this study effectively fulfills its objectives by integrating optimization functionalities into the time series modeling process, leading to

improvements in model accuracy. While the introduction of the SDUC.jl package represents a noteworthy addition to the literature, the primary contribution of this work lies in broadening the discourse on how the adoption of modern optimization techniques can enhance time series modeling.

On the flip side, treating score-driven models as optimization problems has proven to be potentially challenging and computationally demanding. Consequently, the proposed package still presents significant opportunities for enhancement. In this regard, the primary area for improvement in future endeavors lies in reformulating the initialization process for the model's decision variables, aiming to improve both computational efficiency and accuracy. Additionally, to further optimize the package's performance, alternative approaches to representing the model via JuMP will be explored. This includes exploring variations in the choice of solver and adjustments to the syntax of the optimization problem itself.

Once this phase is completed, the focus will shift to integrating additional distributions into the package, thereby enhancing its applicability and impact. Initially, the plan is to incorporate distributions from the TSLab software, with further expansion of options expected thereafter. Furthermore, there is parallel interest in extending the covered dynamics and exploring the inclusion of time-varying effects of explanatory variables, alongside their automatic selection using integer optimization.

6
Bibliography

1 BOX, G.; JENKINS, G. M. **Time Series Analysis: Forecasting and Control**. [S.l.]: Holden-Day, 1970.

2 HARVEY, A. C. **Forecasting, Structural Time Series Models and the Kalman Filter**. [S.l.]: Cambridge University Press, 1990.

3 CREAL, D.; KOOPMAN, S. J.; LUCAS, A. Generalized autoregressive score models with applications. **Journal of Applied Econometrics**, Wiley Online Library, v. 28, n. 5, p. 777–795, 2013.

4 HARVEY, A. C. **Dynamic Models for Volatility and Heavy Tails: With Applications to Financial and Economic Time Series**. [S.l.]: Cambridge University Press, 2013. (Econometric Society Monographs).

5 NELDER, J. A.; MEAD, R. A simplex method for function minimization. **The Computer Journal**, v. 7, n. 4, p. 308–313, 1965. Disponível em: <https://doi.org/10.1093/comjnl/7.4.308>.

6 LIU, D. C.; NOCEDAL, J. On the limited memory bfgs method for large scale optimization. **Mathematical Programming**, v. 45, p. 503–528, 1989. Disponível em: <https://doi.org/10.1007/BF01589116>.

7 BERTSIMAS, D.; DUNN, J. **Machine Learning Under a Modern Optimization Lens**. Dynamic Ideas LLC, 2019. ISBN 9781733788502. Disponível em: <https://books.google.com.br/books?id=g3ZWygEACAAJ>.

8 FUENTES, F.; HERRERA, R.; CLEMENTS, A. Forecasting extreme financial risk: A score-driven approach. **International Journal of Forecasting**, v. 39, n. 2, p. 720–735, 2023. ISSN 0169-2070. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0169207022000310>.

9 SUCARRAT, G.; GRøNNEBERG, S. Risk estimation with a time-varying probability of zero returns. **Journal of Financial Econometrics**, v. 20, n. 2, p. 278–309, 2022. Spring. Disponível em: <https://doi.org/10.1093/jjfinec/nbaa014>.

10 PATTON, A. J.; ZIEGEL, J. F.; CHEN, R. Dynamic semiparametric models for expected shortfall (and value-at-risk). **Journal of Econometrics**, v. 211, n. 2, p. 388–413, 2019. ISSN 0304-4076. Disponível em: <https://www.sciencedirect.com/science/article/pii/S030440761930048X>.

11 XU, Y.; LIEN, D. Optimal futures hedging for energy commodities: An application of the gas model. **Journal of Futures Markets**, v. 40, p. 1090–1108, 2020. Disponível em: <https://doi.org/10.1002/fut.22118>.

12 LAPORTA, A. G.; MERLO, L.; PETRELLA, L. Selection of value at risk models for energy commodities. **Energy Economics**, v. 74, p. 628–643, 2018. ISSN 0140-9883. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0140988318302548>.

13 HOELTGEBAUM, H.; FERNANDES, C.; STREET, A. Generating joint scenarios for renewable generation: The case for non-gaussian models with time-varying parameters. **IEEE Transactions on Power Systems**, IEEE, v. 33, n. 6, p. 7011–7019, 2018.

14 ALANYA-BELTRAN, W. Modelling stock returns volatility with dynamic conditional score models and random shifts. **Finance Research Letters**, v. 45, p. 102121, 2022. ISSN 1544-6123. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1544612321002026>.

15 HEIL, T. L.; PETER, F. J.; PRANGE, P. Measuring 25 years of global equity market co-movement using a time-varying spatial model. **Journal of International Money and Finance**, v. 128, p. 102708, 2022. ISSN 0261-5606. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0261560622001115>.

16 AYALA, A.; BLAZSEK, S. Score-driven copula models for portfolios of two risky assets. **The European Journal of Finance**, Routledge, v. 24, n. 18, p. 1861–1884, 2018. Disponível em: <https://doi.org/10.1080/1351847X.2018.1464488>.

17 SARLO, R.; FERNANDES, C.; BORENSTEIN, D. Lumpy and intermittent retail demand forecasts with score-driven models. **European Journal of Operational Research**, v. 307, n. 3, p. 1146–1160, 2023. ISSN 0377-2217. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0377221722007743>.

18 HOELTGEBAUM, H. et al. A score-driven model of short-term demand forecasting for retail distribution centers. **Journal of Retailing**, v. 97, n. 4, p. 715–725, 2021. ISSN 0022-4359. SI: Metrics and Analytics. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0022435921000361>.

19 ARDIA, D.; BOUDT, K.; CATANIA, L. Generalized autoregressive score models in r: The gas package. **Journal of Statistical Software**, v. 88, n. 6, p. 1–28, 2019. Disponível em: <https://www.jstatsoft.org/index.php/jss/article/view/v088i06>.

20 BODIN, G. et al. Scoredrivenmodels.jl: a julia package for generalized autoregressive score models. Online, 2020. Disponível em: <https://github.com/LAMPSPUC/ScoreDrivenModels.jl.>

21 R Core Team. **R: A Language and Environment for Statistical Computing**. Vienna, Austria, 2021. Disponível em: <https://www.R-project.org/>.

22 BEZANSON, J. et al. Julia: A fresh approach to numerical computing. **SIAM Review**, v. 59, n. 1, p. 65–98, 2017. Disponível em: <https://doi.org/10.1137/141000671>.

23 LIT, R.; KOOPMAN, S.; HARVEY, A. **Time Series Lab**. 2019–2021. <https://timeserieslab.com>.

24 SOYSTER, A. L. Technical note—convex programming with set-inclusive constraints and applications to inexact linear programming. **Operations Research**, v. 21, n. 5, p. 1154–1157, 1973. Disponível em: <https://doi.org/10.1287/opre.21.5.1154>.

25 BEN-TAL, A.; NEMIROVSKI, A. Robust solutions of uncertain linear programs. **Operations Research Letters**, v. 25, n. 1, p. 1–13, 1999. ISSN 0167-6377. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0167637799000164>.

26 BERTSIMAS, D.; SIM, M. The price of robustness. **Operations Research**, INFORMS, v. 52, n. 1, p. 35–53, 2004. ISSN 0030364X, 15265463. Disponível em: <http://www.jstor.org/stable/30036559>.

27 FERNANDES, B. et al. An adaptive robust portfolio optimization model with loss constraints based on data-driven polyhedral uncertainty sets. **European Journal of Operational Research**, v. 255, n. 3, p. 961–970, 2016. ISSN 0377-2217. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0377221716303757>.

28 BERTSIMAS, D.; PASKOV, I. Time series that are robust to regime changes. **Journal of Machine Learning Research**, v. 21, p. 1–31, 2020.

29 COX, D. R. et al. Statistical analysis of time series: Some recent developments [with discussion and reply]. **Scandinavian Journal of Statistics**, [Board of the Foundation of the Scandinavian Journal of Statistics, Wiley], v. 8, n. 2, p. 93–115, 1981. ISSN 03036898, 14679469. Disponível em: <http://www.jstor.org/stable/4615819>.

30 BOLLERSLEV, T. Generalized autoregressive conditional heteroskedasticity. **Journal of Econometrics**, v. 31, n. 3, p. 307–327, 1986. ISSN 0304-4076. Disponível em: <https://www.sciencedirect.com/science/article/pii/0304407686900631>.

31 FABIÁN, Z. Score function of distribution and revival of the moment method. **Communications in Statistics - Theory and Methods**, Taylor and Francis, v. 45, n. 4, p. 1118–1136, 2016. Disponível em: <https://doi.org/10.1080/03610926.2013.857688>.

32 RUDIN, W. **Principles of Mathematical Analysis**. [S.l.]: McGraw-Hill, 1976.

33 BLASQUES, F.; GORGI, P.; KOOPMAN, S. J. Conditional score residuals and diagnostic analysis of serial dependence in time series models. n. 2021-098/III, November 18 2021. Disponível em: <https://ssrn.com/abstract=3966281>.

34 HYNDMAN, R. J.; ATHANASOPOULOS, G. Forecasting: Principles and practice. In: _____. 2nd. ed. [S.l.]: OTexts, 2018. cap. 7, p. 123–145.

35 BERKOWITZ, J. Testing density forecasts, with applications to risk management. **Journal of Business & Economic Statistics**, Taylor & Francis, v. 19, n. 4, p. 465–474, 2001. Disponível em: <https://doi.org/10.1198/07350010152596718>.

36 DUNN, P. K.; SMYTH, G. K. Randomized quantile residuals. **Journal of Computational and Graphical Statistics**, v. 5, p. 236–244, 1996.

37 BLASQUES, F. et al. In-sample confidence bands and out-of-sample forecast bands for time-varying parameters in observation-driven models. **International Journal of Forecasting**, v. 32, n. 3, p. 875–887, 2016. ISSN 0169-2070. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0169207016000303>.

38 BERTSIMAS, D.; DUNN, J. Machine learning under a modern optimization lens. In: _____. [S.l.]: Dynamic Ideas LLC, 2019. cap. 1, p. 1–18.

39 BERTSIMAS, D.; DUNN, J. Machine learning under a modern optimization lens. In: _____. [S.l.]: Dynamic Ideas LLC, 2019. cap. 17, p. 333–348.

40 BREIMAN, L. Bagging predictors. **Machine Learning**, Springer, v. 24, n. 2, p. 123–140, 1996.

41 HAMILTON, J. D. A new approach to the economic analysis of nonstationary time series and the business cycle. **Econometrica**, JSTOR, v. 57, n. 2, p. 357–384, 1989.

42 BERTSIMAS, D.; BROWN, D. B.; CARAMANIS, C. Theory and applications of robust optimization. **SIAM Review**, v. 53, n. 3, p. 464–501, 2011. Disponível em: <https://doi.org/10.1137/080734510>.

43 DUNNING, I.; HUCHETTE, J.; LUBIN, M. Jump: A modeling language for mathematical optimization. **SIAM Review**, v. 59, 08 2015.

44 Revels, J.; Lubin, M.; Papamarkou, T. Forward-mode automatic differentiation in Julia. **arXiv:1607.07892 [cs.MS]**, 2016. Disponível em: <https://arxiv.org/abs/1607.07892>.

45 WÄCHTER, A.; BIEGLER, L. T. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. **Mathematical programming**, Springer, v. 106, n. 1, p. 25–57, 2006.

46 SAAVEDRA, R.; BODIN, G.; SOUTO, M. Statespacemodels.jl: a julia package for time-series analysis in a state-space framework. **arXiv preprint arXiv:1908.01757**, 2019.

47 LIT, R.; KOOPMAN, S.; HARVEY, A. **Time Series Lab - Score Edition**. 2019–2020. <https://timeserieslab.com>.

48 GIACOMINI, R.; WHITE, H. Tests of conditional predictive ability. **Econometrica**, v. 74, n. 6, p. 1545–1578, 2006. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1468-0262.2006.00718.x>.

49 PETROPOULOS, F.; SIEMSEN, E. Forecast selection and representativeness. **Manag. Sci.**, v. 69, p. 2672–2690, 2022. Disponível em: <https://api.semanticscholar.org/CorpusID:250658702>.

50 LAGO, J. et al. Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark. **Applied Energy**, v. 293, p. 116983, 2021.

A
Location Scale t Distribution Local Search

The Student's t distribution is a continuous probability distribution that extends the standard normal distribution, displaying a bell-shaped curve symmetrically centered around the mean. Unlike the Gaussian distribution, the t distribution features heavier tails, where the amount of probability mass in these tails is regulated by the degrees of freedom (ν) parameter.

In the location-scale parameterization, the t distribution expands upon the normal distribution by incorporating a location parameter μ and a scale parameter σ^2. Notably, the conventional Student's t distribution arises as a specific case of the location-scale t distribution when $\mu = 0$ and $\sigma^2 = 1$.

The addition of a model based on this distribution in the SDUC.jl package is crucial because of its ability to handle data with heavier tails. Initially, the integration of these models followed the same procedure as other distributions, with parameters designated as decision variables and optimized using the chosen solver. However, initial tests revealed that the optimization process was excessively sensitive to changes in the degrees of freedom parameter, compromising the overall performance of the model. This prompted the exploration of an alternative approach for estimating such models.

While the mathematical formulation of a t-student distribution defines the ν parameter as a positive real number, it's conventionally treated as an integer due to its interpretation as the difference between the sample size and the number of model parameters. Leveraging this discrete interpretation of the ν parameter, a local search strategy was devised to tackle this challenge. Initially, given an initial estimate for the degree of freedom parameter ν_0, the specified score-driven model is estimated three times: once with ν fixed at $\nu_0 - 1$, once with ν fixed at ν_0, and once with ν fixed at $\nu_0 + 1$. If the model with $\nu = \nu_0$ yields the best result in terms of the corrected Akaike Information Criterion (AICc), ν_0 is considered the optimal

degree of freedom parameter value. Conversely, if an alternative model achieves a superior AICc score, a local search process is initiated. This process iteratively adjusts the fixed value of ν by one unit, continuously evaluating model performance using the AICc metric. The process continues as long as successive changes in the ν parameter lead to improved AICc values. Importantly, to ensure the well-defined moments of the t distribution, only models with $\nu > 2$ are considered.

Furthermore, the determination of the initial value ν_0 warrants discussion. Initially, a simple approach sets $\nu_0 = T-1$, where T denotes the series size. In other words, since T is generally greater than 30, in this approach, the local search starts from a Gaussian model. Although effective for numerous applications, this approach exhibited instability for some longer time series. To address this concern, a second approach was introduced, where ν_0 is derived as the maximum likelihood estimate of the ν parameter, obtained through an optimization problem. Consequently, the local search procedure begins by evaluating which value of ν_0 performs best based on the AICc metric. Subsequently, for the optimal value, the aforementioned local search approach is applied. Algorithm 1 provides a concise overview of the local search steps.

By employing this approach, users can determine the locally optimal value of the ν parameter, considering its integer nature, without the need to handle mixed-integer non-linear problems.

Algorithm 1: Local Search Procedure

Input: Specified score-driven model and the time series of interest

Output: A fitted score-driven model considering the best value of degree of freedom parameter ν_{best}

Estimate a score-driven model with $\nu = T - 1$
Estimate a score-driven model with $\nu = \nu_{ml}$
if *Model with* $\nu = T - 1$ *has the best AICc result* **then**
    $\nu_0 = T - 1$
else
    $\nu_0 = \nu_{ml}$
Estimate score-driven model with $\nu = \nu_0 - 1$
Estimate score-driven model with $\nu = \nu_0 + 1$
if *Model with* $\nu = \nu_0$ *has the best AICc result* **then**
    final model $\leftarrow$ model with $\nu = \nu_0$
else
    if *Model with* $\nu = \nu_0 - 1$ *has the best AICc result* **then**
        $\lambda = -1$
    else
        $\lambda = 1$
    $\nu_{current} = \nu_0 + \lambda$
    while *AICc improves and* $\nu_{current} > 2$ **do**
        Estimate score-driven model with $\nu = \nu_{current}$
        if *New model has better AICc result* **then**
            best model $\leftarrow$ model with $\nu = \nu_{current}$
            $\nu_{current} \leftarrow \nu_{current} + \lambda$
        else
            final model $\leftarrow$ best model