Pontifícia Universidade Católica
do Rio de Janeiro

Eduardo Ravaglia Campos Queiroz

Variable Step-Size evolving Participatory Learning with Kernel Recursive Least Squares model applied to Gas Prices Forecasting in Brazil

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. Fernando Luiz Cyrino Oliveira
Co-advisor: Prof. Eduardo Pestana de Aguiar

Rio de Janeiro
March 2021

Pontifícia Universidade Católica
do Rio de Janeiro

Eduardo Ravaglia Campos Queiroz

Variable Step-Size evolving Participatory Learning with Kernel Recursive Least Squares model applied to Gas Prices Forecasting in Brazil

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

Prof. Eduardo Pestana de Aguiar
Coadvisor
UFJF

Marley Maria Bernardes Rebuzzi Vellasco
Departamento de Engenharia Elétrica – PUC-Rio

Daniel Furtado Leite
UFLA

Rio de Janeiro, March 26th, 2021

Eduardo Ravaglia Campos Queiroz

The author is a bachelor in Exact Sciences and Mechanical Engineer from Federal University of Juiz de Fora (UFJF) in 2017 and 2018, respectively. He also specialized in Work Safety Engineering from Estácio de Sá University (UNESA) in 2020.

Bibliographic data

Queiroz, Eduardo Ravaglia Campos

Variable Step-Size evolving Participatory Learning with Kernel Recursive Least Squares model applied to Gas Prices Forecasting in Brazil / Eduardo Ravaglia Campos Queiroz; advisor: Fernando Luiz Cyrino Oliveira; co-advisor: Eduardo Pestana de Aguiar. – 2021.

61 f: il. color. ; 30 cm

Dissertação (mestrado) - Pontifical Catholic University of Rio de Janeiro, Departamento de Engenharia Industrial, 2021.

Inclui bibliografia

1. Engenharia Industrial – Teses. 2. Previsão de Séries Temporais. 3. Modelos Nebulosos Evolutivos. 4. Passo de Adaptação Variável. I. Oliveira, Fernando Luiz Cyrino. II. Aguiar, Eduardo Pestana de. III. Pontifical Catholic University of Rio de Janeiro. Departamento de Engenharia Industrial. IV. Título.

CDD: 658.5

Acknowledgments

I thank God for giving me health and strength to overcome difficulties.
To my mother Rita Ravaglia for her love and incentive.
To my girlfriend Xênia Souza for her love and support during this journey.
To my friends who were part of my formation. Especially you Guilherme Guimarães, the "Canelinha" (*in memoriam*), a warrior that will never be forgotten.
To my advisor professor Fernando Cyrino, my co-advisor professor Eduardo Aguiar and my coworkers that made this work possible.
To PUC-Rio and UFJF for essential support during this work and to the anonymous referees for their valuable comments.
This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

Abstract

Queiroz, Eduardo Ravaglia Campos; Oliveira, Fernando Luiz Cyrino (Advisor); Aguiar, Eduardo Pestana de (Co-Advisor). **Variable Step-Size evolving Participatory Learning with Kernel Recursive Least Squares model applied to Gas Prices Forecasting in Brazil**. Rio de Janeiro, 2021. 61p. Dissertação de Mestrado – Departamento de Engenharia Industrial, Pontifícia Universidade Católica do Rio de Janeiro.

A prediction model is an indispensable tool in business, helping to make decisions, whether in the short, medium, or long term. In this context, the implementation of machine learning techniques in time series forecasting models has a notorious relevance, as information processing and efficient and dynamic knowledge uncovering are increasingly demanded. This work develops a model called Variable Step-Size evolving Participatory Learning with Kernel Recursive Least Squares, VS-ePL-KRLS, applied to the forecast of weekly prices for S500 and S10 diesel oil, at the Brazilian level, for biweekly and monthly horizons. The presented model demonstrates a better accuracy compared with analogous models in the literature, without loss of computational performance for all time series analyzed.

Keywords

Time Series Forecasting; Evolving Fuzzy Models; Variable Step-Size.

Resumo

Queiroz, Eduardo Ravaglia Campos; Oliveira, Fernando Luiz Cyrino; Aguiar, Eduardo Pestana de. **Modelo Variable Step-Size evolving Participatory Learning with Kernel Recursive Least Squares aplicado à previsão de preços do Óleo Diesel no Brasil**. Rio de Janeiro, 2021. 61p. Dissertação de Mestrado – Departamento de Engenharia Industrial, Pontifícia Universidade Católica do Rio de Janeiro.

Um modelo de previsão é uma ferramenta indispensável nos negócios, ajudando na tomada de decisões, seja a curto, médio ou longo prazo. Neste contexto, a implementação de técnicas de aprendizagem de máquina em modelos de previsão de séries temporais assume notória relevância, visto que o processamento da informação e a extração de conhecimento são cada vez mais exigidos de forma eficiente e dinâmica. Este trabalho desenvolve um modelo denominado Variable Step-Size evolving Participatory Learning with Kernel Recursive Least Squares, VS-ePL-KRLS, aplicado à previsão de preços do óleo diesel S500 e S10. O modelo apresentado demonstra uma melhor acurácia em comparação com os modelos análogos na literatura, sem perda de desempenho computacional para todas as séries temporais analisadas.

Palavras-chave

Previsão de Séries Temporais; Modelos Nebulosos Evolutivos; Passo de Adaptação Variável.

Table of contents

List of figures

List of tables

List of Abreviations

ANP – National Agency of Petroleum, Natural Gas, and Biofuels

CPU – Central Processing Unit

ePL – evolving Participatory Learning

ePL+ – enhanced evolving Participatory Learning

ePL-KRLS – evolving Participatory Learning with Kernel Recursive Least Squares

ESM-ePL-KRLS – Enhanced Set-Membership evolving Participatory Learning with Kernel Recursive Least Squares

eTS – evolving Takagi-Sugeno

eTS+ – enhanced evolving Takagi-Sugeno

eTS-KRLS – evolving Takagi-Sugeno with Kernel Recursive Least Squares

eTS-LS-SVM – evolving Takagi-Sugeno with Least Squares Support Vector Machine

GK – Gustafson-Kessel

KRLS – Kernel Recursive Least Squares

LMS – Least Mean Squares

LS – Least Squares

MAE – Mean Absolute Error

MFKM – Modified Fuzzy K-Means

MGN – Morgan Granger Newbold

Mod_eTS – Modified evolving Takagi-Sugeno

NDEI – Non-Dimensional Index Error

RKHS – Reproducing Kernel Hilbert Space

RLS – Recursive Least Squares

RMSE – Root Mean Square Error

Simpl_eTS – Simplified evolving Takagi-Sugeno

SM-ePL-KRLS – Set-Membership evolving Participatory Learning with Kernel Recursive Least Squares

SVM – Support Vector Machine

TSK – Takagi-Sugeno-Kang

VS – Variable Step

VS-ePL-KRLS – Variable Step-Size evolving Participatory Learning with Kernel Recursive Least Squares

xTS – extended Takagi-Sugeno

Don't worry about failure; you only have to be right once.

Drew Houston.

1
Introduction

Forecasting is an essential tool in business, where it helps in making decisions about operational (short term), tactical (medium term), and strategic (long term) planning [1]. In a world that demands information processing and knowledge extraction in an increasingly efficient and dynamic way, the combination of Forecasting and Machine Learning techniques has gained notorious relevance in modeling real-world phenomena, which are characterized by complexity and uncertainties. Several authors have developed methods to achieve an optimal model setup under some circumstances, that is, a solution that trades off the quality of decisions and response times.

In the logistical context, the ability to make predictions with high accuracy and low computational cost is a competitive advantage. In transport management, for example, responsible for connecting suppliers, production, and customers, it is important to predict the price of the diesel oil, which may be responsible for approximately 21% of the logistics costs [2, 3]. This importance is maximized in a country of continental dimensions such as Brazil, which primarily uses road networks to transport cargo and people.

Several models in the Machine Learning field have been developed with the ability to make predictions in real-time. These are driven mainly by the increasing availability of data and the need to make quick decisions. The Evolving Intelligent Systems are a notorious segment of learning models that concept can be considered a higher level adaptation that concerns model structure as well as model parameters and performed as an effective tool to address the problem of modeling non-stationary, highly non-linear processes online in real-time [4].

Evolving fuzzy rule-based models can be used for clustering, classification, forecasting, control, diagnosis, and regression where learning and model development should be performed incrementally to deal with data stream, that increases continuously and changes rapidly over time. Their differential is to simultaneously manage any significant changes (drift, shifts, non-stationary behaviors, environmental conditions) in the system by using parameter and structural adaptation algorithms to process a data sample at most once [5].

A renowned approach to develop evolving fuzzy rule-based models is

based on the idea of participatory learning. It is a means to learn and revise beliefs based on what is already known about the system itself through the induction of unsupervised dynamic fuzzy clustering algorithms for online, real-time domains and applications, resulting in an effective alternative construction evolving functional fuzzy models and adaptive fuzzy systems [6]. This work aims to propose the Variable Step-Size evolving Participatory Learning with Kernel Recursive Least Squares (VS-ePL-KRLS) model that makes two changes in this approach: The Kernel Recursive Least Squares (KRLS) technique to estimate consequent parameters as in [7] and The Variable Step-Size (VS) technique to update the parameter responsible for control the rate of change of arousal, as a novelty.

The Kernel Recursive Least Squares acts as an adaptation mechanism to maintain and store past knowledge in a robust and efficient manner whose advantage is a greater sensitivity to variations in the input data and a better ability to approximate nonlinear systems accurately and with moderate computational cost [7].

The Variable Step updates parameters recursively based on estimation errors. This technique, discussed in [8], is an extension of earlier ideas in stochastic approximation for varying the step size in the method of steepest descent. According to [9], VS are adaptive filtering, which can provide improved performance while maintaining the simplicity and robustness of conventional Fixed Step-Size. This approach is easier to implement and, consequently, additional computational complexity savings are obtained as noted by [10] in a fault classification problem in an electromechanical switch machine, equipment used for handling railroad switches.

1.1 Objectives

The general objective of this research is to propose a new Evolving Fuzzy modeling approach with high accuracy and low computational cost, with a focus on prediction tasks. The specific objective of this research is to propose an incremental learning method we called Variable Step-Size evolving Participatory Learning with Kernel Recursive Least Squares (VS-ePL-KRLS) for time series forecasting. It expands the ePL-KRLS and replaces the SM-ePL-KRLS and ESM-ePL-KRLS with an error-based Variable Step-Size parameter-updating method.

1.2 Contributions

The major contributions of this dissertation are an online modeling approach that has shown encouraging results in different applications and datasets compared to state-of-the-art evolving approaches; a method with a low percentage of system-wide Central Processing Unit (CPU) utilization, suggesting that the forecasting model can deal with complex dynamics; and an unexplored and important application, which is the fuels prices forecasting, more specifically Brazilian S500 and S10 diesel oil, with weekly data periodicity for biweekly and monthly horizons and monthly data periodicity for monthly, semiannual and yearly horizons.

1.3 Text Organization

The dissertation is divided into six chapters. This chapter introduces the key subjects and describes the motivation and relevance of conducting this research, the general and specific objectives, and the contributions related to the new online modeling approach. Chapter 2 addresses the literature review of intelligent models, emphasizing the Evolving Fuzzy model. Chapter 3 describes the proposed model, VS-ePL-KRLS, as well as its online learning algorithm. Chapter 4 presents the methodology, that is, how we structure the oil diesel price forecasting problem. We provide a historical context, statistical analysis, and metrics used to Forecasting Evaluation and Computational Performance. Chapter 5 gives experimental and comparative results for the algorithm. Chapter 6 concludes this text and suggests further research issues.

2
Literature Review

In recent years, several methods have emerged in the context of online machine learning. The area has become prominent in nonstationary time series prediction. It was stimulated due to the increasing availability of data and the need to real-time predict their behavior to support decision-making. There are three main model groups of this class. Evolutionary models are driven by concepts of biological evolution, which uses selection, crossover, and mutation operators as adaptation mechanisms. Adaptive models are usually referred to as models with a fixed structure and variable parameters. This means that the overall structure is defined before the online learning process. Evolving models are able to process information and extract knowledge in real-time, resulting in a higher level adaptation of their structure and parameters.

This work is focused on evolving fuzzy models due to their characteristic of reacting to data changes by creating, merging, updating, and deleting local models, e.g., fuzzy rules [4]. A wide variety of algorithms and models can be found in the literature. The authors in [11] discussed recent advances that have improved stability, reliability, and useability as well as aspects related to greater interpretability in real-world applications such as online condition monitoring, visual inspection, human-machine interaction, smart sensors, production systems, and other applications. Another important contribution is presented by [5], a systematic survey on evolving intelligent systems with a focus on fuzzy and neuro-fuzzy methods for clustering, classification, regression and system identification in real-world applications as online trading, financial analysis, e-commerce and business, smart home, health care, transportation systems, global supply logistic chains, smart grids, industrial control, cybersecurity, and many other areas.

The evolving Takagi-Sugeno (eTS) model [12] is considered the precursor of many evolving fuzzy systems. An incremental learning algorithm updates the eTS model recursively by adding new rules or updating existing rules. Antecedent terms of Takagi-Sugeno (TS) rules are determined through a subtractive clustering approach [13] based on the notion of potential function. Consequent parameters of rules are updated through Recursive Least Squares (RLS) [14]. It has been evaluated on data from a residential air-conditioning

installation and Mackey-Glass time series. The results illustrate the viability, efficiency, and potential of the approach when used with a limited amount of initial information, especially important in autonomous systems and robotics. Other models have arisen as variations of the eTS model, such as Simpl_eTS [15], xTS [16], Mod_eTS [17], eTS+ [18], Simpl_eTS+ [19], eTS-LS-SVM [20] and eTS-KRLS [21].

The Simpl_eTS model is a simplified version of the eTS. It addresses new concepts such as Cauchy type antecedent membership functions instead of Gaussian, scatter as a measure of data density and summarization ability of rules instead of potential, and aging indicator as a representation of the stationarity of the rules. It has been evaluated on data from the Box-Jenkins time series. The results don't negatively affect the performance of the algorithm but contributed substantially towards its simplification.

The xTS model is an extended version of the eTS. It introduces a new perspective to the online classification of streaming data (e.g., video, speech, sensory data generated from robotic, advanced industrial applications, financial and retail chain transactions, and intruder detection). It has been evaluated on data from intrusion detection and several benchmark problems such as Sonar, Credit Card, Ten-Digit, Ionosphere, Pima, and Wine datasets. The results demonstrate that a flexible evolving classifier can be generated online from streaming data achieving high classification rates and using limited computational resources.

The Mod_eTS model is a modified version of the eTS. It incorporates a modification in the dynamic update of cluster radii while accommodating newly available data is proposed. It has been evaluated on data from a leakage faced by one of the leading United Kingdom water supplying companies. The results show higher accuracy and a smaller number of clusters than standard fuzzy forecasting methods.

The eTS+ model is an enhanced version of the eTS. It uses criteria such as age, utility, local density, and zone of influence to update a rule base structure. This arose as a result of the need to increase the quality, autonomy, and robustness of its rules base, thus enabling a smaller number of local models could be created to represent knowledge of a particular type of problem. The Simpl_eTS+ model is also an enhanced version of the Simpl_eTS, which uses the same idea as eTS+.

The eTS-LS-SVM and eTS-KRLS are an altered versions of the eTS in relation to how consequent parameters are estimated. The eTS updates their parameters by Recursive Least Squares (RLS) while eTS-LS-SVM uses Least Squares (LS) Support Vector Machine (SVM) [22], and eTS-KRLS uses Kernel

Recursive Least Squares (KRLS) [23]. They have been evaluated on data from the Mackey-Glass time series. The results present the smallest number of rules in eTS-LS-SVM and the smallest metrics of errors in eTS-KRLS.

The aforementioned updating procedures, as noted by [24], are not robust in the presence of white noise, that is, noisy data and outliers tend to be incorporated into the model instead of being regularized. In a way to fill that gap, [6] proposed an algorithm called evolving Participatory Learning (ePL), which uses an unsupervised dynamic fuzzy clustering algorithm for Antecedent terms. Consequent parameters of rules are updated through Recursive Least Squares (RLS). This approach mimics human learning [25], which updating the rule base of a model depends on a function whose arguments are the existing rules and new information received. A similar process occurs with human learning, which is amplified if there exists prior knowledge on a given subject. It has been evaluated on data from a workload of electrical energy and Box-Jenkins time series. The results illustrate the competitiveness, efficiency, and parsimony of the approach. Other models have arisen as variations of the ePL model, such ePL+ [26], ePL-KRLS [7], SM-ePL-KRLS [27] and ESM-ePL-KRLS [27].

The ePL+ model is an enhanced version of the eTS. Similarly to [18], it uses criteria such as utility, age, area of influence, and local density to update a rule base structure. It has been evaluated on data from finance variables, Mackey-Glass time series, Box-Jenkins time series, and Non-linear synthetic time series time-variant. The results show the adequacy of the probabilistic fuzzy modeling given the high precision and computational efficiency. Both models, ePL and ePL+, use the RLS to estimate consequent parameters such as eTS and eTS+.

The ePL-KRLS model is an altered version of the ePL in relation to how consequent parameters are estimated. The ePL updates their parameters by Recursive Least Squares (RLS) while ePL-KRLS uses Kernel Recursive Least Squares (KRLS), inspired by [21]. The advantage of using this approach is Kernel-based methods that are more sensitive to variations in the input data and are able to approximate nonlinear systems accurately and efficiently with moderate computational cost. It has been evaluated on data from a wind speed and Mackey-Glass time series. The results present more accuracy (especially for several steps ahead), less quantity and variations of rules, and less sensitivity to changes in its control parameters.

The SM-ePL-KRLS and ESM-ePL-KRLS models are an extended version of the ePL-KRLS. These models have their parameters updated incrementally according to the estimation errors, which, in turn, are based on the concept

of Set-Membership filtering [10]. The first is the implementation of Set-Membership in the evolving Participatory Learning with Kernel Recursive Least Squares and the second is a combination of the evolving Participatory Learning with Kernel Recursive Least Squares and the improved version of the Set-Membership concept. They have been evaluated on data from a hot-spot temperature of power transformers. The results demonstrate lower errors and a competitive number of final rules and, consequently, failure prevention, cost reduction, maintenance of safety, and providing reliable service to consumers.

All in all, we notice that there is ample discussion regarding ways of improving evolving fuzzy models in two lines of research: eTS and ePL. This work aims to propose a variation in the recursively updating parameters based on estimation errors, replacing the Set-Membership filtering with the Variable Step-Size technique. The main goal of this change is to improve the algorithm, that is, to reduce errors and computational complexity.

3
Variable Step-Size evolving Participatory Learning with Kernel Recursive Least Squares Algorithm

The online learning flowchart of the modeling approach we propose, namely, Variable Step-Size evolving Participatory Learning with Kernel Recursive Least Squares, VS-ePL-KRLS, is shown in Figure 3.1. Fundamentally, we have extended known evolving fuzzy models such as ePL [6], ePL-KRLS [7], SM-ePL-KRLS [27] and ESM-ePL-KRLS [27]. The new forecasting model is based on the Variable Step-Size [8] to adjust the parameter that controls the rate of change of arousal as a function of the error. The flowchart consists of input-output streaming data pairs and five main stages. The learning procedures within each stage are described in the next sections.

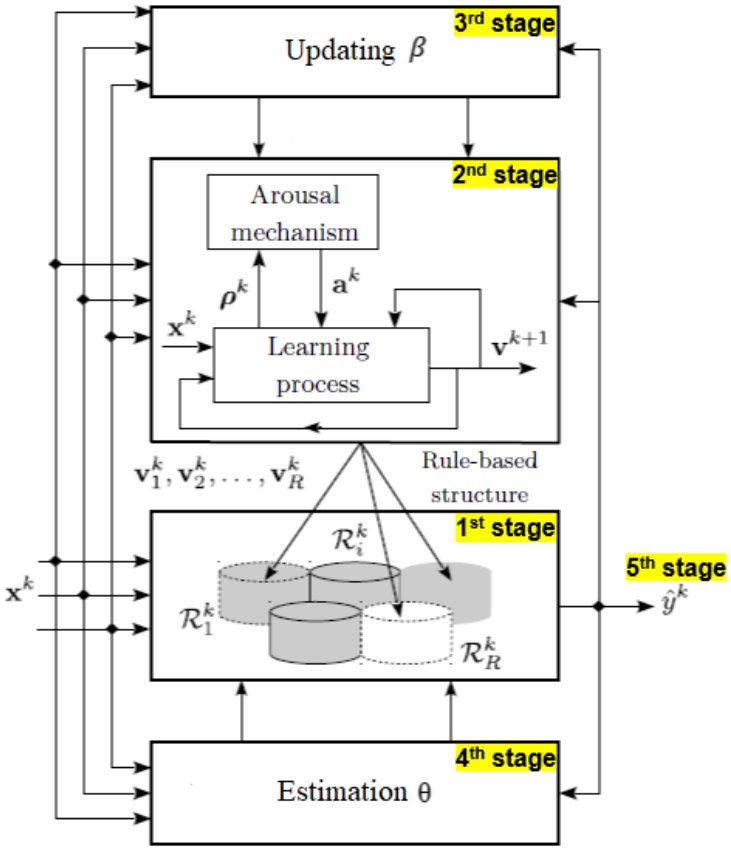

Figure 3.1: VS-ePL-KRLS online learning procedure to construct and update prediction models from scratch

In general, Figure 3.1 shows an evolving fuzzy model based on Takagi-Sugeno-Kang (TSK) rules [28, 29]. The rules are of the type: "IF <antecedent>, THEN <consequent>" as shown in Equation 3-1.

$$\mathcal{R}_i : \texttt{IF x is } \mathcal{A}_i \texttt{ THEN } \hat{y}_i = f_i\left(\mathtt{x}, \theta_i\right) \tag{3-1}$$

in which $\mathcal{R}_i$ is the i-th fuzzy rule, $\mathtt{x} = [x_1, \ldots, x_m]^T \in \mathbb{R}^m$ is the input data vector, $\theta_i = [\theta_{i1}, \ldots, \theta_{in_i}]^T \in \mathbb{R}^{n_i}$ is the consequent parameters vector of the i-th rule, $\mathcal{A}_i\left(\mathtt{x}\right) : \mathbb{R} \rightarrow [0, 1]$ is the membership function in the antecedent fuzzy set of the i-th rule, and $\hat{y}_i : \mathbb{R}^m \ \mathtt{x} \ \mathbb{R}^{n_i} \rightarrow \mathbb{R}$ is the local output of the i-th rule.

The antecedent part of the rule is expressed by a set of premise terms which can be associated to linguistic values; and the consequent part of the rule is expressed by a mathematical function. A feature of TSK fuzzy models is that they are universal approximators of functions [30]. In other words, TSK models are capable of approximating, with any pre-specified accuracy, a general nonlinear continuous function on a compact set through the combination of local functions.

3.1 First Stage: Unsupervised Fuzzy Clustering

This stage of the model performs the unsupervised fuzzy clustering proposed by [31]. It proved to be as efficient as the Gustafson-Kessel (GK) algorithm [32] and the Modified Fuzzy K-Means (MFKM) algorithm [33] – two major fuzzy clustering algorithms. The objective of this learning stage is to update the number of fuzzy rules and the centers of the Gaussian membership functions that describe the antecedent part of fuzzy rules. The algorithm is initialized with a single rule ($R = 1$) whose center is the first input ($v_1^1 = x^1$). During the computational procedure, new rules can be added and existing rules can be updated or merged and it will be explained in the next stage.

3.2 Second Stage: Evolving Participatory Learning

This stage of the model carries out the evolving participatory learning proposed by [6]. Such learning paradigm has a convergent conception with that of the human learning [25]. In humans, the greater the prior knowledge in a given subject, the more amplified the learning will be. The computer learns in the same way in which each new information will be related to the existing model, and the greater the similarity, the greater the learning. This stage is divided in two steps.

In the first step consists in calculating the compatibility $\left(\rho_i^k\right)$ and arousal $\left(a_i^k\right)$ indices using Equations 3-2 and 3-3, respectively.

$$\rho_i^k = 1 - \frac{\left\|x^k - v_i^k\right\|}{m} \tag{3-2}$$

in which $\rho_i^k \in [0,1]$ is the compatibility index of the i-th rule at the k-th step, $x^k \in \mathbb{R}^m$ is the m-dimensional input data vector at the k-th step, and $v_i^k \in [0,1]^m$ is the center of the i-th rule at the k-th step.

$$a_i^k = a_i^{k-1} + \beta^{k-1}\left(1 - \rho_i^k - a_i^{k-1}\right) \tag{3-3}$$

in which $a_i^k \in [0,1]$ is the arousal index of the i-th rule at the k-th step, and $\beta^k \in [0,1]$ is a parameter that controls the rate of change of arousal. The default initialization values is $a_1^0 = 0$.

The compatibility index measures how well data and model rules are compatible. The arousal index evaluates whether a new rule should be added to the model, or an existing rule should be updated by comparing it with a parameter $\tau^k \in [0,1]$. If $\left|a_i^k\right| > \tau^k$, a new rule is added. Otherwise, an existing rule with more compatibility is updated. The center of a new rule added $\left(v_R^{k+1}\right)$ or an existing rule updated $\left(v_i^{k+1}\right)$ is represented by Equation 3-4.

$$\begin{cases} v_R^{k+1} = x^k, \quad if \quad \left|a_i^k\right| > \tau^k \\ v_i^{k+1} = v_i^k + \alpha\left(v_i^k\right)^{1-a_i^{k+1}}\left(x^k - v_i^k\right), \quad otherwise \end{cases} \tag{3-4}$$

in which $\alpha \in [0,1]$ is the learning rate. At each step, the parameter is updated as $\tau^k = \beta^{k-1}$.

In the second step the compatibility index between two rules $\left(\rho_{ij}^k\right)$ is calculated as in Equation 3-5.

$$\rho_{ij}^k = 1 - \frac{1}{m}\sum_{l=1}^{m}\left|v_{il}^k - v_{jl}^k\right| \tag{3-5}$$

in which $\rho_{ij}^k \in [0,1]$ is the compatibility index between the i-th and j-th rules at the k-th step, for $i \neq j$.

This index measures how redundant the rules are, and evaluates whether an existing rule should be removed by comparing it with a parameter $\gamma^k \in [0,1]$. If $\rho_{ij}^k > \gamma^k$, the least compatible rule is removed, and its center $\left(v_j^k\right)$ is merged with the most compatibility rule center $\left(v_i^k\right)$ according to Equation 3-6.

$$\left\{v_{ij}^k = \frac{v_i^k + v_j^k}{2}, \quad if \quad \rho_{ij}^k > \gamma^k \right. \tag{3-6}$$

in which v_{ij}^k is the merged center between the i-th and j-th rules at the k-th step, for $i \neq j$. At each step, the parameter is updated as $\gamma^k = 1 - \beta^{k-1}$.

3.3 Third Stage: β Parameter Updating

This stage of the model updates the value of β, which is the great innovation in relation to previous eTS and ePL approaches. The β parameter, which controls the rate of change of arousal, is updated with the concept of Variable Step proposed by [8]. The normalized error is compared with a parameter $\bar{\gamma} \in [0, 1]$. If $\left|\tilde{e}^k\right| > \bar{\gamma}$, β increases and arousal index increases. Otherwise, β decreases and the arousal index decreases. Equation 3-7 represents this procedure:

$$\beta^{k+1} = \begin{cases} \frac{\beta^k}{\alpha_{VS1}}, & if \quad \left|\tilde{e}^k\right| > \bar{\gamma} \\ \beta^k \ \alpha_{VS2}, & otherwise \end{cases} \tag{3-7}$$

in which $\alpha_{VS1} \in \ ]0, 1[$ and $\alpha_{VS2} \in \ ]0, 1[$ are the factors that increases or decreases parameter β, respectively.

3.4 Fourth Stage: θ Parameter Estimation

This stage of the model estimates the value of θ according to the strategy described in [7]. As shown in Equation 3-1, each rule results in a local output $(\hat{y}_i)$, which can be rewritten as:

$$\begin{aligned} \hat{y}_i^k &= \sum_{j=1}^{n_i} \theta_{ij}^k \ \kappa\left(d_{ij}^k, x^k\right) \\ \hat{y}_i^k &= \sum_{j=1}^{n_i} \theta_{ij}^k exp\left(-\frac{\left\|d_{ij}^k - x^k\right\|}{\sqrt{2}\nu_{ij}^k}\right)^2 \end{aligned} \tag{3-8}$$

in which $\hat{y}_i^k$ is the local output of the i-th rule at the k-th step, $\theta_{ij}^k \in \mathbb{R}$ is the j-th consequent parameters of the i-th rule at the k-th step, and $\kappa(\cdot)$ is the Gaussian kernel function and its parameter ν_{ij}^k is the kernel size. A dictionary with n_i elements $\left(\mathcal{D}_i^k\right)$ is defined for the i-th rule at the k-th step and $d_{ij}^k \in \mathbb{R}^m$ is the j-th element of the i-th dictionary.

The consequent parameters are estimated using an adaptive method known as Kernel Recursive Least Squares (KRLS). It is an extension of Recursive Least Squares (RLS) that increases the dimension of the input data space using a nonlinear function, generating simpler solutions. These values are calculated by determining the weight vector (ω) with the minimization cost function $(L(\omega))$. The optimization is represented by:

$$\min_{\omega} L(\omega) = \sum_{j=1}^{k} \left|y^j - \omega^T \phi^j\right|^2 + \lambda \left|\left|\omega\right|\right|^2 \tag{3-9}$$

in which $\lambda \in [10^{-5}, 10^{-2}]$ is a regularization parameter [34], $\phi : \mathbb{R}^m \to \mathbb{H}$ is a nonlinear function, where operations with the input data is viewed in a high dimensional Hilbert space (neither the ϕ nor the $\mathbb{H}$ needs to be explicitly found due to kernel trick [35]) and y^j is the output at the k-th step.

The solution of Equation 3-9 is obtained from:

$$\begin{aligned} \omega^k &= \left[\lambda \mathbf{I} + \Phi^k \left(\Phi^k\right)^T\right]^{-1} \Phi^k Y^k \\ \omega^k &= \Phi^k \left[\lambda \mathbf{I} + \left(\Phi^k\right)^T \Phi^k\right]^{-1} Y^k \\ \omega^k &= \Phi^k \left[\lambda \mathbf{I} + \mathbf{K}^k\right]^{-1} Y^k \\ \omega^k &= \Phi^k Q^k Y^k \\ \omega^k &= \Phi^k \theta^k \end{aligned} \tag{3-10}$$

in which $\mathbf{I}$ is the identity matrix, $\Phi = \left[\phi^1, \ldots, \phi^k\right]$ is the high dimensional space vector, and $Y = \left[y^1, \ldots, y^k\right]^T$ is the output vector.

The kernel function can be used for generalizations of linear methods written as internal products in a Reproducing Kernel Hilbert Space (RKHS) [36], reducing its dimension and making it possible to solve Equation 3-10. It is represented by:

$$\begin{aligned} \mathbf{K}^k &= \left(\Phi^k\right)^T \Phi^k \\ \mathbf{K}^k &= \begin{bmatrix} \kappa\left(x^1, x^1\right) & \ldots & \kappa\left(x^1, x^k\right) \\ \vdots & \ddots & \vdots \\ \kappa\left(x^k, x^1\right) & \ldots & \kappa\left(x^k, x^k\right) \end{bmatrix} \\ \mathbf{K}^k &= \begin{bmatrix} \mathbf{K}^{k-1} & g^k \\ \left(g^k\right)^T & 1 \end{bmatrix} \end{aligned} \tag{3-11}$$

in which $\mathbf{K}^k$ is the kernel matrix and $g^k = \left[\Phi^{k-1}\right]^T \phi^k = \left[\kappa\left(x^1, x^k\right), \ldots, \kappa\left(x^{k-1}, x^k\right)\right]^T$.

The calculation of matrix $\left[\lambda \mathbf{I} + \mathbf{K}^k\right]$ is computationally expensive. Thus, Q^k variable is defined to recursively approximate it as follows:

$$\begin{aligned} Q^k &= \left[\lambda \mathbf{I} + \mathbf{K}^k\right]^{-1} \\ Q^k &= \left(r^k\right)^{-1} \begin{bmatrix} Q^{k-1} r^k + z^k \left(z^k\right)^T & -z^k \\ -\left(z^k\right)^T & 1 \end{bmatrix} \end{aligned} \tag{3-12}$$

in which $z^k = Q^{k-1} g^k$, $r^k = \lambda + \left[\phi^k\right]^T \phi^k - \left[z^k\right]^T g^k$, and $\tilde{e}^k$ is the error, that is, the difference between a real value (y^k) and a predicted value ($\hat{y}^k$).

The updated solution of consequent parameters is expressed by:

$$\theta^k = Q^k Y^k$$
$$\theta^k = \begin{bmatrix} \theta^{k-1} - z^k [r^k]^{-1} \tilde{e}^k \\ [r^k]^{-1} \tilde{e}^k \end{bmatrix} \tag{3-13}$$

in which θ^k contains the consequent parameters at the k-th step.

It is important to note that the size of the kernel matrix increases quadratically in relation to the amount of data and, consequently, Q^k increases in the same way. This implies more and more expensive operations over time. To get around this situation, a sparcification procedure is used as in [7], because it significantly decreases the required simulation time and memory, and also increases the generalization ability [36]. The dictionary belonging to the rule with the highest compatibility is changed, which means that only the subset of the most relevant samples is considered to update the kernel matrix and parameters vector.

In this model, the novelty criteria [37] are used because they compactly represent knowledge in each local dictionary, being therefore computationally inexpensive. The technique consists in calculating the minimum distance (ψ) between an input and each element of the dictionary by means of:

$$\psi = \min_{(\forall d_{ij} \in \mathcal{D}_i^k)} \left\| x^k - d_{ij}^k \right\| \tag{3-14}$$

in which ψ is the minimum distance between an input and each element in the dictionary.

Two error scenarios are possible. The first considers the new sample at a local dictionary (ε); and the second assumes that the sample is not at the local dictionary (ϵ). If $\psi < \delta$, the sample and all elements of the local dictionary are coherent and are not added to the local dictionary. Otherwise, the sample and all elements of the local dictionary are not coherent and are added to the local dictionary if and only if $\varepsilon < \epsilon$. The value of $\delta = \frac{v_{ij}^k}{10}$ is calculated by an heuristic method [37], where ν_{ij}^k is the kernel size of the element d_{ij}^k, which is the nearest in relation to x^k.

The process for choosing an appropriate value of kernel size is complex. If it is too large, then the inner products for all data become closer to one. If it is too small, then all the data look distinct. The first results in a linear regression and the second results in over-fitting. In this model, the recursive Levenberg-Marquardt algorithm [38] is used. It is a non-linear optimization method that is an intermediate version between gradient and Newton and it has better convergence properties among the three algorithms. Each element ν_{ij}^k in the vector of kernel parameters $\left(\nu_i^k = \left[\nu_{i1}^k, ..., \nu_{in_i}^k\right]\right)$ is associated with an element d_{ij}^k in the local dictionary vector $\mathcal{D}_i^k = \left[d_{i1}^k, ..., d_{in_i}^k\right]$. The objective

is to minimize the local error function. This value is initialized with 0.5 for a new rule. Additionally, this value for an existing rule is expressed by:

$$\nu_i^k = \nu_i^{k-1} + P_i^k \nabla_i^k \tilde{e}_i^k \tag{3-15}$$

in which $P_i^k = \left[P_i^{k-1} - \frac{P_i^{k-1} \nabla_i^k [\nabla_i^k]^T P_i^{k-1}}{1 + [\nabla_i^k]^T P_i^{k-1} \nabla_i^k} \right]$,

$\nabla_i^k = \Lambda_i^k \begin{bmatrix} \theta_{i1}^{k-1} - \frac{\left\| x^k - d_{i1}^k \right\|^2}{(v_{i1}^{k-1})^3} k(x^k, d_{i1}^k) \\ \vdots \\ \theta_{in_1}^{k-1} - \frac{\left\| x^k - d_{in_1}^k \right\|^2}{(v_{in_1}^{k-1})^3} k(x^k, d_{in_1}^k) \end{bmatrix}$ and

$\Lambda_i^k \in [0, 1]$ is the normalized activation degree of the i-th rule.

3.5 Fifth Stage: Global Output

This stage consists of providing a global output. The estimated value is the weighted average of local outputs considering individual rules contributions, resulting in the non-linear nature of the model. The global estimation is given by Equation 3-16:

$$\hat{y} = \sum_{i=1}^{R} \hat{y}_i \Lambda_i \tag{3-16}$$

in which, $\hat{y}$ is the global output, $\hat{y}_i$ is the local output of the i-th rule and Λ_i is the normalized activation degree of the i-th rule.

Next we summarize the learning algorithm to produce and keep a VS-ePL-KRLS model updated from an online data stream.

Algorithm 1: VS-ePL-KRLS

input : $x^k, k = 1, \ldots, n$
output: $\hat{y}^k, k = 1, \ldots, n$

for $k = 2, 3, \ldots, n$ **do**
  for $i = 1, 2, \ldots, R$ **do**
    calculate compatibility: $\rho_i^k = 1 - \frac{\left\|x^k - v_i^k\right\|}{m}$
    calculate arousal: $a_i^k = a_i^{k-1} + \beta^{k-1}\left(1 - \rho_i^k - a_i^{k-1}\right)$
    calculate local output: $\hat{y}_i^k = \sum_{j=1}^{|\mathcal{D}_i|} \theta_{ij}^{k-1}\ \kappa\left(d_{ij}^k, x^k\right)$
  calculate parameter controls rate of change of arousal:
  if $\left|\tilde{e}^k\right| > \bar{\gamma}$ **then**
    $\beta^k = \frac{\beta^{k-1}}{\alpha_{VS1}}$
  else
    $\beta^k = \beta^{k-1}\alpha_{VS2}$
  if $\beta^k < IL$ **then**
    $\beta^k = IL$
  if $\beta^k > SL$ **then**
    $\beta^k = SL$
  $\tau^k = \beta^{k-1}$
  $\gamma^k = 1 - \beta^{k-1}$
  if $\min\left|a^k\right| > \tau$ **then**
    calculate added center: $v_R^{k+1} = x^k$
    initialize local dictionary with sample value: $\mathcal{D}_i^k = x^k$
    calculate kernel size: $\nu_{R1}^k = 0.5$
    calculate parameter consequent: $\theta_i^k = \left[\lambda + \kappa(x^k, x^k)\right]^{-1} y^{k-1}$
  else
    calculate updated center: $v_i^{k+1} = v_i^k + \alpha\left(v_i^k\right)^{1-a_i^{k+1}}\left(x^k - v_i^k\right)$
    calculate minimum distance: $\psi = \min_{(\forall d_{ij} \in \mathcal{D}_i^k)} \left\|x^k - d_{ij}^k\right\|$
    calculate errors: $\tilde{e}_i^k = \left|y_i^k - \hat{y}_i^k\right|$
    if $\psi \geq \delta\ \ \&\ \ \varepsilon < \epsilon$ **then**
      add sample at local dictionary: $\mathcal{D}_i^k = \mathcal{D}_i^k \cup x^k$
    calculate kernel size: $\nu_{R1}^k = 0.5$
    calculate parameter consequent: $\theta_i^k = \begin{bmatrix} \theta_i^{k-1} - z_i^k [r_i^k]^{-1} \tilde{e}_i^k \\ [r_i^k]^{-1} \tilde{e}_i^k \end{bmatrix}$
  for $i, j = 1, 2, \ldots, c, i \neq j$ **do**
    calculate centers compatibility: $\rho_{ij}^k = 1 - \frac{1}{m}\sum_{l=1}^{m}\left|v_{il}^k - v_{jl}^k\right|$
    if $\rho_{ij}^k > \gamma$ **then**
      calculate merged center: $v_i^k = \frac{v_i^k + v_j^k}{2}$
  calculate global output: $\hat{y}^k = \sum_{i=1}^{R} \hat{y}_i \Lambda_i$

4
Methodology

Transport management is a main logistical process responsible for integrating the links in a supply chain, that is, the supply of inputs and raw materials between suppliers and producers and the distribution of final products between producers and customers, respectively. Such management must be done efficiently and effectively to minimize costs and maximize the quality of service.

According to [2], transportation costs correspond to approximately 60% of a company's logistical costs. Of these expenses, 40% represent fixed costs, such as salaries for drivers, helpers, and mechanics. The remaining 60% represents variable costs, particularly maintenance, tolls, and fuel, in the case of a heavy truck used to transport cargo [3]. This study estimates that the fuel is equivalent to about 58% of the variable costs and, consequently, 35% of the transport expenses. In a country of continental dimensions such as Brazil, which primarily uses road networks to transport cargo and people, a computational tool that allows reliable forecasts of the prices of the S500 and S10 diesel oil becomes a competitive differential for logistic operations. As an example, the model can be used to optimize delivery, defining better dates and locations for supply, aiming at minimizing the costs of delivery and, consequently, maximizing the profit of a company.

The fuel price dataset used to evaluate the VS-ePL-KRLS modeling approach described in Chapter 3 was obtained from the website of the National Agency for Petroleum, Natural Gas and Biofuels (ANP) [39]. Resale and distribution prices are considered. The main difference between diesel oils is related to the concentration of sulfur that each contains: S500 has 500 parts per million and S10 has 10 parts per million. This directly affects engine performance and the environment, in which the S10 diesel oil has better ignition and combustion functioning, suffers less from corrosion and is less polluting.

The diesel oil market in Brazil is regulated by the ANP and Federal Law 9.478/97. The price of diesel oil is composed of: 40% pure diesel, 11% biodiesel, 11% state taxes, 16% national taxes, and 20% profit (the final product must contain a mixture of 88% of pure diesel and 12% of biodiesel) [40].

4.1
Historic Contextualization of the Fuel Price Policy in Brazil

Until 2016, the diesel oil price in Brazil was controlled by Petrobras, which delayed the transfer of its international variations as a tool to control inflation. After several financial losses, Petrobras, as a state-owned company, changed this policy. The price started to be controlled by the barrel of oil in the international market, which depends on the price of the dollar and oil. As the company prices are due to several political and economic factors on a global scale, which has dynamic characteristics, some nonstationarities are noticeable since 2017.

Successive increases occurred in fuel prices, especially S500 diesel oil and S10 diesel oil which increased R$ 0.293 (8.383%) and R$ 0.278 (7.755%) in the first three weeks of May 2018, respectively. This resulted in the truckers' strike (also called the diesel crisis) which lasted from May 21st, 2018 to May 30th, 2018 throughout the national territory which had several claims, including a reduction in the price of diesel.

The international price of a barrel of oil plummeted, reached -US$ 37.63 on April 20th, 2020, an unprecedented event in history. Its dynamic character is an example of behavior that requires prediction models to be adapted or redesigned to keep their accuracies up.

4.2
Statistical Analysis

The behavior of historical prices can be observed in Figure 4.1 and a preliminary statistical analysis is presented in Table 4.1 for weekly data periodicity. The dataset is from December 30th, 2012 to May 30th, 2020.

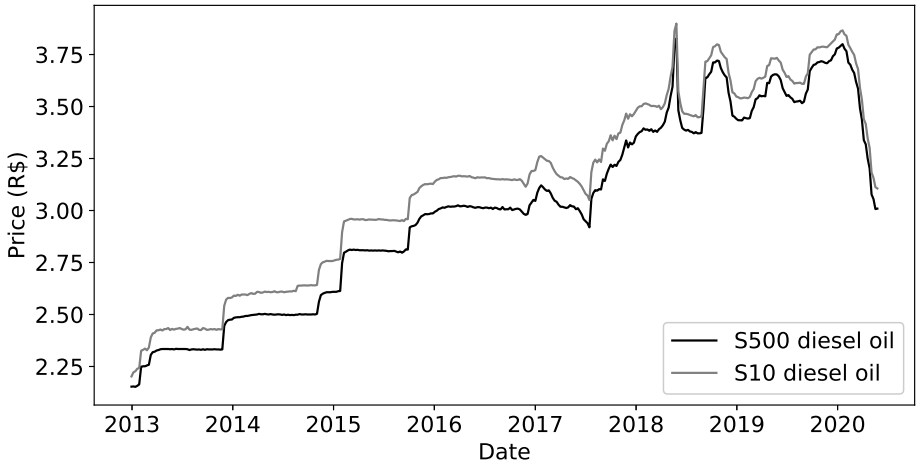

Figure 4.1: Historical diesel oil prices for weekly data periodicity

Table 4.1: Statistical Analysis for weekly data periodicity

	S500 diesel oil	S10 diesel oil
Data Samples	386	386
Mean	3.008	3.124
Standard Deviation	0.454	0.447
Minimum	2.152	2.202
First Quartile	2.518	2.660
Median	3.014	3.155
Third Quartile	3.391	3.504
Maximum	3.828	3.899

The behavior of historical prices can be observed in Figure 4.2 and a preliminary statistical analysis is presented in Table 4.2 for monthly data periodicity. The dataset is from December 2012 to May 2020.

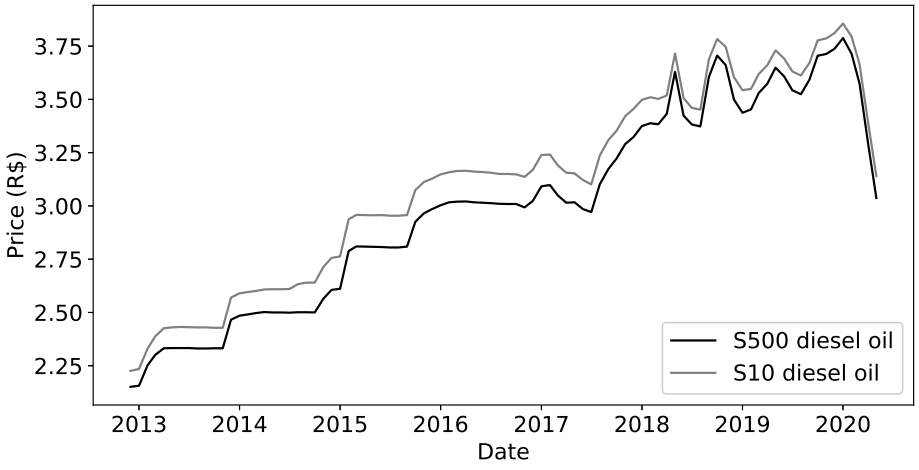

Figure 4.2: Historical diesel oil prices for monthly data periodicity

Table 4.2: Statistical Analysis for monthly data periodicity

	S500 diesel oil	S10 diesel oil
Data Samples	90	90
Mean	2.999	3.114
Standard Deviation	0.460	0.453
Minimum	2.151	2.226
First Quartile	2.518	2.658
Median	3.014	3.152
Third Quartile	3.387	3.505
Maximum	3.788	3.856

4.3 Forecasting Evaluation and Computational Performance

The forecasting evaluation is measured by means of three error metrics: Root Mean Square Error (RMSE), Mean Absolute Error (MAE) and Non-Dimensional Error Index (NDEI), expressed by Equations 4-1, 4-2 and 4-3, respectively.

$$RMSE = \sqrt{\frac{1}{n}\sum_{k=1}^{n}(y^k - \hat{y}^k)^2} \tag{4-1}$$

in which n is the sample size, y^k is the actual value and $\hat{y}^k$ is the predicted value.

$$MAE = \frac{1}{n}\sum_{k=1}^{n}\left|y^k - \hat{y}^k\right| \tag{4-2}$$

in which n is the sample size, y^k is the actual value and $\hat{y}^k$ is the predicted value.

$$NDEI = \frac{1}{n}\sum_{k=1}^{n}\left|\frac{y^k - \hat{y}^k}{y^k}\right| \tag{4-3}$$

in which n is the sample size, y^k is the actual value and $\hat{y}^k$ is the predicted value.

The RMSE and MAE are in the same unit that dataset (for this case, it is the Real (R$) unit). The main difference between them is that the RMSE penalizes large errors and benefits small errors (between 0 and 1). The NDEI is a percentage error. The training data and test data are divided into 80% and 20% of the entire dataset, respectively.

This approach was statistically validated with Morgan Granger Newbold (MGN) Test [41] to evaluate if the accuracy between models is equivalent or not. MGN test follows a student's t distribution with $n-1$ degrees of freedom and it is represented by Equation 4-4:

$$MGN = \frac{\hat{\varrho}}{\sqrt{\frac{1-\hat{\varrho}}{n-1}}} \tag{4-4}$$

in which $\hat{\varrho}$ is the correlation coefficient (Pearson coefficient) between sum and difference of errors for a time series with a size equal to n. The null hypothesis is that two forecast error variances are equivalent.

The computational performance evaluation is measured by calculating the average value in 30 simulations of the percentage of the current system-wide CPU utilization. The test is performed in a computer with Intel(R) Core(TM) Processor i3-7020U CPU @ 2.30GHz 2.30 GHz, 4.00 GB installed memory(RAM) (with 3.87 GB usable) and 64-bit Operating System.

5
Experimental Results

To evaluate the effectiveness of the method proposed in Chapter 3, the VS-ePL-KRLS is applied for a real time series (S500 and S10 diesel oil Prices) and a benchmark time series (Box-Jenkins Gas Furnace). Thus, it is compared with the best models presented in this context as ePL-KRLS [7], SM-ePL-KRLS [27] and ESM-ePL-KRLS [27].

The hyperparameters are defined using an optimized method in which every possible combination of a finite number of these values is tested and the composition that results in the smallest RMSE is chosen. This technique is used to define the learning rate (α) and the parameter that controls the rate of change of arousal (β) for ePL-KRLS, both varying between 0.01 and 1.00 (in 0.01 increments). The threshold value for creating or updating the rules (τ) and the threshold value for merging the rules (γ) are set equal to β value and β complement as in SM-ePL-KRLS, ESM-ePL-KRLS and VS-ePL-KRLS). For these three models, the same α, β, τ, and γ were used. However, γ, α_{VS1} and α_{VS2} are optimized. The first varying between 0.001 and 0.050 (in 0.001 increments) for SM-ePL-KRLS, ESM-ePL-KRLS and VS-ePL-KRLS, the second and third varying between 0.01 and 1.00 (in 0.01 increments) for VS-ePL-KRLS.

The results are measured in terms of Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Non-Dimensional Error Index (NDEI). Then, they are statistically validated with Morgan Granger Newbold (MGN) Test. Finally, the computational performance is measured by calculating the average value in 30 simulations of the percentage of the current system-wide CPU utilization, and the rules are described as a rounded average through the process.

5.1
Diesel Oil Price

The real application is composed of ten time series divided into two data periodicity: weekly and monthly. The first is to predict the biweekly and monthly horizons, both S500 and S10 diesel oil. The second is to predict the monthly, semiannual and yearly horizons, both S500 and S10 diesel oil.

The hyperparameters for S500 diesel oil weekly prices for a biweekly forecasting horizon (two steps ahead) are represented at Table 5.1. In addition, $\alpha = 0.10$, $\beta = 0.02$, $\tau = 0.02$, and $\gamma = 0.98$.

Table 5.1: Hyperparameters – S500 diesel oil for a biweekly forecasting horizon and weekly data periodicity

Algorithm	$\bar{\gamma}$	α_{VS1}	α_{VS2}
SM-ePL-KRLS	0.010	-	-
ESM-ePL-KRLS	0.016	-	-
VS-ePL-KRLS	0.003	0.79	0.64

The experimental results are represented in Table 5.2, which shows the average number of rules, error metrics (RMSE, MAE, and NDEI), and computational complexity (percentage of CPU cycles).

Table 5.2: Results – S500 diesel oil for a biweekly forecasting horizon and weekly data periodicity

Algorithm	Rules	RMSE	MAE	NDEI	%CPU
ePL-KRLS	**2**	0.04834	0.03750	0.10736	19.02 ± 5.84
SM-ePL-KRLS	9	0.04753	0.03367	0.10557	18.44 ± 3.07
ESM-ePL-KRLS	3	0.05038	0.03921	0.11189	17.09 ± 2.87
VS-ePL-KRLS	3	**0.03865**	**0.03018**	**0.08584**	**16.65 ± 2.76**

The VS-ePL-KRLS model presents the lowest errors with a reduction of 18.68% of the RMSE, 10.37% of the MAE, and 18.69% of the NDEI about the second-best model (SM-ePL-KRLS). The lowest percentage of the current system-wide CPU utilization is seen in VS-ePL-KRLS. Concerning the average number of rules, ePL-KRLS, ESM-ePL-KRLS, and VS-ePL-KRLS return the lowest values. Figures 5.1 and 5.2 illustrate the prediction values and rules evolution, respectively.

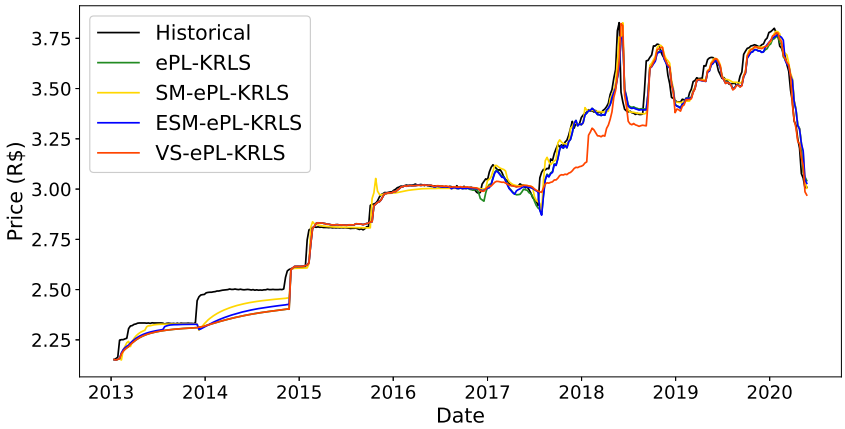

Figure 5.1: Prediction – two steps ahead S500 diesel oil price using weekly data

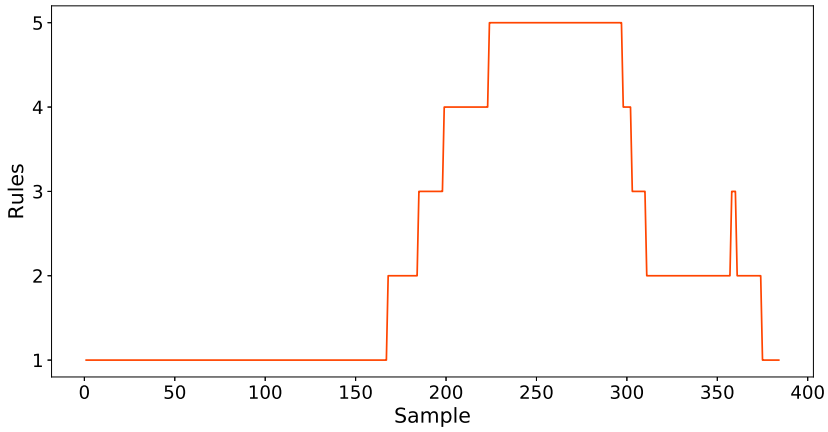

Figure 5.2: VS-ePL-KRLS rules evolution – two steps ahead S500 diesel oil price using weekly data

This result is analyzed statistically in Table 5.3. The null hypothesis (which assumes the models have equal accuracy) is rejected with a significance of 5% when comparing the VS-ePL-KRLS with all the other three models.

Table 5.3: MGN test – S500 diesel oil for a biweekly forecasting horizon and weekly data periodicity

Comparison	MGN	p-value	Observation
VS-ePL-KRLS x ePL-KRLS	-4.178	8.06×10^{-5}	H_0 **is rejected**
VS-ePL-KRLS x SM-ePL-RKLS	-4.966	4.34×10^{-6}	H_0 **is rejected**
VS-ePL-KRLS x ESM-ePL-KRLS	-5.328	1.06×10^{-6}	H_0 **is rejected**

The hyperparameters for S10 diesel oil weekly prices for a biweekly forecasting horizon (two steps ahead) are represented at Table 5.4. In addition, $\alpha = 0.02$, $\beta = 0.05$, $\tau = 0.05$, and $\gamma = 0.95$.

Table 5.4: Hyperparameters – S10 diesel oil for a biweekly forecasting horizon and weekly data periodicity

Algorithm	$\bar{\gamma}$	α_{VS1}	α_{VS2}
SM-ePL-KRLS	0.008	-	-
ESM-ePL-KRLS	0.015	-	-
VS-ePL-KRLS	0.001	0.94	0.74

The experimental results are represented in Table 5.5, which shows the average number of rules, error metrics (RMSE, MAE, and NDEI), and computational complexity (percentage of CPU cycles).

Table 5.5: Results – S10 diesel oil for a biweekly forecasting horizon and weekly data periodicity

Algorithm	Rules	RMSE	MAE	NDEI	%CPU
ePL-KRLS	**2**	0.03974	0.03087	0.08979	**15.13 ± 2.75**
SM-ePL-KRLS	11	0.04238	0.02920	0.09575	17.54 ± 2.36
ESM-ePL-KRLS	7	0.04402	0.03316	0.09947	19.01 ± 6.36
VS-ePL-KRLS	**2**	**0.03616**	**0.02755**	**0.08170**	15.27 ± 2.69

The VS-ePL-KRLS model presents the lowest errors with a reduction of 9.01% of the RMSE, 10.75% of the MAE, and 9.01% of the NDEI about the second-best model (ePL-KRLS). The lowest percentage of the current system-wide CPU utilization is seen in ePL-KRLS. Concerning the average number of rules, ePL-KRLS and VS-ePL-KRLS return the lowest values. Figures 5.3 and 5.4 illustrate the prediction values and rules evolution, respectively.

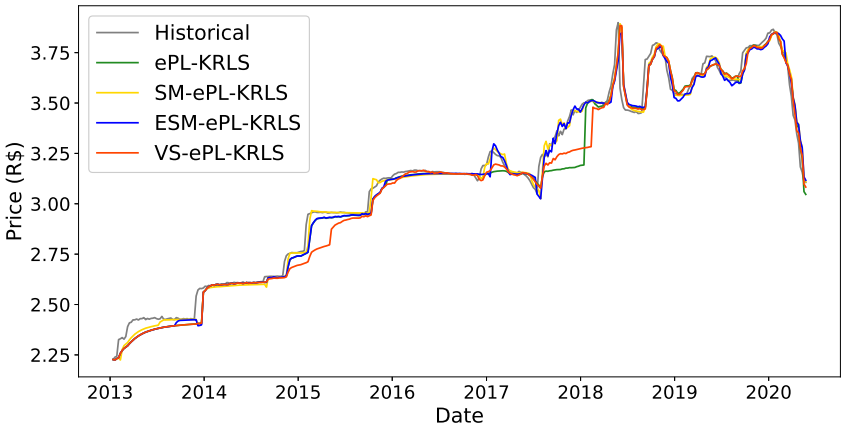

Figure 5.3: Prediction – two steps ahead S10 diesel oil price using weekly data

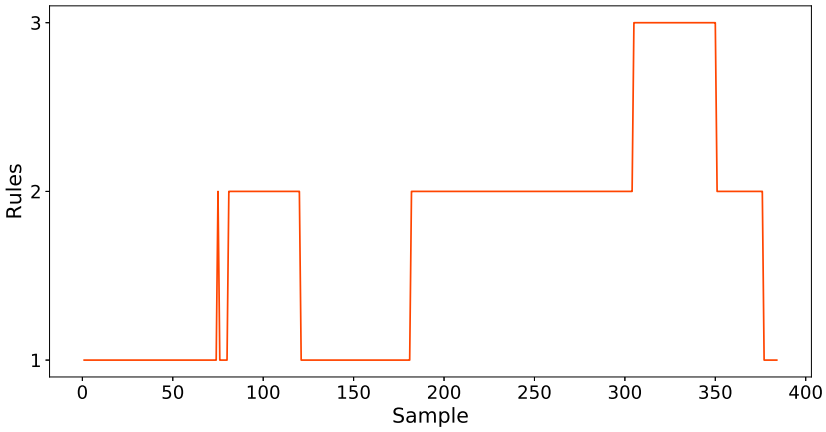

Figure 5.4: VS-ePL-KRLS rules evolution – two steps ahead S10 diesel oil price using weekly data

This result is analyzed statistically in Table 5.6. The null hypothesis (which assumes the models have equal accuracy) is rejected with a significance of 5% when comparing the VS-ePL-KRLS with all the other three models.

Table 5.6: MGN test – S10 diesel oil for a biweekly forecasting horizon and weekly data periodicity

Comparison	MGN	p-value	Observation
VS-ePL-KRLS x ePL-KRLS	-2.672	9.31×10^{-3}	H_0 **is rejected**
VS-ePL-KRLS x SM-ePL-RKLS	-2.544	1.31×10^{-2}	H_0 **is rejected**
VS-ePL-KRLS x ESM-ePL-KRLS	-2.605	1.11×10^{-2}	H_0 **is rejected**

The hyperparameters for S500 diesel oil weekly prices for a monthly forecasting horizon (four steps ahead) are represented at Table 5.7. In addition, $\alpha = 0.28$, $\beta = 0.01$, $\tau = 0.01$, and $\gamma = 0.99$.

Table 5.7: Hyperparameters – S500 diesel oil for a monthly forecasting horizon and weekly data periodicity

Algorithm	$\bar{\gamma}$	α_{VS1}	α_{VS2}
SM-ePL-KRLS	0.028	-	-
ESM-ePL-KRLS	0.002	-	-
VS-ePL-KRLS	0.008	0.57	0.34

The experimental results are represented in Table 5.8, which shows the average number of rules, error metrics (RMSE, MAE, and NDEI), and computational complexity (percentage of CPU cycles).

Table 5.8: Results – S500 diesel oil for a monthly forecasting horizon and weekly data periodicity

Algorithm	Rules	RMSE	MAE	NDEI	%CPU
ePL-KRLS	2	0.08350	0.06421	0.18675	**16.22 ± 2.38**
SM-ePL-KRLS	17	0.08725	0.06780	0.19512	20.07 ± 5.70
ESM-ePL-KRLS	**1**	0.09526	0.07033	0.21304	17.39 ± 4.54
VS-ePL-KRLS	4	**0.06011**	**0.04886**	**0.13442**	17.78 ± 3.17

The VS-ePL-KRLS model presents the lowest errors with a reduction of 28.01% of the RMSE, 23.91% of the MAE, and 28.02% of the NDEI about the second-best model (ePL-KRLS). The lowest percentage of the current system-wide CPU utilization is seen in ePL-KRLS. Concerning the average number of rules, ESM-ePL-KRLS and ePL-KRLS return the lowest values. Figures 5.5 and 5.6 illustrate the prediction values and rules evolution, respectively.

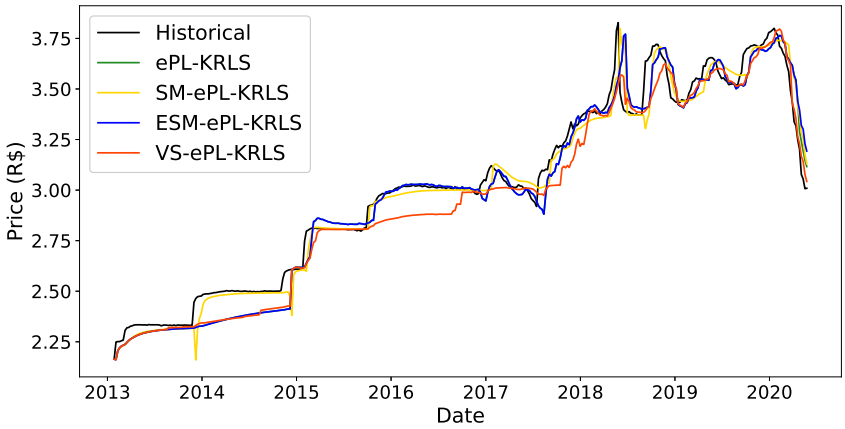

Figure 5.5: Prediction – four steps ahead S500 diesel oil price using weekly data

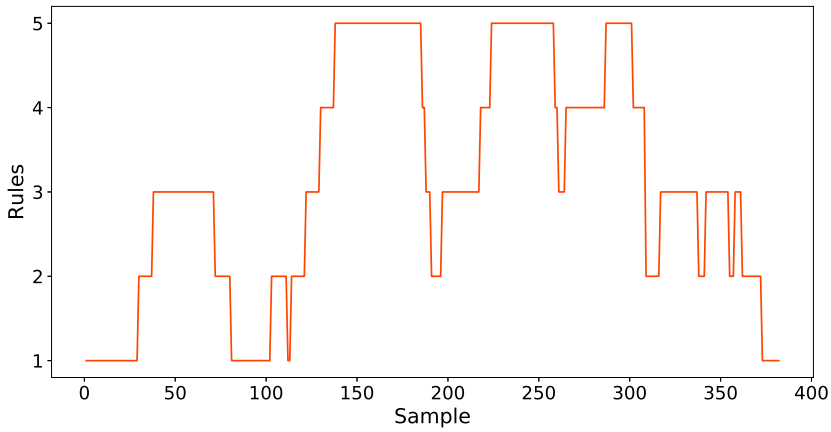

Figure 5.6: VS-ePL-KRLS rules evolution – four steps ahead S500 diesel oil price using weekly data

This result is analyzed statistically in Table 5.9. The null hypothesis (which assumes the models have equal accuracy) is rejected with a significance of 5% when comparing the VS-ePL-KRLS with all the other three models.

Table 5.9: MGN test – S500 diesel oil for a monthly forecasting horizon and weekly data periodicity

Comparison	MGN	p-value	Observation
VS-ePL-KRLS x ePL-KRLS	-5.593	3.67×10^{-7}	H_0 **is rejected**
VS-ePL-KRLS x SM-ePL-RKLS	-5.445	6.63×10^{-7}	H_0 **is rejected**
VS-ePL-KRLS x ESM-ePL-KRLS	-7.364	2.19×10^{-10}	H_0 **is rejected**

The hyperparameters for S10 diesel oil weekly prices for a monthly forecasting horizon (four steps ahead) are represented at Table 5.10. In addition, $\alpha = 0.26$, $\beta = 0.01$, $\tau = 0.01$, and $\gamma = 0.99$.

Table 5.10: Hyperparameters – S10 diesel oil for a monthly forecasting horizon and weekly data periodicity

Algorithm	$\bar{\gamma}$	α_{VS1}	α_{VS2}
SM-ePL-KRLS	0.020	-	-
ESM-ePL-KRLS	0.027	-	-
VS-ePL-KRLS	0.002	0.89	0.62

The experimental results are represented in Table 5.11, which shows the average number of rules, error metrics (RMSE, MAE, and NDEI), and computational complexity (percentage of CPU cycles).

Table 5.11: Results – S10 diesel oil for a monthly forecasting horizon and weekly data periodicity

Algorithm	Rules	RMSE	MAE	NDEI	%CPU
ePL-KRLS	**2**	0.07830	0.06252	0.17840	**15.76 ± 3.93**
SM-ePL-KRLS	16	0.08045	0.06023	0.18330	18.81 ± 2.44
ESM-ePL-KRLS	6	0.08170	0.06526	0.18613	17.43 ± 3.15
VS-ePL-KRLS	3	**0.06190**	**0.04841**	**0.14104**	17.33 ± 5.58

The VS-ePL-KRLS model presents the lowest errors with a reduction of 20.95% of the RMSE, 22.57% of the MAE, and 20.94% of the NDEI about the second-best model (ePL-KRLS). The lowest percentage of the current system-wide CPU utilization is seen in ePL-KRLS. Concerning the average number of rules, ePL-KRLS and VS-ePL-KRLS return the lowest values. Figures 5.7 and 5.8 illustrate the prediction values and rules evolution, respectively.

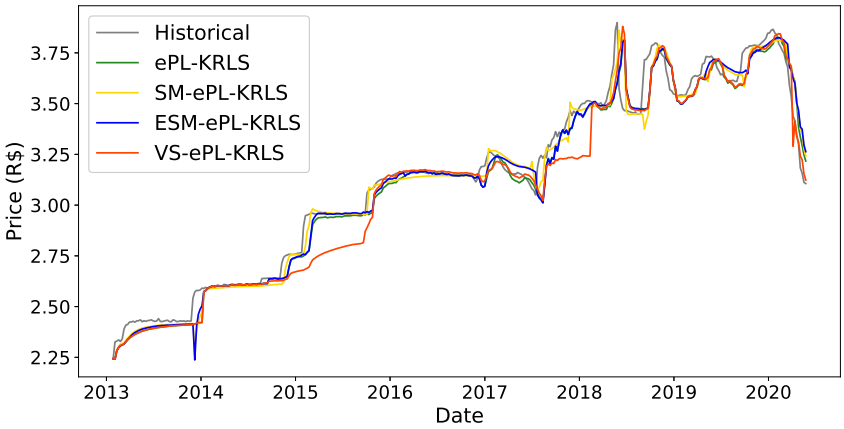

Figure 5.7: Prediction – four steps ahead S10 diesel oil price using weekly data

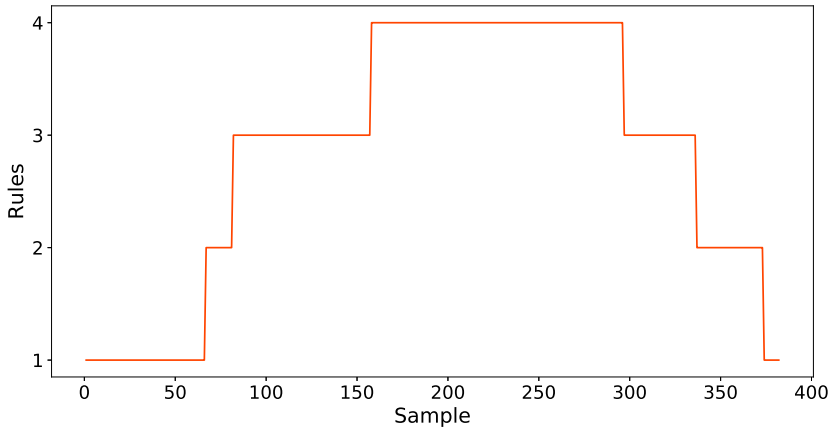

Figure 5.8: VS-ePL-KRLS rules evolution – four steps ahead S10 diesel oil price using weekly data

This result is analyzed statistically in Table 5.12. The null hypothesis (which assumes the models have equal accuracy) is rejected with a significance of 5% when comparing the VS-ePL-KRLS with all the other three models.

Table 5.12: MGN test – S10 diesel oil for a monthly forecasting horizon and weekly data periodicity

Comparison	MGN	p-value	Observation
VS-ePL-KRLS x ePL-KRLS	-3.381	1.16×10^{-3}	H_0 **is rejected**
VS-ePL-KRLS x SM-ePL-RKLS	-3.718	3.91×10^{-4}	H_0 **is rejected**
VS-ePL-KRLS x ESM-ePL-KRLS	-3.716	3.93×10^{-4}	H_0 **is rejected**

The hyperparameters for S500 diesel oil monthly prices for a monthly forecasting horizon (one step ahead) are represented at Table 5.13. In addition, $\alpha = 0.12$, $\beta = 0.01$, $\tau = 0.01$, and $\gamma = 0.99$.

Table 5.13: Hyperparameters – S500 diesel oil for a monthly forecasting horizon and monthly data periodicity

Algorithm	$\bar{\gamma}$	α_{VS1}	α_{VS2}
SM-ePL-KRLS	0.016	-	-
ESM-ePL-KRLS	0.001	-	-
VS-ePL-KRLS	0.015	0.49	0.60

The experimental results are represented in Table 5.14, which shows the average number of rules, error metrics (RMSE, MAE, and NDEI), and computational complexity (percentage of CPU cycles).

Table 5.14: Results – S500 diesel oil for a monthly forecasting horizon and monthly data periodicity

Algorithm	Rules	RMSE	MAE	NDEI	%CPU
ePL-KRLS	2	0.09682	0.07434	0.21475	**14.92 ± 3.34**
SM-ePL-KRLS	4	0.09335	0.07206	0.20704	15.62 ± 2.61
ESM-ePL-KRLS	**1**	0.10189	0.07537	0.22599	15.51 ± 2.51
VS-ePL-KRLS	2	**0.07002**	**0.06026**	**0.15530**	16.92 ± 6.19

The VS-ePL-KRLS model presents the lowest errors with a reduction of 24.99% of the RMSE, 16.38% of the MAE, and 24.99% of the NDEI about the second-best model (SM-ePL-KRLS). The lowest percentage of the current system-wide CPU utilization is seen in ePL-KRLS. Concerning the average number of rules, ESM-ePL-KRLS, ePL-KRLS, and VS-ePL-KRLS return the lowest values. Figures 5.9 and 5.10 illustrate the prediction values and rules evolution, respectively.

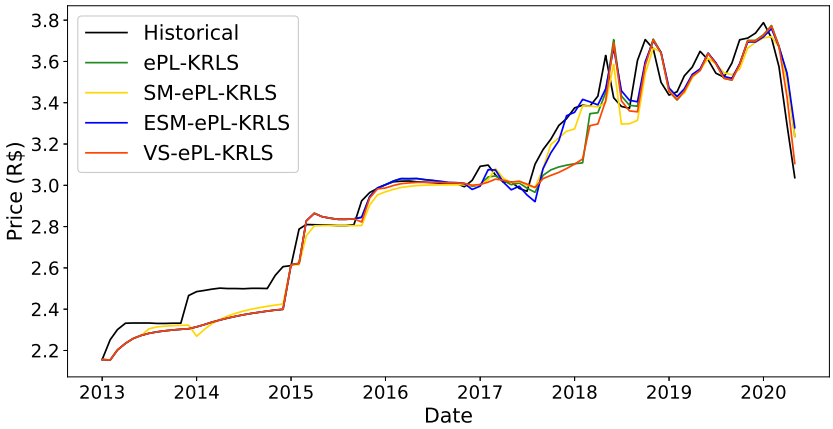

Figure 5.9: Prediction – one step ahead S500 diesel oil price using monthly data

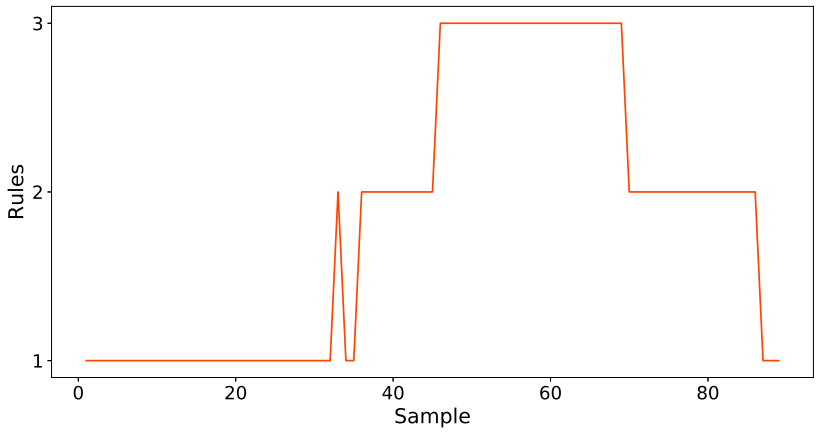

Figure 5.10: VS-ePL-KRLS rules evolution – one step ahead S500 diesel oil price using monthly data

This result is analyzed statistically in Table 5.15. The null hypothesis (which assumes the models have equal accuracy) is rejected with a significance of 5 % when comparing the VS-ePL-KRLS with all the other three models.

Table 5.15: MGN test – S500 diesel oil for a monthly forecasting horizon and monthly data periodicity

Comparison	MGN	p-value	Observation
VS-ePL-KRLS x ePL-KRLS	-4.016	9.98×10^{-4}	H_0 **is rejected**
VS-ePL-KRLS x SM-ePL-RKLS	-2.915	1.01×10^{-2}	H_0 **is rejected**
VS-ePL-KRLS x ESM-ePL-KRLS	-3.711	1.90×10^{-3}	H_0 **is rejected**

The hyperparameters for S10 diesel oil monthly prices for a monthly forecasting horizon (one step ahead) are represented at Table 5.16. In addition, $\alpha = 0.13$, $\beta = 0.01$, $\tau = 0.01$, and $\gamma = 0.99$.

Table 5.16: Hyperparameters – S10 diesel oil for a monthly forecasting horizon and monthly data periodicity

Algorithm	$\bar{\gamma}$	α_{VS1}	α_{VS2}
SM-ePL-KRLS	0.014	-	-
ESM-ePL-KRLS	0.001	-	-
VS-ePL-KRLS	0.006	0.91	0.89

The experimental results are represented in Table 5.17, which shows the average number of rules, error metrics (RMSE, MAE, and NDEI), and computational complexity (percentage of CPU cycles).

Table 5.17: Results – S10 diesel oil for a monthly forecasting horizon and monthly data periodicity

Algorithm	Rules	RMSE	MAE	NDEI	%CPU
ePL-KRLS	2	0.08736	0.06752	0.19709	16.30 ± 3.78
SM-ePL-KRLS	4	0.08221	0.06369	0.18549	18.04 ± 7.29
ESM-ePL-KRLS	**1**	0.09232	0.06737	0.20829	**14.97 ± 3.03**
VS-ePL-KRLS	2	**0.05612**	**0.04889**	**0.12661**	15.20 ± 2.69

The VS-ePL-KRLS model presents the lowest errors with a reduction of 31.74% of the RMSE, 23.24% of the MAE, and 31.74% of the NDEI about the second-best model (SM-ePL-KRLS). The lowest percentage of the current system-wide CPU utilization is seen in ESM-ePL-KRLS. Concerning the average number of rules, ESM-ePL-KRLS, ePL-KRLS, and VS-ePL-KRLS return the lowest values. Figures 5.11 and 5.12 illustrate the prediction values and rules evolution, respectively.

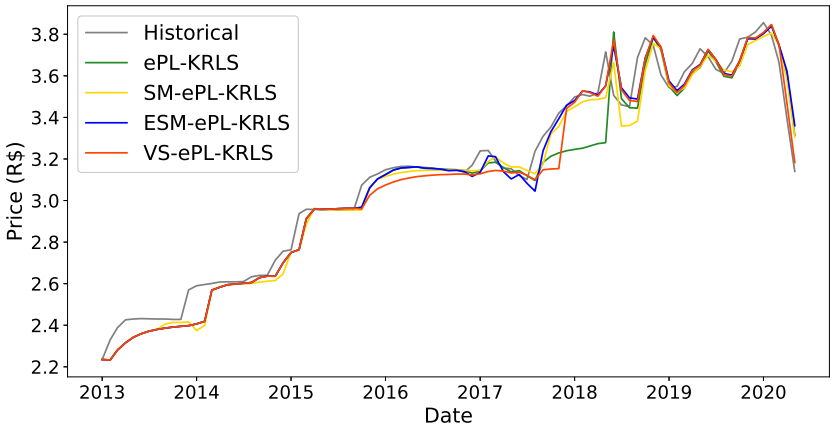

Figure 5.11: Prediction – one step ahead S10 diesel oil price using monthly data

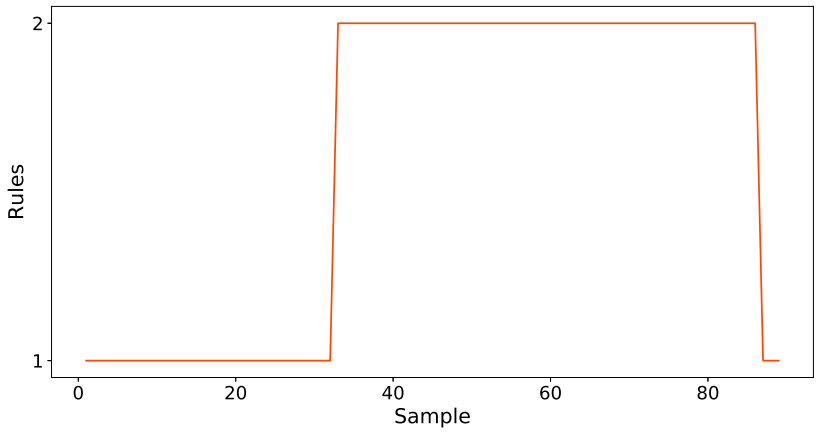

Figure 5.12: VS-ePL-KRLS rules evolution – one step ahead S10 diesel oil price using monthly data

This result is analyzed statistically in Table 5.18. The null hypothesis (which assumes the models have equal accuracy) is rejected with a significance of 5 % when comparing the VS-ePL-KRLS with all the other three models.

Table 5.18: MGN test – S10 diesel oil for a monthly forecasting horizon and monthly data periodicity

Comparison	MGN	p-value	Observation
VS-ePL-KRLS x ePL-KRLS	-3.861	1.38×10^{-3}	H_0 **is rejected**
VS-ePL-KRLS x SM-ePL-RKLS	-3.113	6.70×10^{-3}	H_0 **is rejected**
VS-ePL-KRLS x ESM-ePL-KRLS	-3.824	1.50×10^{-3}	H_0 **is rejected**

The hyperparameters for S500 diesel oil monthly prices for a semiannual forecasting horizon (six steps ahead) are represented at Table 5.19. In addition, $\alpha = 0.02$, $\beta = 0.07$, $\tau = 0.07$, and $\gamma = 0.93$.

Table 5.19: Hyperparameters – S500 diesel oil for a semiannual forecasting horizon and monthly data periodicity

Algorithm	$\bar{\gamma}$	α_{VS1}	α_{VS2}
SM-ePL-KRLS	0.046	-	-
ESM-ePL-KRLS	0.040	-	-
VS-ePL-KRLS	0.009	0.26	0.12

The experimental results are represented in Table 5.20, which shows the average number of rules, error metrics (RMSE, MAE, and NDEI), and computational complexity (percentage of CPU cycles).

Table 5.20: Results – S500 diesel oil for a semiannual forecasting horizon and monthly data periodicity

Algorithm	Rules	RMSE	MAE	NDEI	%CPU
ePL-KRLS	**2**	0.16321	0.11897	0.38342	19.08 ± 12.93
SM-ePL-KRLS	11	0.17548	0.13178	0.41223	17.08 ± 5.46
ESM-ePL-KRLS	4	0.19936	0.12122	0.46835	16.48 ± 2.86
VS-ePL-KRLS	4	**0.11720**	**0.09244**	**0.27534**	**16.31 ± 3.07**

The VS-ePL-KRLS model presents the lowest errors with a reduction of 28.19% of the RMSE, 22.30% of the MAE, and 28.19% of the NDEI about the second-best model (ePL-KRLS). The lowest percentage of the current system-wide CPU utilization is seen in VS-ePL-KRLS. Concerning the average number of rules, ePL-KRLS returns the lowest values. Figures 5.13 and 5.14 illustrate the prediction values and rules evolution, respectively.

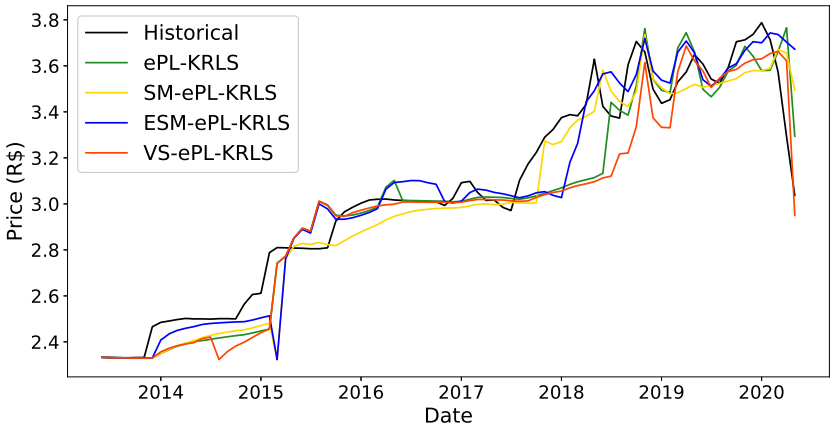

Figure 5.13: Prediction – six steps ahead S500 diesel oil price using monthly data

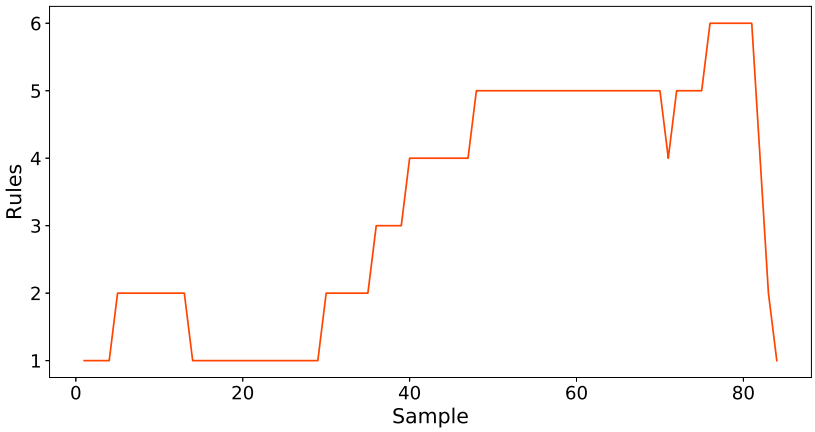

Figure 5.14: VS-ePL-KRLS rules evolution – six steps ahead S500 diesel oil price using monthly data

This result is analyzed statistically in Table 5.21. The null hypothesis (which assumes the models have equal accuracy) is rejected when comparing the VS-ePL-KRLS with ESM-ePL-KRLS and is accepted when comparing the VS-ePL-KRLS with ePL-KRLS and SM-ePL-KRLS, with a significance of 5%.

Table 5.21: MGN test – S500 diesel oil for a semiannual forecasting horizon and monthly data periodicity

Comparison	MGN	p-value	Observation
VS-ePL-KRLS x ePL-KRLS	-2.105	5.15×10^{-2}	H_0 **is accepted**
VS-ePL-KRLS x SM-ePL-RKLS	-1.962	6.74×10^{-2}	H_0 **is accepted**
VS-ePL-KRLS x ESM-ePL-KRLS	-2.154	4.69×10^{-2}	H_0 **is rejected**

The hyperparameters for S10 diesel oil monthly prices for a semiannual forecasting horizon (six steps ahead) are represented at Table 5.22. In addition, $\alpha = 0.17$, $\beta = 0.01$, $\tau = 0.01$, and $\gamma = 0.99$.

Table 5.22: Hyperparameters – S10 diesel oil for a semiannual forecasting horizon and monthly data periodicity

Algorithm	$\bar{\gamma}$	α_{VS1}	α_{VS2}
SM-ePL-KRLS	0.038	-	-
ESM-ePL-KRLS	0.049	-	-
VS-ePL-KRLS	0.004	0.54	0.08

The experimental results are represented in Table 5.23, which shows the average number of rules, error metrics (RMSE, MAE, and NDEI), and computational complexity (percentage of CPU cycles).

Table 5.23: Results – S10 diesel oil for a semiannual forecasting horizon and monthly data periodicity

Algorithm	Rules	RMSE	MAE	NDEI	%CPU
ePL-KRLS	**2**	0.16829	0.11702	0.40672	**14.11 ± 2.19**
SM-ePL-KRLS	10	0.14097	0.10742	0.34069	15.88 ± 2.93
ESM-ePL-KRLS	6	0.18468	0.11673	0.44634	15.46 ± 2.46
VS-ePL-KRLS	3	**0.11922**	**0.09442**	**0.28813**	15.56 ± 2.58

The VS-ePL-KRLS model presents the lowest errors with a reduction of 15.43% of the RMSE, 12.10% of the MAE, and 15.43% of the NDEI about the second-best model (SM-ePL-KRLS). The lowest percentage of the current system-wide CPU utilization is seen in ePL-KRLS. Concerning the average number of rules, ePL-KRLS and VS-ePL-KRLS return the lowest values. Figures 5.15 and 5.16 illustrate the prediction values and rules evolution, respectively.

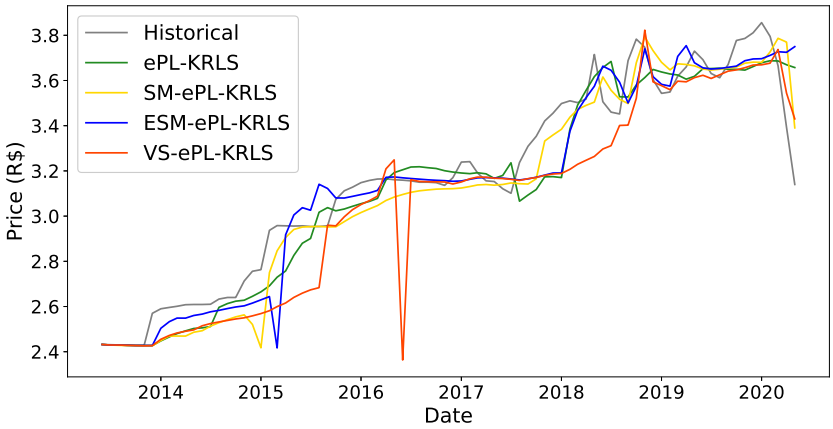

Figure 5.15: Prediction – six steps ahead S10 diesel oil price using monthly data

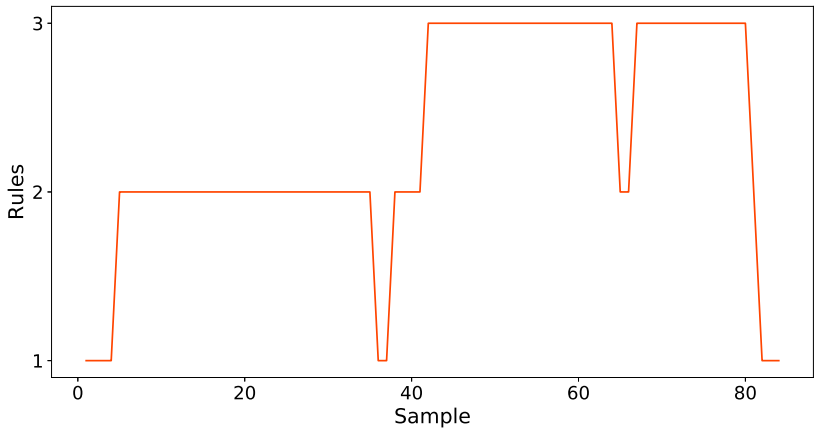

Figure 5.16: VS-ePL-KRLS rules evolution – six steps ahead S10 diesel oil price using monthly data

This result is analyzed statistically in Table 5.24. The null hypothesis (which assumes the models have equal accuracy) is rejected when comparing the VS-ePL-KRLS with ePL-KRLS and ESM-ePL-KRLS and is accepted when comparing the VS-ePL-KRLS with SM-ePL-KRLS, with a significance of 5%.

Table 5.24: MGN test – S10 diesel oil for a semiannual forecasting horizon and monthly data periodicity

Comparison	MGN	p-value	Observation
VS-ePL-KRLS x ePL-KRLS	-6.775	4.46×10^{-6}	H_0 **is rejected**
VS-ePL-KRLS x SM-ePL-RKLS	-1.872	7.96×10^{-2}	H_0 **is accepted**
VS-ePL-KRLS x ESM-ePL-KRLS	-5.402	5.87×10^{-5}	H_0 **is rejected**

The hyperparameters for S500 diesel oil monthly prices for a yearly forecasting horizon (twelve steps ahead) are represented at Table 5.25. In addition, $\alpha = 0.02$, $\beta = 0.06$, $\tau = 0.06$, and $\gamma = 0.94$.

Table 5.25: Hyperparameters – S500 diesel oil for a yearly forecasting horizon and monthly data periodicity

Algorithm	$\bar{\gamma}$	α_{VS1}	α_{VS2}
SM-ePL-KRLS	0.047	-	-
ESM-ePL-KRLS	0.009	-	-
VS-ePL-KRLS	0.009	0.49	0.28

The experimental results are represented in Table 5.26, which shows the average number of rules, error metrics (RMSE, MAE, and NDEI), and computational complexity (percentage of CPU cycles).

Table 5.26: Results – S500 diesel oil for a yearly forecasting horizon and monthly data periodicity

Algorithm	Rules	RMSE	MAE	NDEI	%CPU
ePL-KRLS	**2**	0.16514	0.12451	0.42327	**16.15 ± 3.10**
SM-ePL-KRLS	10	0.18711	0.13054	0.47957	17.42 ± 4.87
ESM-ePL-KRLS	**2**	0.24931	0.18501	0.63900	16.90 ± 4.54
VS-ePL-KRLS	4	**0.11889**	**0.08874**	**0.30474**	17.95 ± 5.40

The VS-ePL-KRLS model presents the lowest errors with a reduction of 28.01% of the RMSE, 28.73% of the MAE, and 28.00% of the NDEI about the second-best model (ePL-KRLS). The lowest percentage of the current system-wide CPU utilization is seen in ePL-KRLS. Concerning the average number of rules, ePL-KRLS and ESM-ePL-KRLS return the lowest values. Figures 5.17 and 5.18 illustrate the prediction values and rules evolution, respectively.

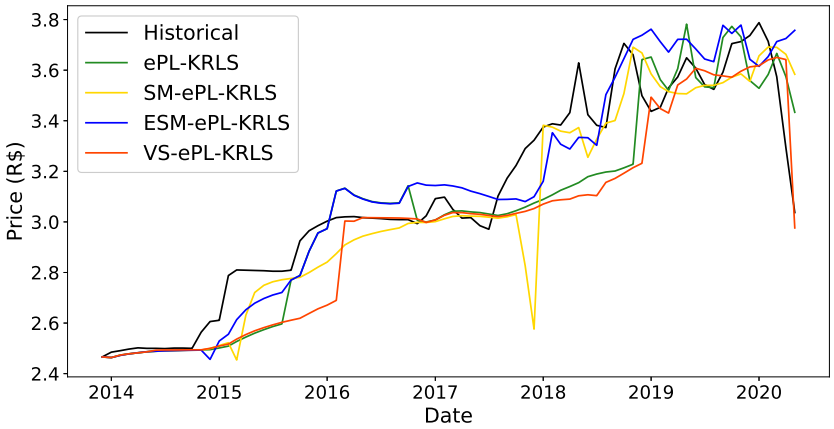

Figure 5.17: Prediction – twelve steps ahead S500 diesel oil price using monthly data

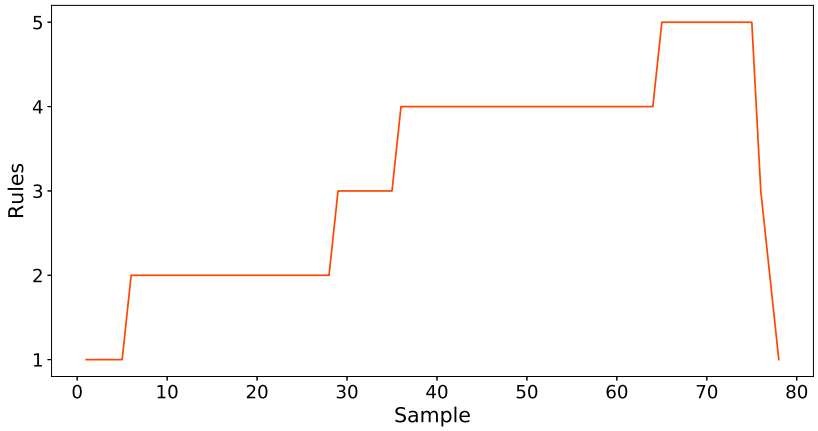

Figure 5.18: VS-ePL-KRLS rules evolution – twelve steps ahead S500 diesel oil price using monthly data

This result is analyzed statistically in Table 5.27. The null hypothesis (which assumes the models have equal accuracy) is rejected when comparing the VS-ePL-KRLS with SM-ePL-KRLS and ESM-ePL-KRLS and is accepted when comparing the VS-ePL-KRLS with ePL-KRLS, with a significance of 5%.

Table 5.27: MGN test – S500 diesel oil for a yearly forecasting horizon and monthly data periodicity

Comparison	MGN	p-value	Observation
VS-ePL-KRLS x ePL-KRLS	-1.375	1.88×10^{-1}	H_0 **is accepted**
VS-ePL-KRLS x SM-ePL-RKLS	-2.340	3.26×10^{-2}	H_0 **is rejected**
VS-ePL-KRLS x ESM-ePL-KRLS	-2.612	1.89×10^{-2}	H_0 **is rejected**

The hyperparameters for S10 diesel oil monthly prices for a yearly forecasting horizon (twelve steps ahead) are represented at Table 5.28. In addition, $\alpha = 0.02$, $\beta = 0.07$, $\tau = 0.07$, and $\gamma = 0.93$.

Table 5.28: Hyperparameters – S10 diesel oil for a yearly forecasting horizon and monthly data periodicity

Algorithm	$\bar{\gamma}$	α_{VS1}	α_{VS2}
SM-ePL-KRLS	0.045	-	-
ESM-ePL-KRLS	0.032	-	-
VS-ePL-KRLS	0.009	0.48	0.29

The experimental results are represented in Table 5.29, which shows the average number of rules, error metrics (RMSE, MAE, and NDEI), and computational complexity (percentage of CPU cycles).

Table 5.29: Results – S10 diesel oil for a yearly forecasting horizon and monthly data periodicity

Algorithm	Rules	RMSE	MAE	NDEI	%CPU
ePL-KRLS	**2**	0.15564	0.11972	0.41732	**15.28 ± 3.10**
SM-ePL-KRLS	10	0.16295	0.12255	0.43692	15.39 ± 2.37
ESM-ePL-KRLS	4	0.19517	0.11999	0.52333	15.82 ± 2.71
VS-ePL-KRLS	4	**0.10946**	**0.08238**	**0.29349**	15.54 ± 2.75

The VS-ePL-KRLS model presents the lowest errors with a reduction of 29.67% of the RMSE, 31.19% of the MAE, and 29.67% of the NDEI about the second-best model (ePL-KRLS). The lowest percentage of the current system-wide CPU utilization is seen in ePL-KRLS. Concerning the average number of rules, ePL-KRLS returns the lowest values. Figures 5.19 and 5.20 illustrate the prediction values and rules evolution, respectively.

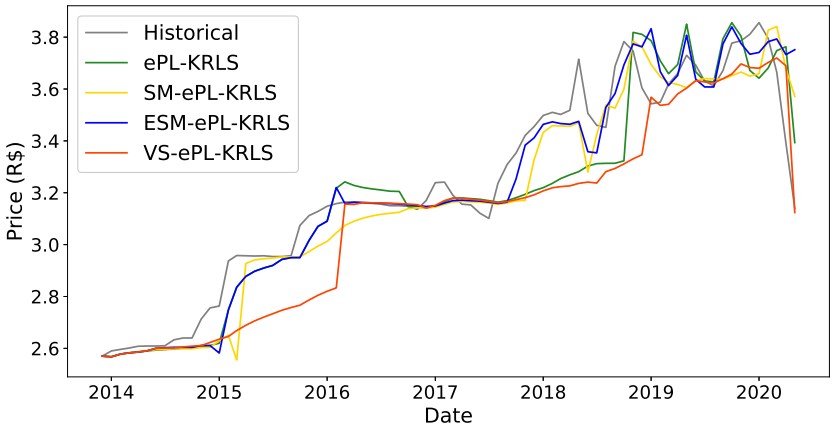

Figure 5.19: Prediction – twelve steps ahead S10 diesel oil price using monthly data

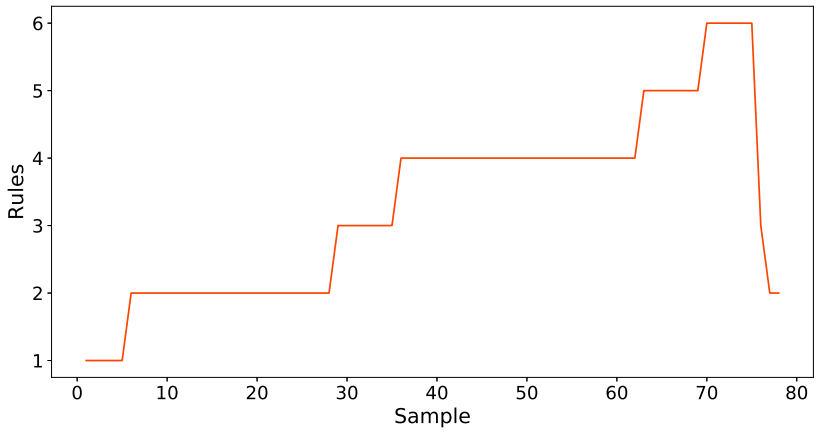

Figure 5.20: VS-ePL-KRLS rules evolution – twelve steps ahead S10 diesel oil price using monthly data

This result is analyzed statistically in Table 5.30. The null hypothesis (which assumes the models have equal accuracy) is rejected when comparing the VS-ePL-KRLS with SM-ePL-KRLS and ESM-ePL-KRLS and is accepted when comparing the VS-ePL-KRLS with ePL-KRLS, with a significance of 5%.

Table 5.30: MGN test – S10 diesel oil for a yearly forecasting horizon and monthly data periodicity

Comparison	MGN	p-value	Observation
VS-ePL-KRLS x ePL-KRLS	-1.974	6.60×10^{-2}	H_0 **is accepted**
VS-ePL-KRLS x SM-ePL-RKLS	-2.694	1.60×10^{-2}	H_0 **is rejected**
VS-ePL-KRLS x ESM-ePL-KRLS	-2.710	1.55×10^{-2}	H_0 **is rejected**

5.2 Box-Jenkins Gas Furnace

The benchmark application consists of the methane flow rate (input variable, x^k) and the carbon dioxide concentration in off gas (output variable, y^k) taken from a laboratory furnace [42]. The equation that best describes the Box-Jenkins dataset is given by:

$$y(k) = f\left(x(k-4), y(k-1)\right) \tag{5-1}$$

The hyperparameters are represented at Table 5.31. In addition, $\alpha = 0.85$, $\beta = 0.07$, $\tau = 0.07$ and $\gamma = 0.93$.

Table 5.31: Hyperparameters – Box-Jenkins Gas Furnace

Algorithm	$\bar{\gamma}$	α_{VS1}	α_{VS2}
SM-ePL-KRLS	0.006	-	-
ESM-ePL-KRLS	0.025	-	-
VS-ePL-KRLS	0.002	0.60	0.30

The experimental results are represented in Table 5.32, which shows the average number of rules, error metrics (RMSE, MAE, and NDEI), and computational complexity (percentage of CPU cycles).

Table 5.32: Results – Box-Jenkins Gas Furnace

Algorithm	Rules	RMSE	MAE	NDEI	%CPU
ePL-KRLS	3	1.08214	0.83738	0.33623	19.53 ± 2.49
SM-ePL-KRLS	**2**	1.11876	0.85494	0.34760	**18.48 ± 3.87**
ESM-ePL-KRLS	8	1.20342	0.90552	0.37391	19.60 ± 2.75
VS-ePL-KRLS	**2**	**0.73350**	**0.52865**	**0.22790**	20.38 ± 3.90

The VS-ePL-KRLS model presents the lowest errors with a reduction of 32.22% of the RMSE, 36.87% of the MAE, and 32.22% of the NDEI about the second-best model (ePL-KRLS). The lowest percentage of the current system-wide CPU utilization is seen in SM-ePL-KRLS. Concerning the average number of rules, VS-ePL-KRLS and SM-ePL-KRLS return the lowest values. Figures 5.21 and 5.22 illustrate the prediction values and rules evolution, respectively.

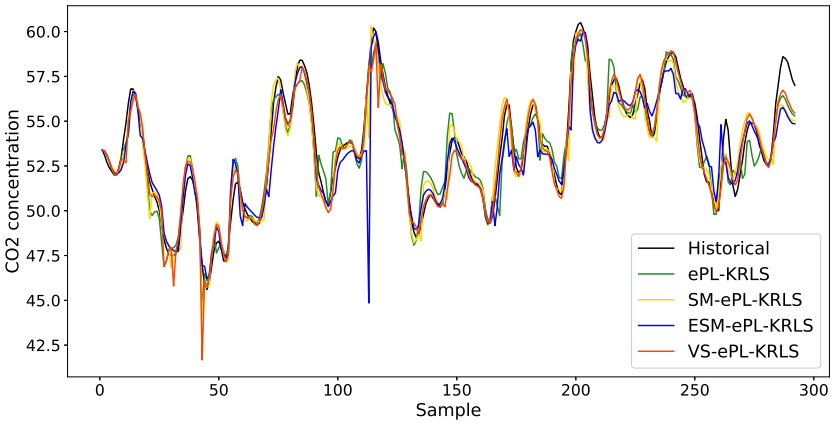

Figure 5.21: Prediction – Box-Jenkins Gas Furnace

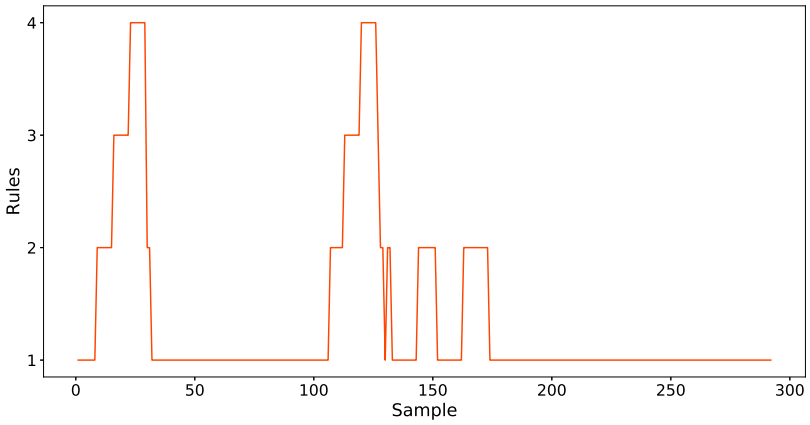

Figure 5.22: VS-ePL-KRLS rules evolution – Box-Jenkins Gas Furnace

This result is analyzed statistically in Table 5.33. The null hypothesis (which assumes the models have equal accuracy) is rejected with a significance of 5 % when comparing the VS-ePL-KRLS with all the other three models.

Table 5.33: MGN test – Box-Jenkins Gas Furnace

Comparison	MGN	p-value	Observation
VS-ePL-KRLS x ePL-KRLS	-4.227	5.61×10^{-5}	H_0 **is rejected**
VS-ePL-KRLS x SM-ePL-RKLS	-6.529	3.70×10^{-9}	H_0 **is rejected**
VS-ePL-KRLS x ESM-ePL-KRLS	-7.028	3.74×10^{-10}	H_0 **is rejected**

6
Conclusions

In this work, a new forecasting model is suggested to deal with the problem of fuel price forecasting: the Variable Step-Size evolving Participatory Learning with Kernel Recursive Least Squares, VS-ePL-KRLS. This model is tested for the prediction of S500 and S10 diesel oil weekly prices in the context of Brazil for biweekly and monthly horizons and it appears as a better choice to integrate a decision support tool to assist the operational, tactical, and strategic logistic planning, for example. In addition, it was also applied to a benchmarks problem which is that of Box-Jenkins Gas Furnace.

The evaluation of this model is done in terms of error, number of final rules, and computational complexity. The proposed model had the smallest error in all time series with gains ranging between 9.01 % and 32.22 % with the second-best model and it was statistically validated in MGN Test through the rejection of the null hypothesis in 28 of the 33 comparisons, indicating that two forecast error variances are not equivalent to a significance level of 5%. The computational cost and the average number of rules were satisfactory in the proposed model.

The proposed model proved to be a decision support tool in planning operational, tactical, and strategic logistics. As an example, the model can be used to optimize delivery, defining better dates and locations for supply, aiming at minimizing the costs of delivery and, consequently, maximizing the profit of the logistic operator.

Suggestions for future work are the application of VS-ePL-KRLS model in other types of time series or the implementation of the Variable Step-Size in other types of Evolving Fuzzy models.

Bibliography

[1] HYNDMAN, R. J.; ATHANASOPOULOS, G.. **Forecasting: principles and practice**. OTexts, 2018.

[2] FLEURY, P. F.; WANKE, P. ; FIGUEIREDO, K. F.. **Logística empresarial: a perspectiva brasileira**. Editora Atlas SA, 2000.

[3] VILARDAGA, V.. **Choque de gestão no transporte rodoviário**. Anuário de gestão de frotas, treinamento e pós-vendas, 4:6–10, 2007.

[4] KASABOV, N.; FILEV, D.. **Evolving intelligent systems: methods, learning, & applications**. In: 2006 INTERNATIONAL SYMPOSIUM ON EVOLVING FUZZY SYSTEMS, p. 8–18. IEEE, 2006.

[5] ŠKRJANC, I.; IGLESIAS, J. A.; SANCHIS, A.; LEITE, D.; LUGHOFER, E. ; GOMIDE, F.. **Evolving fuzzy and neuro-fuzzy approaches in clustering, regression, identification, and classification: a survey**. Information Sciences, 490:344–368, 2019.

[6] LIMA, E.; HELL, M.; BALLINI, R. ; GOMIDE, F.. **Evolving fuzzy modeling using participatory learning**. Evolving intelligent systems: methodology and applications, p. 67–86, 2010.

[7] VIEIRA, R.; GOMIDE, F. ; BALLINI, R.. **Kernel evolving participatory fuzzy modeling for time series forecasting**. In: 2018 IEEE INTERNATIONAL CONFERENCE ON FUZZY SYSTEMS (FUZZ-IEEE), p. 1–9. IEEE, 2018.

[8] HARRIS, R.; CHABRIES, D. ; BISHOP, F.. **A variable step (vs) adaptive filter algorithm**. IEEE Transactions on Acoustics, Speech, and Signal Processing, 34(2):309–316, 1986.

[9] GELFAND, S. B.; WEI, Y. ; KROGMEIER, J. V.. **The stability of variable step-size lms algorithms**. IEEE transactions on signal processing, 47(12):3277–3288, 1999.

[10] DE AGUIAR, E. P.; FERNANDO, M. D. A.; VELLASCO, M. M. ; RIBEIRO, M. V.. **Set-membership type-1 fuzzy logic system applied to fault**

classification in a switch machine. IEEE Transactions on Intelligent Transportation Systems, 18(10):2703–2712, 2017.

[11] LUGHOFER, E.. **Evolving fuzzy systems—fundamentals, reliability, interpretability, useability, applications**. In: HANDBOOK ON COMPUTATIONAL INTELLIGENCE: VOLUME 1: FUZZY LOGIC, SYSTEMS, ARTIFICIAL NEURAL NETWORKS, AND LEARNING SYSTEMS, p. 67–135. World Scientific, 2016.

[12] ANGELOV, P. P.; FILEV, D. P.. **An approach to online identification of takagi-sugeno fuzzy models**. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 34(1):484–498, 2004.

[13] CHIU, S. L.. **Fuzzy model identification based on cluster estimation**. Journal of Intelligent & Fuzzy Systems, 2(3):267–278, 1994.

[14] YOUNG, P. C.. **Recursive estimation and time-series analysis: an introduction**. Springer Science & Business Media, 2012.

[15] ANGELOV, P.; FILEV, D.. **Simpl_ets: A simplified method for learning evolving takagi-sugeno fuzzy models**. In: THE 14TH IEEE INTERNATIONAL CONFERENCE ON FUZZY SYSTEMS, 2005. FUZZ'05., p. 1068–1073. IEEE, 2005.

[16] ANGELOV, P. P.; ZHOU, X.. **Evolving fuzzy-rule-based classifiers from data streams**. IEEE Transactions on Fuzzy Systems, 16(6):1462–1475, 2008.

[17] BIREK, L.; PETROVIC, D. ; BOYLAN, J.. **Water leakage forecasting: the application of a modified fuzzy evolving algorithm**. Applied Soft Computing, 14:305–315, 2014.

[18] ANGELOV, P.; FILEV, D. P. ; KASABOV, N.. **Evolving intelligent systems: methodology and applications**, volumen 12. John Wiley & Sons, 2010.

[19] ANGELOV, P.. **Fuzzily connected multimodel systems evolving autonomously from data streams**. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 41(4):898–910, 2011.

[20] KOMIJANI, M.; LUCAS, C.; ARAABI, B. N. ; KALHOR, A.. **Introducing evolving takagi–sugeno method based on local least squares support vector machine models**. Evolving Systems, 3(2):81–93, 2012.

[21] SHAFIEEZADEH-ABADEH, S.; KALHOR, A.. **Evolving takagi–sugeno model based on online gustafson-kessel algorithm and kernel recursive least square method**. Evolving Systems, 7(1):1–14, 2016.

[22] DE BRABANTER, J.; DE MOOR, B.; SUYKENS, J. A.; VAN GESTEL, T. ; VANDEWALLE, J. P.. **Least squares support vector machines**. World Scientific, 2002.

[23] ENGEL, Y.; MANNOR, S. ; MEIR, R.. **The kernel recursive least-squares algorithm**. IEEE Transactions on Signal Processing, 52(8):2275–2285, 2004.

[24] LEMOS, A.; CAMINHAS, W. ; GOMIDE, F.. **Multivariable gaussian evolving fuzzy modeling system**. IEEE Transactions on Fuzzy Systems, 19(1):91–104, 2010.

[25] YAGER, R. R.. **Participatory learning: A paradigm for more human like learning**. In: 2004 IEEE INTERNATIONAL CONFERENCE ON FUZZY SYSTEMS (IEEE CAT. NO. 04CH37542), volumen 1, p. 79–84. IEEE, 2004.

[26] MACIEL, L.; VIEIRA, R.; PORTO, A.; GOMIDE, F. ; BALLINI, R.. **Evolving participatory learning fuzzy modeling for financial interval time series forecasting**. In: 2017 EVOLVING AND ADAPTIVE INTELLIGENT SYSTEMS (EAIS), p. 1–8. IEEE, 2017.

[27] ALVES, K. S. T. R.; HELL, M.; CYRINO OLIVEIRA, F. L. ; DE AGUIAR, E. P.. **An enhanced set-membership evolving participatory learning with kernel recursive least squares applied to thermal modeling of power transformers**. Electric Power Systems Research, 184:106334, 2020.

[28] TAKAGI, T.; SUGENO, M.. **Fuzzy identification of systems and its applications to modeling and control**. IEEE Transactions on Systems, Man, and Cybernetics, (1):116–132, 1985.

[29] SUGENO, M.; KANG, G.. **Structure identification of fuzzy model**. Fuzzy Sets and Systems, 28(1):15–33, 1988.

[30] AFRAVI, M.; KREINOVICH, V.. **Fuzzy systems are universal approximators for random dependencies: A simplified proof**. In: DECISION MAKING UNDER CONSTRAINTS, p. 1–5. Springer, 2020.

[31] SILVA, L.; GOMIDE, F. ; YAGER, R.. **Participatory learning in fuzzy clustering**. In: THE 14TH IEEE INTERNATIONAL CONFERENCE ON FUZZY SYSTEMS, 2005. FUZZ'05., p. 857–861. IEEE, 2005.

[32] GUSTAFSON, D. E.; KESSEL, W. C.. **Fuzzy clustering with a fuzzy covariance matrix**. In: 1978 IEEE CONFERENCE ON DECISION AND CONTROL INCLUDING THE 17TH SYMPOSIUM ON ADAPTIVE PROCESSES, p. 761–766. IEEE, 1979.

[33] GATH, I.; ISKOZ, A. S. ; VAN CUTSEM, B.. **Data induced metric and fuzzy clustering of non-convex patterns of arbitrary shape**. Pattern Recognition Letters, 18(6):541–553, 1997.

[34] HAYKIN, S. S.; OTHERS. **Neural networks and learning machines/simon haykin.**, 2009.

[35] LIU, W.; PRINCIPE, J. C. ; HAYKIN, S.. **Kernel adaptive filtering: a comprehensive introduction**, volumen 57. John Wiley & Sons, 2011.

[36] SCHOLKOPF, B.; SMOLA, A. J.. **Learning with kernels: support vector machines, regularization, optimization, and beyond**. MIT press, 2001.

[37] RICHARD, C.; BERMUDEZ, J. C. M. ; HONEINE, P.. **Online prediction of time series data with kernels**. IEEE Transactions on Signal Processing, 57(3):1058–1067, 2008.

[38] NGIA, L. S.; SJOBERG, J. ; VIBERG, M.. **Adaptive neural nets filter using a recursive levenberg-marquardt search direction**. In: CONFERENCE RECORD OF THIRTY-SECOND ASILOMAR CONFERENCE ON SIGNALS, SYSTEMS AND COMPUTERS (CAT. NO. 98CH36284), volumen 1, p. 697–701. IEEE, 1998.

[39] AGÊNCIA NACIONAL DO PETRÓLEO, G. N. E. B.. **Série histórica do levantamento de preços e de margens de comercialização de combustíveis**, 2020.

[40] PETROBRAS. **Diesel - composição de preços ao consumidor**, 2020.

[41] GRANGER, C. W. J.; NEWBOLD, P.. **Forecasting economic time series**. Academic Press, 2014.

[42] BOX, G. E.; JENKINS, G. M.. **Time series analysis: Forecasting and control san francisco**. Calif: Holden-Day, 1976.