UNIVERSIDADE FEDERAL DE JUIZ DE FORA
FACULDADE DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM COMPUTACIONAL

Kaike Sa Teles Rocha Alves

A novel rule-based evolving fuzzy system applied to the thermal modeling of power transformers

Juiz de Fora
2021

Kaike Sa Teles Rocha Alves

A novel rule-based evolving fuzzy system applied to the thermal modeling of power transformers

Dissertação apresentada ao Programa de Pós-Graduação em Modelagem Computacional da Universidade Federal de Juiz de Fora, na área de concentração em Modelagem Computacional, como requisito parcial à obtenção do título de Mestre em Modelagem Computacional.

Orientador: Prof. D. Sc. Eduardo Pestana de Aguiar

Juiz de Fora

2021

Ficha catalográfica elaborada através do Modelo Latex do CDC da UFJF
com os dados fornecidos pelo(a) autor(a)

Alves, Kaike Sa Teles Rocha.
A novel rule-based evolving fuzzy system applied to the thermal modeling of power transformers / Kaike Sa Teles Rocha Alves. – 2021.
49 f. : il.

Orientador: Eduardo Pestana de Aguiar
Dissertação (Mestrado) – Universidade Federal de Juiz de Fora, Faculdade de Engenharia. Programa de Pós-Graduação em Modelagem Computacional, 2021.

1. Time series forecasting. 2. evolving Fuzzy Systems. 3. Artificial Intelligence. I. Aguiar, Eduardo Pestana de, orient. II. Título.

Kaike Sa Teles Rocha Alves

A novel rule-based evolving fuzzy system applied to the thermal modeling of power transformers

Dissertação apresentada ao Programa de Pós-Graduação em Modelagem Computacional da Universidade Federal de Juiz de Fora, na área de concentração em Sistemas Computacionais Aplicados, como requisito parcial à obtenção do título de Mestre em Modelagem Computacional.

Aprovada em 28 de setembro de 2021

BANCA EXAMINADORA

Prof. D. Sc. Eduardo Pestana de Aguiar - Orientador
Universidade Federal de Juiz de Fora

Prof. D. Sc. André Paim Lemos
Universidade Federal de Minas Gerais

Prof. PhD Direnc Pekaslan
University of Nottingham

Prof. D. Sc. Leonardo Goliatt da Fonseca
Universidade Federal de Juiz de Fora

To my parents, Gilterney and Rita

ACKLOWLEDGMENT

I thank Jesus Christ for all he has done for me and all his love.

To my parents, Gilterney and Rita, for giving me precious teachings and for supporting me in my career.

To my sisters, Keyla and Anna Klara, with I have learned much from.

To my love, Carmen Goulart.

To all the friends who have made and are still part of my life.

To my advisor Eduardo Aguiar for guide me in this process.

To the professors of the department of Computational Modeling from the Federal University of Juiz de Fora.

To all people who contributed to my progress.

I acknowledge the Federal University of Juiz de Fora for essential support during this work an to the anonymous referees for their valuable comments.

I also thank the financial support of CAPES - Finance Code 001.

ABSTRACT

Big Data advancements motivate researchers to develop and improve intelligent models to deal efficiently and effectively with data. In this scenario, time series forecasting obtains even more attention. The literature demonstrated the better performance of such models in this subject. Forecasting is widely used in strategic planning to support decision-making, providing competitive differential to organizations. In this work, a novel rule-based evolving Fuzzy System is proposed for time series forecasting. This is a robust model able to develop and update its structure in unknown environments, capture dynamics and changes of streams, and produce accurate results even when dealing with complex data. The introduced model implements the distance correlation to improve the rules' quality by reducing their standard deviation. The model is evaluated using two synthetic datasets: the Mackey-Glass time-series and the nonlinear dynamic system identification. And finally, the introduced system is implemented to predict the hot spot temperature using three datasets from a real power transformer. Hot spot monitoring is necessary to maximize the load capacity and the lifespan of power transformers. The proposed method is evaluated in terms of root-mean-square error, non-dimensional index error, mean absolute error, runtime, and the number of final rules. The results are compared with traditional forecasting models and with some related state-of-the-art rule-based evolving Fuzzy Systems. The new evolving Fuzzy System outperformed the compared models for the Mackey-Glass time-series and the power transformers datasets concerning the errors. A statistical test comprised the superior performance of the introduced model. The algorithm also obtained a competitive execution time and number of final rules. The results demonstrate the high level of autonomy and adaptation of the model to predict accurately complex and non-stationary data. Seeing the importance of accurate models to deal with data to support decision-making, the results suggest the model's implementation as a forecasting tool in strategic planning.

Keywords: Time series forecasting. evolving Fuzzy Systems. Artificial Intelligence.

RESUMO

Os avanços em *Big Data* motivaram pesquisadores a desenvolver e aprimorar modelos inteligentes para lidar de forma eficiente e eficaz com os dados. Nesse cenário, a previsão de séries temporais vem ganhando ainda mais atenção. A literatura científica demonstra o melhor desempenho de tais modelos nesse assunto. A previsão de séries temporais é amplamente utilizada no planejamento estratégico para apoiar a tomada de decisões, proporcionando diferencial competitivo às organizações. Neste trabalho, um novo sistema nebuloso evolutivos baseado em regras é proposto para a previsão de séries temporais. Este é um modelo robusto capaz de desenvolver e atualizar sua estrutura em ambientes desconhecidos, capturar dinâmicas e mudanças de fluxo em dados e produzir resultados precisos mesmo quando se trata de dados complexos. O modelo introduzido implementa a correlação para melhorar a qualidade dos *clusters*, reduzindo seu desvio padrão. O modelo é avaliado usando dois conjuntos de dados sintéticos: a série temporal Mackey-Glass e a identificação do sistema dinâmico não linear. E, finalmente, o sistema introduzido é implementado para prever a temperatura do ponto quente, usando três conjuntos de dados de um transformador de potência real. O monitoramento de pontos quentes é necessário para maximizar a capacidade de carga e a vida útil dos transformadores. O método proposto é avaliado em termos de erro quadrático médio, erro de índice adimensional, erro absoluto médio, tempo de execução e número de regras finais. Os resultados são comparados com modelos de previsão tradicionais e com alguns sistemas nebuloso evolutivo baseados em regras. O novo sistema nebuloso evolutivos superou os modelos comparados para a série temporal Mackey-Glass e os conjuntos de dados de transformadores de potência, considerando as métricas de erro. Um teste estatístico comprovou o desempenho superior do modelo introduzido. O algoritmo também obteve um tempo de execução e número de regras finais competitivo. Os resultados demonstram o alto nível de autonomia e adaptação do modelo para prever dados complexos e não estacionários com precisão. Vendo a importância de modelos precisos para lidar com dados no apoio à tomada de decisão, os resultados sugerem a implementação do modelo como ferramenta de previsão favorecendo planejamento estratégico.

Palavras-chave: Previsão de séries temporais. Sistemas nebulosos evolutivos. Inteligência Artificial.

CONTENTS

1 INTRODUCTION

The world is experiencing an era in which technological advances have substantially changed how data is collected, driving expressive Big Data advancements. Data are usually produced at high speed and presents complicated structures [1]. Such challenges increase the requirements for robust tools to deal with data and extract useful information of them [2]. In this scenario, artificial intelligence (AI) is gaining much visibility and is attracting even more researchers' interests [3, 4]. As data usually assume a stream form, time series forecasting using accurate models can be implemented to support management, planning, and decision-making, improving the competitive performance of organizations [5, 6, 7]. The capability to analyze data streams is based on the premise that a data's portion carries information that explains the nature of the underlying system [8]. In this sense, evolving Fuzzy Systems (eFSs) emerge as a powerful AI technique in time series forecasting. They can extract knowledge from a data stream and simultaneously adapt its functionality and structure [9, 10]. Furthermore, eFSs usually have self-learning for some of their parameters [1, 11]. The fuzzy concept is an essential pillar of the eFSs. Fuzzy operators are implemented to define the interdependency of fuzzy input and form fuzzy rules from them [12]. These rules consist of clusters formed from the input space and are used to compute the model's output. Three noticeable characteristics of the eFS models can be highlighted: i) their self-ability to develop and update its structure in unknown environments; ii) their ability to capture dynamics and changes of streams; and iii) their capacity to produce accurate results even when dealing with nonlinear data [13, 14].

This work introduces a novel rule-based eFS model, so-called evolving Participatory Learning with Kernel Recursive Least Square and Distance Correlation (ePL-KRLS-DISCO), for time series forecasting. This model has its origins in the evolving Takagi-Sugeno (eTS), proposed by Angelov and Filev [15, 16], a model with a small number of parameters. The eTS algorithm computes the consequent parameters using the recursive least square (RLS) method. RLS usually requires less data to adjust the model parameters and is computationally efficient. However, RLS is a linear regression that performs inaccurate outputs when dealing with nonlinear data [17, 18]. Angelov and Filev also presented a simplified version of eTS, called Simpl_eTS [19]. Simpl_eTS uses the scatter concept instead of potential and the Cauchy type function instead of the Gaussian exponential membership functions to reduce the computational cost and improve the model's errors concerning the eTS. Angelov and Zhou [20] proposed the evolving extended Takagi-Sugeno (exTS). This model has a mechanism to update the spread of the membership functions recursively. Angelov and Filev introduced an enhanced version of eTS, the eTS+ [21], a model that uses the utility measure to remove underused rules. Lima *et al.* [22] introduced the evolving Participatory Learning (ePL), a model that combines the eTS structure with the participatory learning (PL) concept to control the creation or updating of the rules. PL is a recursive unsupervised clustering algorithm that implements convex combinations between input data and the closest cluster center. This clustering mechanism avoids the curse of dimensionality. The ePL model also implements the RLS method to calculate consequent parameters in the same way as eTS and, consequently, presents the same limitation when dealing with nonlinear data. Maciel *et al.* [23] proposed the enhanced evolving Participatory Learning (ePL+), which adds the concepts of clusters quality measurement and adaptation of the zone of influence as same as eTS+.

Lemos *et al.* [9] implemented the evolving Multivariable Gaussian (eMG), a model that inherits the ePL structure. The main novelty of eMG is that it forms the fuzzy sets using multivariable Gaussian membership functions to prevent waste of knowledge concerning the inputs. However, this model needs a great number of rules to make accurate estimations [24]. Shafieezadeh-Abadeh and Kalhor [25, 26] addressed the evolving Takagi-Sugeno with Kernel Recursive Least Square (eTS-KRLS), and Maciel *et al.* [27] proposed the evolving Participatory Learning with Kernel Recursive Least Square (ePL-KRLS). Both models use the Kernel Recursive Least Square (KRLS) to update the consequent parameters. KRLS consists in to perform an inner product between two samples in the high-dimensional Hilbert space. Moreover, the kernel function maps all the features in the high-dimensional space so that the nonlinear relationship of them is linearly separable in the high-dimensional space [28]. Then, these models easier cope with data manipulating them in the new space [29]. The benefit of using this approach is that Kernel-based methods are more sensitive to variations in the input data and can approximate nonlinear systems accurately and efficiently with a moderate computational cost [10]. Vieira *et al.* [30] implemented the enhanced ePL-KRLS (ePL-KRLS+), which implements the same mechanisms of eTS+ and ePL+ concerning the removal of rules and the adaptation of the zone of influence.

The ePL-KRLS-DISCO model intends to overcome the following limitations of ePL-KRLS: i) ePL-KRLS computes the global output using local outputs of all rules. Consequently, the local output of unlike rules disturbs the global output; ii) ePL-KRLS only considers the Euclidean distance to create/update the rules. So, if the input is very close to an existing cluster, it probably will be included in that rule, even if the spatial distribution is different [31]; and iii) the updating of the threshold that defines the inclusion of new inputs into the dictionaries is slow. So, the model may discard some relevant characteristics of the observations. Therefore, the proposed model introduces the following improvements to overcome such shortcomings: i) ePL-KRLS-DISCO calculates the global output using only the most compatible rule; ii) ePL-KRLS-DISCO adds the distance correlation (DISCO) to form the clusters; and iii) ePL-KRLS-DISCO implements a new mechanism to update the kernel size as an error function as same as [1, 32, 33]. These improvements make ePL-KRLS-DISCO a robust forecasting model. As the DISCO is a measure of dependence between vectors [34, 35], this mechanism avoids grouping two different structure vectors, even if these data are close to one another [36, 37]. Then, the DISCO forms rules with a reduced standard deviation, improving the quality of the clusters. So, the rules are composed of inputs with very like characteristics. Every cluster becomes specialized in similar inputs, holding in dictionaries relevant information about the past entries. When a new observation enters the system, KRLS computes the output using the more suitable rule. As the DISCO improves the rules' quality, and the model's performance is directly related to the quality of the clusters and its capacity of learning, the introduced model has a substantially superior performance compared to the previous models and performs precise simulations even with complex data.

The introduced model is evaluated using two benchmark for time series. And finally, ePL-KRLS-DISCO is applied in the thermal modeling of power transformers. The aim is to predict the hot spot temperature using data from a real power transformer. The hot spot, which is the highest temperature near the top of the windings, directly impacts the insulation's aging [38]. Power transformers are one of the most expensive equipment in power distribution, and the insulation defect may determine its end life. For this reason, hot spot monitoring is necessary to define the load capacity that maximizes the cost-benefit [39]. The most precise method to capture the hot spot temperature adopts sensors set near the high-voltage (HV)/low-voltage (LV) windings. However, this method is costly and hard to maintain in practice restricting to test transformers [40]. A classical model based on the IEEE Standard C57.91-2011 consists of differential equations [41]. Unfortunately, this model assumes a series of simplifications, making the transformers' capacity underutilized [11, 42]. Thus, intelligent models arise as a modeling tool of power transformers, aiming to optimize the transformers' life and load capacity [43]. Daponte *et al.* and Galdi *et al.* [42, 44] proposed neural networks to estimate the hot spot temperature. Ipolito [45] implemented fuzzy sets, Hell *et al.* [46, 47] used neuro-fuzzy systems, and Alves *et al.*, Souza *et al.*, and Rocha *et al.* [11, 24, 43] applied eFS models for this task. The model's performance is shown in terms of errors, runtime, and the number of final rules. And finally, the results are compared with traditional forecasting models and some related state-of-the-art evolving fuzzy modeling approaches.

The main contributions of this work are summarized as follows:

- This work introduces a novel rule-based eFS model, so-called evolving Participatory Learning with Kernel Recursive Least Square and Distance Correlation (ePL-KRLS-DISCO), for time series forecasting.
- The ePL-KRLS-DISCO introduces the distance correlation (DISCO) to compute the compatibility measure, the calculation of the global output as the local output of the most compatible rule, and a new mechanism to update the kernel size.
- The model's performance is evaluated using error metrics, runtime, and the number

of final rules. Additionally, the ePL-KRLS-DISCO performance is compared with other eFS models using synthetic data and data collected from an experimental power transformer.

The remainder of this work is organized as follows: Section 2 presents a valuable literature review on evolving Fuzzy Systems. Section 3 detailed explains the ePL-KRLS-DISCO. Section 4 shows and discusses the model's results. And finally, Section 5 concludes this work and suggests future researches.

2 LITERATURE REVIEW ON EVOLVING FUZZY SYSTEMS

The eFS models have three main classes: i) neural; ii) rule-based; and iii) tree. Neural networks started the discussion of the evolving systems paradigm, see Fritzke [48], Williamson [49], and Kwok and Yeung [50]. Juang and Lin proposed the neural fuzzy inference network (SONFIN) [51], a model based on the Takagi-Sugeno-Kang (TSK). SONFIN produces small networks and is proper to fast online learning. This approach has three main sensitive parameters that perform an essential role: i) the initial weights of the nodes; ii) the width of each fuzzy set; and iii) the number of fuzzy rules in the online phase [52]. Juang and Lin also introduced the recurrent self-organizing neural fuzzy inference network (RSONFIN) [53, 54]. RSONFIN employs internal memories called context elements to deal with temporal problems. This model has a flexible structure where the number of rules is not predefined, reducing the design effort. Kasabov [55] proposed a model with a fast adaptive local learning that is robust against catastrophic forgetting called evolving fuzzy neural network (EFuNN). EFuNN is a five-layer architecture with adaptive and incremental evolving capability, making the learning more effective [56]. Kasabov and Song [57] introduced a neural-fuzzy hybrid termed dynamic evolving neural fuzzy inference system (DENFIS), which produces the output comparing the position of the input vector with the created rules. DENFIS has the advantage of being a fast algorithm in the training and test phase, learn new features at any stage in an incremental way, define parameters to scale the attributes of the input space and provide explainable results [58].

Rubio [59] explored the self-organizing fuzzy modified least-square (SOFMLS) network. SOFMLS has a reduced number of parameters due to the implementation of the unidimensional membership functions and is a stable model able to reorganize its structure according to the changes in the data [60]. Soleimani-B *et al.* [61, 62] developed the evolving neuro-fuzzy model (ENFM), which uses a recursive extension of Gath–Geva to construct elliptical clusters. ENFM produces accurate results with a few neurons due to the implementation of general structures for covariance matrices. Leite *et al.* [63] depicted the evolving granular neural network (eGNN). The eGNN model can process online data streams searching for results that combine precision and interpretability. Pratama *et al.* [64] exposed a model termed parsimonious network based on fuzzy inference system (PANFIS). PANFIS forms ellipsoidal clusters from the input space to construct the antecedent parts, improving its ability to cope with nonlinear data. The consequent parameters use the extended recursive least square (ERLS). Unfortunately, it presents high complexity concerning the parameters selection, and programming language [65]. Pratama *et al.* also proposed the Generic Evolving Neuro-Fuzzy Inference System (GENEFIS) [66], the recurrent classifier (rClass) [67], and the GENERIC-Classifier (gClass) [68]. GENEFIS is an algorithm able to make accurate predictions with a few rules. And rClass and gClass are two meta-cognitive classifiers. Silva *et al.* [69] developed a variation of the EFuNN to detect high impedance faults in distribution systems. Artificial neural network approaches perform parallel processing and identify and generalize patterns in datasets. Unfortunately, neural networks usually demand high-quality training data and time-consuming offline learning. Generally speaking, this class of model gives black box results, nontransparent and hard to interpret [63]. The black-box property limits its usage from applying in high-stake areas [70].

Angelov and Buswell [71] introduced the evolving fuzzy rule-based model. As compared to pure neural networks, a notorious advantage of rule-based systems is their evident and explainable results [21, 72]. Rong *et al.* [73] discussed a Sequential Adaptive Fuzzy Inference System that uses an extended Kalman filter to update the fuzzy rules called SAFIS. SAFIS has two main shortcomings: i) the complex calculation of the rule's influence, defined through statistical means; and ii) difficulty to cope with high-dimensional input spaces. Rong *et al.* [74] proposed the extended version of SAFIS, termed ESAFIS, to overcome those limitations. Rubio and Bouchachia [75] introduced the modified sequential adaptive fuzzy inference system (MSAFIS). The difference is that MSAFIS uses the stable gradient descent algorithm (SGD) for the parameters' updating instead of the Kalman filter used in SAFIS. SGD improves learning performance. Lughofer [76] depicted the flexible fuzzy inference systems (FLEXFIS), which uses a linear polynomial to exploits the Takagi-Sugeno model. FLEXFIS demands low computational cost and is adequate for on-line operations [77]. Lughofer *et al.* presented the FLEXFIS+, an enhanced version of the FLEXFIS with mechanisms to detect and eliminate redundancies [78]. Lughofer and Kindermann [79] presented the sparse fuzzy inference systems (SparseFIS), a model that optimizes the consequent parameters and sparses out unimportant rules. SparseFis uses a numerical optimization mechanism to define a compact ruleset [80]. Leite *et al.* [8] exploited the fuzzy set based evolving modeling (FBeM), a framework that employs fuzzy granular models to provide a more intelligible exhibition of the data. FBeM provides a transparent and interpretable description of the input data due to the fuzzy hyperboles that form the antecedent part. Angelov and Yager [81] contributed with the AnYa, a simplified model that constructs the antecedent part implicitly. AnYa implements a non-parametric vectorized antecedent part [82].

Dovzăn *et al.* [83] presented a novel evolving fuzzy model (eFuMo) for monitoring and fault detection of waste-water treatment plants (WWTPs). This model is effective for the online identification of fuzzy rule-based models [84]. Lughofer *et al.* [85] constructed a model derived from the FLEXFIS named Generalized Smart Evolving Fuzzy Systems (Gen-Smart-EFS). Gen-Smart-EFS can produce accurate outputs with a compact structure because of its ability to model local correlations by intervals between variables. Two main limitations of Gen-Smart-EFS can be highlighted: i) it may not capture a slight shift in the data, making the ellipsoid of a rule increase much; and ii) incorrectly set the parameter that defines the rules creation usually makes the model not create enough rules to represent the input space. Lughofer *et al.* [14] suggested an incremental rule splitting strategy to overcome the Gen-Smart-EFS limitations. The model can immediately forget samples after their processing. Škrjanc and Dovzăn [86] implemented the evolving Gustafson-Kessel possibilistic c-Means clustering (eGKPCM), a model indicated to deal with highly noisy data. Maciel *et al.* [87] built the evolving possibilistic fuzzy modeling approach (ePFM), which uses the evolving Gustafson-Kessel-Like algorithm (eGKL) to adapt its structure. The ePFM modeling approach aggregates the advantages of the Gustafson-Kessel (GK) clustering algorithm of identifying clusters with different shapes and orientations while recursively processing data. Such a model also uses utility measure to evaluate the quality of the current cluster structure. This model is robust to modeling volatility dynamics and nonlinear volatility forecasting with jumps.

Ge and Zeng [32] detailed the self-evolving fuzzy system (SEFS), a model with self-learning parameters. SEFS auto-adjust the rule's adding speed to reduce underfitting or over-fitting, improving the model's accuracy. Leite *et al.* [88] displayed the evolving optimal granular system (eOGS). An advantage of using eOGS is the possibility of trading off multiple objectives. Alves *et al.* [11] implemented the Set-Membership (SM) and the Enhanced Set-Membership (ESM) to update the rate of change of the arousal index in the ePL-KRLS. The SM and ESM implementation control the rule creation speed to enhance the performance of the models. Recently, Ge and Zeng [1] presented the evolving fuzzy system self-learning/adaptive thresholds (EFS-SLAT), a model with self-learning thresholds. Lemos *et al.* [89] presented a fuzzy evolving linear regression trees. The model creates the tree from the data stream using a statistical model selection test based on a hypothesis test. The model evolves the tree recursively, replacing the leaves with subtrees and consequently improving the quality of the model. The parameters of each leaf are adapted using the conventional least-squares algorithm. The algorithm produces efficient models robust against over fitting. Hapfelmeier *et al.* [90] discussed the Guarded Incremental Pruning (GuIP), which prunes the tree to improve the model's performance in the sense of overfitting and overly large structure avoidance.

3 THE PROPOSED MODEL

The current section introduces and describes the ePL-KRLS-DISCO. First, it is presented an overview of the model's structure. Following, the mechanism implemented to define the rules is given. After, the correlation coefficient is briefly discussed. Next, the method to prune underused rules is exploited. Then, it is detailed the method to define the consequent parameters. Next, it is shown the method to update the kernel size. And finally, the complete algorithm is presented.

3.1 AN OVERVIEW OF THE PL-KRLS-DISCO

The ePL-KRLS-DISCO model is a fuzzy rule-based system. This model uses Takagi-Sugeno (TS) fuzzy rules [91, 92], as following expressed:

$$\mathcal{R}_i : \quad \textbf{IF} \quad \underbrace{x \quad \text{is} \quad \mathcal{A}_i}_{\text{Antecedent}} \quad \textbf{THEN} \quad \underbrace{y_i = f_i(x, \theta_i)}_{\text{Consequent}} \tag{3.1}$$

where $\mathcal{R}_i$ is the i-th fuzzy rule, $i = 1, 2, \ldots, R$, R is the number of fuzzy rules, $x = [x_1, \ldots, x_m]^T \in \mathbb{R}^m$ is the input, m is the number of attributes in the input vector, $\mathcal{A}_i$ is the fuzzy set of the i-th fuzzy rule, and y_i is the output of the i-th rule calculated as a function of the input and the consequent parameters.

TS fuzzy rules have two main parts, the antecedent part and the consequent one. The antecedent consists of rules, which are clusters formed from the input space. Each rule is composed of similar inputs, i.e., ePL-KRLS-DISCO clusters the input space according to the degree of similarity of the input vectors to form the rules. The proposed model implements the participatory learning (PL) concept to define the rules. PL uses the inputs to construct the model's structure and functionality as part of the learning process when a new input is available [93]. As a result, learning from new input data depends on what the system has learned and will impact the learning from future inputs [22]. Furthermore, ePL-KRLS-DISCO uses the utility measure to avoid underused rules, keep a compact structure, and maintain a low computational cost [30, 81].

The consequent part of TS fuzzy rules consists of parameters associated with each rule. The set of parameters associated with a rule is called consequent parameters. The model computes the output as a function of the input and the consequent parameters. The proposed model uses the KRLS method to define the consequent parameters. After the model adds a new input into a rule, it updates the parameters of this rule. When a new rule is created, the KRLS initializes the consequent parameters of this rule.

3.2 IDENTIFICATION OF THE ANTECEDENT PART

Each rule is characterized by a center, which is an estimation of the rule's mean. When a new input enters the system, the model compares the new vector with the center of all created rules. If this input is similar enough to the data of the most like cluster, the model includes the new input into that rule and updates the cluster center. Otherwise, the model creates a new rule [9, 94]. The model initializes the center of a new rule with the input vector, i.e., $v_i^k = x^k$, where v_i^k, is the center of the i-th rule at the k-th iteration, $x^k = [x_1, \ldots, x_m]^T$ is the input vector of attributes with m elements. When a rule receives a new input vector, ePL-KRLS-DISCO updates the cluster center recursively according to Equation (3.2).

$$v_i^k = v_i^{k-1} + \alpha (c_i^k)^{(1-a_i^k)} (x^k - v_i^{k-1}) \tag{3.2}$$

where $\alpha \in [0, 1]$ is the learning rate.

The proposed model defines the rules using the arousal index and the compatibility measure. When an input enters the system, the model calculates the compatibility measure of all rules. The $c_i^k \in [0, 1]$ is the compatibility measure of the i-th rule at the k-th iteration, calculated as a function of the k-th input vector (x^k) and the cluster center of the i-th rule at the k-th iteration (v_i^k) according to Equation (3.3). It indicates the degree of similarity between the new input and the cluster. A compatibility measure zero shows no similarity between the rule and the input, and a value of the compatibility measure equal to one represents the maximum similarity.

$$c_i^k = \left(1 - \frac{\left\| x^k - v_i^k \right\|}{m}\right) \left(\frac{\rho_{x^k, v_i^k} + 1}{2}\right) \tag{3.3}$$

where $\rho_{x^k, v_i^k} \in [-1, 1]$ is the correlation between x^k and v_i^k. The correlation coefficient is presented in the next subsection.

On the other hand, $a_i^k \in [0, 1]$ is the arousal index of the i-th rule at the k-th iteration, calculated as a function of a_i^{k-1} and c_i^k, according to Equation (3.4). The arousal index can be seen as a complement to the compatibility measure, indicating the needing to create a new rule and reducing the effect of outliers. The algorithm creates a new rule if the smallest arousal index is greater than a threshold, i.e., $a_i^k > \tau$, where $i = \arg\min_i \{a_i^k\}$ and $\tau = \beta$ as same as [11] and the model has not excluded any rule. Otherwise, the model assigns the input vector in the rule with the highest compatibility measure.

$$a_i^k = a_i^{k-1} + \beta(1 - c_i^k - a_i^{k-1}) \tag{3.4}$$

where $\beta \in [0, 1]$ controls the growth rate of a_i^k.

3.3 THE CORRELATION COEFFICIENT

The ePL-KRLS model forms the rules based on the Euclidean distance but disregards the spatial distribution of the vectors' elements. This work introduces the correlation to construct the rules to overcome that limitation. Thus, the mechanism to define the clusters uses the DISCO, improving the performance of the clusters. The correlation is a value in the range $[-1, 1]$, which detects an interdependency between distinct observations [95, 96]. The stronger the relationship between the variables, the closer to 1 the correlation is. The closer the correlation is to -1, the stronger is the inverse proportion. Furthermore, a correlation 0 indicates no association between the two vectors [31, 97]. Given the two vectors $A = [A_1, A_2, \ldots, A_m]$ and $B = [B_1, B_2, \ldots, B_m]$, the correlation coefficient is computed as follows:

$$\rho(A, B) = \frac{Cov(A, B)}{\sqrt{Var(A)}\sqrt{Var(B)}} \tag{3.5}$$

where $Cov(A, B)$ is the covariance between features A and B, calculated as Equation (3.6), and $Var(A)$ and $Var(B)$ are the variances of A and B respectively, as Equation (3.7).

$$Cov(A, B) = \frac{\sum_{l=1}^{m}\left(\left(A_l - \bar{A}\right)\left(B_l - \bar{B}\right)\right)}{m-1} \tag{3.6}$$

where m is the number of observations in A and B, and $\bar{A}$ and $\bar{B}$ are the mean of A and B respectively.

$$Var(A) = \frac{\sum_{l=1}^{m}\left(A_l - \bar{A}\right)^2}{m-1} \tag{3.7}$$

It is observed that the compatibility measure will be sufficiently great if the Euclidean distance is small and the correlation is high. So, it is noticeable that higher correlations between two observations come from higher covariance and reduced variances. Therefore, the correlation produces rules with reduced standard deviations. As the accuracy of the output is directly related to the mean and the standard deviation of the clusters, DISCO improves the model's performance.

3.4 PRUNING THE RULES

Pruning is implemented to avoid overfitting noisy data. The main idea is to eliminate complex rules with low coverage that contain irrelevant literals that have only been created to enclose noisy examples [98]. Unnecessarily large rule structure wastes time, as it generally slows down instance processing. Complex structures also require more processing memory, wasting space [59, 90]. The ePL-KRLS-DISCO algorithm removes underused rules using the utility measure rules at each iteration as presented in Equation

(3.8). When a utility of a rule gets lower than a threshold ($U_i^k < \epsilon$), the model eliminates the rule.

$$U_i^k = \frac{\sum_{l=1}^{k} \lambda_i^l}{k - I_i} \tag{3.8}$$

where λ_i^l is the normalized activation level of the i-th rule at the l-th iteration calculated as Equation (3.9), k-th is the current iteration, and I_i is the iteration when the i-th rule was created.

$$\lambda_i^k = \frac{\tau_i}{\sum_{rule=1}^{R^k} \tau_{rule}} \tag{3.9}$$

where τ_i is the activation level of the i-th rule calculated as Equation (3.10), and R^k is the number of rules at the current iteration.

$$\tau_i = \mu_{i1} \times \mu_{i2} \times \ldots \mu_{ij} \tag{3.10}$$

where μ_{ij} is calculated according to Equation (3.11).

$$\mu_{ij} = e^{-\frac{\left\|x_j^k - v_{ij}^k\right\|^2}{2\sigma^2}} \tag{3.11}$$

where v_{ij}^k is the j-th element of the i-th rule center, $j = 1, 2, \ldots, m$, and σ defines the spread of the antecedent part.

3.5 IDENTIFICATION OF THE CONSEQUENT PART

The model estimates the consequent parameters using the Kernel Recursive Least Square (KRLS) [29]. KRLS is a mechanism that uses a collection of past inputs to compute the consequent parameters. Each collection forms the local dictionary, i.e., $\mathcal{D}_i^k = [d_{i1}, \ldots, d_{in_i}]$ is the local dictionary of the i-th rule at the k-th iteration, n_i is the number of input vectors stored in the dictionary, and $d_{ij} = [d_1, \ldots, d_m]^T$ is the j-th input vector of the dictionary. If the dictionaries included all inputs, the computational cost would increase much and would unviable the simulations. Hence, the model uses a sparcification procedure to store in the local dictionaries only relevant input vectors. The algorithm defines the inputs' importance based on the distance from it and the nearest element in the local dictionary, as shown in Equation (3.12). This sparcification procedure is called the novelty criterion (NC), and its implementation is essential to reduce the computational cost and improve the model's accuracy [99, 100].

$$dis_x = \min_{\forall d_{ij} \in \mathcal{D}_i^k} \left\| x^k - d_{ij}^k \right\| \tag{3.12}$$

If the distance is greater than a threshold, the model adds the input vector into the local dictionary, i.e., if $dis_x \geq 0.1\nu_i^k$, where $j = \arg\min_j \left\| x^k - d_{ij}^k \right\|$, then $\mathcal{D}_i^k = [\mathcal{D}_i^{k-1} \cup x^k]$, ν_i^k is the kernel size of the dictionary. In this work, ν_i^k is initialized with σ and updated as shown in the next subsection. The smaller the value of ν_i^k, the more inputs will be stored in the dictionaries.

When the model adds a vector to the dictionary, the consequent parameters are updated as Equation (3.13).

$$\theta_i^k = \begin{bmatrix} \theta_i^{k-1} - z_i^k [r_i^k]^{-1} \hat{e}^k \\ [r_i^k]^{-1} \hat{e}^k \end{bmatrix} \tag{3.13}$$

where $\theta_i^{k-1} = [\theta_{i1}^{k-1}, \ldots, \theta_{in_1}^{k-1}]^T$, $z^k = Q_i^{k-1} g^k$, Q_i^k is updated according to the Equation (3.14), $g^k = [\kappa\langle d_{i1}^k, x^k \rangle, \ldots, \kappa\langle d_{in_i}^k, x^k \rangle]^T$, $\kappa\langle . \, , \, . \rangle$ is the Gaussian-Kernel function calculated as Equation (3.15) [101], $r^k = \lambda + \kappa\langle x^k, x^k \rangle - (z^k)^T g^k$, and $\hat{e}^k = y^k - g^k \theta_i^{k-1}$ is the error estimator.

$$Q_i^k = (r^k)^{-1} \begin{bmatrix} Q_i^{k-1} r^k + z^k (z^k)^T & -z^k \\ -(z^k)^T & 1 \end{bmatrix} \tag{3.14}$$

$$\kappa\langle x^i, x^j \rangle = exp\left(-\frac{\left\| x^i - x^j \right\|^2}{2\sigma^2} \right) \tag{3.15}$$

where σ is the kernel width and controls the linearity of the model. The greater σ, the more linear the function will be [102]. Simulations of Fan *et al.* [103] suggest initial values of σ between 0.2 and 0.5.

Furthermore, the matrix P is updated as follows:

$$P_i^k = \begin{bmatrix} P_i^{k-1} & 0 \\ 0^T & 1 \end{bmatrix} \tag{3.16}$$

where P_i^k is initialized with one.

Otherwise, if the input isn't included into the dictionary, the consequent parameters are updated as Equation (3.17), and the matrix P as Equation (3.20).

$$\theta_i^k = \theta_i^{k-1} + Q_i^k q_i^k \hat{e}^k \tag{3.17}$$

where Q_i^k is obtained from the Equation (3.18), q_i^k from Equation (3.19).

$$Q_i^k = Q_i^{k-1} \tag{3.18}$$

$$q_i^k = \frac{P_i^{k-1} z^k}{1 + (z^k)^T P_i^{k-1} (z^k)} \tag{3.19}$$

$$P_i^k = P_i^{k-1} - \frac{P_i^{k-1} z^k (z^k)^T P_i^{k-1}}{1 + (z^k)^T P_i^{k-1} (z^k)} \tag{3.20}$$

The consequent parameter of a new rule is initialized as follows:

$$\theta_i^k = [\lambda + \kappa\langle x^k, x^k\rangle]^{-1} y^k \tag{3.21}$$

where $\lambda \in [0, 1]$ is a parameter of regularization and y^k is the desired output.

And finally, ePL-KRLS-DISCO computes the output using the most compatible rule, i.e., $\hat{y} = \hat{y}_i | i = \arg\max_i \{c_i^k\}$. The calculation of the output is as follows:

$$\hat{y}_i = \sum_{j=1}^{n_i} \theta_{ij}^k \times \kappa\langle x^k, d_{ij}^k\rangle \tag{3.22}$$

3.6 UPDATING THE KERNEL SIZE

The model updates the kernel size of the rule according to papers [1, 32, 33]. When the model updates an existing rule, the kernel size is updated as follows:

$$\nu_i^k = \sqrt{\left(\nu_i^{k-1}\right)^2 + \frac{\left\|x^k - \upsilon_i^k\right\|^2 - \left(\nu_i^{k-1}\right)^2}{N_i^k} + \frac{\left(N_i^k - 1\right)\left\|\upsilon_i^k - \upsilon_i^{k-1}\right\|^2}{N_i^k}} \tag{3.23}$$

where N_i^k is the number of inputs in the i-th rule at the k-th iteration.

Otherwise, if a new rule is created, the model initializes the kernel size of the rule as follows:

$$\nu_{R+1}^k = \frac{\left\|x^k - \upsilon_i^k\right\|}{\sqrt{-2\log\left(\eta_{max}\right)}} \tag{3.24}$$

where η_{max} is the maximum value of η_i, for $i = 1, 2, \ldots, k$, k is the current iteration, and η is calculated recursively as Equation (3.25).

$$\eta^k = e^{-0.5}\left(\frac{2}{1 + e^{-\tilde{e}^k}} - 1\right) \tag{3.25}$$

where $\tilde{e}^k = 0.8\tilde{e}^{k-1} + \left\|y^k - \hat{y}^k\right\|$ and $\tilde{e}^1 = 0$.

As noted above, the model updates the kernel size based on the error. It improves the model's performance, seeing that higher errors will make the model store more input data into the dictionary.

3.7 THE EPL-KRLS-DISCO ALGORITHM

The Algorithm 1 presents ePL-KRLS-DISCO.

Algorithm 1: ePL-KRLS-DISCO

Input: $x, y, \alpha, \beta, \lambda, \tau, \omega, \sigma, \epsilon$

Output: $\hat{y}$

1 **Initialization:**
$v_1^1 = x^1, \mathcal{D}_1^1 = x^1, \nu_1^1 = \sigma, P_1^1 = 1, \theta_1^1 = [\lambda + \kappa\langle x^1, x^1\rangle]^{-1} y^1, a_1^1 = 0,$
$Excluded\_Rule = false$

2 **for** $k = 2, 3, \ldots, n$ **do**

3   **for** $i = 1, 2, \ldots, R$ **do**

4   Compute the compatibility measure: Equation (3.3)

5   Compute the arousal index: $a_i^k = a_i^{k-1} + \beta(1 - c_i^k - a_i^{k-1})$

6   **if** $\left(a_i^k\right) > \tau | i = arg\ min_i\ \{a_i^k\}$ *and* $Excluded\_Rule == false$ **then**

7   Create a new rule: $R = R + 1$

8   Initialize $v_R^k, \mathcal{D}_R^k, \theta_i^k$: $v_R^k = [x^k]$, $\mathcal{D}_R^k = x^k$, $\theta_R^k = [\lambda + \kappa\langle x^k, x^k\rangle]^{-1} y^k$

9   Initialize ν_R^k: Equation (3.24)

10   **else**

11   Find the most compatible rule: $i = \arg\max_i \{c_i^k\}$

12   Update the rule center: $v_i^k = v_i^{k-1} + \alpha(c_i^k)^{(1-a_i^k)}(x^k - v_i^{k-1})$

13   Compute g, z, and r: $g^k = [\kappa\langle d_{i1}^k, x^k\rangle, \ldots, \kappa\langle d_{in_i}^k, x^k\rangle]^T$, $z^k = Q_i^{k-1} g^k$,
$r^k = \lambda + \kappa\langle x^k, x^k\rangle - (z^k)^T g^k$

14   **if** $\min_{(\forall d_{ij} \in \mathcal{D}_i^k)} \left\| x^k - d_{ij}^k \right\| \geq 0.1\nu_{ij}^k$ **then**

15   Include x^k into the dictionary: $\mathcal{D}_i^k = \mathcal{D}_i^k \cup x^k$

16   Compute Q_i^k, P_i^k, θ_i^k, ν_i^k: Equations (3.14), (3.16), (3.13), (3.23)

17   **else**

18   Compute Q_i^k, P_i^k, q_i^k, and θ_i^k: Equations (3.18), (3.20), (3.19), (3.17)

19   **for** $i = 1, 2, \ldots, R$ **do**

20   **if** $U_i^k < \epsilon$ **then**

21   Remove underused rules: Remove (i)

22   Excluded_Rule = true

23   Compute the output: $\hat{y} = \sum_{j=1}^{n_i} \theta_{ij} \kappa\langle d_{ij}^k, x^k\rangle | i = \arg\max_i \{c_i^k\}$

The first line shows the initialization of the cluster center, the dictionary, the kernel size, the matrix P, the consequent parameters, and the arousal index. In the second line, it is started a loop for all input vectors in the training phase. The compatibility measure and the arousal index are calculated in lines 4 and 5 for all rules. After the model calculates the arousal index for all rules, it compares the smallest arousal index with τ, as expressed in line 6. When the smallest arousal index is greater than τ and none rule was excluded, the model creates a new rule (line 7). The model initializes the cluster center, the dictionary, and the consequent parameters, as shown in line 8 and the kernel size according to line 9.

Otherwise, if the smallest arousal index doesn't exceed τ, the model searches the rule with the greater compatibility measure as shown in line 11, and updates the center of this rule. The model also calculates the g, z, and r of the updated rule. If the input vector is distant enough to all past inputs in the dictionary (line 14), the model adds the input into the dictionary and updates Q, P, θ, and ν according to line 16. On the other hand, if the input doesn't attend the expression presented in line 14, the model does not include the input into the dictionary and updates Q, P, θ, and ν as expressed in line 18.

Furthermore, the model computes the utility measure for all rules to eliminate underused rules. When the algorithm gets the utility measure smaller than ϵ, the model removes this rule, as written in lines 19, 20, and 21. And finally, the model computes the output using the most compatible rule according to line 22. For the test phase, the model only executes the commands shown in lines 3, 4, and 22.

4 EXPERIMENTAL RESULTS AND DISCUSSIONS

The model's performance is evaluated using two benchmark time series: the Mackey-Glass chaotic time series (widely used as a benchmark for online models) and the nonlinear system identification. These examples are applied to demonstrate that ePL-KRLS-DISCO can accurately solve online regression problems. These series are largely used in the literature of eFSs to test the model's performance, especially the Mackey-Glass. It is possible to observe the adoption of these two datasets in papers [1, 9, 10, 16, 20, 21, 22, 23, 25, 26, 33, 43, 51, 52, 55, 57, 59, 61, 62, 66, 73, 75, 76, 83, 88]. Moreover, the model is applied to predict the hot spot temperature using three datasets from a real power transformer. These data are the same used in [9, 11, 43, 44, 46, 47]. The root-mean-square error (RMSE), non-dimensional index error (NDEI), and mean absolute error (MAE) measures the precision of the models, calculated according to Equations (4.1), (4.2), and (4.3), respectively.

$$RMSE = \sqrt{\frac{1}{T}\sum_{k=1}^{T}(y^k - \hat{y}^k)^2} \tag{4.1}$$

$$NDEI = \frac{RMSE}{std([y^1, ..., y^T])} \tag{4.2}$$

$$MAE = \frac{1}{T}\sum_{k=1}^{T}|y^k - \hat{y}^k| \tag{4.3}$$

where y^k is the $k-$th actual value, $\hat{y}^k$ is the k-th predicted value, T is the sample size, and $std()$ is the standard deviation function.

Another relevant measure is computational complexity, which means predicting the resources the algorithm requires. The complexity estimation usually uses time [104, 105]. Practical usability and faster computational speed of the models increase the acceptability and applicability of evolving systems in the applications of real-world problems [72]. Thus, the models' execution time is estimated in seconds, computing the mean runtime and the standard deviation of thirty simulations. Furthermore, the number of final rules/neurons of each model is presented. The results of ePL-KRLS-DISCO are compared with traditional forecasting models such as ARIMA [106], adaptive-network-based fuzzy inference system (ANFIS) [107], Multi-Layer Perceptron (MLP) [108] trained by backpropagation, and Support Vector Machine (SVM) [109] and with the state-of-the-art of the evolving fuzzy modeling approaches, such as eTS [16], ePL [22], eMG [9], ePL+ [23], ePL-KRLS [27], and ePL-KRLS+ [30]. Additionally, a statistical test validates the obtained results. All codes were executed using Matlab 2018a in a PC device that has Intel Core i7-8565U, 1.99 GHz Turbo, and 8 GB RAM. The hyper-parameters are heuristically defined by computational experiments aiming to produce the lowest RMSE, NDEI, and MAE. The following guidelines were implemented: $\beta = \tau$, $\alpha \leq 0.1$, $\lambda \leq 10^{-2}$, $r \leq 0.5$, $\omega = 1$, $\epsilon \in [0.03, 0.05]$.

4.1 MACKEY–GLASS TIME-SERIES FORECASTING

Mackey and Glass [110] introduced a long-term time-series, proposed as a model of white blood cell production, obtained through the following differential equation:

$$\frac{dx(t)}{dt} = \frac{0.2x(t-\tau)}{1 + x^{10}(t-\tau)} - 0.1x(t-1) \quad (4.4)$$

where $x(0) = 1.2$ and $\tau = 17$.

The goal is to predict x^{k+85} using as input vector $[x^k, x^{k+6}, x^{k+12}, x^{k+18}]$ for any k value. The simulations were trained using 3000 data samples, for $k \in [201, 3200]$, and then, 500 data samples were collected to test the model for $k \in [5001, 5500]$. Table 1 shows the parameter values.

Table 1 – Models' parameters for the Mackey-Glass time-series

Model	Parameters
ARIMA [106]	$p = 1$, $d = 0$ and $q = 0$
ANFIS [107]	10 epochs
MLP [108]	a hidden layer with three neurons
SVM [109]	$C = 1$, $\gamma = 1$
eTS [16]	$r = 0.1$ and $\omega = 1000$
ePL [22]	$\beta = 0.3$, $\tau = 0.1$, $alpha = 0.1$, $\lambda = 0.8$, $r = 0.25$, and $\omega = 1000$
eMG [9]	$\alpha = 0.01$, $\lambda = 0.1$, $w = 10$, and $\Sigma_{init} = 10^{-3} I_4$
ePL+ [23]	$\beta = 0.3$, $\tau = 0.1$, $alpha = 0.1$, $\lambda = 0.8$, $\omega = 1000$, $\epsilon = 0.07$, and $\pi = 0.4$
ePL-KRLS [27]	$\beta = 0.2$, $\tau = 0.09$, $\gamma = 0.91$, $alpha = 0.1$, $\lambda = 0.0001$, $r = 0.5$, and $\omega = 1$
ePL-KRLS+ [30]	$\beta = 0.16$, $\tau = 0.16$, $\gamma = 0.84$, $alpha = 0.1$, $\lambda = 10^{-10}$, $r = 0.5$, $\omega = 1$, and $\epsilon = 0.07$
ePL-KRLS-DISCO	$\beta = 0.06$, $\tau = 0.06$, $\alpha = 0.001$, $\lambda = 10^{-7}$, $\sigma = 0.3$, $\omega = 1$, and $\epsilon = 0.05$

Source: Prepared by the author (2021)

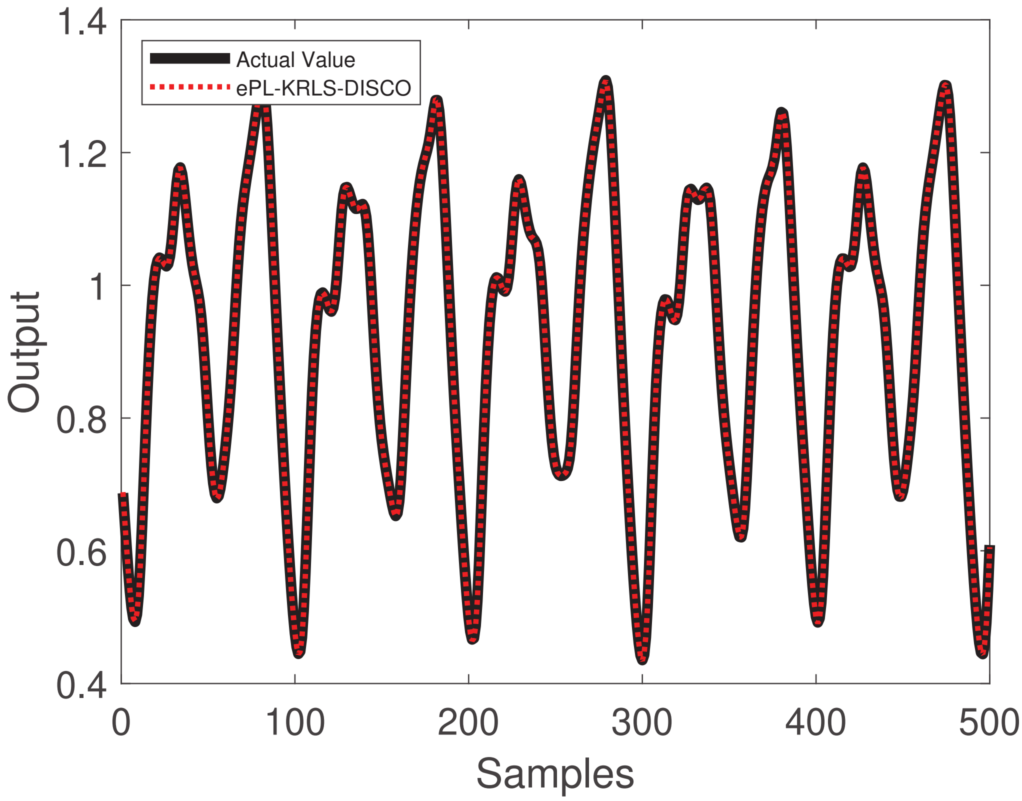
Table 2 shows the models' results. The PL-KRLS-DISCO model obtained the RMSE of 0.0013, the NDEI of 0.0057, the MAE of 0.0008, the runtime of 3.00 ± 0.08 seconds, and 14 final rules, achieving the lowest errors among all the models. This model reached errors approximately 92% smaller than ePL-KRLS+, which performed the second-best results concerning the errors and the smallest number of final rules, and ePL-KRLS-DISCO performed shorter runtime than MLP, eMG, ePL-KRLS, and ePL-KRLS+. The eMG model has the third-best errors but achieved a runtime of 34.52 ± 0.82 seconds and 89 final rules, the greatest among all models. Among the eFS models, eTS performed the shortest execution time and the highest errors. ANFIS obtained the lowest errors and the highest number of final rules among the traditional forecasting models, achieving similar errors of eMG with much inferior runtime and number of final rules. SVM performed the predictions with the shortest runtime, and MLP had the highest error values. The results suggest that ePL-KRLS-DISCO is a robust algorithm able to treat complex data with high accuracy. Figure 1 shows the graphic with the predictions of ePL-KRLS-DISCO.

Table 2 – Simulations' results of Mackey-Glass time-series

Model	RMSE	NDEI	MAE	Runtime	Rules/Neurons
ARIMA [106]	0.0918243	0.4076627	0.0736333	0.50 ± 0.07	-
ANFIS [107]	0.0259086	0.1150235	0.0209717	1.33 ± 0.05	16
MLP [108]	0.1043437	0.4632439	0.0832548	15.86 ± 1.05	8
SVM [109]	0.0948219	0.4209707	0.0750996	**0.15 ± 0.02**	-
eTS [16]	0.0848845	0.3768529	0.0684699	0.25 ± 0.06	8
ePL [22]	0.0922248	0.4094407	0.0744546	0.49 ± 0.13	7
eMG [9]	0.0188697	0.0837737	0.0122970	34.52 ± 0.82	89
ePL+ [23]	0.0924519	0.4104489	0.0746511	0.44 ± 0.09	6
ePL-KRLS [27]	0.0227922	0.1011883	0.0188428	9.12 ± 0.73	1
ePL-KRLS+ [30]	0.0145419	0.0645604	0.0111955	8.24 ± 0.41	**1**
ePL-KRLS-DISCO	**0.0012738**	**0.0056551**	**0.0008272**	3.00 ± 0.08	14

Source: Prepared by the author (2021)

Figure 1 – Predictions of ePL-KRLS-DISCO for the Mackey–Glass time-series



Source: Prepared by the author (2021)

The modified Morgan-Granger-Newbold test (MGN) [111] is implemented to validate the results. MGN is a Diebold-Mariano (DM) Test proposed to overcome the limitations of the original MGN test [112]. Considering two sets of samples $\mathcal{G}_1$ and $\mathcal{G}_2$, the MGN test allows us to infer assumptions from two independent data samples and to verify their statistical validity. The hypothesis is given as follows:

$$\begin{cases} \mathcal{H}_0 : \mathcal{G}_1 = \mathcal{G}_2 \\ \mathcal{H}_1 : \mathcal{G}_1 \neq \mathcal{G}_2 \end{cases} \tag{4.5}$$

Given a significance level α, usually around 0.05, the p-value represents the lowest value of α to reject the null hypothesis $\mathcal{H}_0$. Thus, values of the p-value below α means that the null hypothesis is not true. The sets of samples $\mathcal{G}_1$ and $\mathcal{G}_2$ are the outputs of the models, where $\mathcal{G}_1$ refers to ePL-KRLS-DISCO and $\mathcal{G}_2$ to the compared model. Table 3 list the results of the MGN test performed for the test accuracy metric in Table 2. The rejection of the null hypothesis is described by letters ‘W’ and ‘L’ meaning respectively win and loss of the method tested, and the non-rejection of the null hypothesis is expressed by ‘E’ which means equality of the tested methods. The results of the statistical tests show that ePL-KRLS-DISCO yields better performance for the Mackey-Glass time-series concerning all compared models, with a 95% confidence level.

Table 3 – MGN test for the results of the Mackey-Glass time-series

$\mathcal{G}_1$	$\mathcal{G}_2$	MGN	p-value	Null hypothesis
ePL-KRLS-DISCO	ARIMA	-17.0195868	1.5×10^{-51}	W
ePL-KRLS-DISCO	ANFIS	-17.0386727	1.2×10^{-51}	W
ePL-KRLS-DISCO	MLP	-16.7557232	2.5×10^{-50}	W
ePL-KRLS-DISCO	SVM	-15.7203781	1.6×10^{-45}	W
ePL-KRLS-DISCO	eTS	-17.7545414	5.0×10^{-55}	W
ePL-KRLS-DISCO	ePL	-17.4031165	2.3×10^{-53}	W
ePL-KRLS-DISCO	eMG	-6.2260174	1.0×10^{-9}	W
ePL-KRLS-DISCO	ePL+	-17.40286747	2.3×10^{-53}	W
ePL-KRLS-DISCO	ePL-KRLS	-18.4851863	1.7×10^{-58}	W
ePL-KRLS-DISCO	ePL-KRLS+	-13.7731833	8.3×10^{-37}	W

Source: Prepared by the author (2021)

4.2 NONLINEAR DYNAMIC SYSTEM IDENTIFICATION

The classic nonlinear dynamic system identification, described in [113], is obtained through the following difference equation [114]:

$$y^k = \frac{y^{k-1}y^{k-2}(y^{k-1} - 0.5)}{1 + (y^{k-1})^2 + (y^{k-2})^2} - u^{k-1} \tag{4.6}$$

where $u^k = sin(\frac{2\pi k}{25})$ and $y^0 = y^1 = 0$.

The goal is to predict y^k using as input vector $[y^{k-2}, y^{k-1}, u^{k-1}]$ for any k value. The k value is set as $k \in [2, 5201]$, where the first 5000 data samples trained the models, and the last 200 data samples tested them. Table 4 presents the models' parameters.

Table 4 – Models' parameters for the nonlinear dynamic system identification

Model	Parameters
ARIMA [106]	$p = 2$, $d = 1$ and $q = 0$
ANFIS [107]	10 epochs
MLP [108]	a hidden layer with five neurons
SVM [109]	$C = 0.01$, $\gamma = 1$
eTS [16]	$r = 0.4$ and $\omega = 750$
ePL [22]	$\beta = 0.3$, $\tau = 0.1$, $alpha = 0.1$, $\lambda = 0.8$, $r = 0.25$, and $\omega = 1000$
eMG [9]	$\alpha = 0.01$, $\lambda = 0.1$, $w = 10$, and $\Sigma_{init} = 10^{-5} I_3$
ePL+ [23]	$\beta = 0.3$, $\tau = 0.1$, $alpha = 0.1$, $\lambda = 0.8$, $\omega = 1000$, $\epsilon = 0.07$, and $\pi = 0.4$
ePL-KRLS [27]	$\beta = 0.24$, $\tau = 0.24$, $\gamma = 0.76$, $alpha = 0.1$, $\lambda = 10^{-10}$, $r = 0.5$, and $\omega = 1$
ePL-KRLS+ [30]	$\beta = 0.3$, $\tau = 0.3$, $\gamma = 0.7$, $alpha = 0.1$, $\lambda = 10^{-10}$, $r = 0.5$, $\omega = 1$, and $\epsilon = 0.07$
ePL-KRLS-DISCO	$\beta = 0.1$, $\tau = 0.1$, $\alpha = 0.1$, $\lambda = 10^{-16}$, $\sigma = 0.5$, $\omega = 1$, and $\epsilon = 0.05$

Source: Prepared by the author (2021)

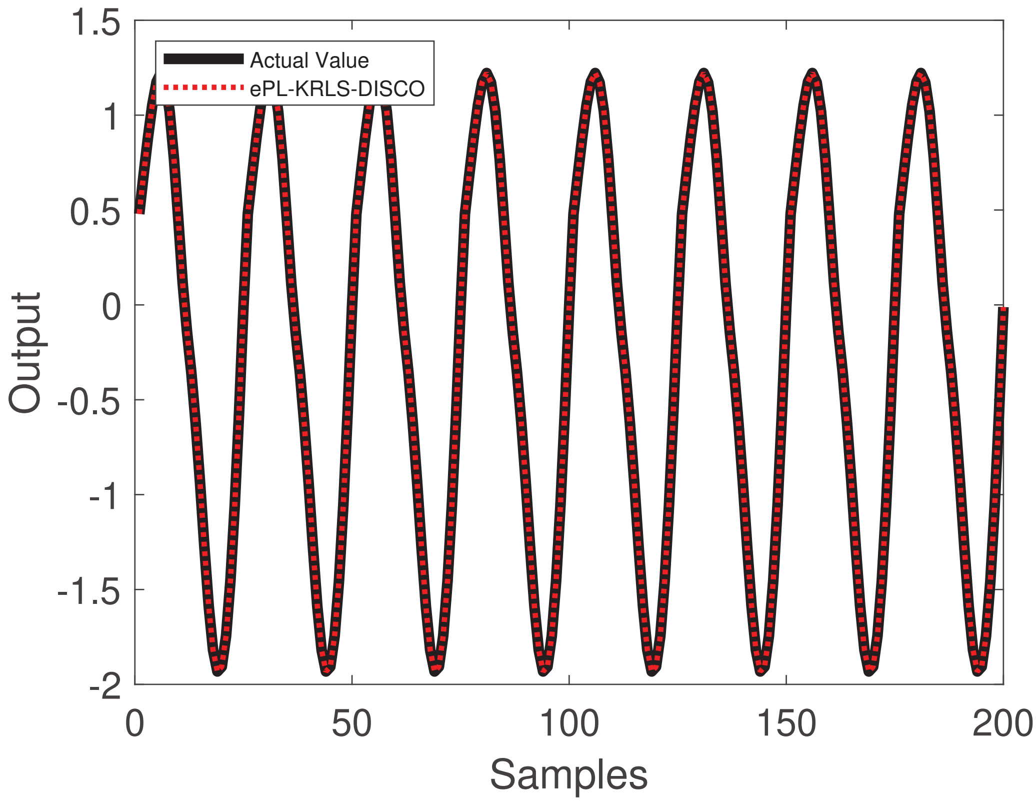
Table 5 shows the results of the simulations. The ePL-KRLS-DISCO model achieved the second-best accuracy among all the models, obtaining the RMSE of 9.0×10^{-10}, the NDEI of 8.2×10^{-10}, and the MAE of 2.3×10^{-10}. These error results correspond to a reduction of more than 99.99% compared to the ePL-KRLS and ePL-KRLS+. This model simulated the nonlinear in 1.88 ± 0.08 seconds and obtained seventeen final rules. The eMG algorithm reached the best error results but performed simulations in 11.09 seconds and achieved 26 final rules, the greatest among all models. The proposed model performed simulations in about a sixth of eMG runtime and half of its final rules. The ePL-KRLS and ePL-KRLS+ approaches obtained the smallest number of final rules and achieved inferior errors than ARIMA, MLP, SVM, eTS, ePL, and ePL+. The eTS algorithm had the shortest runtime among the eFS models, but it performed predictions with the highest errors among all the models. The shortest runtime is performed by the SVM. Considering only the traditional forecasting models, MLP obtained the highest errors, and ANFIS performed the smallest one, getting errors inferior to ePL-KRLS and ePL-KRLS+. These results indicate the superiority of the proposed model to deal with nonlinear data. Figure 2 shows the graphic of the predictions of ePL-KRLS-DISCO.

Table 5 – Simulations' results of nonlinear dynamic system identification

Model	RMSE	NDEI	MAE	Runtime (s)	Rules/Neurons
ARIMA [106]	0.0717446	0.0656251	0.0575244	0.74 ± 0.06	-
ANFIS [107]	0.0000309	0.0000283	0.0000208	0.49 ± 0.02	8
MLP [108]	0.1937605	0.1772336	0.1443096	29.66 ± 62.14	9
SVM [109]	0.1024126	0.0936773	0.0929539	$\mathbf{0.10 \pm 0.02}$	-
eTS [16]	0.2857880	0.2614116	0.2527342	0.23 ± 0.02	5
ePL [22]	0.0486007	0.0444553	0.0338808	0.39 ± 0.04	6
eMG [9]	$\mathbf{9.0 \times 10^{-11}}$	$\mathbf{8.2 \times 10^{-11}}$	$\mathbf{1.8 \times 10^{-11}}$	11.09 ± 1.16	26
ePL+ [23]	0.0486047	0.0444589	0.0338827	0.43 ± 0.03	5
ePL-KRLS [27]	0.0000904	0.0000827	0.0000465	0.96 ± 0.16	**1**
ePL-KRLS+ [30]	0.0000904	0.0000827	0.0000465	0.88 ± 0.03	**1**
ePL-KRLS-DISCO	9.0×10^{-10}	8.2×10^{-10}	2.3×10^{-10}	1.88 ± 0.08	17

Source: Prepared by the author (2021)

Figure 2 – Predictions of ePL-KRLS-DISCO for the nonlinear dynamic system identification



Source: Prepared by the author (2021)

Table 6 shows the results of the MGN test, supporting that ePL-KRLS-DISCO yields better performance for the nonlinear system identification concerning all models, except for eMG.

Table 6 – MGN test for the results of the nonlinear dynamic system

$\mathcal{G}_1$	$\mathcal{G}_2$	MGN	p-value	Null hypothesis
ePL-KRLS-DISCO	ARIMA	-12.6121967	3.5×10^{-27}	W
ePL-KRLS-DISCO	ANFIS	-8.5998942	2.4×10^{-15}	W
ePL-KRLS-DISCO	MLP	-19.3042450	1.8×10^{-47}	W
ePL-KRLS-DISCO	SVM	-24.2009144	3.4×10^{-61}	W
ePL-KRLS-DISCO	eTS	-15.1368173	6.0×10^{-35}	W
ePL-KRLS-DISCO	ePL	-6.1845240	3.5×10^{-9}	W
ePL-KRLS-DISCO	eMG	3.2378872	0.0014109	L
ePL-KRLS-DISCO	ePL+	-6.1843327	3.5×10^{-9}	W
ePL-KRLS-DISCO	ePL-KRLS	-5.6952260	4.4×10^{-8}	W
ePL-KRLS-DISCO	ePL-KRLS+	-5.6952260	4.4×10^{-8}	W

Source: Prepared by the author (2021)

4.3 HOT SPOT TEMPERATURE FORECASTING

And finally, the model is applied in the thermal modeling of the power transformer. Table 7 presents the characteristics of the transformer.

Table 7 – Characteristics of the experimental power transformer

Copper losses	776 W
Factory year	MACE/1987
Iron losses	195 W
Nameplate rating	25 kVA
Tank dimensions	$64 \times 16 \times 80\ cm^3$
Top oil temperature rise at full load	73.1 °C
Type of cooling	ONAN
$V_{primary}/V_{secondary}$	10 kV / 380 kV
Weight of core and coil assembly	136 kg
Weight of oil	62 kg

Source: [46]

The aim is to predict the hot spot temperature using as inputs the load current (K), the top oil temperature (Θ_{TO}), and one step delayed load current ($q^{-1}K$, where q^{-1} is the delay operator). Three datasets were implemented in the simulations. Each one consists of measurements taken every 5 minutes for 24 hours. The firsts two datasets have no overload conditions (dataset 1 and 2), and the last one is with overload conditions (dataset 3). The Augmented Dickey-Fuller (ADF) test [115] for a unit root is implemented to verify the stationarity of the datasets. For a p-value smaller than 0.05, the test rejects the null hypothesis of a unit root against the autoregressive alternative, with a 95% confidence level. Table 8 shows the tests' results. All tests fail to reject the null hypothesis, indicating that all databases are non-stationary.

Furthermore, the Tsay test [116] is performed to verify the linearity of the datasets. For a p-value smaller than 0.05, the test rejects the null hypothesis that the true model is

Table 8 – ADF test for the transformers datasets

	Dataset 1	Dataset 2	Dataset 3
p-value	0.62	0.65	0.62

Source: Prepared by the author (2021)

an autoregressive (AR) process, with a 95% level of confidence. Table 9 shows the results. All tests fail to reject the null hypothesis, indicating that all databases output variable depends linearly on its previous values.

Table 9 – Tsay test for the transformers datasets

	Dataset 1	Dataset 2	Dataset 3
p-value	0.79	0.75	0.84

Source: Prepared by the author (2021)

For the dataset 1, ARIMA has $p = 2$, $d = 1$, and $q = 5$; ANFIS considers 10 epochs; MLP has a hidden layer with three neurons; SVM has $C = 1$, and $\gamma = 1$; eTS has $r = 0.9$ and $\omega = 1000$; ePL has $\beta = 0.04$, $\tau = 0.04$, $alpha = 0.1$, $\lambda = 0.9$, $r = 0.08$, and $\omega = 1000$; ePL+ has $\beta = 0.3$, $\tau = 0.3$, $alpha = 0.01$, $\lambda = 0.0001$, $\omega = 1000$, $\epsilon = 0.03$, and $\pi = 0.3$; eMG has $\alpha = 0.05$, $\lambda = 0.3$, $w = 10$, and $\Sigma_{init} = 3 \times 10^{-2} I_3$; ePL-KRLS has $\beta = 0.1$, $\tau = 0.1$, $\gamma = 0.99$, $alpha = 0.1$, $\lambda = 0.0001$, $r = 0.9$, and $\omega = 1$; ePL-KRLS+ has $\beta = 0.09$, $\tau = 0.09$, $\gamma = 0.91$, $alpha = 0.1$, $\lambda = 10^{-10}$, $r = 0.1$, $\omega = 1$, and $\epsilon = 0.07$; ePL-KRLS-DISCO has $\beta = 0.04$, $\tau = 0.04$, $\alpha = 0.05$, $\lambda = 10^{-10}$, $\sigma = 0.5$, $\omega = 1$, and $\epsilon = 0.05$.

For the dataset 2, ARIMA has $p = 1$, $d = 1$, and $q = 4$; ANFIS considers 10 epochs; MLP has a hidden layer with three neurons; SVM has $C = 1$, and $\gamma = 1$; eTS has $r = 0.4$ and $\omega = 1000$; ePL has $\beta = 0.08$, $\tau = 0.08$, $alpha = 0.1$, $\lambda = 0.8$, $r = 0.2$, and $\omega = 1000$; ePL+ has $\beta = 0.1$, $\tau = 0.1$, $alpha = 0.1$, $\lambda = 0.6$, $\omega = 1000$, $\epsilon = 0.07$, and $\pi = 0.4$; eMG has $\alpha = 0.01$, $\lambda = 0.7$, $w = 10$, and $\Sigma_{init} = 6 \times 10^{-2} I_3$; ePL-KRLS has $\beta = 0.14$, $\tau = 0.14$, $\gamma = 0.86$, $alpha = 0.1$, $\lambda = 0.0001$, $r = 0.5$, and $\omega = 1$; ePL-KRLS+ has $\beta = 0.16$, $\tau = 0.16$, $\gamma = 0.84$, $alpha = 0.1$, $\lambda = 10^{-10}$, $r = 0.5$, $\omega = 1$, and $\epsilon = 0.07$; ePL-KRLS-DISCO has $\beta = 0.6$, $\tau = 0.6$, $\alpha = 0.15$, $\lambda = 10^{-10}$, $\sigma = 0.5$, $\omega = 1$, and $\epsilon = 0.03$.

For the dataset 3, ARIMA has $p = 2$, $d = 2$, and $q = 2$; ANFIS considers 10 epochs; MLP has a hidden layer with three neurons; SVM has $C = 1$, and $\gamma = 1$; eTS has $r = 0.7$ and $\omega = 1000$; ePL has $\beta = 0.001$, $\tau = 0.003$, $alpha = 0.1$, $\lambda = 0.72$, $r = 0.2$, and $\omega = 1000$; ePL+ has $\beta = 0.001$, $\tau = 0.003$, $alpha = 0.1$, $\lambda = 0.8$, $\omega = 1000$, $\epsilon = 0.07$, and $\pi = 0.4$; eMG has $\alpha = 0.2$, $\lambda = 0.05$, $w = 30$, and $\Sigma_{init} = I_4$; ePL-KRLS has $\beta = 0.005$, $\tau = 0.005$, $\gamma = 0.995$, $alpha = 0.9$, $\lambda = 0.00001$, $r = 0.37$, and $\omega = 1$; ePL-KRLS+ has

$\beta = 0.18$, $\tau = 0.18$, $\gamma = 0.82$, $alpha = 0.1$, $\lambda = 10^{-10}$, $r = 0.5$, $\omega = 1$, and $\epsilon = 0.07$; ePL-KRLS-DISCO has $\beta = 0.1$, $\tau = 0.1$, $\alpha = 0.1$, $\lambda = 10^{-8}$, $\sigma = 0.5$, $\omega = 1$, and $\epsilon = 0.04$.

Table 10 presents the performance of the models for dataset 1. The ePL-KRLS-DISCO model obtained the RMSE of 0.0015, the NDEI of 0.0118, the MAE of 0.0006, the runtime of 0.08 ± 0.01 seconds, and seventeen final rules. The proposed model reached the smallest error values, obtaining errors approximately 54% lower than ePL-KRLS+ and more than 92% lower than ePL-KRLS. The ePL-KRLS+ performed the second-best error values with just one final rule, the same number of final rules as ePL+ and ePL-KRLS. The eTS and ePL algorithms performed the simulations with the shortest runtime among the eFSs, and eTS achieved the highest number of final rules. MLP obtained the worst error values, and ARIMA performed the best error values among the traditional forecasting models.

Table 10 – Simulations' results for the dataset 1 of the power transformer

Model	RMSE	NDEI	MAE	Runtime	Rules/Neurons
ARIMA [106]	0.0075217	0.0610153	0.0039499	0.50 ± 0.07	-
ANFIS [107]	0.0147961	0.1200251	0.0113695	0.04 ± 0.01	8
MLP [108]	0.0377404	0.3061469	0.0305381	8.25 ± 1.00	7
SVM [109]	0.0231634	0.1878998	0.0174993	**0.01 ± 0.02**	-
eTS [16]	0.0174631	0.1416590	0.0112957	0.02 ± 0.00	16
ePL [22]	0.0126353	0.1024968	0.0091833	0.02 ± 0.00	5
eMG [9]	0.0156586	0.1270215	0.0117626	0.07 ± 0.01	3
ePL+ [23]	0.0195684	0.1587369	0.0151500	0.01 ± 0.00	**1**
ePL-KRLS [27]	0.0190857	0.1548216	0.0139502	0.06 ± 0.01	**1**
ePL-KRLS+ [30]	0.0031561	0.0256019	0.0013618	0.22 ± 0.01	**1**
ePL-KRLS-DISCO	**0.0014586**	**0.0118321**	**0.0006085**	0.08 ± 0.01	17

Source: Prepared by the author (2021)

Table 11 shows the performance of the models for dataset 2. The ePL-KRLS-DISCO model obtained the RMSE of 0.0003, the NDEI of 0.0048, the MAE of 8.1^{-5}, the runtime of 0.10 ± 0.02 seconds, and two final rules, achieving the smallest error values, about 85% inferior concerning the ePL-KRLS+ and approximately 96% lower than ePL-KRLS with less runtime. It also reached shorter runtime than ARIMA and MLP. The ePL+, ePL-KRLS, and ePL-KRLS+ approaches obtained just one final rule, and eTS achieved the highest number of final rules. MLP performed the highest runtime, SVM, eTS, and ePL the shortest one, and ANFIS obtained the third-best error values with a shorter runtime.

Table 12 presents the results of the simulations for dataset 3, which has overload conditions. The ePL-KRLS-DISCO model reached the RMSE of 0.0007, NDEI of 0.0036, MAE of 0.0002, the runtime of 0.07 ± 0.01 seconds, and eight final rules, obtaining the smallest errors. The ePL-KRLS-DISCO model achieved error values approximately 80% smaller than the ePL-KRLS+ and performed simulations faster than ePL-KRLS and ePL-

Table 11 – Simulations' results for the dataset 2 of the power transformer

Model	RMSE	NDEI	MAE	Runtime	Rules/Neurons
ARIMA [106]	0.0067408	0.1059581	0.0049175	0.16 ± 0.04	-
ANFIS [107]	0.0026448	0.0415743	0.0017017	0.04 ± 0.01	8
MLP [108]	0.0194530	0.3057816	0.0168367	42.32 ± 3.34	7
SVM [109]	0.0118426	0.1861537	0.0087845	**0.01 ± 0.02**	-
eTS [16]	0.0082422	0.1295587	0.0052521	**0.01 ± 0.00**	9
ePL [22]	0.0069890	0.1098594	0.0049435	**0.01 ± 0.00**	2
eMG [9]	0.0069270	0.1088853	0.0047034	0.09 ± 0.01	4
ePL+ [23]	0.0075628	0.1188787	0.0048492	0.02 ± 0.01	**1**
ePL-KRLS [27]	0.0069614	0.1094266	0.0029991	0.14 ± 0.01	**1**
ePL-KRLS+ [30]	0.0020022	0.0314721	0.0009663	0.12 ± 0.01	**1**
ePL-KRLS-DISCO	**0.0003078**	**0.0048386**	**0.0000811**	0.10 ± 0.02	2

Source: Prepared by the author (2021)

KRLS+. The ePL-KRLS+ algorithm performed the second-best results concerning the errors, reaching just one final rule, the smallest number of final rules among all simulations, and eTS performed the highest number of final rules. The ePL model performed the shortest runtime, and MLP the longest one. ANFIS reached the third-best errors with a short runtime. Figures 3 4 and 5 show the graphics of the predictions of ePL-KRLS-DISCO for datasets 1, 2, and 3, respectively.

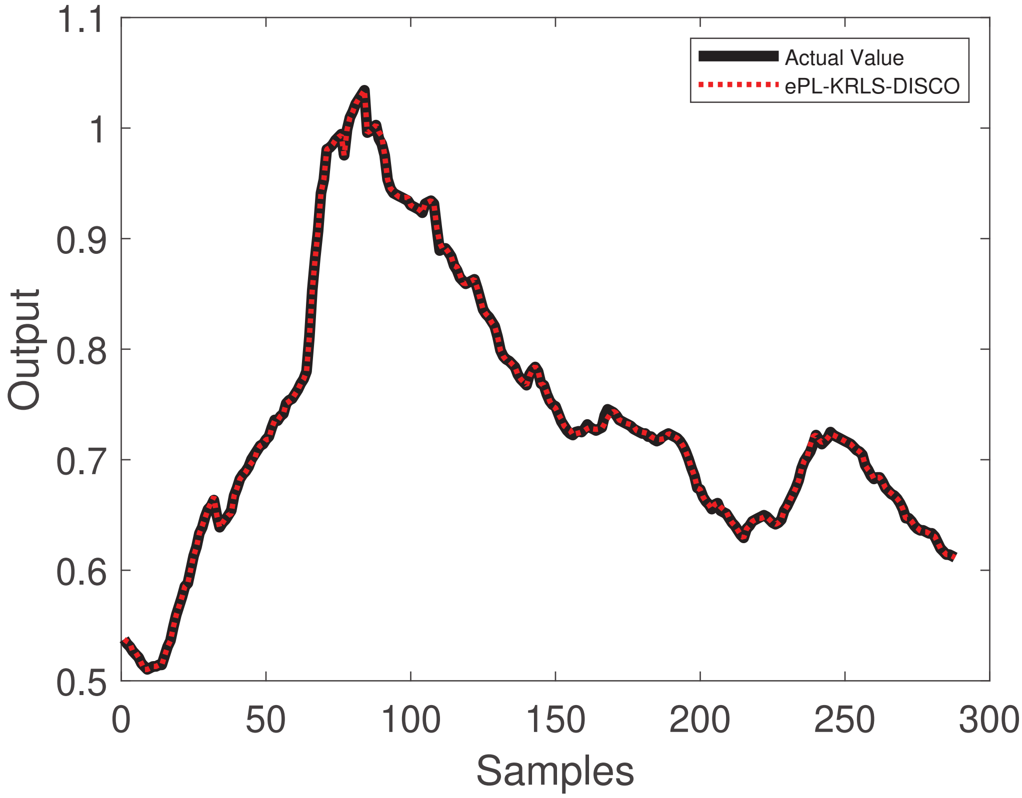
Table 12 – Simulations' results for the dataset 3 of the power transformer

Model	RMSE	NDEI	MAE	Runtime	Rules/Neurons
ARIMA [106]	0.0074403	0.0362909	0.0028487	0.49 ± 0.06	-
ANFIS [107]	0.0110810	0.0540488	0.0068150	0.04 ± 0.01	8
MLP [108]	0.0548766	0.2676667	0.0344566	2.58 ± 0.19	7
SVM [109]	0.0434348	0.2118577	0.0309763	0.02 ± 0.02	-
eTS [16]	0.0244274	0.1191473	0.0129252	0.04 ± 0.01	26
ePL [22]	0.0185969	0.0907086	0.0085326	**0.01 ± 0.00**	2
eMG [9]	0.0190197	0.0927704	0.0121184	0.05 ± 0.01	3
ePL+ [23]	0.0202123	0.0985875	0.0099453	0.02 ± 0.00	2
ePL-KRLS [27]	0.0171763	0.0837791	0.0080920	0.10 ± 0.01	2
ePL-KRLS+ [30]	0.0035913	0.0175170	0.0018634	0.12 ± 0.01	**1**
ePL-KRLS-DISCO	**0.0007307**	**0.0035643**	**0.0002468**	0.07 ± 0.01	8

Source: Prepared by the author (2021)

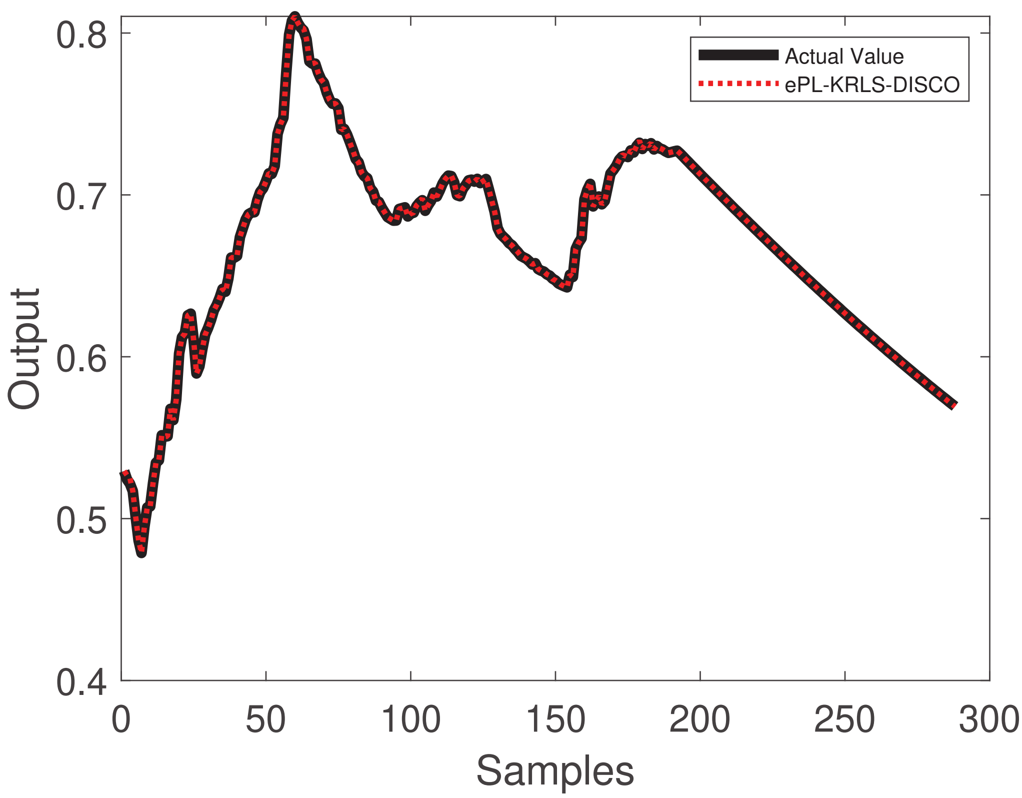
The results of the statistical tests, presented in Tables 13, 14, and 15, show that ePL-KRLS-DISCO yields better performance for datasets 1, 2, and 3 concerning all compared models, with a 95% confidence level.

Figure 3 – Estimation of hot spot temperature without overload condition for all models



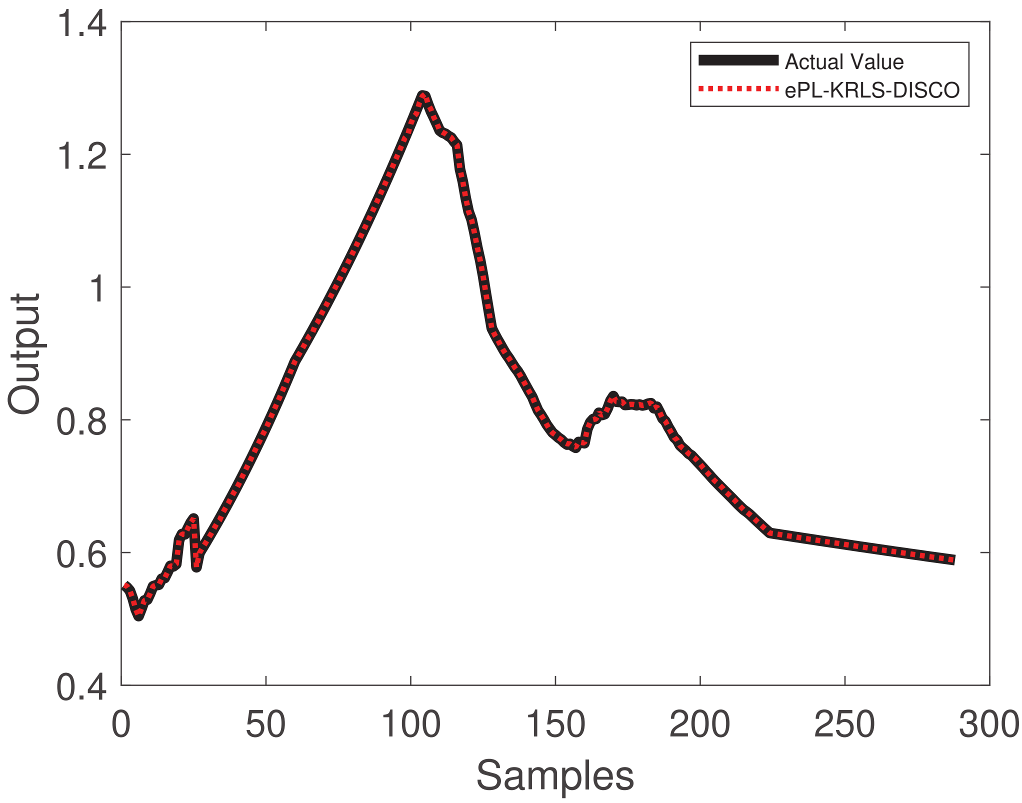
Source: Prepared by the author (2021)

Figure 4 – Estimation of hot spot temperature without overload condition



Source: Prepared by the author (2021)

Figure 5 – Estimation of hot spot temperature with overload condition



Source: Prepared by the author (2021)

Table 13 – MGN test for the results of the dataset 1 of the power transformer

$\mathcal{G}_1$	$\mathcal{G}_2$	MGN	p-value	Null hypothesis
ePL-KRLS-DISCO	ARIMA	-2.8681142	0.0044345	W
ePL-KRLS-DISCO	ANFIS	-10.9625888	1.3×10^{-23}	W
ePL-KRLS-DISCO	MLP	-13.9645529	3.4×10^{-34}	W
ePL-KRLS-DISCO	SVM	-8.5264028	8.7×10^{-16}	W
ePL-KRLS-DISCO	eTS	-5.3536283	1.8×10^{-7}	W
ePL-KRLS-DISCO	ePL	-6.9623190	2.3×10^{-11}	W
ePL-KRLS-DISCO	eMG	-7.5805032	7.6×10^{-13}	W
ePL-KRLS-DISCO	ePL+	-10.6531145	1.5×10^{-22}	W
ePL-KRLS-DISCO	ePL-KRLS	-5.8677983	1.2×10^{-8}	W
ePL-KRLS-DISCO	ePL-KRLS+	-3.8743454	0.0001324	W

Source: Prepared by the author (2021)

Table 14 – MGN test for the results of the dataset 2 of the power transformer

$\mathcal{G}_1$	$\mathcal{G}_2$	MGN	p-value	Null hypothesis
ePL-KRLS-DISCO	ARIMA	-7.1593994	1.9×10^{-12}	W
ePL-KRLS-DISCO	ANFIS	-5.5446575	6.7×10^{-8}	W
ePL-KRLS-DISCO	MLP	-16.5174974	1.4×10^{-43}	W
ePL-KRLS-DISCO	SVM	-7.5779066	4.8×10^{-13}	W
ePL-KRLS-DISCO	eTS	-5.1149601	5.7×10^{-7}	W
ePL-KRLS-DISCO	ePL	-7.8264477	9.6×10^{-14}	W
ePL-KRLS-DISCO	eMG	-7.8967298	6.1×10^{-14}	W
ePL-KRLS-DISCO	ePL+	-6.4557110	4.6×10^{-10}	W
ePL-KRLS-DISCO	ePL-KRLS	-2.6126780	0.0094552	W
ePL-KRLS-DISCO	ePL-KRLS+	-5.1840136	4.1×10^{-7}	W

Source: Prepared by the author (2021)

Table 15 – MGN test for the results of the dataset 3 of the power transformer

$\mathcal{G}_1$	$\mathcal{G}_2$	MGN	p-value	Null hypothesis
ePL-KRLS-DISCO	ARIMA	-2.2572225	0.0247433	W
ePL-KRLS-DISCO	ANFIS	-5.9328417	8.5×10^{-9}	W
ePL-KRLS-DISCO	MLP	-7.4814592	9.0×10^{-13}	W
ePL-KRLS-DISCO	SVM	-8.1805141	9.2×10^{-15}	W
ePL-KRLS-DISCO	eTS	-3.5483855	0.0004524	W
ePL-KRLS-DISCO	ePL	-3.6012074	0.0003729	W
ePL-KRLS-DISCO	eMG	-8.1101470	1.4×10^{-14}	W
ePL-KRLS-DISCO	ePL+	-4.0951969	9.6×10^{-5}	W
ePL-KRLS-DISCO	ePL-KRLS	-3.9570239	0.0094552	W
ePL-KRLS-DISCO	ePL-KRLS+	-4.3687168	1.7×10^{-5}	W

Source: Prepared by the author (2021)

4.4 DISCUSSIONS

The simulations proved the robustness of the proposed model to predict complex data. The ePL-KRLS-DISCO model obtained far superior results concerning the error metrics in the synthetic and power transformers datasets. The proposed model outperformed all compared models concerning the RMSE, NDEI, and MAE for the Mackey-Glass time-series and the power transformers' datasets. The MGN test validates the results statistically, supporting the accuracy of the proposed model. The DISCO implementation to compute the compatibility measure makes the model create clusters with a smaller standard deviation. Consequently, the model forms rules with similar characteristics. So, the algorithm can extract useful information from each cluster and can make accurate predictions through the KRLS approach. The output is calculated based on the most similar rule since each one holds similar attributes extracted from inputs. These attributes are stored in the local dictionaries to don't lose useful information about past inputs. Consequently, the proposed model can model complex behaviors from streams, dealing accurately with non-stationary data.

And the superior performance of ePL-KRLS-DISCO didn't impact directly in the runtime and the number of final rules. The ePL-KRLS-DISCO approach performed a longer runtime than ePL-KRLS and ePL-KRLS+ in just two simulations. In this sense, simulations demonstrate that the introduced model performs predictions with competitive runtime. The small number of rules from ePL-KRLS-DISCO provides explainable and interpretable results. The small computational cost accrues from the KRLS algorithm. KRLS makes nonlinear predictions by making linear operations in the kernel space. KRLS also uses sparcification procedures to keep small the computational cost. The results demonstrate the high level of autonomy and adaptation of the proposed model to deal with complex and non-stationary data. The model advances its structure autonomously during the life span of the system. An accurate estimation of the hot spot temperature supports defining the load capacity by maximizing the long-term cost-benefit. Thus, the results support the adoption of ePL-KRLS-DISCO to monitor the hot spot temperature.

5 CONCLUSIONS

This work introduced a novel rule-based eFS model termed ePL-KRLS-DISCO. The proposed model is evaluated in terms of errors, runtime, and the number of final rules. Two benchmark time series are implemented to evaluate the models. Finally, the model is applied to predict the hot spot temperature using three datasets from a real power transformer. The simulations' results are compared with traditional forecasting models and some related state-of-the-art evolving fuzzy modeling approaches.

The ePL-KRLS-DISCO model outperformed all compared models for the Mackey-Glass time-series and the power transformers datasets concerning the errors. The MGN test validates statistically the results, proving the superior performance of the proposed model. The results demonstrate the ability of ePL-KRLS-DISCO to model the underlying system of streams, dealing accurately with non-stationary time series. The model also obtained a competitive runtime and number of final rules, supporting that the models have a competitive computational complexity. The results support that the ePL-KRLS-DISCO is a robust model able to adapt its structure and functionality according to the data change, predicting data precisely and with competitive computational cost.

Considering the importance of accurate models, the results suggest the ePL-KRLS-DISCO implementation as a time series forecasting tool to support decision-making. As future work, it is suggested to use optimization algorithms to define the model parameters to improve its performance in terms of errors.

REFERENCES

[1] D. Ge, X.-J. Zeng, Learning data streams online — an evolving fuzzy system approach with self-learning/adaptive thresholds, Information Sciences 507 (2020) 172–184. doi:https://doi.org/10.1016/j.ins.2019.08.036.

[2] Y. Hong, M. Zhang, W. Q. Meeker, Big data and reliability applications: The complexity dimension, Journal of Quality Technology 50 (2) (2018) 135–149. doi:https://doi.org/10.1080/00224065.2018.1438007.

[3] Y. Duan, J. S. Edwards, Y. K. Dwivedi, Artificial intelligence for decision making in the era of big data – evolution, challenges and research agenda, International Journal of Information Management 48 (2019) 63–71. doi:https://doi.org/10.1016/j.ijinfomgt.2019.01.021.

[4] Z. Allam, Z. A. Dhunny, On big data, artificial intelligence and smart cities, Cities 89 (2019) 80–91. doi:https://doi.org/10.1016/j.cities.2019.01.032.

[5] A. Agrawal, J. S. Gans, A. Goldfarb, Artificial intelligence: the ambiguous labor market impact of automating prediction, Journal of Economic Perspectives 33 (2) (2019) 31–50. doi:https://doi.org/10.1257/jep.33.2.31.

[6] P. R. Daugherty, H. J. Wilson, Human+ machine: Reimagining work in the age of AI, Harvard Business Press, 2018.

[7] C. Hartmann, M. Hahmann, W. Lehner, F. Rosenthal, Exploiting big data in time series forecasting: A cross-sectional approach, in: 2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA), IEEE, 2015, pp. 1–10. doi:https://doi.org/10.1109/dsaa.2015.7344786.

[8] D. Leite, R. Ballini, P. Costa, F. Gomide, Evolving fuzzy granular modeling from nonstationary fuzzy data streams, Evolving Systems 3 (2) (2012) 65–79. doi:https://doi.org/10.1007/s12530-012-9050-9.

[9] A. Lemos, W. Caminhas, F. Gomide, Multivariable gaussian evolving fuzzy modeling system, IEEE Transactions on Fuzzy Systems 19 (1) (2010) 91–104. doi:https://doi.org/10.1109/tfuzz.2010.2087381.

[10] R. Vieira, F. Gomide, R. Ballini, Kernel evolving participatory fuzzy modeling for time series forecasting, in: 2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), IEEE, 2018, pp. 1–9. doi:https://doi.org/10.1109/fuzz-ieee.2018.8491484.

[11] K. S. T. R. Alves, M. Hell, F. L. C. Oliveira, E. P. de Aguiar, An enhanced set-membership evolving participatory learning with kernel recursive least squares applied to thermal modeling of power transformers, Electric Power Systems Research 184 (2020) 106334. doi:https://doi.org/10.1016/j.epsr.2020.106334.

[12] A. Sujatha, L. Govindaraju, N. Shivakumar, V. Devaraj, Fuzzy knowledge based system for suitability of soils in airfield applications, Civil Engineering Journal 7 (1) (2021) 140–152. doi:https://doi.org/10.28991/cej-2021-03091643.

[13] E. Lughofer, On-line assurance of interpretability criteria in evolving fuzzy systems – achievements, new concepts and open issues, Information Sciences 251 (2013) 22–46. doi:https://doi.org/10.1016/j.ins.2013.07.002.

[14] E. Lughofer, M. Pratama, I. Skrjanc, Incremental rule splitting in generalized evolving fuzzy systems for autonomous drift compensation, IEEE Transactions on Fuzzy Systems 26 (4) (2017) 1854–1865. doi:https://doi.org/10.1109/tfuzz.2017.2753727.

[15] P. Angelov, D. Filev, On-line design of takagi-sugeno models, in: International Fuzzy Systems Association World Congress, Springer, 2003, pp. 576–584. doi:https://doi.org/10.1007/3-540-44967-1_69.

[16] P. P. Angelov, D. P. Filev, An approach to online identification of takagi-sugeno fuzzy models, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 34 (1) (2004) 484–498. doi:https://doi.org/10.1109/tsmcb.2003.817053.

[17] J. Jiang, Y. Zhang, A revisit to block and recursive least squares for parameter estimation, Computers & Electrical Engineering 30 (5) (2004) 403–416. doi:https://doi.org/10.1016/s0045-7906(04)00021-7.

[18] X. Yu, K. Turksoy, M. Rashid, J. Feng, N. Hobbs, I. Hajizadeh, S. Samadi, M. Sevil, C. Lazaro, Z. Maloney, et al., Model-fusion-based online glucose concentration predictions in people with type 1 diabetes, Control Engineering Practice 71 (2018) 129–141. doi:https://doi.org/10.1016/j.conengprac.2017.10.013.

[19] P. Angelov, D. Filev, Simpl_ets: A simplified method for learning evolving takagi-sugeno fuzzy models, in: The 14th IEEE International Conference on Fuzzy Systems, 2005. FUZZ'05., IEEE, 2005, pp. 1068–1073. doi:https://doi.org/10.1109/fuzzy.2005.1452543.

[20] P. Angelov, X. Zhou, Evolving fuzzy systems from data streams in real-time, in: 2006 International Symposium on Evolving Fuzzy Systems, IEEE, 2006, pp. 29–35. doi:https://doi.org/10.1109/isefs.2006.251157.

[21] P. Angelov, Evolving Takagi-Sugeno fuzzy systems from streaming data (eTS+), in: Evolving Intelligent Systems: Methodology and Applications, Vol. 12, Wiley Online Library, 2010, p. 21. doi:https://doi.org/10.1002/9780470569962.ch2.

[22] E. Lima, M. Hell, R. Ballini, F. Gomide, Evolving fuzzy modeling using participatory learning, in: Evolving Intelligent Systems: Methodology and Applications, Wiley Online Library, 2010, pp. 67–86. doi:https://doi.org/10.1002/9780470569962.ch4.

[23] L. Maciel, F. Gomide, R. Ballini, An enhanced approach for evolving participatory learning fuzzy modeling, in: 2012 IEEE Conference on Evolving and Adaptive Intelligent Systems, IEEE, 2012, pp. 23–28. doi:https://doi.org/10.1109/eais.2012.6232799.

[24] L. Souza, A. P. Lemos, W. M. Caminhas, W. Boaventura, Thermal modeling of power transformers using evolving fuzzy systems, Engineering Applications of

Artificial Intelligence 25 (5) (2012) 980–988. doi:https://doi.org/10.1016/j.engappai.2011.12.007.

[25] S. Shafieezadeh-Abadeh, A. Kalhor, Evolving Takagi-Sugeno model based on online Gustafson-Kessel algorithm and kernel recursive least square method, in: 2014 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS), IEEE, 2014, pp. 1–8. doi:https://doi.org/10.1109/eais.2014.6867467.

[26] S. Shafieezadeh-Abadeh, A. Kalhor, Evolving Takagi–Sugeno model based on online Gustafson-Kessel algorithm and kernel recursive least square method, Evolving Systems 7 (1) (2016) 1–14. doi:https://doi.org/10.1007/s12530-015-9129-1.

[27] L. Maciel, R. Vieira, A. Porto, F. Gomide, R. Ballini, Evolving participatory learning fuzzy modeling for financial interval time series forecasting, in: 2017 Evolving and Adaptive Intelligent Systems (EAIS), IEEE, 2017, pp. 1–8. doi:https://doi.org/10.1109/eais.2017.7954826.

[28] J. Li, G. Wen, J. Gan, L. Zhang, S. Zhang, Sparse nonlinear feature selection algorithm via local structure learning, Emerging Science Journal 3 (2) (2019) 115–129. doi:https://doi.org/10.28991/esj-2019-01175.

[29] Y. Engel, S. Mannor, R. Meir, The kernel recursive least-squares algorithm, IEEE Transactions on Signal Processing 52 (8) (2004) 2275–2285. doi:https://doi.org/10.1109/tsp.2004.830985.

[30] R. Vieira, L. Maciel, R. Ballini, F. Gomide, Stock market price forecasting using a kernel participatory learning fuzzy model, in: North American Fuzzy Information Processing Society Annual Conference, Springer, 2018, pp. 361–373. doi:https://doi.org/10.1007/978-3-319-95312-0_31.

[31] H. Gunduz, Y. Yaslan, Z. Cataltepe, Intraday prediction of Borsa Istanbul using convolutional neural networks and feature correlations, Knowledge-Based Systems 137 (2017) 138–148. doi:http://dx.doi.org/10.1016/j.knosys.2017.09.023.

[32] D. Ge, X.-J. Zeng, A self-evolving fuzzy system which learns dynamic threshold parameter by itself, IEEE Transactions on Fuzzy Systems 27 (8) (2018) 1625–1637. doi:https://doi.org/10.1109/tfuzz.2018.2886154.

[33] D. Ge, X.-J. Zeng, Learning evolving T-S fuzzy systems with both local and global accuracy – a local online optimization approach, Applied Soft Computing 68 (2018) 795–810. doi:https://doi.org/10.1016/j.asoc.2017.05.046.

[34] R. Sun, T. Ma, S. Liu, Portfolio selection based on semivariance and distance correlation under minimum variance framework, Statistica Neerlandica 73 (3) (2019) 373–394. doi:https://doi.org/10.1111/stan.12174.

[35] G. Kasieczka, D. Shih, Robust jet classifiers through distance correlation, Physical Review Letters 125 (12) (2020) 122001. doi:https://doi.org/10.1103/PhysRevLett.125.122001.

[36] S. Chormunge, S. Jena, Correlation based feature selection with clustering for high dimensional data, Journal of Electrical Systems and Information Technology 5 (3) (2018) 542–549. doi:https://doi.org/10.1016/j.jesit.2017.06.004.

[37] S. Zahra, M. A. Ghazanfar, A. Khalid, M. A. Azam, U. Naeem, A. Prugel-Bennett, Novel centroid selection approaches for KMeans-clustering based recommender systems, Information Sciences 320 (2015) 156–189. doi:http://dx.doi.org/10.1016/j.ins.2015.03.062.

[38] M. Mikha-Beyranvand, J. Faiz, B. Rezaeealam, A. Rezaei-Zare, M. Jafarboland, Thermal analysis of power transformers under unbalanced supply voltage, IET Electric Power Applications 13 (4) (2019) 503–512. doi:https://doi.org/10.1049/iet-epa.2018.5799.

[39] Z. Radakovic, K. Feser, A new method for the calculation of the hot-spot temperature in power transformers with ONAN cooling, IEEE Transactions on Power Delivery 18 (4) (2003) 1284–1292. doi:https://doi.org/10.1109/tpwrd.2003.817740.

[40] Q. Zheng, S. Jiaode, Y. Kedao, S. Diansheng, On-line monitoring of hot-spot temperature in transformer winding, High Voltage Engineering 29 (9) (2003) 26–29.

[41] IEEE guide for loading mineral-oil-immersed transformers and step-voltage regulators, IEEE Std C57.91-2011 (Revision of IEEE Std C57.91-1995) (2012) 1–123. doi:https://doi.org/10.1109/ieeestd.2012.6166928.

[42] P. Daponte, D. Grimaldi, A. Piccolo, D. Villacci, A neural diagnostic system for the monitoring of transformer heating, Measurement 18 (1) (1996) 35–46. doi:https://doi.org/10.1016/0263-2241(96)00043-7.

[43] M. V. G. da Rocha, M. B. Hell, K. S. T. R. Alves, F. L. C. Oliveira, E. P. de Aguiar, Power transformers thermal modeling using an enhanced set-membership multivariable gaussian evolving fuzzy system, Electric Power Systems Research 194 (2021) 107088. doi:https://doi.org/10.1016/j.epsr.2021.107088.

[44] V. Galdi, L. Ippolito, A. Piccolo, A. Vaccaro, Neural diagnostic system for transformer thermal overload protection, IEE Proceedings - Electric Power Applications 147 (5) (2000) 415–421. doi:https://doi.org/10.1049/ip-epa:20000519.

[45] L. Ippolito, An adaptive fuzzy approach to predictive overload protection systems for power transformers, Automatika: časopis za automatiku, mjerenje, elektroniku, računarstvo i komunikacije 45 (3-4) (2004) 169–178.

[46] M. Hell, P. Costa, F. Gomide, Recurrent neurofuzzy network in thermal modeling of power transformers, IEEE Transactions on Power Delivery 22 (2) (2007) 904–910. doi:https://doi.org/10.1109/tpwrd.2006.874613.

[47] M. Hell, P. Costa, F. Gomide, Participatory learning in power transformers thermal modeling, IEEE Transactions on Power Delivery 23 (4) (2008) 2058–2067. doi:https://doi.org/10.1109/tpwrd.2008.923994.

[48] B. Fritzke, Growing cell structures—A self-organizing network for unsupervised and supervised learning, Neural Networks 7 (9) (1994) 1441–1460. doi:https://doi.org/10.1016/0893-6080(94)90091-4.

[49] J. R. Williamson, Gaussian ARTMAP: A neural network for fast incremental learning of noisy multidimensional maps, Neural Networks 9 (5) (1996) 881–897. doi:https://doi.org/10.1016/0893-6080(95)00115-8.

[50] T.-Y. Kwok, D.-Y. Yeung, Constructive algorithms for structure learning in feedforward neural networks for regression problems, IEEE Transactions on Neural Networks 8 (3) (1997) 630–645. doi:https://doi.org/10.1109/72.572102.

[51] C.-F. Juang, C.-T. Lin, An online self-constructing neural fuzzy inference network and its applications, IEEE Transactions on Fuzzy Systems 6 (1) (1998) 12–32. doi:https://doi.org/10.1109/91.660805.

[52] M. Prasad, C.-T. Lin, D.-L. Li, C.-T. Hong, W.-P. Ding, J.-Y. Chang, Soft-boosted self-constructing neural fuzzy inference network, IEEE Transactions on Systems, Man, and Cybernetics: Systems 47 (3) (2015) 584–588. doi:https://doi.org/10.1109/tsmc.2015.2507139.

[53] C.-F. Juang, C.-T. Lin, A recurrent self-organizing neural fuzzy inference network, in: Proceedings of 6th International Fuzzy Systems Conference, Vol. 3, IEEE, 1997, pp. 1369–1374. doi:https://doi.org/10.1109/fuzzy.1997.619743.

[54] C.-F. Juang, C.-T. Lin, A recurrent self-organizing neural fuzzy inference network, IEEE Transactions on Neural Networks 10 (4) (1999) 828–845. doi:https://doi.org/10.1109/72.774232.

[55] N. Kasabov, Evolving fuzzy neural networks for supervised/unsupervised online knowledge-based learning, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 31 (6) (2001) 902–918. doi:https://doi.org/10.1109/3477.969494.

[56] S. Heddam, Fuzzy neural network (EFuNN) for modelling dissolved oxygen concentration (DO), in: Intelligence Systems in Environmental Management: Theory and Applications, Springer, 2017, pp. 231–253. doi:https://doi.org/10.1007/978-3-319-42993-9_11.

[57] N. K. Kasabov, Q. Song, DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction, IEEE Transactions on Fuzzy Systems 10 (2) (2002) 144–154. doi:https://doi.org/10.1109/91.995117.

[58] S. Heddam, Modelling hourly dissolved oxygen concentration (DO) using dynamic evolving neural-fuzzy inference system (DENFIS)-based approach: case study of Klamath River at Miller Island Boat Ramp, OR, USA, Environmental Science and Pollution Research 21 (15) (2014) 9212–9227. doi:https://doi.org/10.1007/s11356-014-2842-7.

[59] J. de Jesús Rubio, SOFMLS: online self-organizing fuzzy modified least-squares network, IEEE Transactions on Fuzzy Systems 17 (6) (2009) 1296–1309. doi:https://doi.org/10.1109/tfuzz.2009.2029569.

[60] J. de Jesús Rubio, Evolving intelligent algorithms for the modelling of brain and eye signals, Applied Soft Computing 14 (2014) 259–268. doi:https://doi.org/10.1016/j.asoc.2013.07.023.

[61] H. Soleimani-B, C. Lucas, B. N. Araabi, Recursive Gath–Geva clustering as a basis for evolving neuro-fuzzy modeling, Evolving Systems 1 (1) (2010) 59–71. doi:https://doi.org/10.1007/s12530-010-9006-x.

[62] H. Soleimani-B, C. Lucas, B. N. Araabi, Recursive Gath-Geva clustering as a basis for evolving neuro-fuzzy modeling, in: International Conference on Fuzzy Systems, IEEE, 2010, pp. 1–7. doi:https://doi.org/10.1109/fuzzy.2010.5584088.

[63] D. Leite, P. Costa, F. Gomide, Evolving granular neural networks from fuzzy data streams, Neural Networks 38 (2013) 1–16. doi:https://doi.org/10.1016/j.neunet.2012.10.006.

[64] M. Pratama, S. G. Anavatti, P. P. Angelov, E. Lughofer, PANFIS: A novel incremental learning machine, IEEE Transactions on Neural Networks and Learning Systems 25 (1) (2013) 55–68. doi:https://doi.org/10.1109/tnnls.2013.2271933.

[65] W. Caesarendra, M. Pratama, B. Kosasih, T. Tjahjowidodo, A. Glowacz, Parsimonious network based on a fuzzy inference system (PANFIS) for time series feature prediction of low speed slew bearing prognosis, Applied Sciences 8 (12) (2018) 2656. doi:https://doi.org/10.3390/app8122656.

[66] M. Pratama, S. G. Anavatti, E. Lughofer, GENEFIS: toward an effective localist network, IEEE Transactions on Fuzzy Systems 22 (3) (2013) 547–562. doi:https://doi.org/10.1109/tfuzz.2013.2264938.

[67] M. Pratama, S. G. Anavatti, J. Lu, Recurrent classifier based on an incremental metacognitive-based scaffolding algorithm, IEEE Transactions on Fuzzy Systems 23 (6) (2015) 2048–2066. doi:https://doi.org/10.1109/tfuzz.2015.2402683.

[68] M. Pratama, J. Lu, S. Anavatti, E. Lughofer, C.-P. Lim, An incremental meta-cognitive-based scaffolding fuzzy neural network, Neurocomputing 171 (2016) 89–105. doi:https://doi.org/10.1016/j.neucom.2015.06.022.

[69] S. Silva, P. Costa, M. Santana, D. Leite, Evolving neuro-fuzzy network for real-time high impedance fault detection and classification, Neural Computing and Applications 32 (12) (2020) 7597–7610. doi:https://doi.org/10.1007/s00521-018-3789-2.

[70] H. Zhang, A. Zhou, X. Lin, Interpretable policy derivation for reinforcement learning based on evolutionary feature synthesis, Complex & Intelligent Systems 6 (3) (2020) 741–753. doi:https://doi.org/10.1007/s40747-020-00175-y.

[71] P. Angelov, R. Buswell, Evolving rule-based models: A tool for intelligent adaptation, in: Proceedings Joint 9th IFSA World Congress and 20th NAFIPS International Conference (Cat. No. 01TH8569), Vol. 2, IEEE, 2001, pp. 1062–1067. doi:https://doi.org/10.1109/nafips.2001.944752.

[72] D. Leite, I. Škrjanc, F. Gomide, An overview on evolving systems and learning from stream data, Evolving Systems 11 (2) (2020) 181–198. doi:https://doi.org/10.1007/s12530-020-09334-5.

[73] H.-J. Rong, N. Sundararajan, G.-B. Huang, P. Saratchandran, Sequential Adaptive Fuzzy Inference System (SAFIS) for nonlinear system identification and prediction, Fuzzy Sets and Systems 157 (9) (2006) 1260–1275. doi:https://doi.org/10.1016/j.fss.2005.12.011.

[74] H.-J. Rong, N. Sundararajan, G.-B. Huang, G.-S. Zhao, Extended sequential adaptive fuzzy inference system for classification problems, Evolving Systems 2 (2) (2011) 71–82. doi:https://doi.org/10.1007/s12530-010-9023-9.

[75] J. de Jesús Rubio, A. Bouchachia, MSAFIS: an evolving fuzzy inference system, Soft Computing 21 (9) (2017) 2357–2366. doi:https://doi.org/10.1007/s00500-015-1946-4.

[76] E. D. Lughofer, FLEXFIS: A robust incremental learning approach for evolving takagi–sugeno fuzzy models, IEEE Transactions on Fuzzy Systems 16 (6) (2008) 1393–1410. doi:https://doi.org/10.1109/tfuzz.2008.925908.

[77] B. Trawiński, K. Trawiński, E. Lughofer, T. Lasota, Investigation of evolving fuzzy systems methods FLEXFIS and eTS on predicting residential prices, in: Fuzzy Logic and Applications, Springer, 2011, pp. 123–130. doi:https://doi.org/10.1007/978-3-642-23713-3_16.

[78] E. Lughofer, J.-L. Bouchot, A. Shaker, On-line elimination of local redundancies in evolving fuzzy systems, Evolving Systems 2 (3) (2011) 165–187. doi:https://doi.org/10.1007/s12530-011-9032-3.

[79] E. Lughofer, S. Kindermann, SparseFIS: Data-driven learning of fuzzy systems with sparsity constraints, IEEE Transactions on Fuzzy Systems 18 (2) (2010) 396–411. doi:https://doi.org/10.1109/tfuzz.2010.2042960.

[80] F. Serdio, E. Lughofer, K. Pichler, T. Buchegger, H. Efendic, Data-driven residual-based fault detection for condition monitoring in rolling mills, IFAC Proceedings Volumes 46 (9) (2013) 1530–1535. doi:https://doi.org/10.3182/20130619-3-ru-3018.00403.

[81] P. Angelov, R. Yager, A new type of simplified fuzzy rule-based system, International Journal of General Systems 41 (2) (2012) 163–185. doi:https://doi.org/10.1080/03081079.2011.634807.

[82] G. Andonovski, G. Mušič, S. Blažič, I. Škrjanc, On-line evolving cloud-based model identification for production control, IFAC-PapersOnLine 49 (5) (2016) 79–84. doi:https://doi.org/10.1016/j.ifacol.2016.07.093.

[83] D. Dovžan, V. Logar, I. Škrjanc, Implementation of an evolving fuzzy model (eFuMo) in a monitoring system for a waste-water treatment process, IEEE Transactions on Fuzzy Systems 23 (5) (2014) 1761–1776. doi:https://doi.org/10.1109/tfuzz.2014.2379252.

[84] L. Maciel, F. Gomide, R. Ballini, Enhanced evolving participatory learning fuzzy modeling: an application for asset returns volatility forecasting, Evolving Systems 5 (2) (2014) 75–88. doi:https://doi.org/10.1007/s12530-013-9099-0.

[85] E. Lughofer, C. Cernuda, S. Kindermann, M. Pratama, Generalized smart evolving fuzzy systems, Evolving Systems 6 (4) (2015) 269–292. doi:https://doi.org/10.1007/s12530-015-9132-6.

[86] I. Škrjanc, D. Dovžan, Evolving Gustafson-Kessel possibilistic c-means clustering, Procedia Computer Science 53 (2015) 191–198. doi:https://doi.org/10.1016/j.procs.2015.07.294.

[87] L. Maciel, R. Ballini, F. Gomide, Evolving possibilistic fuzzy modeling for realized volatility forecasting with jumps, IEEE Transactions on Fuzzy Systems 25 (2) (2016) 302–314. doi:https://doi.org/10.1109/tfuzz.2016.2578338.

[88] D. Leite, G. Andonovski, I. Škrjanc, F. Gomide, Optimal rule-based granular systems from data streams, IEEE Transactions on Fuzzy Systems 28 (3) (2019) 583–596. doi:https://doi.org/10.1109/tfuzz.2019.2911493.

[89] A. Lemos, W. Caminhas, F. Gomide, Fuzzy evolving linear regression trees, Evolving Systems 2 (1) (2011) 1–14. doi:https://doi.org/10.1007/s12530-011-9028-z.

[90] A. Hapfelmeier, B. Pfahringer, S. Kramer, Pruning incremental linear model trees with approximate lookahead, IEEE Transactions on Knowledge and Data Engineering 26 (8) (2013) 2072–2076. doi:https://doi.org/10.1109/tkde.2013.132.

[91] T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, IEEE Transactions on Systems, Man, and Cybernetics: Systems SMC-15 (1) (1985) 116–132. doi:https://doi.org/10.1109/tsmc.1985.6313399.

[92] M. Sugeno, T. Takagi, Fuzzy identification of systems and its applications to modelling and control, Readings in Fuzzy Sets for Intelligent Systems 15 (1) (1993) 387–403. doi:https://doi.org/10.1016/b978-1-4832-1450-4.50045-6.

[93] R. R. Yager, A model of participatory learning, IEEE Transactions on Systems, Man, and Cybernetics: Systems 20 (5) (1990) 1229–1234. doi:https://doi.org/10.1109/21.59986.

[94] I. Škrjanc, J. A. Iglesias, A. Sanchis, D. Leite, E. Lughofer, F. Gomide, Evolving fuzzy and neuro-fuzzy approaches in clustering, regression, identification, and classification: A survey, Information Sciences 490 (2019) 344–368. doi:https://doi.org/10.1016/j.ins.2019.03.060.

[95] R. Arora, H. Garg, A robust correlation coefficient measure of dual hesitant fuzzy soft sets and their application in decision making, Engineering Applications of Artificial Intelligence 72 (2018) 80–92. doi:https://doi.org/10.1016/j.engappai.2018.03.019.

[96] K. Hemming, M. Taljaard, A. Forbes, Analysis of cluster randomised stepped wedge trials with repeated cross-sectional samples, Trials 18 (1) (2017) 101. doi:https://doi.org/10.1186/s13063-017-1833-7.

[97] R. Taylor, Interpretation of the correlation coefficient: a basic review, Journal of Diagnostic Medical Sonography 6 (1) (1990) 35–39. doi:https://doi.org/10.1177/875647939000600106.

[98] J. Fürnkranz, Pruning algorithms for rule learning, Machine Learning 27 (2) (1997) 139–172. doi:https://doi.org/10.1023/A:1007329424533.

[99] C. Richard, J. C. M. Bermudez, P. Honeine, Online prediction of time series data with kernels, IEEE Transactions on Signal Processing 57 (3) (2008) 1058–1067. doi:https://doi.org/10.1109/tsp.2008.2009895.

[100] W. Liu, J. C. Principe, S. Haykin, Kernel adaptive filtering: a comprehensive introduction, John Wiley & Sons, 2010. doi:https://doi.org/10.1002/9780470608593.

[101] B. Scholkopf, A. J. Smola, Learning with kernels: support vector machines, regularization, optimization, and beyond, The MIT Press, 2018. doi:https://doi.org/10.7551/mitpress/4175.001.0001.

[102] C. Soares, P. B. Brazdil, P. Kuba, A meta-learning method to select the kernel width in support vector regression, Machine Learning 54 (3) (2004) 195–209. doi:https://doi.org/10.1023/b:mach.0000015879.28004.9b.

[103] H. Fan, Q. Song, S. B. Shrestha, Kernel online learning with adaptive kernel width, Neurocomputing 175 (2016) 233–242. doi:https://doi.org/10.1016/j.neucom.2015.10.055.

[104] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, Introduction to algorithms, MIT Press, 2009.

[105] P. S. Oliveto, X. Yao, Runtime analysis of evolutionary algorithms for discrete optimization, in: Theory of Randomized Search Heuristics: Foundations and Recent Developments, World Scientific, 2011, pp. 21–52. doi:https://doi.org/10.1142/9789814282673_0002.

[106] G. E. Box, G. M. Jenkins, G. C. Reinsel, G. M. Ljung, Time series analysis: forecasting and control, John Wiley & Sons, 2015.

[107] J.-S. Jang, ANFIS: adaptive-network-based fuzzy inference system, IEEE Transactions on Systems, Man, and Cybernetics 23 (3) (1993) 665–685. doi:https://doi.org/10.1109/21.256541.

[108] F. Rosenblatt, Principles of neurodynamics. perceptrons and the theory of brain mechanisms, Tech. rep., Cornell Aeronautical Lab Inc Buffalo NY (1961).

[109] V. Vapnik, The support vector method of function estimation, in: Nonlinear Modeling, Springer, 1998, pp. 55–85. doi:https://doi.org/10.1007/978-1-4615-5703-6_3.

[110] M. C. Mackey, L. Glass, Oscillation and chaos in physiological control systems, Science 197 (4300) (1977) 287–289. doi:https://doi.org/10.1126/science.267326.

[111] D. Harvey, S. Leybourne, P. Newbold, Testing the equality of prediction mean squared errors, International Journal of Forecasting 13 (2) (1997) 281–291. doi:https://doi.org/10.1016/S0169-2070(96)00719-4.

[112] F. Diebold, R. Mariano, Comparing predictive accuracy, Tech. rep. (nov 1994). doi:https://doi.org/10.3386/t0169.

[113] L. Wang, J. Yen, Extracting fuzzy rules for system modeling using a hybrid of genetic algorithms and kalman filter, Fuzzy Sets and Systems 101 (3) (1999) 353–362. doi:https://doi.org/10.1016/s0165-0114(97)00098-5.

[114] L.-X. Wang, J. M. Mendel, Back-propagation fuzzy system as nonlinear dynamic system identifiers, in: [1992 Proceedings] IEEE International Conference on Fuzzy Systems, IEEE, 1992, pp. 1409–1418. doi:https://doi.org/10.1109/fuzzy.1992.258711.

[115] D. A. Dickey, W. A. Fuller, Likelihood ratio statistics for autoregressive time series with a unit root, Econometrica: Journal of the Econometric Society (1981) 1057–1072 doi:https://doi.org/10.2307/1912517.

[116] R. S. Tsay, Nonlinearity tests for time series, Biometrika 73 (2) (1986) 461–466. doi:https://doi.org/10.1093/biomet/73.2.461.

APPENDIX A – PUBLICATIONS

This Appendix presents the list of publications.

The list of papers published in journals during the master period are as follows:

- **K. S. T. R. Alves**, E. P. de Aguiar, A novel rule-based evolving Fuzzy System applied to the thermal modeling of power transformers, Applied Soft Computing 112 (2021) 107764. doi: https://doi.org/10.1016/j.asoc.2021.107764.
- M. V. G. da Rocha, M. B. Hell, **K. S. T. R. Alves**, F. L. C. Oliveira, E. P. de Aguiar, Power transformers thermal modeling using an enhanced set-membership multivariable gaussian evolving fuzzy system, Electric Power Systems Research 194 (2021) 107088. doi: https://doi.org/10.1016/j.epsr.2021.107088.
- E. R. C. Queiroz, **K. S. T. R. Alves**, F. L. C. Oliveira, E. P. de Aguiar, Variable step-size evolving participatory learning with kernel recursive least squares applied to gas prices forecasting in Brazil, Evolving Systems (2021) 1–10. doi: https://doi.org/10.1007/s12530-021-09388-z.
- M. V. G. da Rocha, **K. S. T. R. Alves**, E. R. C. Queiroz, F. L. C. Oliveira, M. B. Hell, E. P. de Aguiar, Power Transformers Thermal Modeling Based on the Modified Set-Membership evolving Multivariable Gaussian and Variable Step-Size evolving Multivariable Gaussian, Journal of Control, Automation and Electrical Systems. *Under review.*
- **K. S. T. R. Alves**, M. Hell, F. L. C. Oliveira, E. P. de Aguiar, An enhanced set-membership evolving participatory learning with kernel recursive least squares applied to thermal modeling of power transformers, Electric Power Systems Research 184 (2020) 106334. doi: http://doi.org/10.1016/j.epsr.2020.106334.

The list of conference papers published during the master period are as follows:

- E. R. C. Queiroz, **K. S. T. R. Alves**, F. L. C. Oliveira, E. P. de Aguiar, Evolving fuzzy modeling with variable step-size applied to gas prices forecasting in Brazil, in: 2021 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), IEEE, 2021, pp. 1–6. doi: http://doi.org/10.1109/FUZZ45933.2021.9494405.
- M. V. da Rocha, **K. S. T. R. Alves**, M. B. Hell, F. L. Oliveira, E. P. de Aguiar, Modelagem térmica de transformadores de potência baseada em um sistema fuzzy evolutivo set-membership gaussiano multivariado, in: Congresso Brasileiro de Automática-CBA, Vol. 2, 2020. doi: http://doi.org/10.48011/asba.v2i1.1024

- E. R. C. Queiroz, **K. S. T. R. Alves**, F. L. C. Oliveira, E. P. de Aguiar, Modelagem nebulosa evolutiva participativa com passo de adaptaçao variável aplicada a previsao de preços do óleo diesel no Brasil, in: Congresso Brasileiro de Automática-CBA, Vol. 2, 2020. doi: http://doi.org/10.48011/asba.v2i1.1039
- D. P. de Souza, E. P. de Aguiar, **K. S. T. R. Alves**, M. V. da Rocha, T. E. Fernandes, M. B. Hell, F. L. Oliveira, E. d. S. Christo, Modelagem térmica de transformadores de potência baseada em soda e sistema de inferência fuzzy otimizado por enxame de partículas, in: Congresso Brasileiro de Automática-CBA, Vol. 2, 2020. doi: http://doi.org/10.48011/asba.v2i1.1542.