



Article An Evolving Fuzzy Neural Network Based on Or-Type Logic Neurons for Identifying and Extracting Knowledge in Auction Fraud⁺

Paulo Vitor de Campos Souza^{1,*}, Edwin Lughofer¹, Huoston Rodrigues Batista²

- ¹ Institute for Mathematical Methods in Medicine and Data Based Modeling, Johannes Kepler University Linz, 4040 Linz, Austria
- ² School of Informatics, Communication and Media, University of Applied Sciences Upper Austria Hagenberg, 4232 Hagenberg im Mühlkreis, Austria
- ³ Specialization of Artificial Intelligence and Machine Learning, Pontifical Catholic University of Minas Gerais, Belo Horizonte 30535-901, Minas Gerais, Brazil
- * Correspondence: paulo.de_campos_souza@jku.at
- + This paper is an extended version of our paper published in 19th World Congress of the International Fuzzy Systems Association and the 12th Conference of the European Society for Fuzzy Logic and Technology, IFSA-EUSFLAT 2021, Bratislava, Slovakia, 19–24 September 2021; pp. 314–321.

Abstract: The rise in online transactions for purchasing goods and services can benefit the parties involved. However, it also creates uncertainty and the possibility of fraud-related threats. This work aims to explore and extract knowledge of auction fraud by using an innovative evolving fuzzy neural network model based on logic neurons. This model uses a fuzzification technique based on empirical data analysis operators in an evolving way for stream samples. In order to evaluate the applied model, state-of-the-art neuro-fuzzy models were used to compare a public dataset on the topic and, simultaneously, validate the interpretability results based on a common criterion to identify the correct patterns present in the dataset. The fuzzy rules and the interpretability criteria demonstrate the model's ability to extract knowledge. The results of the model proposed in this paper are superior to the other models evaluated (close to 98.50% accuracy) in the test.

Keywords: evolving fuzzy neural network; or-neuron; auction fraud; knowledge extraction

MSC: 68T35

1. Introduction

Intelligent machine-learning-based models have assumed a prominent role in solving complex problems common in modern society. Intelligent methods can dynamically solve problems and optimize industry processes, disease diagnoses, or actions related to Internet transactions [1].

With the advent of the Internet in people's lives, and the popularization of e-commerce, a current problem is online shopping fraud. Some websites promote user interaction based on bids placed on a product or service. This online auction encourages users to buy products through bids that can be made remotely, according to the standards set by the platforms that offer this service. However, purchases involving high-value goods or rare items often suffer from fraud attempts. In general, online auction fraud has specific characteristics, ranging from products of dubious origin, suspicious and repetitive bidding on a particular item, overvaluation of a product through constant bidding, and users registering on the platform to cause trouble [2]. These problems can make the experience of buying a product at an online auction appalling, preventing new users from opting for this form of purchase and discrediting the platform that offers such services [3]. Considering the actuality of the



Citation: de Campos Souza, P.V.; Lughofer, E.; Rodrigues Batista, H.; Junio Guimaraes, A. An Evolving Fuzzy Neural Network Based on Or-Type Logic Neurons for Identifying and Extracting Knowledge in Auction Fraud. *Mathematics* 2022, *10*, 3872. https:// doi.org/10.3390/math10203872

Academic Editors: Antonin Dvorak and Vilém Novák

Received: 20 September 2022 Accepted: 14 October 2022 Published: 18 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). problem, some researchers have built a database of factors that exist in auctions to identify fraud during transactions to prevent such problems from occurring [4]. These studies support significant areas of current research, and researchers worldwide are working to identify corresponding patterns and propose solutions, including approaches that use clustering and machine learning [5].

Machine-learning-based models can act to assist in the detection of fraud in online auctions. In particular, a model that can combine the efficiency of training artificial neural networks and interpreting fuzzy systems can solve this problem and extract knowledge of auction fraud through linguistic rules about the problem [6]. Fuzzy neural networks can process input data in fuzzy sets and, using neural network training, provide answers to problems with a high degree of interpretability, because this model is comprised of a fuzzy inference system that can represent data in a linguistic and interpretable way using fuzzy operations [7].

When accompanied by logical interpretability about the functioning of the models that act in this solution, this problem solving can facilitate humans' understanding and acceptance of results. Fuzzy natural logic (FNL) is a formal theory that aims to provide a mathematical model of natural human reasoning whose typical feature is using natural language. Its central basis is based on the results of linguistics, artificial intelligence, and logical analysis of natural language [8].

This paper proposes a new evolving fuzzy neural network architecture based on an empirical data analysis fuzzification process that applies the concepts of self-organized direction-aware data partitioning [9]. Logic neurons based on t-conorm (or-neurons) are in the model's second layer. The model's training is founded on Extreme Learning Machine [10] in the offline phase, and the evolving phase is established on indicator-based Recursive Weighted Least Squares [11]. The three-layer model has a fuzzy inference system connected to a neural aggregation network represented in its third (and last) layers. This combination of factors allows the third layer of the model to provide the network's output (defuzzification process) with answers about frauds in online auctions and fuzzy rules that express the logical and interpretable relations present in the evaluated dataset. The model also includes sophisticated capabilities for the behavior's interpretability throughout its analyses, demonstrating tendencies important to interpretability and information extraction. In addition, this paper proposes a presentation of knowledge about a dataset of fraud in auctions with IF–THEN rules, presenting the logical relationships between the dimensions of the problems and the possible impacts of discovering fraudulent behavior. The behavior of the weights of each dimension linked to the fuzzy sets generated gives an interpretative dimension to the issues present in identifying fraud.

The motivation for applying evolving fuzzy neural networks in dynamic behavior problem areas (such as auction fraud) is connected to the dynamic ability of these types of approaches to identify new behaviors from data, incorporating new neural structures into their architecture. Auction fraud problems have an extremely dynamic behavior, as the types of fraud are constantly changing (either by creating new techniques or when previous frauds are discovered). This evolution of knowledge on the models' components helps solve complex problems that can dynamically change over time. Stream data assessments require models to obtain assertive and adaptive outputs. Thus, evolving fuzzy neural networks act this way, with the advantage of demonstrating the evolution of knowledge and identifying behaviors of the elements of their fuzzy rules (antecedents and consequents).

This paper's main highlight is connected with an assertive and interpretable approach to the problem of fraud in auctions. Relations on validating interpretability criteria and generating fuzzy rules are presented to the reader. Another evolving fuzzy neural network model has already been used to solve these demands [12]. However, this paper presents the first neuro-fuzzy model to solve problems and extract advantageous interpretations of the dataset.

The paper is organized as follows: In Section 2, the literature supplies details about evolving fuzzy neural networks, interpretability of evolving fuzzy systems, and auction

fraud. Section 3 presents the model employed in this paper, highlighting its layers, training method, and interpretability criteria applied in the model's results. Section 4 presents the test features, the state-of-the-art model to reveal auction fraud, and the results acquired with due arguments and interpretation. Finally, Section 6 highlights the paper's conclusions and possible future work.

2. Literature Review

2.1. Artificial Neural Network and Applications

Artificial neural networks (ANN) are mathematical models that can learn through computational means and are based on human networks of biological neurons. These networks process information using storage and processing structures, with each processor element representing a neuron. They are fed a set of weighted inputs, and their output is obtained by applying an activation function to simulate the degree of stimulation of the biological neuron in order to obtain the responses [13]. According to Fausett and others [14], ANN is a set of computational procedures that provide a mathematical representation inspired by the neural composition of intelligent organisms and that gain knowledge through expertise, allowing tasks typically performed by reasonable beings to be performed in computational environments.

Neural networks are constructed of many interconnected processing units known as neurons. This method is based on the nature of the human brain and how learning and problem solving work. Weighted bonds called synapses connect artificial neurons. Each neuron receives multiple input signals and generates an output signal [13].

Since the 1980s, studies on neural networks have undergone significant evolution, and this area of research has become well-known, either because of the promising features introduced by neural network models or because of the general technological requirement of implementation that allows the audacious development of applications of architectures with parallel neural networks in dedicated hardware, thus acquiring excellent fulfillment of these systems (foremost to conventional methods). Deep neural networks represent the state of the art in neural network development (or Deep Learning) [15]. Where the behavior of variables is not purely known, ANNs are frequently employed to solve complicated issues. Their capacity to learn from examples and generalize the knowledge acquired, producing a nonlinear model, is one of their key properties, which greatly increases the effectiveness of their use in spatial analysis [13].

The selection of the optimum architecture is one of the most challenging aspects of using neural networks since it is experimental and takes a long time to complete. The procedure must be used in practice to evaluate multiple learning models as well as the many topology configurations a network might have to address a particular issue [13].

In two different ways, a neural network approximates the human brain: information is obtained through knowledge stages, and synaptic weights are employed to gather experience. The existing connection between neurons is referred to as a synapse. Values called synaptic weights are assigned to the connections. This demonstrates that artificial neural networks are made up of a succession of artificial (or virtual) neurons that are coupled to form a system of processing components. Their acknowledgments are determined by how the model's underlying structures are linked, and their parameters are predicted [13].

At least one hidden layer is required in ANNs. If the model has multiple hidden layers, each one is accountable for increasing its capacity. The ANN architecture is capable of executing activities that aid in producing model outputs. The neurons in the hidden layers can be activated using activation functions (e.g., Gaussian, triangular, sigmoidal, sine, and hyperbolic tangent). These activation functions are in charge of introducing the nonlinear aspect of the model's activities. Backpropagation or Gradient Descent [16], Extreme Learning Machine [10], and others are examples of training methods for updating model parameters. An example of an artificial neural network is shown in Figure 1.



Figure 1. Artificial neural network with five layers.

2.2. Fuzzy Neural Network

Fuzzy neural networks are based on the theory of fuzzy sets and are trained by learning algorithms derived from neural networks. Being able to unify in a model the generalization of parameters and the training capacity provided by artificial neural networks together with the interpretability characteristics of fuzzy systems [17], fuzzy neural networks can work to solve problems of different varieties, making it easier to find solutions for a parameter update in the smart models. By interpreting aspects of the user's database of the problem, they identify existing patterns that make up a set of fuzzy rules capable of transforming numerical information into linguistic contexts that can be interpreted by people who do not necessarily know the main techniques of artificial intelligence. Incorporating fuzzy logic into the neural network can alleviate the insufficiency of each technique, approaching them as more economical, robust, and simple to understand problems.

These models have multilayer architectures, commonly, the three-layer ones stand out, but there may be differences in their architecture. Fuzzy neural networks have also been used for the detection of Pulmonary Tuberculosis [18]. They are also present in the detection of COVID-19 cases from chest X-rays [19]. Most of the models listed above have differences in their architecture. Each of the functions of these layers incorporates the concepts of fuzzification–defuzzification systems and artificial neural networks. The first layer is the one that receives the input data, converting them into fuzzy logic neurons. Fuzzy versions of c-means, adaptive-network-based fuzzy inference systems, and cloud clustering are usually implemented [7]. These techniques usually have at least one hidden layer. In the training algorithms, models based on backpropagation, genetic and evolutionary models, and Extreme Learning Machines stand out [7]. A model with an optimized structure determines its architecture from the fuzzy neural network and an output layer, which is also known as the defuzzification layer.

The architecture of a fuzzy neural network can be expressed in layers, where each one is responsible for a function in the model. Figure 2 represents a fuzzy neural networks in layers where one is responsible for the fuzzification process, the other takes the aggregation of fuzzy neurons, and the last layer is responsible for the model outputs. Recent proposals for fuzzy neural networks have addressed time-series forecasting [20,21], harmonic suppression [22], autonomous vehicles [23], and breast cancer [24].



Figure 2. Example of a layered fuzzy neural network.

2.3. Fuzzy Natural Logic

Fuzzy natural logic (FNL) is a class of mathematical theories that includes evaluative expression theory, theory of fuzzy/linguistic IF–THEN rules, and approximate reasoning based on them. The theory of generalized fuzzy quantifiers and their syllogisms, models of natural human reasoning, are also highlighted. FNL aims to develop models of human thought whose most relevant feature is the use of natural language [25].

Many works have been proposed to explore this theory to facilitate the understanding of the results by its users. We highlight works that addressed the theory of intermediate quantifiers in FNL [26], forecasting in analogous time series [27], integration of probabilistic uncertainty to fuzzy systems [28], and interpretable rules with medical data [29], among others.

2.4. Auction Fraud

Online markets present themselves as a very popular way to trade online. E-commerce's main feature is to unite the real and virtual market for the online shopping model, being able to be divided into several sales schemes [30]. The most common business models are Business to Business, which indicates a company that does business with other companies, Business to Consumer, which is the trade carried out directly between the producer, seller, or service provider and the final customer, and Consumer to Consumer, which is based on the direct relationship between consumer and consumer.

According to the Global Payments 2022 Report (https://worldpay.globalpaymentsrep ort.com/en (accessed on 12 August 2022)) published by FIS Worldpay that examines current and future payment trends in 40 countries across five regions, the global e-commerce market will grow 55% by 2025. Among the many user benefits, C2C offers minimal costs while maintaining higher margins for sellers and lower prices for buyers. There is also the convenience factor, and customers can list their products online and wait for buyers to come to them. A great example would be an auction.

Large companies promote online auctions, being the only intermediaries to correspond to consumers [30]. The most prominent example is eBay, a successful site since its launch in 1995. Anyone can sign up and start selling or buying. However, fraudsters use it to take advantage by finding alternatives to take advantage of. The leading illicit practices in online auctions are [31]:

- Advertising illegal (stolen) or black market goods;
- Fraud that happens during the bidding period, such as Shill Bidding;
- Post-auction fraud, such as the exaggerated collection of fees, insurance, and even the nondelivery of the purchased goods.

According to the FBI's 2019 Internet Crime Report (FBI's Internet Crime Complaint Center (IC3) in its 2019 Internet Crime Report (available in: https://pdf.ic3.gov/2019_IC3 Report.pdf (accessed on 15 August 2022))), there were 467,361 complaints in 2019—and more than USD 3.5 billion in losses for individual and commercial victims.

The practice of Shill Bidding, the focus of this paper, is when a seller uses a separate account, be it themselves, a friend, or a family member, or someone else, and asks them to bid on their auction to raise the auction price artificially. The item's price is higher than if a legitimate buyer placed a bid and purchased the item. Machine learning techniques have been working on solving fraud identification in auctions. Recent works proposed in [32–35] stand out.

2.5. Evolving Fuzzy Systems and Interpretability

The evolving fuzzy systems are models capable of working with advanced problem solving with a certain degree of interpretability. This problem-interpreting capacity comes from its structure, which can transform data into representations people can read and interpret. A range of models extracts knowledge through fuzzy rules, where each applies techniques and has different architectures for this purpose. The most famous examples of existing evolving fuzzy systems are evolving models based on Takagi–Sugeno, fuzzy classifiers, and fuzzy neural networks.

They differ in the type of fuzzy rules and possible interpretations of them. Several works in the evolving fuzzy system literature work on pattern classification, linear regression, and time-series forecasting (see examples in [36]). In particular, this work deals with interpretable aspects generated by an evolving fuzzy neural network.

Regarding aspects of evolution over time, several proposals were made to facilitate the understanding of the behavior of models as they analyze data. Lughofer [37] proposed criteria that ensure that the fuzzy rules generated can bring certainty to users regarding their actions during the execution of their activities. One should look for more straightforward and distinguishable models, that is, models with a smaller number of fuzzy rules solving the target problems in a coherent and interpretable way, avoiding ambiguities, and that each structure that composes the model avoids redundancy (overlapping of Gaussians may be an example of redundancy present in these models). Other criteria deal with overlapping by evaluating consequents and rule antecedents. The relationship between these two fuzzy rule components determines whether there are inconsistent rules (which can confuse evaluators). Moreover, the evaluation of completeness (a criterion that evaluates the contribution of rules with a significant distance to the sample) and coverage is noteworthy, which verifies whether all samples evaluated by the model are covered by the space of characteristics generated in the fuzzification processes. However, there are also criteria related to assessing the dimensions of the problems. These criteria for interpretability facilitate the identification of dimensions relevant to the issue, which can facilitate understanding a new problem.

On the other hand, this evaluation can also identify less irrelevant features that do not need to be in constructing fuzzy rules as they do not significantly contribute to the resolution of the situation itself. Finally, the criteria for identifying the importance of rules (how much a rule contributes or not to the identification of a target class of the problem), the interpretation of rule consequents (allowing to assess locally and globally the impacts of a fuzzy rule response for the problem solving), and knowledge expansion (identifying when the model identified new patterns and expanded its knowledge base) are fundamental to affirm that a fuzzy neural network model is interpretable. By guaranteeing these criteria, analyzing the generated fuzzy rules with a more outstanding guarantee that they represent solid knowledge about the analyzed problem is possible.

3. Evolving Fuzzy Neural Network Based on Self-Organizing Direction-Aware Data Partitioning (SODA) Approach and Or-Neurons-eFNN-SODA

The development of evolving fuzzy neural networks, models with unprecedented adaptability and freedom, allows acquiring knowledge through the information presented in a data set. This approach assists the precise demonstration of how to build a model able to identify some patterns in a problem analyzed. In this paper, the leading layer of the model proposes a calculation with data density to foster consistent neurons with Gaussian membership function through the idea of or-neurons.

These neurons are responsible for extracting knowledge based on fuzzy rules. The third layer of the model is represented by an artificial neural network model that can process the defuzzification approach. Its training is based on the concept of weight definition through an incremental approach based on recursive least squares. The model's engineering is visible in Figure 3, which is introduced in the following section.



Figure 3. Architecture of the fuzzy neural network proposed in this paper.

3.1. First Layer

The model fuzzification layer is present in the first layer. It is responsible for fuzzifying the data set entries through a clustering process across clouds that identifies them through data density concepts. Therefore, this procedure is responsible for creating Gaussian neurons representing neurons of the first layer. These neurons have weights in the range [0,1] that are defined according to the separability criteria of the dimensions of the problem. For each input variable x_{ij} , L neurons are defined A_{lj} , l = 1, ..., L, whose activation functions

are composed of membership functions in the corresponding neurons. The Gaussian neurons created in this layer are expressed by:

$$a_{jl} = \mu_{A_l}, \ j = 1 \dots n, \ l = 1 \dots L.$$
 (1)

where a_{jl} (μ_{A_l}) represents the degree of association related to the inputs submitted to the model, N is the number of inputs (features), and L represents the number of neurons. For each input variable x_{ij} , L neurons are defined A_{lj} , l = 1, ..., L, whose activation functions are composed of membership functions in the corresponding neurons [38].

The respective weights of each Gaussian neuron formed by the fuzzification process are created using feature weight separability criteria developed by [39]. These weights are expressed by:

$$w_{il}, i = 1...N, l = 1...L$$
 (2)

The fuzzification process used by the model is described below.

3.1.1. Self-Organizing Direction-Aware Data Partitioning

Self-Organizing Direction-Aware Data Partitioning is a data partitioning approach based on empirical data analysis [40], which can identify peaks/modes of the input frequency and apply them as focal points. Data clouds can be considered a particular variety of clouds that possess an especially peculiar contrast. They are non-parametric, as they do not observe a pre-defined data distribution that is commonly unknown. The technique operates a magnitude component based on a universal distance metric and a directional/angular component based on the cosine similarity [9]. This approach is suitable for the online processing of streaming data.

The definitions of the empirical data analysis [41] operators used in the fuzzification approaches are listed below. SODA, adopted in this paper, acts to update clouds as new information changes its data density. These modifications develop new cloud formats and directly impact the model's formation of the outputs to be acquired. For the SODA approach, consider:

- $\mathbf{x} = \{x_1, x_2, \dots, x_k\} \in \mathbb{R}^d$: Input variables (where the index *k* indicates the time instance at which the data point arrives).
- **un** = { $un_1, un_2, ..., un_{\Psi}$ } $\in \mathbb{R}^d$: Set of unique data point locations.
- $f_1, f_2, \ldots, f_{\Psi}$: Number of times that different data points occupy the same unique locations.
- K^c : The number of data samples $\{x\}_{K^c}^c$ belonging to the *c*-th class.
- Un_K^c : The number of unique data samples belonging to the *c*-th class.
- $\sum_{c=1}^{C} K^c = K$ and $\sum_{c=1}^{C} Un_K^c = Un_K$.

Based on $un_1, un_2, ..., un_{\Psi}$ and $f_1, f_2, ..., f_{\Psi}$, it is possible to reconstruct the dataset $x_1, x_2, ..., x_k$ exactly if necessary, regardless of the order of arrival of the data points [41].

The first empirical data analysis operator is cumulative proximity. These elements are defined as the distance between the samples present in the model evaluation. The following is the definition of cumulative proximity [41]:

$$\pi_K(\mathbf{x}_i) = \sum_{j=1}^K d^2(\mathbf{x}_i, \mathbf{x}_j); \quad i = 1, 2, \dots, K$$
(3)

where $d(x_i, x_j)$ denotes the distance between x_i and x_j , which can be Euclidean or cosine, among others [42].

The second operator for the determination of data clouds is the unimodal (or local) density, which in turn is determined by [41]:

$$D_{K}(\mathbf{x}_{i}) = \frac{\sum_{l=1}^{K} \pi_{K}(\mathbf{x}_{l})}{2K \pi_{K}(\mathbf{x}_{i})} = \frac{\sum_{l=1}^{K} \sum_{j=1}^{K} d^{2}(\mathbf{x}_{l}, \mathbf{x}_{j})}{2K \sum_{j=1}^{K} d^{2}(\mathbf{x}_{i}, \mathbf{x}_{j})}; \qquad i = 1, 2, \dots, K$$
(4)

where, for Euclidean distance, $\sum_{l=1}^{K} d^2(\mathbf{x}_i, \mathbf{x}_l) = \sum_{l=1}^{K} || \mathbf{x}_i - \mathbf{x}_l ||^2$ and $\sum_{l=1}^{K} \sum_{j=1}^{K} d^2(\mathbf{x}_l, \mathbf{x}_j) = \sum_{l=1}^{K} \sum_{j=1}^{K} || \mathbf{x}_l - \mathbf{x}_j ||^2$. It can be simplified using the average of $\{x\}_K$, φ_K and the average scalar product, X_K , as in [43]:

$$\sum_{l=1}^{K} \|\boldsymbol{x}_{l} - \boldsymbol{x}_{l}\|^{2} = K \Big(\|\boldsymbol{x}_{l} - \boldsymbol{\varphi}_{K}\|^{2} + X_{K} - \|\boldsymbol{\varphi}_{K}\|^{2} \Big)$$
(5)

$$\sum_{l=1}^{K} \sum_{j=1}^{K} ||\mathbf{x}_{l} - \mathbf{x}_{j}||^{2} = 2K^{2} \left(X_{K} - ||\boldsymbol{\varphi}_{K}||^{2} \right)$$
(6)

Finally, the third empirical data analysis operator is global density (D_K^G) . It is necessary to identify in the data the weighted sum of the local density by a similar occurrence in $f_1, f_2, \ldots, f_{\Psi K}$. It is defined for unique data samples and their corresponding number of repetitions in the data set/stream. It is expressed in (7).

$$D_{K}^{G}(un_{k}) = \frac{f_{k}}{\sum_{j=1}^{\Psi_{K}} f_{j}} D_{K}(un_{k}) = \frac{f_{k}}{1 + \frac{\|un_{k} - \varphi_{K}\|^{2}}{X_{K} - \|\varphi_{K}\|^{2}}}$$
(7)

The essence of an evolving model is to have its parameters evolve as new samples emerge with new information. The empirical data analysis operators can be updated recursively. This update can be seen below:

$$\boldsymbol{\varphi}_{k} = \frac{k-1}{k} \boldsymbol{\varphi}_{k-1} + \frac{1}{k} \boldsymbol{x}_{k}; \quad \boldsymbol{\varphi}_{1} = \boldsymbol{x}_{1}; \quad k = 1, 2, \dots, K$$
 (8)

$$X_{k} = \frac{k-1}{k} X_{k-1} + \frac{1}{k} \|\boldsymbol{x}_{k}\|^{2}; \ X_{1} = \|\boldsymbol{x}_{1}\|^{2}; \ k = 1, 2, \dots, K$$
(9)

$$D_K(\mathbf{x}_k) = \frac{1}{1 + \frac{\|\mathbf{x}_k - \boldsymbol{\varphi}_K\|^2}{X_K - \|\boldsymbol{\varphi}_K\|^2}}$$
(10)

The considerable typical distance metric used, Euclidean, was adopted in SODA as the magnitude component and, consequently, can be represented between x_i and x_j by [9]:

$$d_M(x_i) = \|x_i - x_j\| = \sqrt{\sum_{k=1}^m (x_{ik} - x_{jk})^2} i, j = 1, 2, \dots, n$$
(11)

The angular component that assumed the ideas of cosine similarity and is provided in the SODA model can be expressed as [9]:

$$d_A(x_i) = \sqrt{1 - \cos(\Theta_{x_i, x_j})} i, j = 1, 2, \dots, n$$
(12)

where $\cos(\Theta_{x_i,x_j}) = \frac{\langle x_i, x_j \rangle}{\|x_i\|, \|x_j\|}$ expresses the value of the angle between x_i and x_j [9].

Simultaneously, one applies the measurement and angular component values mutually. Considerable problems can be estimated on a 2D plane, called the direction-aware plane [9]. The empirical data analytics operators [40] employed in this approach are the cumulative proximity (3), local density (4), and global density (7).

The recursive update of these parameters also follows the identical concepts as empirical data analysis operators. Their formulation can be seen in Equations (8) and (9). The angular component updates similarly, as expressed below [9]:

$$\boldsymbol{\varphi}_{n}^{A} = \frac{n-1}{n} \boldsymbol{\varphi}_{n-1}^{A} + \frac{1}{n} \frac{x_{n}}{\|\mathbf{x}_{\mathbf{n}}\|}; \quad \boldsymbol{\varphi}_{1}^{A} = \frac{x_{1}}{\|\mathbf{x}_{1}\|}$$
(13)

When Euclidean distance is used for Local Density in the SODA approach, it can be represented as follows [9]:

$$\sum_{j=1}^{n} \pi_n^M(\mathbf{x}_j) = 2n^2 \left(X_n^M - \left\| \varphi_{\mathbf{n}}^{\mathbf{M}} \right\|^2 \right)$$
(14)

The initial phases of the SODA algorithm regard, firstly, composing different directionaware planes of the recognized data samples operating both the magnitude-based and angular-based densities; secondly, distinguishing focal points; and finally, handling the focal points to partition the data space into data clouds [9]. The algorithm is executed in the following steps:

Stage 1—Preparation: Estimate the average values between every pair of data samples, $x_1, x_2, ..., x_n$ for both the square Euclidean components, d_M and square angular components, d_A [9].

$$\overline{d}_{M}^{2} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} d_{M}^{2}(\mathbf{x}_{i}, \mathbf{x}_{j})}{n^{2}} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} \|\mathbf{x}_{i} - \mathbf{x}_{j}\|^{2}}{n^{2}} = 2\left(X_{n}^{M} - \|\boldsymbol{\varphi}_{\mathbf{n}}^{\mathbf{M}}\|^{2}\right)$$
(15)

$$\overline{d}_{A}^{2} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} d_{A}^{2}(\mathbf{x}_{i}, \mathbf{x}_{j})}{n^{2}} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} \left\| \frac{\mathbf{x}_{i}}{\|\mathbf{x}_{i}\|} - \frac{\mathbf{x}_{j}}{\|\mathbf{x}_{i}\|} \right\|^{2}}{2n^{2}} = 1 - \left\| \boldsymbol{\varphi}_{n}^{A} \right\|^{2}$$
(16)

After computing the global density, SODA reclassifies the problem samples in a decreasing way and renames them as $\{\widehat{\Phi}_1, \widehat{\Phi}_2, \dots, \widehat{\Phi}_{n_u n}\}$ [9].

Stage 2—Direction-Aware Plane Projection: The direction aware projection process starts with the unique data sample with the highest global density, namely $\bar{\vartheta}_1$. It is initially set to be the first reference, $\varphi_1 = \hat{\Phi}_j$, which is also the origin point of the first direction-aware plane, denoted by P1 ($\Psi = 1$, and Ψ is the number of existing direction-aware planes in the data space. The following rule can describe the second step of the algorithm [9]:

Condition 1
$$IF\left(\frac{d_M(\boldsymbol{\varphi}_l, \boldsymbol{\hat{\Phi}}_j)}{\overline{d}_M} < \frac{1}{\vartheta}\right) AND\left(\frac{d_A(\boldsymbol{\varphi}_l, \boldsymbol{\hat{\Phi}}_j)}{\overline{d}_A} < \frac{1}{\vartheta}\right) THEN\left(\mathbf{P}_l = \widehat{\mathbf{\Phi}}_j\right)$$
(17)

where ϑ assists in the deduction of the granularity of the cloud used. When considerable direction-aware planes satisfy the Equation (17) criteria at the same time, $\hat{\Phi}_j$ will be assigned to the one nearest to them according to the following equation [9]:

$$i = \arg\min_{l=1,2,\dots,\Psi} \left(\frac{d_M(\boldsymbol{\varphi}_l, \hat{\boldsymbol{\Phi}}_j)}{\overline{d}_M} + \frac{d_A(\boldsymbol{\varphi}_l, \hat{\boldsymbol{\Phi}}_j)}{\overline{d}_A} \right)$$
(18)

In this step, the mean, the support (number of samples of the problem), and the sum of the global density (φ_i , Sp_i and D_i , respectively) are updated as follows [9]:

$$\boldsymbol{\varphi}_i = \frac{S_i}{S_i+1} \boldsymbol{\varphi}_i + \frac{1}{S_i+1} \widehat{\boldsymbol{\Phi}}_j \tag{19}$$

$$Sp_i = Sp_i + 1 \tag{20}$$

$$\mathbf{D}_{i} = \mathbf{D}_{i} + D_{n}^{G}\left(\widehat{\mathbf{\Phi}}_{j}\right) \tag{21}$$

Nonetheless, if Equation (17) is not fulfilled, the parameters involved in the analysis will be updated after the creation of a new direction-aware plane ($P_{\Psi+1}$) and new references, as follows [9]:

$$\Psi = \Psi + 1 \tag{22}$$

$$\boldsymbol{\varphi}_{\Psi} = \widehat{\boldsymbol{\Phi}}_{i} \tag{23}$$

$$S_{\Psi} = 1 \tag{24}$$

$$\mathbf{D}_{\Psi} = D_n^G \left(\widehat{\mathbf{\Phi}}_j \right) \tag{25}$$

This procedure occurs until all problem samples are organized. In this situation, some data samples may be located on several direction-aware planes simultaneously, depending on the behavior of the problem. In this sense, the final establishment of these samples is defined by the distances between them and the points of origin of the following direction-aware plans [9].

Stage 3—Identifying the Focal Points: For each direction-aware plane, denoted as P_{β} , find the neighboring direction-aware planes ({**P**}^{*n*}_{β}, $l = 1, 2, ..., L, l \neq \beta$). The subsequent association can define the rule for determining new plans [9]:

Condition 2
$$IF\left(\frac{d_M(\boldsymbol{\varphi}_{\beta},\boldsymbol{\varphi}_l)}{\overline{d}_M} \leq \frac{2}{\vartheta}\right) AND\left(\frac{d_A(\boldsymbol{\varphi}_{\beta},\boldsymbol{\varphi}_l)}{\overline{d}_A} \leq \frac{2}{\vartheta}\right)$$
$$THEN\left(\{\mathbf{P}\}_{\beta}^n = \{\mathbf{P}\}_{\beta}^n \cup \{\mathbf{P}_l\}\right)$$
(26)

The central mode/peak of data density (\mathbf{P}_{β}) will be selected if the following equation is attended (assuming the corresponding \mathbf{D} of $\{\mathbf{P}\}_{\beta}^{n}$, $l = 1, 2, ..., \Psi$, $l \neq \beta$ is expressed by $\{\mathbf{D}\}_{\beta}^{n}$ [9]):

Condition 3
$$IF(\mathbf{D}_{\beta} > \max\left(\{\mathbf{D}\}_{\beta}^{n}\right))$$
 THEN $(\mathbf{P}_{\beta} \text{ is a mode/ peak of } \mathbf{D})$ (27)

All peaks are found when conditions 2 and 3 perform mutually.

Stage 4—Forming Data Clouds: After all the direction-aware planes encountering the modes/peaks of the data density are appointed, it reflects their origin points, represented by φ_o , as the focal points and uses them to form data clouds according to a Voronoi tessellation [44]. The term that represents the data cloud is suggested as follows [9]:

Condition 4
$$\frac{IF\left(\nvDash = \underset{j=1,2,\dots,\xi}{\operatorname{arg\,min}} \left(\frac{d_M(\mathbf{x}_l, \boldsymbol{\varphi}_j^o)}{\overline{d}_M} + \frac{d_A(\mathbf{x}_l, \boldsymbol{\varphi}_j^o)}{\overline{d}_A} \right) \right) }{THEN(\mathbf{x}_l \text{ is assigned to the \diamondsuit^{th} data cloud}) }$$
(28)

where ξ is seen as the number of focal points. The concept of clouds is equivalent to clusters. Regardless, there are distinctions because they are non-parametric, do not have a typical shape, and can express any real data distributions following local density criteria [45].

This approach can also perform with the evolution of their clouds due to the ability to update their parameters recursively. For the evolving strategy used in SODA, some actions are essential for new samples to influence changes in the data clouds. Therefore, the following steps, which are related to the steps listed previously, help in describing the update of the SODA fuzzification method [9]:

Stage 5—Update Problem Parameters: For each new sample introduced to training $(x_{n+1}, \varphi_n^M, \varphi_n^A, \text{ and } X_n^M)$, they are updated by Equations (9), (13), and (14). Likewise, the Euclidean components and the angular components between the new sample and the established centers of the direction-aware planes are also updated for each new sample employing Equations (11) and (12) and are currently described by $d_M(x_{n+1}, \varphi_l)$ and $d_A(x_{n+1}, \varphi_l)$ for $l = 1, 2, ..., \Psi$, sequentially.

At this point in the procedure, the direction-aware plane projection stage is triggered, letting a joint analysis of Condition 1 and Equation (18) determine whether the new sample belongs to an actual data set (model parameters are updated based on Equations (19) and (20), or, if it is a new sample with different behavior, consequently forming the demand to produce a new direction-aware plane (and hence the parameters will be updated based on Equations (22), (23) and (24)) [9]. Stage 6—The Fusion of Overlapping Direction-Aware Planes: After the realization of Stage 5, a new requirement is examined on the fuzzification approach, recognizing strongly overlapping direction-aware planes [9]:

Condition 5
$$IF\left(\frac{d_M(\varphi_i,\varphi_j)}{\overline{d}_M} < \frac{1}{2\vartheta}\right) AND\left(\frac{d_A(\varphi_i,\varphi_j)}{\overline{d}_A} < \frac{1}{2\vartheta}\right)$$
 (29)
 $THEN(\mathbf{P}_i \text{ and } \mathbf{P}_j \text{ are strongly overlapping})$

This situation is not advisable, so these circumstances are solved by assembling a new direction-aware plane on P_j by combining the analyzed direction-aware plane. This merger criterion is described by [9]:

Ψ

$$f = \Psi - 1 \tag{30}$$

$$\boldsymbol{\varphi}_{j} = \frac{Sp_{j}}{Sp_{j}+Sp_{i}}\boldsymbol{\varphi}_{j} + \frac{S_{i}}{S_{j}+S_{i}}\boldsymbol{\varphi}_{i}$$
(31)

$$Sp_i = Sp_i + Sp_i \tag{32}$$

The procedure occurs until all the direction-aware planes are not highly overlapping. This technique occurs by eliminating the parameters of P_i and returning to Stage 5. This flow occurs until the conditions reported in Condition 5 are not satisfied, thus flowing to the final stage of the evolving fuzzification process [9]. This situation is essential for using the SODA technique, it is a benchmark for certifying the interpretability of the results.

Stage 7—Forming Data Clouds: Once all samples are examined, SODA defines all the focal points of the existent centers in the direction-aware planes resulting from Stage 6. The subsequent steps are correlated to the global density estimation of the centers of the direction-aware plane through Equation (7), operating the supports of each direction-aware plane as the number of repetitions to be performed. Therefore, it obtains the global density, which is called $D_n^G(\varphi_l)$ [9]. Second, each direction-aware plane is examined using Condition 2 to find its neighboring direction-aware plane. In this circumstance, Condition 3 is used in order to confirm whether the $D_n^G(\varphi_e)$ underneath investigation is one of the local maxima of $(D_n^G(\varphi_l))$, so that eventually all the identified maximum $D_n^G(\varphi_l)$ zones and the centers of the corresponding direction-aware planes, together with Condition 4, will act in the establishment of a focal point, and hence in the data clouds [9].

It is meaningful that at the first moment of the model's training, the SODA algorithm's demeanor will be similar to the offline behavior of the algorithm (Stages 1–4). From the subsequent data presented to the model, SODA will act in its evolving state, primarily concentrating on the stages 5–7 of the SODA algorithm [9].

Figure 4 presents an example of the construction of clouds and identification of the centers that guide the construction of the structures resulting from the fuzzification process.

3.1.2. Incremental Feature Weights Learning

This paper addresses a different creation in the definition of weights when compared with various models of evolving fuzzy neural networks that define them randomly. Here, the weights are defined by an online and incremental criterion for evaluating the relevance of the problem features when determining the best separation of the target classes. This proposal increases the reducibility of fuzzy rules, as irrelevant weights can be discarded from a visual and operational evaluation and, at the same time, bring more interpretability to the Gaussian neurons of the first layer of the model. It should be noted that this procedure has already been successfully applied in evolving fuzzy neural networks, as in [11,46].



Figure 4. Example of the SODA approach with 17 clouds.

The approach proposed in [39] uses the Dy–Brodley separability criterion [47] for the classes analyzed in the problem. Either compute as a single feature (using each feature (and only one of them)) or discard each feature from the complete set to identify the feature in question that best defines the separation of the classes analyzed. This procedure sets values closer to 1 to features that are more satisfactory in separate classes [11]. On the other hand, dimensions that are ineffective at identifying labels correctly are assigned values close to zero. Accordingly, if a dimension is nonessential to the model, it can be excluded from the model to reduce interpretation complexity.

The class separability criterion can be represented as an extension of Fisher's separability criterion [11]:

$$=\delta(S_w^{-1}S_b) \tag{33}$$

where $\delta(S_w^{-1}S_b)$ is the sum of the diagonal elements of the matrix $S_w^{-1}S_b$. S_b means the dispersion matrix between classes that estimate the class's dispersion averages to the total average, and S_w denotes the within scattering matrix (that measures the samples' dispersion in their class averages) [11].

This paper uses the leave-one-feature-out approach. It works by discarding each feature from the complete set and then calculates Equation (33) for each obtained subset \rightarrow again obtaining J_1, \ldots, J_N for N features in the data set. It can be expressed by [39]:

$$w_{j} = 1 - \frac{J_{j} - \min_{i=1,\dots,N} J_{i}}{\max_{i=1,\dots,N} J_{i}}$$
(34)

The lower the value of J_i becomes, the more important feature *i* is, because feature *i* was discarded. Hence, and seeking comparable importance among all features (relative to the most noteworthy feature, which should acquire a weight of 1), the feature weights are assigned by [39]:

In particular, S_b can be updated by updating class-wise and the overall mean through an incremental mean formula. Recursive covariance update formulas update the covariance matrices per class with rank-1 change for more powerful and faster convergence to the batch calculation [48]. For all formulas and details, check [39].

14 of 31

3.2. Second Layer

The second layer comprises fuzzy logic neurons that use fuzzy operators [49]. The most well-known is the *t*-norm, $(T : [0,1]^2 \rightarrow [0,1])$, which represents the intersection of fuzzy sets, and the *t*-conorm $(S : [0,1]^2 \rightarrow [0.1])$, capable of the union of fuzzy sets. In this paper, these operations are represented by [49]:

$$t(x,y) = x \times y \tag{35}$$

$$s(x,y) = x + y - x \times y \tag{36}$$

Neurons that use t-norm and t-conorm are extremely common in constructing evolving fuzzy neural networks; and-neuron and or-neuron stand out. These fuzzy neural structures, proposed first by Hirota and Pedrycz [50], can aggregate the model inputs (in the case of this paper, the Gaussian neurons of the first layer) with their respective weights. They are considered operational units capable of nonlinear multivariate operations in unit hypercubes ($[0,1] \rightarrow [0,1]^n$) with fuzzy inputs and weights. This process generates a unique value that allows the construction of fuzzy rules in the IF–THEN format. Their interpretability is directly related to the operators applied in the procedures of these fuzzy neuronal structures.

In these neurons with two levels, the first one executes aggregate operations, and the last completes the procedures with fuzzy operators. This operation simplifies the aggregation of the fuzzy input with its respective weight, creating interpretable connectivity between the antecedent connectives [51]. In the case of or-neurons (the neuron structure used in this paper), all or-type connectors are present in the connection of all antecedents of a fuzzy rule generated by this structure [50]. It can be exemplified by:

$$\mathbf{z} = orneuron(\mathbf{w}; \mathbf{a}) = S_{i=1}^{n}(w_i \ t \ a_i)$$
(37)

where *S* is a *t*-conorms and *t* is *t*-norms, $\mathbf{a} = [a_1, a_2, ..., a_3, ...a_N]$ is the neuron's inputs (values of fuzzy relevance (Gaussian neurons)), and $\mathbf{w} = [w_1, w_2, ..., w_3, ...w_N]$ represents the weights, $\mathbf{a}, \mathbf{w} \in [0, 1]^n$, defined in the range between 0 and 1. Finally, \mathbf{z} is the neuron's output, which is also seen as a fuzzy rule. This neuron can generate fuzzy rules as shown below:

$$Rule_{1}: If x_{1} is A_{1}^{1} with impact w_{11} \dots$$

$$or x_{2} is A_{1}^{2} with impact w_{21} \dots$$

$$or x_{n} is A_{1}^{n} with impact w_{n1} \dots$$

$$Then y_{1} is v_{1}$$

$$Rule_{L}: If x_{1} is A_{L}^{1} with impact w_{1L} \dots$$

$$or x_{2} is A_{L}^{2} with impact w_{2L} \dots$$

$$or x_{n} is A_{L}^{n} with impact w_{nL} \dots$$

$$Then y_{L} is v_{L}$$

$$(38)$$

with A_i^1, \ldots, A_i^n fuzzy sets represented as linguistic terms for the *n* inputs appearing in the *i*th rules, and y_1, \ldots, y_L are consequent terms (output); *L* is the number of rules, and **v** represents the correspondence value for the output [17].

3.3. Third Layer

The third layer of the fuzzy neural network is composed of the neural network that aggregates all the fuzzy rules generated in the first layer. Its inputs are the fuzzy neurons with their respective weights generated in the training process, to be explained later. An artificial neuron aggregates these fuzzy rules and processes them with weights to generate model outputs. This singleton neural network (composed of a single neuron) has a linear activation function, and a signal function transforms the values obtained by the model into

the expected outputs (-1 and 1), indicating whether or not there was fraud in the auction. This defuzzification process can be mathematically described by [38]:

$$y = \Omega\left(\sum_{j=0}^{l} f_{\Gamma}(z_j, v_j)\right)$$
(39)

where $z_0 = 1$, v_0 is the bias, and z_j and v_j , j = 1, ..., l are the output of each fuzzy neuron of the second layer and their corresponding weight, respectively. f_{Γ} represents the linear activation function and Ω the signal function [38].

The linear and signal function are represented, respectively, by:

$$f_{\Gamma}(\mathbf{z}) = \mathbf{z}^* \mathbf{w} \tag{40}$$

$$\Omega = \begin{cases} 1, \ if \ \sum_{j=0}^{l} f_{\Gamma}(z_{j}, v_{j}) > 0 \\ -1, \ if \ \sum_{j=0}^{l} f_{\Gamma}(z_{j}, v_{j}) < 0 \end{cases}$$
(41)

3.4. Training

The neural network training process facilitates the definition of existing parameters in an evolving fuzzy neural network. In the model proposed in this article, this training takes place in two stages. The first, offline, is performed using the concept of the Extreme Learning Machine [10], and in the evolving phase of the model, a recursive weighted least squares (RWLS) approach [52] is used. The first process guarantees a definition of weights based on the initial architecture, and the second incrementally updates these weights as new fuzzy rules, and samples with essential information are evaluated. The initial definition of the weights of the third layer carried out in the offline training stage can be expressed by [38]:

$$\vec{v}_k = Z^+ \vec{y}_m \quad \forall m = 1, \dots, 2 \tag{42}$$

where *m* is the number of classes, and $Z^+ = Z^T Z$ is the pseudo-inverse of the Moore–Penrose matrix [53] of *Z* (output of the second layer). This procedure is commonly applied in training artificial neural networks and sets the weights in a single step, allowing the training to be fast and accurate.

In the evolving phase of the model (new incoming online stream samples), the recursive weighted least squares (RWLS) approach is used [52] with formulas for updating the $\vec{v_k}$, and it can be expressed as follow:

$$\eta = \vec{z}^{t} Q^{t-1} \left(\psi + (\vec{z}^{t})^{T} Q^{t-1} \vec{z}^{t} \right)^{-1}$$
(43)

$$Q^{t} = (\mathbf{I}_{L^{t}} - \eta^{T} \vec{z}^{t}) \psi^{-1} Q^{t-1}$$
(44)

$$\vec{v}_k^t = \vec{v}_k^{t-1} + \eta^T (y_m^t - \vec{z}^t \vec{v}_k^{t-1})$$
(45)

where the index *k* again denotes the class index k = 1, ..., C. \vec{z}^t denotes the regressor vector of the current sample, η is the current kalman gain vector, $I_{L_s^t}$ is an identity matrix based on the number of neurons in the second layer, and $L_s^t \times L_s^t$; $\psi \in]0, 1]$ denotes forgetting factor (1 per default). Q denotes the inverse Hessian matrix $Q = (Z_{sel}^T Z_{sel})^{-1}$ and is set initially as $\omega I_{L_s^t}$, where $\omega = 1000$ [54].

This matrix is directly and incrementally updated by the second equation above without requiring (time-consuming and possibly unpredictable) matrix re-inversion. This procedure identifies existing changes in the dataset and updates the weight values without losing the previous reference. Thus, the concept of memory is applied to the weights, allowing the model's training to be consistent with the premises of evolving fuzzy systems.

The eFNN-SODA can be synthesized, as represented in Algorithm 1. It has one parameter during the initial batch learning phase:

Granularity of the cloud results (grid size), ϑ ;

Algorithm 1 eFNN-SODA Training and Update Algorithm

Initial Batch Learning Phase (Input: data matrix X):

(1) Define granularity of the cloud, ϑ .

(2) Extract *L* clouds in the first layer using the SODA approach (based on ϑ).

(3) Construct *L* fuzzy neurons with Gaussian membership functions with \vec{c} and σ values derived from SODA.

(4) Calculate the combination (feature) weights \vec{w} for neuron construction using Equation (34).

(5) Construct *L* logic neurons on the second layer of the network by welding the *L* fuzzy neurons of the first layer, using or-neurons (Equation (37)) and the centers \vec{c} and widths $\vec{\sigma}$. (6)

for i = 1, ..., K do

(6.1) Calculate the regression vector $z(x_i)$.

(6.2) Store it as one row entry into the activation level matrix *z*.

end for

(7) Extract activation level matrix z according to the L neurons.

(8) Estimate the weights of the output layer for all classes k = 1, ..., C by Equation (42) using z and indicator vectors \vec{y}_k .

Update Phase (Input: single data sample \vec{x}_t):

(11) Update *L* clouds and evolving new ones on demand (based on Stages 5, 6, and 7 of the SODA approach).

(12) Update the feature weights \vec{w} by updating the within- and between-class scatter matrix and recalculating Equation (34).

(13) Perform Steps (3) and (5).

(14) Calculate the interpretability criteria (Section 3.5).

(15) Calculate the regression vector $z(\vec{x}_t)$.

(16) Update the output layer weights by Equation (45).

The computational complexity of eFNN-SODA encloses the number of flops demanded to process one single sample through the updated algorithm (second part in Algorithm 1) because this affects the online speed of the algorithm. In this sense, the main steps involved can be listed below:

- SODA algorithm: The complexity of *O*(*p*) (*p* is the dimensionality of the input space when updating with a single sample).
- Or-neurons: Complexity O(mp) (constructing them from *m* clouds).
- Weights in the first-layer Gaussian neurons: Complexity Kp^3 (where K is the number of classes). That happens because the between- and within-class scatter matrices need to be updated for each class for each feature separately and independently, having a complexity of $O(p^2)$ (matrices have a size of $p \times p$).
- Neuron activation in the third-layer model: The complexity of O(mp) (in each sample, the activation levels to all *m* neurons (with dimensionality *p*) require to be estimated).
- Output-layer neuron: The complexity of $O(mp^2)$ (because of the weighted method in each rule individually, demanding this complexity [55].)

3.5. Interpretability Criteria

The interpretability criteria are essential for evaluating the model's behavior throughout the evaluations. In this case, the eFNN-SODA model uses some approaches to validate the interpretability criteria proposed in [37], so that the generated fuzzy rules are reliable. In this case, the model applies some evaluations throughout the experiments to ensure that the generated fuzzy rules can add some knowledge about the evaluated dataset. These procedures are listed below:

Simplicity and Distinguishability: These two criteria verify whether the proposed model is the simplest and has its structures distinguishable during training. This

means that the evaluation revolves around low complexity and high accuracy. Regarding the simplicity of the eFNN-SODA, the model is expected to have a more compact structure and a high degree of assertiveness. The criterion defined for the identification of model simplicity (in the comparison between models) can be expressed by:

$if\{(L_a < L_b) \lor (accuracy_{model_a} \ge accuracy_{model_b})\} then model_a is simpler than model_b$ (46)

where L_a and L_b are, respectively, the number of fuzzy rules of the *model*_a and *model*_b. Regarding distinguishability, it is expected to assess whether there is an overlapping in the structures formed in the fuzzification process. The SODA approach uses an assessment of overlapping in the evolution process in its sixth stage, thus ensuring that this situation does not occur. For the evaluation of the distinguishability criterion, eFNN-SODA uses the comparison of similarity between the Gaussian neurons formed in the first layer (termed as $z_i(bef)$ and $z_i(after)$) dimension-wise, and similarity (S_{im}) degree $S_{im}(z_i(bef), z_i(after))$ can be used for calculating an amalgamated value. The degree of change (\ltimes) is then presented by [11]:

$$\ltimes (z_i) = 1 - S_{im}(z_i(bef), z_i(after))$$
(47)

bef = N - n and after = N, assuming that n new samples have passed the datastream-based transformation phase with N samples operated so far for model training and adaptation [11].

Therefore, it is feasible to conclude that two rules are only identical if all their antecedent parts are equivalent. The x coordinates of the points of intersection of two Gaussians used as fuzzy sets in the identical antecedent part of the Gaussian rule i (here for the *j*th) before and after its update can be estimated by [56]:

$$inter_{x}(1,2) = -\frac{\vec{c}_{bef,j}\vec{\sigma}_{after,j}^{2} - \vec{c}_{after,j}\vec{\sigma}_{bef,j}^{2}}{\vec{\sigma}_{bef,j}^{2} - \vec{\sigma}_{after,j}^{2}} + -\sqrt{\left(\frac{\vec{c}_{bef,j}\vec{\sigma}_{bef,j}^{2} - \vec{c}_{bef,j}\vec{\sigma}_{after,j}^{2}}{\vec{\sigma}_{after,j}^{2} - \vec{\sigma}_{bef,j}^{2}}\right)^{2} - \frac{\vec{c}_{bef,j}^{2}\vec{\sigma}_{after,j}^{2} - \vec{c}_{after,j}^{2}\vec{\sigma}_{bef,j}^{2}}{\vec{\sigma}_{after,j}^{2} - \sigma_{bef,j}^{2}}}$$
(48)

with $\vec{c}_{bef,j}$, $\vec{\sigma}_{bef,j}$ being the *j*th center coordinate and standard deviation of the Gaussian neuron before its update, and $\vec{c}_{after,j}$ and $\vec{\sigma}_{after,j}$ the *j*th center coordinate and standard deviation of the Gaussian neuron after its update [11].

The maximal membership degree of the two Gaussian membership functions in the intersection coordinates is then used as overlap. Consequently, the similarity degree of the corresponding rules' antecedent parts in the *j*th dimension is [11]:

$$S_{im}^{bef,after}(j) = \max(\mu_i(inter_x(1)), \mu_i(inter_x(2)))$$
(49)

with $\mu_i(inter_x(1))$ being the membership degree of the *j*th fuzzy set in Rule *i* in the intersection point *inter*_x(1). The amalgamation of overall rule antecedent parts leads to the final similarity degree between the rule before and after its update:

$$S_{im}(z_i(bef), z_i(after)) = T_{j=1}^p S_{im}^{bef, after}(j)$$
(50)

where *T* denotes a t-norm operator, and *p* is the number of inputs, as a robust nonoverlap along one single dimension is sufficient for the clouds to not overlap at all [56].

 Consistency, Coverage, and Completeness: The concept of consistency in evolving fuzzy systems is attending when their fuzzy rule-set does not deliver a high noise level or an inconsistently learned output behavior. Therefore, a set of fuzzy rules is considered inconsistent when two or more rules overlap in the antecedents, but not in the consequents. In this paper, the consistency of fuzzy rules (comparing a rule before

$$Rule \ z_1 \ is \ inconsistent \ to \ Rule \ z_2 \ if \ and \ only \ if$$

$$S_{ante}(z_1, z_2) \ge S_{cons}(z_1, z_2) \ with \ S_{ante}(z_1, z_2) \ge thr.$$

$$where \ consistency = \begin{cases} 1 \ if \ Equation \ (51) \ is \ false \\ 0 \ if \ Equation \ (51) \ is \ true \end{cases}$$
(51)

where S_{ante} and S_{cons} are close to 1 invariably can be assumed to indicate a heightened similarity, and when they are close to 0, a low similarity [37]. *thr* is a threshold value usually set at 0.8 or 0.9.

The coverage criterion identifies whether there are holes in the resource space by generating undefined input states. This criterion can be solved by applying Gaussian functions, which have unlimited support. In this case, eFNN-SODA guarantees this criterion using this type of function throughout the model's training [37].

Finally, the ϵ -completeness criterion in the eFNN-SODA is defined by [37]:

$$\left(\forall \vec{x} \; \exists i \; (z_i = \underset{j=1,\dots,rl}{T} (\mu_{ij}) > \epsilon)\right) \Rightarrow \left(\forall \vec{x} \; \exists i \; (\forall j \; \mu_{ij} > \epsilon)\right)$$
(52)

where μ_{ij} is the membership degree of a fuzzy set A_j appearing in the *j*th antecedent part of the *i*th rule, *rl* is the rule length, and $\epsilon = 0.135$ according to definitions made in other research, which is considered an evaluation standard for this criterion [37].

- Antecedent Interpretation: The evaluation of rule antecedents is also a fundamental
 part of interpreting and validating the results. In the case of eFNN-SODA, this
 evaluation is performed with the behavior of the evolution of the weights and the
 evaluation of the similarity of the Gaussian neurons formed in the first layer.
- Consequent Interpretation: The evaluation of rule consequents in eFNN-SODA is performed by graphically evaluating changes in the class to which the rule is connected. This is because the values of \vec{v} can change as the model evaluates new samples, and the consequent of the respective rule can be changed.
- Feature Importance Level: The evaluation of the features of the problem is also a fundamental part of understanding the model's behavior. Graphical analyses of the variations generated by the weights in Equation (34) are obtained, allowing us to identify the behavior of the features throughout the experiment.
- Knowledge Expansion: The expansion of knowledge takes place by evaluating the evolution of fuzzy rules throughout the experiment. The SODA approach acts on the evolution and reduction in irrelevant rules. The eFNN-SODA model can also appreciate this behavior throughout the experiment.

4. Auction Fraud Testing

The following sections presents the details and procedures of the tests that were performed, as well as the models and data set. All tests were performed on a computer with the following settings: Intel (R) Core (TM) i7-6700 CPU 3.40 GHz with 16 GB RAM. In the execution and elaboration of the models present in the tests of this paper, Matlab was used, and for data analysis, the Orange Data Mining tool was used (developed in phyton (https://orangedatamining.com/ (accessed on 17 August 2022))).

4.1. Data Set Features

As mentioned previously, this study aims to analyze Shill Bidding fraud. The analyzed dataset was collected by [4] and published in the Machine Learning Repository—UCI (arch ive.ics.uci.edu/ml/datasets/Shill+Bidding+Dataset (accessed on 17 August 2022)). The data correspond to fraudulent bids on one of the largest auction sites on the Internet, eBay.

The collected database [4], features completed iPhone 7 auctions for three months (March to June 2017). The original dataset contains 12 input features. However, for the studies described in this paper, the dimensions related to personal IDs (Record ID, Auction ID, and Bidder ID) were removed since they are ID values and, therefore, irrelevant to the experiments. The remaining nine dimensions are Bidder Tendency, Bidding Ratio, Successive Outbidding, Last Bidding, Auction Bids, Auction Starting Price, Early Bidding, Winning Ratio, Auction Duration, and Class (1 to normal and -1 to fraud). Figure 5 presents statistical values and the data distribution by class involved in the problem.



Figure 5. Statistical values and the data distribution—Auction Fraud data set.

As shown in Figure 5, the dataset used in this experiment is imbalanced, making the classification task challenging.

Figure 6 presents a representation of the data using the FreeViz technique [57]. In it, points in the same class attract each other, while those from different classes repel each other, and the resulting forces are exerted on the anchors of the attributes, that is, on unit vectors of each dimensional axis. With this technique, it is possible to identify projections of unit vectors that are very short compared with the others. This indicates that their associated attribute is not very informative for a particular classification task.



Figure 6. FreeViz projection—Auction Fraud data set.

Based on Figure 6, the most representative dimensions for fraud classification in auctions are successive outbidding, bidding, and winning ratios. In Figure 7, a data visualization according to the Radviz technique [58] is presented to demonstrate a nonlinear multidimensional visualization that can display data defined by three or more variables in a two-dimensional projection. Visualized variables are presented as anchor points evenly spaced around the perimeter of a unit circle. Data instances close to a variable anchor set have higher values for those variables than for others.



Figure 7. Radviz projection—Auction Fraud data set.

4.2. Models and Hyperparameters

The models used in the experiment are evolving fuzzy systems considered state-of-theart in classifying binary patterns. They have different architectures and training methods. The parameters of each of these models were defined through pre-tests that used 10-folds in a group of parameters that make up each approach. The models used in the experiment are listed below:

EFDDC—Evolving fuzzy data density cluster. The evolving model uses data-densitybased clustering based on empirical data analysis operators and nullneuron. The model's training is based on the Extreme Learning Machine and Bolasso technique to select the best neurons. The model parameters are $\rho = 0.01$, bootstraps bt = 16, and consensus threshold $\lambda = 0.7$ (the best parameters are defined between: $\rho = \{0.01, 0.02, 0.03, 0.04\}$, $bt = \{4, 8, 16, 32\}$, and $\lambda = \{0.5, 0.6, 0.7\}$) [59].

EFNHN—Evolving fuzzy neural hydrocarbon network. The model combines an evolving fuzzification technique based on data density (Autonomous Data Partitioning), training based on Extreme Learning Machine, and unineurons. The defuzzification process is based on an Artificial Hydrocarbon network. The model parameter is the Learning rate = 0.1 (the best parameter is defined between: Learning rate = $\{0.01, 0.05, 0.1, 0.2\}$) [60].

EFNNS—Evolving fuzzy neural network and Self-Organized direction aware. The evolving fuzzy neural network model uses the Self-Organized direction aware for fuzzification, unineurons, Extreme Learning Machine, and pruning technique. The only parameter used in the model is the ϑ =3 (the best parameter is defined between: Learning rate = {2, 3, 4, 5}) [61].

ALMMo-0*—Autonomous zero-order multiple learning with pre-processing. The model is a neuro-fuzzy approach to autonomous zero-order multiple learning with pre-processing that improves the classifier's accuracy, as it creates stable models. The parameter is radius = $\sqrt{2 - 2\cos(30^\circ)}$ (the best parameter is defined between: radius = $\left\{\sqrt{2 - 2\cos(15^\circ)}, \sqrt{2 - 2\cos(30^\circ)}, \sqrt{2 - 2\cos(45^\circ)}, \sqrt{2 - 2\cos(60^\circ)}\right\}\right)$ [62].

ALMMo—Autonomous zero-order multiple learning. A neuro-fuzzy approach for autonomous zero-order multiple models without pre-processing. The parameter is radius = $\sqrt{2 - 2\cos(30^\circ)}$ (the best parameter defined between: radius = { $\sqrt{2 - 2\cos(30^\circ)}$, $\sqrt{2 - 2\cos(45^\circ)}$, $\sqrt{2 - 2\cos(60^\circ)}$ }) [63].

eGNN—Evolving Granular Neural Network. A model that uses concepts of hyberboxes and nullnorm. The parameters are Rho = 0.85, eta = 2, hr = 40, Beta = 0, chi = 0.1, zeta = 0.9, c = 0, counter = 1, and alpha = 0 (for this model, the reference values proposed by the author of the model were used. More information can be seen at https://sites.google.com/view/dleite-evolving-ai/algorithms (accessed on 18 August 2022)) [64].

4.3. Evaluation Criteria of Experiments

Evolving fuzzy systems approaches can act in the evaluation of stream data, where each sample is considered separately, and the accuracy of the result is evaluated individually and incrementally added to the final result. In this case, the evaluation of accuracy in trend lines is suitable for this type of context as it allows the evolution of the model results to be seen as new samples are evaluated.

These trend lines were calculated using the following criteria: [65]:

$$Accuracy(K+1) = \frac{Accuracy(K) * K + I_{\hat{y}=y}}{K+1},$$
(53)

where accuracy:

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} * 100.$$
(54)

where TP = true positive, TN = true negative, FN = false negative, and FP = false positive.

4.4. Results

The results of the evaluation of the trend lines are presented in Figure 8. This evaluation was carried out with 20% of the total samples (1264) for initial training and the rest (5057) for online adaptation, evolution, and evaluation of the model. For the experiments, the models listed above were compared with the results obtained by eFNN-SODA with $\vartheta = 2$ (the values were defined through a cross-validation procedure for $\vartheta = \{2, 3, 4, 5\}$).



Figure 8. Trend lines for the Auction Fraud dataset.

The results presented in Figure 8 corroborate the efficiency of the approach in correctly classifying the situations involved in the data set that deal with fraud in auctions. The model proposed in this paper could dynamically and more assertively resolve cases of attack or nonexistence in evaluating the samples. Next to the results obtained by eFNN-SODA are the results generated by Autonomous zero-order multiple learning with pre-processing and Autonomous zero-order multiple learning. Numerically, they are similar, with a slight advantage for eFNN-SODA. A little less effective, but with good results, is the evolving fuzzy neural network with Self-Organizing Direction-Aware Data Partitioning model. The other models did not perform as effectively as the eFNN-SODA models and the Autonomous zero-order multiple learning variations.

The eFNN-SODA model solved the problem of fraud in auctions with three rules. In this case, the Self-Organizing Direction-Aware Data Partitioning fuzzification approach did not generate new neurons, as it was only necessary to change centers in the rule antecedents to solve the problem. The three fuzzy rules are presented in the following:

Rule 1. If Bidder Tendency is small with impact 0.41 or Bidding Ratio is small with impact 0.59 or Successive Outbidding is small with impact 1.00 or Last Bidding is medium with impact 0.39 or Auction Bids is small with impact 0.38 or Auction Starting Price is medium with impact 0.38 or Early Bidding is medium with impact 0.38 or Winning Ratio is small with impact 0.53 or Auction Duration is small with impact 0.38 then output is normal.

Rule 2. If Bidder Tendency is medium with impact 0.41 or Bidding Ratio is medium with impact 0.59 or Successive Outbidding is medium with impact 1.00 or Last Bidding is high with impact 0.39 or Auction Bids is high with impact 0.38 or Auction Starting Price is high with impact 0.38 or Early Bidding is high with impact 0.38 or Winning Ratio is medium with impact 0.53 or Auction Duration is medium with impact 0.38 then output is fraud.

Rule 3. If Bidder Tendency is high with impact 0.41 or Bidding Ratio is high with impact 0.59 or Successive Outbidding is high with impact 1.00 or Last Bidding is small with impact 0.39 or Auction Bids is medium with impact 0.38 or Auction Starting Price is small with impact 0.38 or Early Bidding is small with impact 0.38 or Winning Ratio is high with impact 0.53 or Auction Duration is high with impact 0.38 then output is normal.

These rules are seen as the extraction of knowledge from the evaluated data set and can serve as a conceptual analysis of the events involved in identifying or not identifying fraud in auctions. The interpretability criteria help understand and validate the generated knowledge to confirm the efficiency of the generated fuzzy rules.

The first evaluation seeks to verify whether the eFNN-SODA model was the simplest in solving auction fraud problems. This factor is confirmed because, according to Equation (46), the eFNN-SODA model had the lowest number of fuzzy rules and the highest accuracy result compared with the other models used in the experiments. eFNN-SODA assertively solved the target problem with only three fuzzy rules. As a comparison, the different models tested in the experiments generated between 10 and 28 fuzzy rules.

The distinguishability criteria (based on Equation (50)) of the formed solution are presented in Figure 9. In this figure, the evolution of the first layer of Gaussian neurons can be observed, and it is identified that the neuron was the one that suffered the most changes during the experiments. It is responsible for representing when there is no fraud in the auction (the largest class of the data set). Therefore, it is expected that with a more significant number of samples in this context, the changes in clouds will be more remarkable, consequently generating a greater impact on the comparative similarity with their previous version (before the evolution training). In this evaluation, it is possible to identify that the fuzzy rules (presented below) are distinguishable from each other, as they have different antecedents and consequents.





Regarding the overlapping criteria of the generated Gaussians, it can be seen in Figure 10 (generated at the beginning of the training) that the centers of the generated fuzzy neurons are not superimposed. This confirms that there are differences between them. This criterion is also guaranteed by steps 5 and 6 of the Self-Organizing Direction-Aware Data Partitioning fuzzification approach. This figure presents the evaluation concerning the successive outbidding and bidding ratio dimensions (more relevant in the model analysis shown below).



Figure 10. Evaluation of overlapping of the three generated fuzzy neurons during the fuzzification process: successive outbidding versus bidding ratio.

The consistency of the generated fuzzy rules (Equation (51)) is visualized in Figure 11. In this figure, it is possible to see that the consistency relation of the fuzzy rules is violated only in a few evaluations in each of the three fuzzy rules generated. At the end of the experiment, all rules are consistent, concluding that model training corrected these inconsistencies when they appeared.





Concerning the criterion of completeness of the fuzzy rules, an evaluation based on ϵ – *completeness* was applied during training, according to Equation (52). The results visualized in Figure 12 confirm that the generated fuzzy ruleset meets this validation criterion throughout the experiment.



Figure 12. ϵ – *completeness* criteria during the training.

Changes in antecedents and consequents of fuzzy rules can also be observed and visualized in eFNN-SODA. Table 1 presents feedback from the model in the last sample evaluation that generated impacts on the first and third fuzzy rules.

Table 1. Interpretability concerning (degree of) changes in fuzzy neurons during the evolution phase.

Rule 1 changed with two membership functions, and this impacts changes in the consequent rule by a degree of 45%.
Rule 2 did not change.
Rule 3 changed with two membership functions, and this impacts changes in the consequent rule by a degree of 0.0003%.

The evaluation of the evolving behavior of the problem features is also a relevant part of the interpretability of the results. This variation can be seen in Figure 13 and identifies how each weight generated according to the separability criterion (Equation (34)) was significant during the fraud classification process. The most relevant dimensions were successive outbidding, bidding ratio, and winning ratio (which makes sense when compared with the analysis in Figure 6). Figures 14 and 15 present a graphical evaluation based on a scatter plot of the two dimensions that best contribute to the class separability criterion.

Finally, the expansion of knowledge given by eFNN-SODA demonstrates a different way of interpreting the problem through fuzzy rules connected with the relevance of the problem's features. The Fuzzy Hoeffding Decision Tree model [66] also generated an interpretative approach, which presents a decision tree according to fuzzy techniques in constructing its leaves. The Explicas [67] online solution (https://demos.citius.usc.es/Ex pliClas/datasets (accessed on 12 August 2022)) facilitates the acquisition of knowledge to be compared with the extraction of knowledge obtained in this paper. The decision tree formed by the Fuzzy Hoeffding Decision Tree is shown in Figure 16.



Figure 13. Feature separability criteria throughout the evaluations of the eFNN-SODA model.



Figure 14. Scatter plot of successive outbidding feature.



Figure 15. Scatter plot of bidding ratio feature.



Figure 16. Fuzzy Hoeffding Decision Tree interpretability.

5. Discussions

Discussions about the results obtained by the model are carried out based on the interpretability criteria to be evaluated in eFNN-SODA. This is required to ensure that the information extraction stated below is trustworthy and coherent, in addition to verifying the accuracy findings.

The results obtained by the model present factors that can be interpreted through fuzzy rules that solve the target problem. The fuzzy rules were validated concerning the criteria of simplicity (fewer rules with greater accuracy), distinguishability (not similar), consistency, with coverage in all samples evaluated, and that meet the criteria of completeness. Through these generated fuzzy rules, it is possible to identify and analyze their antecedents and consequents and the relationship between them, and finally, it is possible to identify the evolution of the model through the creation of new fuzzy rules.

An interpretive evaluation of fuzzy rules leads us to identify that all dimensions of the problem (except successive outbidding) have similar final relevance values for separating fraudulent behavior from normal behavior. This indicates that there is not such a clear separability between the two groups of samples in these dimensions. The evolving behavior of the weights (Figure 13) also demonstrates stability in the order of relevance of the features of the problem. They did not change their position of relevance throughout the experiment. The second highest weight value for the feature weight technique is the Bidding Ratio of 0.59. This demonstrates that, except for the Successive Outbidding dimension, all the others involved in the problem have poorly separable samples according to the Dy–Brodley separability criteria.

The comparison of interpretability extracted by the eFNN-SODA model and the Fuzzy Hoeffding Decision Tree model (Figure 16) has some similarities and distinctions. The tree model is easier for visualizing the relationships, but the fuzzy rule model is closer to human reasoning. Regarding their similarities to the root of the tree shown in Figure 16 that has the Successive Outbidding as its central dimension, other nodes arise for the following evaluations. This corroborates that this is the first value to be analyzed in the tree, thus being the most relevant dimension. The generated fuzzy rules also identify this factor, as Successive Outbidding significantly impacts rule antecedents.

The three most relevant dimensions to the problem were determined based on the fuzzy rules generated. With them, it is possible to elaborate future studies and technological tools that can avoid Successive Outbidding, Bidding Ratio, and Winning Ratio considered fraudulent. Successive Outbidding Technologies to avoid subsequent bids can be implemented in digital solutions that provide auction services. For example, if the fuzzy neural network identifies a user at high risk of fraud, they cannot place successive bids for a certain period. In the same way, when possible fraud in the bids is identified, they may undergo an audit (human or through some expert system) or be the subject of extra validation, such as the confirmation of some data or limitation on the number of bids that they can give for a product until their veracity is proven. Finally, different mechanisms can be provided for Winning Ratio fraud if the fuzzy neural network identifies that that winner has a fraudulent profile.

6. Conclusions

This work presented an evaluation through evolving fuzzy neural network models to address auction fraud problems. The eFNN-SODA model proposed in this paper achieved more than 98% in correctly classifying fraudulent situations or not within a highly unbalanced data set. eFNN-SODA can be considered the most straightforward approach compared with the other models evaluated in the experiment, as it performed the fraud classification tasks with fewer fuzzy rules and greater assertiveness. The model also presented distinguishable fuzzy rules because their antecedents and consequents are dissimilar.

Another factor verified in the experiments is that the inconsistencies of the fuzzy rules during the experiment were minimal and promptly corrected by training eFNN-SODA. Regarding the coverage criterion, as they are Gaussian neurons formed in the fuzzification process, eFNN-SODA also meets the requirement. Finally, the model meets the criterion of completeness throughout the experiment. Therefore, it can be concluded that the model's fuzzy rules presented at the end of the training meet the criteria of consistency, coverage, and completeness.

The evaluation of problem features helped the classifier to correctly identify the classes by assigning to each fuzzy rule antecedent a weight corresponding to its relevance in correctly identifying the problem classes. Factors such as the expansion of knowledge through a new approach, the visualization of the interpretability of the problem, and the analysis of antecedents and consequents were also presented in this article. They corroborate the statement that the fuzzy rules generated by the eFNN-SODA model are efficient and interpretable in solving fraud identification problems in an auction. The model proposed in this paper can generate technological products that evolve their knowledge as the behavior of fraud changes. This can happen by creating auxiliary expert systems to validate or investigate certain behaviors in the auctions. Fuzzy rules can be shown to system administrators and users, allowing corrective or analysis actions. With this type of system, both parties have clear and understandable logical relationships. Finally, creating an expert system to assist in data validation after the auction is also possible.

Possible future work can be conducted to develop new training techniques for model evolution and to use other neural structures to improve model accuracy. Another interesting topic to address is to evaluate or propose new methods for verifying the interpretability of models evolving neuro-fuzzy networks. The validation of the generated fuzzy rules can also be a potential focus of future research. Other (less unbalanced) data sets and auction fraud assessments are also encouraged to verify the model's adaptability in solving the problems of this behavior.

Author Contributions: Conceptualization, P.V.d.C.S. and E.L.; methodology, P.V.d.C.S. and H.R.B.; software, P.V.d.C.S.; validation, E.L.; formal analysis, E.L.; investigation, P.V.d.C.S. and A.J.G.; resources, P.V.d.C.S.; data curation, E.L.; writing—original draft preparation, P.V.d.C.S., H.R.B., and A.J.G.; writing—review and editing, E.L. and H.R.B.; visualization, P.V.d.C.S., A.J.G., and H.R.B.; supervision, E.L.; project administration, E.L.; funding acquisition, E.L. All authors have read and agreed to the published version of the manuscript.

Funding: Open Access Funding by the Austrian Science Fund (FWF), contract number P32272-N38, acronym IL-EFS.

Data Availability Statement: Publicly available datasets were analyzed in this study. These data can be found here: https://archive.ics.uci.edu/ml/datasets/Shill+Bidding+Dataset (accessed on 12 August 2022).

Conflicts of Interest: The authors declare no conflict of interest.

References

- 1. Mitchell, R.; Michalski, J.; Carbonell, T. *An Artificial Intelligence Approach*; Springer: Berlin/Heidelberg, Germany, 2013.
- 2. Chua, C.E.H.; Wareham, J. Fighting internet auction fraud: An assessment and proposal. *Computer* **2004**, *37*, 31–37. [CrossRef]
- 3. Button, M.; Nicholls, C.M.; Kerr, J.; Owen, R. Online frauds: Learning from victims why they fall for these scams. *Aust. N. Z. J. Criminol.* **2014**, *47*, 391–408. [CrossRef]
- 4. Alzahrani, A.; Sadaoui, S. Scraping and preprocessing commercial auction data for fraud classification. *arXiv* 2018, arXiv:1806.00656.
- Alzahrani, A.; Sadaoui, S. Clustering and labeling auction fraud data. In *Data Management, Analytics and Innovation*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 269–283.
- 6. Buckley, J.; Hayashi, Y. Fuzzy neural networks: A survey. *Fuzzy Sets Syst.* **1994**, *66*, 1–13. [CrossRef]
- 7. De Campos Souza, P.V. Fuzzy neural networks and neuro-fuzzy networks: A review the main techniques and applications used in the literature. *Appl. Soft Comput.* **2020**, *92*, 106275. [CrossRef]
- 8. Novák, V. Fuzzy natural logic: Towards mathematical logic of human reasoning. In *Towards the Future of Fuzzy Logic*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 137–165.
- Gu, X.; Angelov, P.; Kangin, D.; Principe, J. Self-Organised direction aware data partitioning algorithm. *Inf. Sci.* 2018, 423, 80–95. [CrossRef]
- 10. Huang, G.B.; Zhu, Q.Y.; Siew, C.K. Extreme learning machine: Theory and applications. *Neurocomputing* **2006**, *70*, 489–501. [CrossRef]
- De Campos Souza, P.V.; Lughofer, E. An evolving neuro-fuzzy system based on uni-nullneurons with advanced interpretability capabilities. *Neurocomputing* 2021, 451, 231–251. [CrossRef]
- De Campos Souza, P.V.; Lughofer, E.; Guimaraes, A.J. Evolving Fuzzy Neural Network Based on Uni-nullneuron to Identify Auction Fraud. In Proceedings of the Joint Proceedings of the 19th World Congress of the International Fuzzy Systems Association (IFSA), the 12th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT), and the 11th International Summer School on Aggregation Operators (AGOP), Bratislava, Slovakia, 19–24 September 2021; pp. 314–321. [CrossRef]
- 13. Haykin, S.S.; Haykin. *Neural Networks and Learning Machines*; Pearson Education Upper Saddle River: Hoboken, NJ, USA, 2009; Volume 3.
- 14. Fausett, L.V. *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications;* Prentice-Hall Englewood Cliffs: Hoboken, NJ, USA, 1994; Volume 3.
- 15. Zhang, Z. Artificial neural network. In *Multivariate Time Series Analysis in Climate and Environmental Research;* Springer: Berlin/Heidelberg, Germany, 2018; pp. 1–35.

- Rumelhart, D.E.; Durbin, R.; Golden, R.; Chauvin, Y. Backpropagation: The basic theory. *Backpropagation Theory Archit. Appl.* 1995, 1–34.
- 17. Pedrycz, W. Neurocomputations in relational systems. IEEE Trans. Pattern Anal. Mach. Intell. 1991, 13, 289–297. [CrossRef]
- Mithra, K.; Sam Emmanuel, W. GFNN: Gaussian-Fuzzy-Neural network for diagnosis of tuberculosis using sputum smear microscopic images. J. King Saud Univ.-Comput. Inf. Sci. 2021, 33, 1084–1095. [CrossRef]
- 19. Hu, Q.; Gois, F.N.B.; Costa, R.; Zhang, L.; Yin, L.; Magaia, N.; de Albuquerque, V.H.C. Explainable artificial intelligence-based edge fuzzy images for COVID-19 detection and identification. *Appl. Soft Comput.* **2022**, *123*, 108966. [CrossRef]
- 20. Salimi-Badr, A.; Ebadzadeh, M.M. A novel learning algorithm based on computing the rules' desired outputs of a TSK fuzzy neural network with non-separable fuzzy rules. *Neurocomputing* **2022**, 470, 139–153. [CrossRef]
- 21. Nasiri, H.; Ebadzadeh, M.M. MFRFNN: Multi-Functional Recurrent Fuzzy Neural Network for Chaotic Time Series Prediction. *Neurocomputing* **2022**, *507*, 292–310. [CrossRef]
- 22. Pan, Q.; Li, X.; Fei, J. Adaptive Fuzzy Neural Network Harmonic Control with a Super-Twisting Sliding Mode Approach. *Mathematics* **2022**, *10*, 1063. [CrossRef]
- Amirkhani, A.; Shirzadeh, M.; Molaie, M. An Indirect Type-2 Fuzzy Neural Network Optimized by the Grasshopper Algorithm for Vehicle ABS Controller. *IEEE Access* 2022, 10, 58736–58751. [CrossRef]
- Algehyne, E.A.; Jibril, M.L.; Algehainy, N.A.; Alamri, O.A.; Alzahrani, A.K. Fuzzy Neural Network Expert System with an Improved Gini Index Random Forest-Based Feature Importance Measure Algorithm for Early Diagnosis of Breast Cancer in Saudi Arabia. *Big Data Cogn. Comput.* 2022, 6, 13. [CrossRef]
- 25. Novák, V.; Perfilieva, I.; Dvorak, A. Insight into Fuzzy Modeling; John Wiley & Sons: Hoboken, NJ, USA, 2016.
- 26. Murinová, P.; Novák, V. The theory of intermediate quantifiers in fuzzy natural logic revisited and the model of "Many". *Fuzzy Sets Syst.* 2020, *388*, 56–89. [CrossRef]
- 27. Novák, V.; Perfilieva, I. Forecasting direction of trend of a group of analogous time series using F-transform and fuzzy natural logic. *Int. J. Comput. Intell. Syst.* 2015, *8*, 15–28. [CrossRef]
- Nguyen, L. Integrating The Probabilistic Uncertainty to Fuzzy Systems in Fuzzy Natural logic. In Proceedings of the 2020 12th International Conference on Knowledge and Systems Engineering (KSE), Can Tho City, Vietnam, 12–14 November 2020; pp. 142–146.
- Xu, X.; Ding, X.; Qin, Z.; Liu, Y. Classification Models for Medical Data with Interpretative Rules. In Proceedings of the International Conference on Neural Information Processing; Springer: Berlin/Heidelberg, Germany, 2021; pp. 227–239.
- 30. Nemat, R. Taking a look at different types of e-commerce. World Appl. Program. 2011, 1, 100–104.
- 31. Trevathan, J. Getting into the mind of an "in-auction" fraud perpetrator. Comput. Sci. Rev. 2018, 27, 1–15. [CrossRef]
- 32. Abidi, W.U.H.; Daoud, M.S.; Ihnaini, B.; Khan, M.A.; Alyas, T.; Fatima, A.; Ahmad, M. Real-Time Shill Bidding Fraud Detection Empowered With Fussed Machine Learning. *IEEE Access* 2021, *9*, 113612–113621. [CrossRef]
- Anowar, F.; Sadaoui, S. Detection of Auction Fraud in Commercial Sites. J. Theor. Appl. Electron. Commer. Res. 2020, 15, 81–98. [CrossRef]
- Anowar, F.; Sadaoui, S.; Mouhoub, M. Auction Fraud Classification Based on Clustering and Sampling Techniques. In Proceedings of the 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), Orlando, FL, USA, 17–20 December 2018; pp. 366–371.
- 35. De Campos Souza, P.V.; Lughofer, E. Evolving fuzzy neural classifier that integrates uncertainty from human-expert feedback. *Evol. Syst.* **2022**, 1–23. [CrossRef]
- Škrjanc, I.; Iglesias, J.A.; Sanchis, A.; Leite, D.; Lughofer, E.; Gomide, F. Evolving fuzzy and neuro-fuzzy approaches in clustering, regression, identification, and classification: A survey. *Inf. Sci.* 2019, 490, 344–368. [CrossRef]
- 37. Lughofer, E. On-line assurance of interpretability criteria in evolving fuzzy systems—Achievements, new concepts and open issues. *Inf. Sci.* 2013, 251, 22–46. [CrossRef]
- 38. De Campos Souza, P.V.; Lughofer, E. An advanced interpretable Fuzzy Neural Network model based on uni-nullneuron constructed from n-uninorms. *Fuzzy Sets Syst.* 2020, 426, 1–26. [CrossRef]
- 39. Lughofer, E. On-line incremental feature weighting in evolving fuzzy classifiers. Fuzzy Sets Syst. 2011, 163, 1–23. [CrossRef]
- 40. Angelov, P.; Gu, X.; Príncipe, J. A generalized methodology for data analysis. IEEE Trans. Cybern. 2017, 48, 2981–2993. [CrossRef]
- 41. Angelov, P.; Gu, X.; Kangin, D. Empirical data analytics. *Int. J. Intell. Syst.* 2017, 32, 1261–1284. [CrossRef]
- 42. Gu, X.; Angelov, P.P. Self-organising fuzzy logic classifier. *Inf. Sci.* **2018**, 447, 36–51. [CrossRef]
- 43. Gu, X.; Angelov, P.P.; Príncipe, J.C. A method for autonomous data partitioning. Inf. Sci. 2018, 460, 65–82. [CrossRef]
- 44. Watson, D.F. Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes. *Comput. J.* **1981**, 24, 167–172. [CrossRef]
- 45. Angelov, P.; Yager, R. A new type of simplified fuzzy rule-based system. Int. J. Gen. Syst. 2012, 41, 163–185. [CrossRef]
- 46. De Campos Souza, P.V.; Lughofer, E.; Rodrigues Batista, H. An Explainable Evolving Fuzzy Neural Network to Predict the k Barriers for Intrusion Detection Using a Wireless Sensor Network. *Sensors* **2022**, *22*, 5446. [CrossRef]
- 47. Dy, J.; Brodley, C. Feature Selection for Unsupervised Learning. J. Mach. Learn. Res. 2004, 5, 845–889.
- 48. Qin, S.; Li, W.; Yue, H. Recursive PCA for Adaptive Process Monitoring. J. Process Control 2000, 10, 471–486.
- 49. Klement, E.P.; Mesiar, R.; Pap, E. Triangular Norms; Springer Science & Business Media: Berlin, Germany, 2013; Volume 8.
- 50. Hirota, K.; Pedrycz, W. OR/AND neuron in modeling fuzzy set connectives. IEEE Trans. Fuzzy Syst. 1994, 2, 151–161. [CrossRef]

- 51. Pedrycz, W.; Reformat, M.; Li, K. OR/AND neurons and the development of interpretable logic models. *IEEE Trans. Neural Netw.* **2006**, *17*, 636–658. [CrossRef]
- 52. Huang, G.B.; Chen, L.; Siew, C.K. Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Trans. Neural Netw.* 2006, 17, 879–892. [CrossRef]
- 53. Albert, A. Regression and the Moore-Penrose Pseudoinverse; Elsevier: Amsterdam, The Netherlands, 1972.
- Rosa, R.; Gomide, F.; Dovzan, D.; Skrjanc, I. Evolving neural network with extreme learning for system modeling. In Proceedings of the 2014 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS), Linz, Austria, 2–4 June 2014; pp. 1–7.
- 55. Lughofer, E. Evolving Fuzzy Systems—Methodologies, Advanced Concepts and Applications; Springer: Berlin/Heidelberg, Germany, 2011.
- 56. Lughofer, E.; Bouchot, J.L.; Shaker, A. On-line elimination of local redundancies in evolving fuzzy systems. *Evol. Syst.* 2011, 2, 165–187. [CrossRef]
- 57. Demšar, J.; Leban, G.; Zupan, B. FreeViz—An intelligent multivariate visualization approach to explorative analysis of biomedical data. *J. Biomed. Inform.* 2007, 40, 661–671. [CrossRef]
- 58. Hoffman, P.; Grinstein, G.; Marx, K.; Grosse, I.; Stanley, E. DNA visual and analytic data mining. In Proceedings of the Visualization '97 (Cat. No. 97CB36155), Phoenix, AZ, USA, 19–24 October 1997; pp. 437–441. [CrossRef]
- de Campos Souza, P.V.; Torres, L.C.B.; Guimaraes, A.J.; Araujo, V.S.; Araujo, V.J.S.; Rezende, T.S. Data density-based clustering for regularized fuzzy neural networks based on nullneurons and robust activation function. *Soft Comput.* 2019, 23, 12475–12489. [CrossRef]
- 60. Souza, P.; Ponce, H.; Lughofer, E. Evolving fuzzy neural hydrocarbon networks: A model based on organic compounds. *Knowl.-Based Syst.* **2020**, *203*, 106099. [CrossRef]
- 61. De Campos Souza, P.V.; Rezende, T.S.; Guimaraes, A.J.; Araujo, V.S.; Batista, L.O.; da Silva, G.A.; Silva Araujo, V.J. Evolving fuzzy neural networks to aid in the construction of systems specialists in cyber attacks. *J. Intell. Fuzzy Syst.* **2019**, *36*, 6743–6763. [CrossRef]
- 62. Soares, E.; Angelov, P.; Gu, X. Autonomous Learning Multiple-Model zero-order classifier for heart sound classification. *Appl. Soft Comput.* **2020**, *94*, 106449. [CrossRef]
- 63. Angelov, P.; Gu, X. Autonomous learning multi-model classifier of 0-Order (ALMMo-0). In Proceedings of the 2017 Evolving and Adaptive Intelligent Systems (EAIS), Ljubljana, Slovenia, 31 May–2 June 2017; pp. 1–7. [CrossRef]
- 64. Leite, D.; Costa, P.; Gomide, F. Evolving granular neural network for semi-supervised data stream classification. In Proceedings of the 2010 International Joint Conference on Neural Networks (IJCNN), Barcelona, Spain, 18–23 July 2010; pp. 1–8. [CrossRef]
- 65. Bifet, A.; Holmes, G.; Kirkby, R.; Pfahringer, B. MOA: Massive Online Analysis. *J. Mach. Learn. Res.* **2010**, *11*, 1601–1604.
- Ducange, P.; Marcelloni, F.; Pecori, R. Fuzzy Hoeffding Decision Tree for Data Stream Classification. Int. J. Comput. Intell. Syst. 2021, 14, 946–964. [CrossRef]
- Alonso, J.M.; Bugarín, A. ExpliClas: Automatic Generation of Explanations in Natural Language for Weka Classifiers. In Proceedings of the 2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Hyderabad, India, 7–10 July 2019; pp. 1–6. [CrossRef]