2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

Knowledge Extraction about beer classification using evolving fuzzy neural networks

1st Paulo Vitor de Campos Souza
Intelligent Digital Agents Research Group-Fondazione Bruno Kessler
Trento, Italia
0000-0002-7343-5844

***Abstract*—Evolving Fuzzy Neural Networks (EFNNs) are well-regarded for their interpretability and proficiency in pattern classification tasks. However, their accuracy may need to be improved when confronted with limited samples for specific classes or the emergence of new classes in the data stream. To overcome this limitation, we applied the EFNN-Gen, a novel approach that integrates a priori knowledge through generalist rules to solve a beer classification problem. These rules are derived from assessing the specificity of Gaussian functions within the first layer neurons of the EFNN. They represent expert knowledge about the classification problem and are aimed at enhancing the network's performance. Experimental tests conducted on the Beer dataset, a real-world multiclass pattern classification dataset, demonstrate that integrating generalist rules leads to a significant accuracy improvement of 97.14%.**

***Index Terms*—Evolving fuzzy neural network, generalist rules, interpretability, EFNN-Gen.**

I. Introduction

Experts' Feedback has proven beneficial for several intelligent models in overcoming their challenges. Such feedback can be obtained from logical propositions, like logical rules, or through evaluating the final results achieved [1]. Consequently, incorporating a priori knowledge while training intelligent models becomes a viable solution to enhance their accuracy and performance in tackling classification problems. This expert knowledge about a problem is crucial in helping intelligent models adapt and improve their performance as new patterns and behaviors emerge from the data. Leveraging this strategy enables the effective implementation of evolving system models, which can continuously adjust their parameters in response to new data for ongoing evaluation. As intelligent models adapt to new information, their knowledge can further contribute to determining a priori knowledge relevant to the problem at hand [2].

Evolving Fuzzy Neural Networks (EFNN) are models capable of dynamically defining their architectural elements and updating factors related to their training as new samples are evaluated [3] [4]. This adaptive behavior empowers EFNNs to tackle complex problems of diverse nature effectively [5]. These challenges span a wide range, from addressing different diseases [6]–[8] to more general areas of study, such as auction fraud detection, power quality disturbance classification [9], and even Typhoon path prediction [10]. Notwithstanding their exceptional problem-solving abilities across various domains, EFNNs encounter a challenge: their performance may decline when faced with a scarcity of samples for specific classes, particularly when new classes emerge in the data stream.

It is worth acknowledging that, in interpretability, Evolving Fuzzy Neural Networks (EFNNs) hold a distinct advantage over other machine-learning approaches documented in the literature. Consequently, it becomes imperative to explore methods that can enhance EFNNs' problem-solving accuracy while retaining their superior interpretability.

This study aims to implement the model proposed in [11] (EFNN-Gen) to address the classification of various types of beer. A key enhancement to EFNN-Gen involves incorporating generalist rules based on the specificity of the first layer Gaussian neurons. These rules enable the model to leverage a priori knowledge in the dataset, allowing for a comparison with the results obtained through fuzzy rules.

Using feedback to incorporate a priori knowledge in model training helps to provide initial knowledge, making the learning process more coherent and adaptable to new data. Expert systems exemplify this approach, where prior knowledge is combined with data-extracted knowledge. The feedback can come from familiar users or domain experts, with the latter providing direct interaction based on a priori knowledge. Professionals can verify the knowledge base extracted from data, creating expert systems and disseminating knowledge through training using hybrid models.

The primary focus of this paper is to showcase the exceptional problem-solving capabilities of EFNN-Gen, particularly in handling complex challenges. By leveraging the beer dataset a priori knowledge, we aim to conduct a comparative analysis, contrasting the results of the model presented in this article with the pre-existing understanding of the problem at hand.

In addition to the introduction, this paper presents the section of theoretical references (Section II), linked to the concepts of evolving models and related work on fuzzy rules ideas. In section III, the layers and training of the model and the new proposition regarding generalist rules are exposed to the reader. The experiments and their results will be highlighted in section IV. Finally, conclusions about the activities performed in the paper will be presented in Section VI.

II. Literature Review

A. Evolving System concepts

Fuzzy neural network models and Neuro-fuzzy systems stand out as significant advancements in the realm of hybrid models [12] [5]. Intelligent models have demonstrated advantages and limitations as researchers develop techniques and algorithms to address various societal challenges. Combining two or more intelligent techniques offers the opportunity to harness the strengths of diverse models, leading to more efficient solutions for complex problems. Evolving systems, encompassing fuzzy neural network models and neuro-fuzzy systems, excel in adaptively solving problems by dynamically adjusting system parameters and internal structures as new samples are presented [13]. This form of training closely mirrors the comparative assessment of human behavior, where problem-solving knowledge adapts to emerging situations [3] [4]. Evolving models have found widespread applications in diverse scientific fields, particularly within data stream mining processes. Unlike traditional batch offline train models, evolving models efficiently process successively arriving data online, owing to fast single-pass update techniques and swift convergence to optimal solutions. Consequently, these models have been successfully employed in various industrial problems with increasing complexity [14], as well as within the realms of healthcare and finance. For a comprehensive overview of the concrete applications of evolving models, please refer to [4] and [3].

B. Specificity

The notion of specificity serves as a metric to quantify the information content within a studied fuzzy subset. It shares similarities with entropy in probability theory, as seen in fuzzy sets and possibilities theory [15]. The specificity measure gauges the extent to which a fuzzy subset unequivocally points to a single element as its member. It is closely linked to the inverse of the set's cardinality. Yager discerns between imprecision and specificity, where the former pertains to the gradual lack of clarity in association within certain sets, while the latter relates to the absence of precise knowledge concerning specific attributes [15].

Specificity measures find applications in decision-making, serving as a metric of tranquility in the decision process. A more specific set of choices reduces decision-related anxiety. Additionally, specificity plays a crucial role in measuring the performance of decision support systems and determining the usefulness of the information they provide. The relationship between increased specificity and enhanced information utility is central in this theory [15]. Correctness and specificity are highly desirable qualities in decision support and knowledge-based systems. Evaluating their performance requires consideration of both these aspects [16]. Specificity also holds significance in deductive reasoning systems, as demonstrated by Dubois and Prade using the concept of minimal specificity [17]. This principle highlights the importance of selecting manifestations resulting in outputs with the least specificity, minimizing unjustified additional information [15].

In the context of this study, our objective is to assess the minimum specificity values of Gaussian membership functions formed within the first layer of the model. This endeavor seeks a generalized understanding of a specific group of characteristics rather than a specialized one. Leite and Škrjanc [18] emphasize that specificity measures tend to approach their maximum values when an object closely aligns with a single element. Therefore, the specificity (Sp) of a Gaussian membership function (Ω) can be expressed as [19]:

$$\mathrm{Sp}(\Omega) = 1, \qquad \text{if}\, \Omega \equiv \{x\}\, \text{(singleton)} \tag{1}$$

$$\mathrm{Sp}(\Omega) = 0, \qquad \text{if}\, \Omega = \emptyset \tag{2}$$

$$\mathrm{Sp}(\Omega^1) \leq \mathrm{Sp}(\Omega^2), \qquad \text{if}\, \Omega^1 \supseteq \Omega^2. \tag{3}$$

and $0 \leq \mathrm{Sp}(\Omega) \leq 1$. X= $(X_1 \times ... \times X_j \times ... \times X_n)$ represents the Cartesian product space assembles of an interval-valued cloud [19].

III. EFNN-Gen architecture and training

The EFNN-Gen model presents a notable advantage by harnessing domain expertise to enhance the beer classification process. The model's architecture comprises three layers and a fuzzy rule structure based on uni-nullneurons, enabling the formation of fuzzy rules with both AND- and OR-connections in their antecedents. This flexibility, distinct from previous EFNN approaches, allows for a more comprehensive knowledge representation. In the first layer, the specificity of the Gaussians originating from the clusters is evaluated after creating the clouds through a data density-based technique. Those that meet the knowledge generalization criterion form the generalist rules, encapsulating expert knowledge about the analyzed dataset. In the offline phase, the pseudo-inverse concepts are employed to define the rule consequents and the weights of the Singleton neuron in the third layer. These Singleton consequents, representing certainty levels of the rules to possible output classes, play a vital role in the model's defuzzification process, providing the expected outputs for pattern classification. In the evolving training phase, the rule consequents are determined using an indicator-based recursive weighted least squares approach. Additionally, an incremental feature weighting technique considers the class separability power of features, assigning weights to Gaussian neurons' features. This technique helps mitigate the curse of dimensionality effects when certain features receive low weights, as they are down-weighted or masked in the neuron construction. An architectural representation of the EFNN-Gen can be seen in Fig. 1.

A. First Layer: Autonomous Fuzzification Method

The process applied in the input layer enables the construction of inference systems based on logical rules, representing knowledge through linguistic expressions [20]. This approach, known as EFNN-Gen, utilizes the Autonomous Data Partitioning (ADPA) fuzzification process proposed by Gu et al. [21]. ADPA is a non-parametric technique that autonomously identifies maximum data-density locations, forming clouds that serve as the basis for Gaussian fuzzy neurons in the first layer [20]. Each input variable (x_j) is associated with several clouds, denoted by A_{lj}, l = 1... L, serving as activation functions for corresponding neurons. Consequently, the first layer outputs

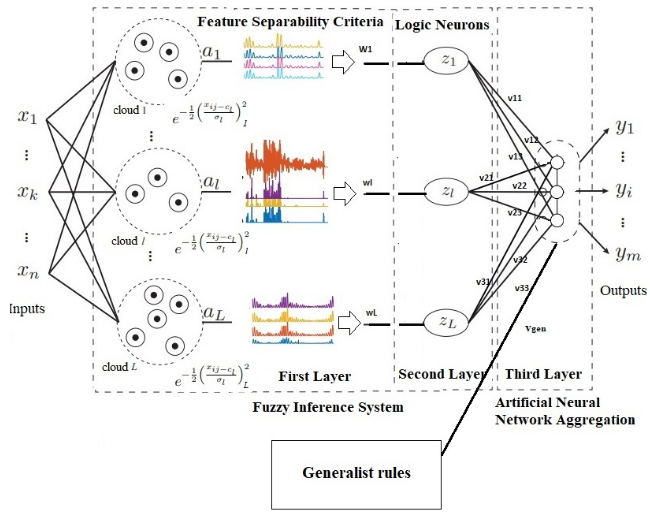


Fig. 1. EFNN-Gen Architecture [11]

membership degrees linked to input values, represented as a_{jl} = μ_l^A for j = 1... N and l = 1 ...L, where N represents the number of inputs, and L signifies the number of clouds for each input. The Gaussian fuzzy neurons of EFNN-Gen can be expressed by:

$$\Omega(x_j, c_{jl}, \sigma_{jl}) = e^{-\frac{1}{2}\left(\frac{x_j - c_{jl}}{\sigma_{jl}}\right)^2},\ for\, j = 1...\, n,\ l = 1\, ...\, L \quad (4)$$

where c_{jl} denotes the jth coordinate of the center of the lth neuron and σ_{jl} denotes the standard deviation of the lth neuron in the jth input dimension.

1) Autonomous fuzzification process: ADPA, introduced by Gu et al. [21], effectively partitions the input space into data clouds using local modes. A key advantage of ADPA is its ability to objectively represent the original data distribution, forming clouds without predefined shapes. This contrasts with traditional prototype-based clustering techniques [22], which rely on predefined cluster centers and their influence ranges, often using predetermined local density functions and cluster numbers. By leveraging data density concepts, ADPA dynamically determines the number of clouds necessary to solve the problem based on the organization and essence of the data [23]. This adaptive process involves evaluating four recursively calculated parameters derived from empirical data analysis (EDA): cumulative proximity, standardized eccentricity, density, and multimodal typicality [23]. About the EDA parameters used by EFNN-Gen, consider seeing more information in [11] and [21].

The offline approach in EFNN-Gen utilizes EDA operators to define the initial clouds during the initial training stage. As new samples are evaluated, the fuzzification strategy recursively updates these EDA parameters, allowing the previously formed clouds to adapt and represent the new data. ADPA operates in several steps: initializing the data cloud by selecting the first sample in the data stream, and then evolving the parameters and distances with the insertion of new samples [21]. The fuzzification approach continuously updates EDA operators as the model evaluates new data, leading to the redefinition of the system structure. This recursive update of EDA parameters is further described by Gu et al. [21]. EFNN-Gen uses C_k as the number of local modes at the k time. More details about cloud formation, please see [11].

Moreover, in [20], an extension of ADPA was introduced to govern the evolution of rules. This extension involves incorporating a relative term to verify whether the output derived from the fuzzification method aligns with the expected label of the evaluated class. This condition can be formulated as follows [20]:

$$IF\ \left(x_k - \varrho_k^n \leq \frac{\Delta_k^c}{2}\right)$$
$$AND\ (argmax(\rho_{n,.}) == (y_k - 1)\ and\ \nu \geq 0.80)$$
$$THEN\ (x_k\ is\ assigned\ to\ \varrho_k^n)$$

let y_k denote the actual label of the k-th stream sample $\boldsymbol{x}k$, and ρ represents a matrix. The (i, j)-th entry in this matrix indicates the number of samples falling into cloud i and belonging to class j. Consequently, $argmax(\rho n, .)$ refers to the argument maximum of the n-th row, which corresponds to the nearest cloud n, with class labels starting from 0. Additionally, ν represents the frequency of the majority class within the closest cloud (n) compared to the frequencies of other classes. Consequently, the majority class must match the class of the current samples and be significant. This condition reduces the probability of significant class overlaps within the clouds.

2) Definition of weights in first layer neurons: Online Incremental Characteristic Weight Calculation: The proposal introduced by Campos Souza and Lughofer [20] incorporates the concepts of relevance and reducibility in neurons within the first layer. In conventional fuzzy neural network approaches, the weights of Gaussian neurons formed during the fuzzification process are randomly defined. However, by identifying the classes of the problem and considering this information to determine the weights, it becomes possible to discern the most relevant dimensions (and subsequently neurons) for class identification. This objective is achieved through an incremental technique that calculates weights based on the problem's dimensions, utilizing the Dy-Brodley separability criterion [24] for the analyzed classes. The feature importance is determined by assessing the contribution of each feature in defining the separation between classes. This approach enables EFNN-Gen neurons to align with the relevance of new data blocks in the dimensions, facilitating a smooth and adaptive learning process for evolving fuzzy classifiers. Dimensions contributing to class separation receive values close to 1, whereas dimensions with negligible contributions are assigned values close to zero. By identifying and detaching irrelevant dimensions, this strategy promotes reducibility in the complexity of interpretation, enhancing interpretability assessments [25]. Further details regarding the definition and calculation of these weights can be found in de Campos Souza and Lughofer [20].

B. Second Layer: Fuzzy Rules

In EFNN-Gen, the second layer employs neurons based on 2-uninorms [26]. This neuron structure enables using uninorm and nullnorm operators within the same group of fuzzy rules. Consequently, uni-nullneurons can create an IF-THEN rule base with different rule connectors, such as AND or OR. The inputs for these neurons consist of Gaussian neurons and their corresponding weights (w_{il} where i = 1... N and l = 1... L), which are estimated through the feature weighting approach described in the preceding subsection. The 2-uninorms are based on conventional uninorms [27], expressed as follows:

$$U(x,y) = \begin{cases} g\ T(\frac{x}{g}, \frac{y}{g}), & x, y\ \in [0, g] \\ g + (1-g)\ S\ (\frac{x-g}{1-g}, \frac{y-g}{1-g}), & x, y\ \in (g, 1]. \end{cases} \quad (5)$$

where T is a *t-norm* (probabilistic sum), S is a *t-conorm* (product) and g is the neutral element of the uninorm. Therefore, the 2-uninorm used in EFNN-Gen is defined by [20]:

$$N^{un}(x,y,\beta,g,\lambda) = \begin{cases} \beta U_1(\frac{x}{\beta}, \frac{y}{\beta}),\ x,y \in [0,\beta] \\ \beta + (1-\beta) U_2(\frac{x-\beta}{1-\beta}, \frac{y-\beta}{1-\beta}),\ x,y \in (\beta, 1] \end{cases}$$

In the context of EFNN-Gen, we utilize two distinct uninorms denoted as U_1 and U_2. The uninorm U_1 is governed by Eq. (5) with a neural element (identity) equal to $\frac{g}{\beta}$, while U_2 is represented by Eq. (5) using $(\frac{\lambda-\beta}{1-\beta})$ as the neutral element [28]. Combining the elements (g, λ, β) within a 2-uninorm allows us to achieve varying rule antecedent connectors, alternating between AND and OR. For a detailed exploration of the combinations of these elements and their respective rule antecedents' connectives, please refer to [20].

The uni-nullneuron creates a key role in computing the model output for each pair (a_i, w_i), effectively transforming them into a single value denoted as $b_i = p(a_i,\ w_i)$ through a conditional transformation (p). The function (p) carries out the unified aggregation of the transformed values, denoted as N^{un} $(b_1, b_2 ... b_n)$, as defined by [20]:

$$p(w,a,\beta,g,\lambda) = \begin{cases} wa + \bar{w}\frac{g}{\beta},\ if\ U_1 \\ wa + \bar{w}\frac{\lambda-\beta}{1-\beta},\ if\ U_2 \end{cases} \quad (6)$$

Finally, the uni-nullneuron (for the l-th neuron) can be expressed by [20]:

$$z_l = UNI^{NUL}(w,a,\beta,g,\lambda) = N^{un}{}_{i=1}^{n} p(w_i, a_{il}, \beta, g, \lambda) \quad (7)$$

The w_i denotes the weight of the i-th feature, estimated using the incremental separability check techniques described in the preceding section, while a_{il} represents the activation level of the l-th neuron in the i-th feature. These neurons offer high interpretability since a group of generated rules can possess diverse meanings based on the connectors used in the rule antecedents. This characteristic enhances the feasibility of comprehending the knowledge extracted from the model, even for individuals who are not experts in artificial intelligence techniques [20].

Uni-nullneurons develop fuzzy rules that can be analyzed as follows:

$$\begin{aligned} R_1:\ If\ x_1\ is\ A_1^1\ with\ impact\ w_{11}... \\ and/or_{(g,\lambda,\beta)}\ x_2\ is\ A_1^2\ with\ impact\ w_{21}... \\ Then\ y_1\ is\ [v_{11}...v_{1C}] \\ R_2:\ If\ x_1\ is\ A_2^1\ with\ impact\ w_{12}... \\ and/or_{(g,\lambda,\beta)}\ x_2\ is\ A_2^2\ with\ impact\ w_{22}... \\ Then\ y_2\ is\ [v_{21}...v_{2C}] \\ ....R_L:\ If\ x_1\ is\ A_L^1\ with\ impact\ w_{1L}... \\ and/or_{(g,\lambda,\beta)}\ x_2\ is\ A_L^2\ with\ impact\ w_{2L}... \\ Then\ y_L\ is\ [v_{L1}...v_{LC}] \end{aligned} \quad (8)$$

In the context of this problem, C represents the number of classes, and $\vec{v}i = [vi1, ..., v_{iC}]$ can be regarded as a certainty vector, where each entry signifies the certainty of the rule to its output class. Consequently, our rules can convey certainty degrees in the outputs, thereby enhancing interpretability, particularly concerning the level of class overlap in the local region represented by the rule.

C. Definition of generalist rules

The determination of generalist rules is contingent upon the evaluation of the specificity of Gaussians generated during the fuzzification process. As the uni-nullneuron performs the aggregation of Gaussian neurons from the first layer, it enables the assessment of data representativeness based on the sensitivity of the extracted Gaussians via the ADPA technique.

In [19], the authors define $Sp(\Omega)$ as a measure of specificity for a set, indicating the extent to which Ω refers exclusively to a single element. When Ω is a subset of X, the general class of specificity measures over continuous domains can be defined as [29]:

$$Sp(\Omega) = \int_0^{\alpha_{\max}} F(\Upsilon(\Omega_\alpha)) d\alpha \quad (9)$$

where $\Omega_\alpha = \{x : \Omega(x) \geq \alpha\}$ is the α-level of Ω, $\alpha_{\max}$ is the height of Ω, Υ is a monotonic measure and $F : [0,1] \to [0,1]$ is a function such that $F(0) = 1, F(1) = 0$, and $0 \leq F(x_1) \leq F(x_2)$ for $x_1 > x_2$.

Level sets of Gaussian membership functions are intervals Ω_α whose centers and radii are of the form $c \pm \sqrt{-2\sigma^2 \ln(\alpha)}$, so when setting [19]:

$$M \triangleq \sqrt{-2\sigma^2 \ln{(\alpha)}} \quad (10)$$

then

$$\Omega_\alpha = [c - M, c + M]. \quad (11)$$

The Lebesgue–Stieltjes measure [30] is used to define Υ in a totally bounded domain $X = [t, d]$:

$$\Upsilon(\Omega_\alpha) = \frac{\text{wdt}(\Omega_\alpha)}{\text{wdt}(X)} = \frac{c + M - (c - M)}{d - t} = \frac{2M}{d - t} \quad (12)$$

where wdt(.) means the interval width which is equal to the absolute difference between the endpoints of the interval [19]. If all $x \in [0,1]$, then $[t,d]=[0,1]$.

Assuming F(Φ)= $1-\Phi$, we get:

$$F(\Upsilon(\Omega_\alpha)) = 1 - \frac{2M}{d-t}. \quad (13)$$

Therefore, using the definition of Specificity and the α-level set Ω_α, we have [19]:

$$\begin{aligned} Sp(\Omega_\alpha) &= \int_0^1 F(\Upsilon(\Omega_\alpha))d\alpha && (14)\\ &= 1 - \int_0^1 \Upsilon(\Omega_\alpha)\mathrm{d}\alpha && (15)\\ &= 1 - \frac{2}{d-t}\int_0^1 M\mathrm{d}\alpha. && (16)\end{aligned}$$

In terms of the dispersion σ and α [19]:

$$Sp(\Omega_\alpha) = 1 - \frac{2\sqrt{2}\sigma\sqrt{-\ln(\alpha)}}{d-t}. \quad (17)$$

Finally, let's consider $\Omega^i = \{\Omega_1^i, ..., \Omega_j^i, ..., \Omega_n^i\}$ as a set of Gaussian functions, where Ω_j^i has a domain X_j, and $X^n = X_1 \times ... \times X_j \times ... \times X_n$. Utilizing α-cuts, we can form an interval Gaussian neuron a^i in X^n. In [19], the authors define the specificity of an n-dimensional Gaussian neuron a^i as the average of the specificity measure of each of its n components, expressed as:

$$Spa(a^i) = 1 - \frac{2\sqrt{2}}{n}\sum_{j=1}^{n}\frac{\sigma_j^i\sqrt{-\ln(\alpha)}}{d_j - t_j}. \quad (18)$$

This indicates that the Gaussians with higher standard deviations can cover a greater number of samples, effectively representing an overall view of the analyzed problem. Higher specificity implies that the 'linguistic value' associated with the cluster's membership functions is more specific. In contrast, lower specificity suggests a broader coverage of the generated Gaussians.

Building upon this concept, our work introduces the concept of generalist rules (φ). These rules are determined by identifying the set of Gaussians (η) with the least specificity among the model's first-layer neurons. Hence, this technique selects several fuzzy rules (η) based on their specificity value. These rules are defined during the offline training phase of the model, and can be mathematically expressed as:

$$\begin{aligned} Z\varphi_{(\eta)} = \{R_i, i \in [\![1, L]\!] \mid \\ |\{j \in [\![1, L]\!], j \neq i \mid Spa(a^j) < Spa(a^i)\}| \leq \eta - 1\} \end{aligned} \quad (19)$$

where $\eta \in \{1, ..., L\}$ defines the number of generalist rules to be considered in the model.

D. Third Layer: Neural Aggregation Network

The third layer of the model plays a crucial role in obtaining the final consequent parameters to determine the output aggregation layer of the fuzzy neural network. This layer consists of a single neuron, which can be mathematically expressed as given in [20]:

$$y = \Xi\left(\sum_{j=0}^{l} f_\Gamma(z_j, v_j)\right) \quad (20)$$

where f_Γ represent the activation function of the neuron, which is linear in this particular study. We define $z_0 = 1$ and v_0 as the bias, while z_j and v_j, for $j = 1, ..., l$, denote the output of the fuzzy neurons in the second layer and their corresponding weights, respectively.

Additionally, the function Ξ serves a purpose similar to the sign function in binary pattern problems, producing a value of 1 for positive numbers and -1 for negative numbers. Mathematically, the Ξ function is expressed as:

$$\Xi = \begin{cases} 1, \ if \ \sum_{j=0}^{l} f_\Gamma(z_j, v_j) \ > 0 \\ -1, \ if \ \sum_{j=0}^{l} f_\Gamma(z_j, v_j) \ < 0 \end{cases} \quad (21)$$

E. Model's Training

The EFNN-Gen training comprises two distinct stages. The first stage involves utilizing the Moore-Penrose pseudo-inversion method [31] to define the network weights. Subsequently, during the evolving phase of the model, the indicator-based recursive weighted least squares (I-RWLS) technique [32] is employed to update the consequent parameters $\vec{v} \in \mathbb{R}^L$. The values of these parameters can be estimated in the offline phase and evolving training, respectively, as described in [20]:

$$\vec{v} = Z^+\vec{y} \quad (22)$$

with $\vec{y}$ the target vector containing all the class labels of all samples.

$$\eta = \vec{z}^t Q^{t-1}\left(\psi + (\vec{z}^t)^T Q^{t-1}\vec{z}^t\right)^{-1} \quad (23)$$

$$Q^t = (\mathrm{I}_{L^t} - \eta^T\vec{z}^t)\psi^{-1}Q^{t-1} \quad (24)$$

$$\vec{v}^t = \vec{v}^{t-1} + \eta^T(y^t - \vec{z}^t\vec{v}^{t-1}) \quad (25)$$

In the EFNN-Gen model, the pseudo-inverse of the Moore-Penrose matrix Z^+ is computed from the activation levels of all neurons stored in matrix Z, where each row represents the activation levels of a neuron, and each column corresponds to a stream sample. The activation levels of all neurons in the current stream sample are denoted by $\vec{z}^t$, and η represents the current Kalman gain (row) vector. The matrix ILs^t is an identity matrix based on the number of neurons in the second layer, denoted as $L_s^t \times L_s^t$. The parameter $\psi \in]0,1]$ can be a forgetting factor but is typically set to 1 (no forgetting) by default. The inverse Hessian matrix Q is initialized as ωILs^t, where ω is a large value (e.g., 1000) [20]. As for managing parameters in EFNN-Gen, it only requires specifying the number of generalist rules (η) that will be incorporated into the model's architecture. The I-RWLS method and the incremental feature weighting technique are completely parameter-free, and for the ADPA clustering approach, the default setting for Δ_k^c is used, as described in the original paper.

IV. Experiments

The experiments to be carried out in this paper seek to prove the increased assertiveness of evolving fuzzy neural networks in classifying patterns through the insertion of generalist knowledge for a beer dataset. Tests will be performed to compare the results with a previous research, allowing a solid comparison of the applicability of the generalist rules proposed in this paper. For this purpose, data sets provided by [33]. After the tests, the generated fuzzy rules will be compared with the problem domain fuzzy rules. All tests were performed on a computer with the following settings: Intel (R) Core (TM) i7-6700 CPU 3.40GHz with 16GB RAM.

A. Model evaluation and approaches used in the test

In the EFNN-Gen experiments, the accuracy evaluation is established through a trend line approach, particularly in stream-mining scenarios. In such cases, the accuracy is updated in a cumulative manner using the following methodology:

$$Accuracy(K+1) = \frac{Accuracy(K) * K + I_{\hat{y}=y}}{K+1}, \qquad (26)$$

The accuracy is determined using the indicator function, denoted as I, in the given context. When the prediction matches the actual label, i.e., $\hat{y} = y$, the indicator function takes the value 1. Otherwise, it takes 0 ($Accuracy(0) = 0$). Following the accuracy update, the model itself undergoes an update, adhering to an interleaved-test-and-then-train protocol, which is a widely adopted approach in the data stream mining community [34]. This protocol assesses the model ahead prediction capabilities while incrementally adapting and evolving.

B. Validation of fuzzy rules results based on a prior knowledge

The study in [33] used a fuzzy inference system to generate the model's rules. They obtained 87.50% accuracy using nine fuzzy rules for beer classification, while expert-taught knowledge achieved 81.25% with eight rules. The authors adopted antecedents and connectives to generate the fuzzy rules in this experiment. However, expert knowledge was insufficient to fully solve the target problem.

In the experiment, 30% of samples were used for training, and the remaining 70% were used for evaluating the model's performance. The simulation was carried out using the EFNN-Gen model, which was chosen based on simplicity. The model used uni-nullneurons and a linear function in the third layer to generate rules with the same antecedent connectives as expert knowledge. For this experiment we decided to highlight 2 expert rules.

The final result achieved 97.14% accuracy in identifying beers, using eight fuzzy rules extracted from the analyzed dataset. Fig. 2 displays the model evaluation results in trend lines.

Fig. 3 presents the confusion matrix generated by the experiment.

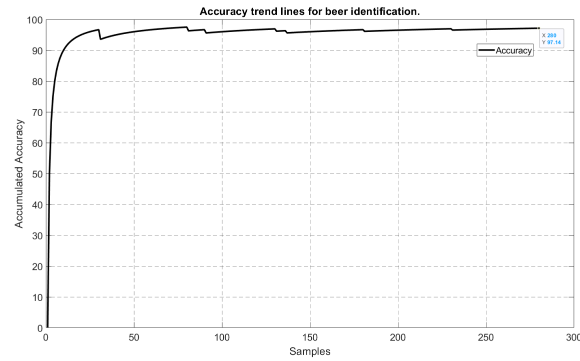
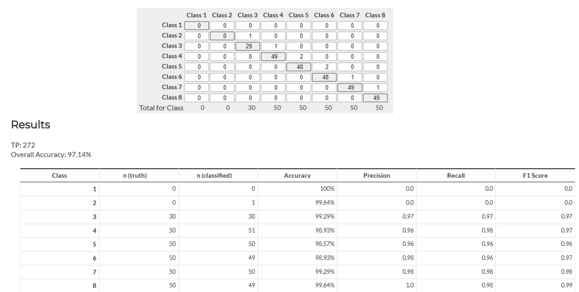


Fig. 2. Accuracy trend lines for beer identification -EFNN-Gen.



	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8
Class 1	0	0	0	0	0	0	0	0
Class 2	0	0	1	0	0	0	0	0
Class 3	0	0	29	1	0	0	0	0
Class 4	0	0	0	49	2	0	0	0
Class 5	0	0	0	0	48	2	0	0
Class 6	0	0	0	0	0	48	1	0
Class 7	0	0	0	0	0	0	49	1
Class 8	0	0	0	0	0	0	0	49
Total for Class	0	0	30	50	50	50	50	50

Results

TP: 272
Overall Accuracy: 97.14%

Class	n (truth)	n (classified)	Accuracy	Precision	Recall	F1 Score
1	0	0	100%	0.0	0.0	0.0
2	0	1	99.64%	0.0	0.0	0.0
3	30	30	99.29%	0.97	0.97	0.97
4	50	51	98.93%	0.96	0.98	0.97
5	50	50	98.57%	0.96	0.96	0.96
6	50	49	98.93%	0.98	0.96	0.97
7	50	50	99.29%	0.98	0.98	0.98
8	50	49	99.64%	1.0	0.98	0.99

Fig. 3. Confusion matrix generated by EFNN-Gen in identifying the type of beer.

V. Evaluation of fuzzy rules generated by the model according to prior knowledge established by an expert.

This subsection presents aspects related to beer classification using EFNN-Gen. Figs. 4, 5, and 6 illustrate the three dimensions of the problem, which were derived from the expert's knowledge mentioned in the paper [33]. This enables a direct comparison between the Gaussian functions generated by EFNN-Gen and the trapezoidal functions extracted by the specialist.

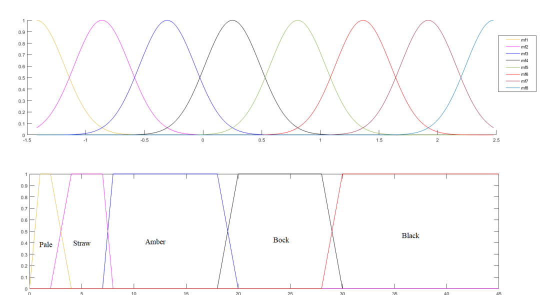


Fig. 4. Comparison of membership functions obtained by the EFNN-Gen model and expert knowledge for the Color dimension.

Eight fuzzy rules were generated at the end of the model training as logical to expect from an eight-class problem. They

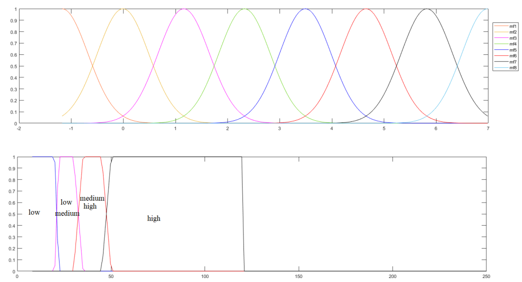


Fig. 5. Comparison of membership functions obtained by the EFNN-Gen model and expert knowledge for the Bitterness dimension.

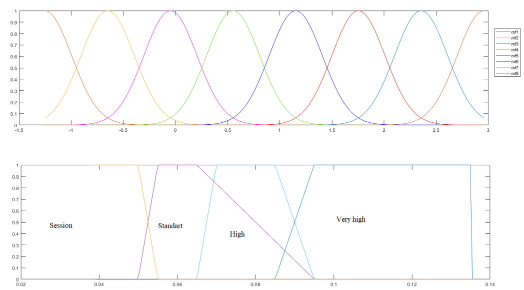


Fig. 6. Comparison of membership functions obtained by the EFNN-Gen model and expert knowledge for the Strength dimension.

are presented below.

1. If (Color is Brown or Black) with impact 1.00 and (Bitterness is high) with impact 0.16 and (Strength is Session) with impact 0.44 then (Beer is out1).

2. If (Color is Pale or Straw) with impact 1.00 and (Bitterness is Medium high) with impact 0.16 and (Strength is Session) with impact 0.44 then (Beer is out2).

3. If (Color is Pale or Straw) with impact 1.00 and (Bitterness is low) with impact 0.16 and (Strength is Session) with impact 0.44 then (Beer is out3).

4. If (Color is Amber or Brown) with impact 1.00 and (Bitterness is high) with impact 0.16 and (Strength is Session or Standart) with impact 0.44 then (Beer is out4).

5. If (Color is Straw or Amber) with impact 1.00 and (Bitterness is medium high or high) with impact 0.16 and (Strength is Session) with impact 0.44 then (Beer is out5).

6. If (Color is Amber) with impact 1.00 and (Bitterness is high) with impact 0.16 and (Strength is Session) with impact 0.44 then (Beer is out6).

7. If (Color is Pale or Straw) with impact 1.00 and (Bitterness is low medium or medium high) with impact 0.16 and (Strength is Session) with impact 0.44 then (Beer is out7).

8. If (Color is Brown or Black) with impact 1.00 and (Bitterness is high) with impact 0.16 and (Strength is Session or Standart or High or very high) with impact 0.44 then (Beer is out8).

Table I presents the outputs for each of the generated fuzzy rules.

TABLE I
CORRESPONDING OUTPUT OF EACH FUZZY NEURON-EFNN-GEN

Rule	Blanche	Lager	Pilsner	IPA	Stout	Barleywine	Porter	Belgian Strong Ale
out1	medium	none	medium	medium	none	none	none	medium
out2	none	medium	none	none	medium	medium	medium	large
out3	large	none	large	none	none	none	none	none
out4	large	none	large	large	none	none	none	none
out5	medium	none	medium	medium	none	none	none	none
out6	medium	none	medium	medium	none	none	none	none
out7	none	medium	none	none	medium	medium	medium	none
out8	none	large	none	none	large	large	large	large

The fuzzy rules generated by EFNN-Gen in this experiment were evaluated based on their ability to identify different types of beers. Each rule's identification capacity was analyzed concerning the three dimensions: Color, Bitterness, and Strength, using the prior knowledge provided by the specialists. Overall, the results indicated that some rules accurately identified specific beer types, while others showed inconsistencies with the expert's knowledge. The rules were divided into categories based on their identification probabilities. Specific rules (e.g., Rules 1, 3, and 4) could correctly identify specific beer types, such as Blanche, Pilsen, and IPA. These identifications aligned well with the characteristics highlighted by the beer specialists. However, some rules (e.g., Rules 2, 6, and 7) showed medium-level identification probabilities and needed consistent with the expert's knowledge, particularly for the Stout, Barleywine, and Belgian Strong Ale types. Despite some inconsistencies, the EFNN-Gen model achieved an impressive accuracy of 97.14In conclusion, the fuzzy rules extracted by EFNN-Gen showed promising results and aligned with the prior knowledge to a considerable extent. The model's high accuracy highlights its potential for beer classification, surpassing the performance of previous studies in this domain.

VI. CONCLUSION

This study presented the EFNN-Gen model, an evolving fuzzy neural network designed to classify different types of beer. By incorporating generalist rules into its architecture, the model can leverage prior knowledge about the data extracted by analyzing the Gaussian specificity generated during the fuzzification process. This approach proved beneficial in enhancing beer identification accuracy. Interestingly, the study identified that the model's learning capacity is subject to a maximum saturation point, determined by the number of pre-defined rules integrated. In future research, it would be valuable to investigate the relationship between the saturation degree of generalist rules and the initial number of generated fuzzy rules in different datasets. Additionally, exploring the evolution of rules during the model's iterative process, rather than relying solely on replacements, could yield further insights. Moreover, evaluating alternative factors for Gaussian assessment could lead to a more robust definition of generalist knowledge. Notwithstanding its contributions, this study acknowledges limitations concerning evaluating the interplay between generalist rules and the model's generalization ability. Addressing these factors in subsequent research endeavors could shed more light on EFNN-Gen's capabilities and potential applications.

ACKNOWLEDGMENT

We acknowledge the support of the PNRR project FAIR - Future AI Research (PE00000013), under the NRRP MUR program funded by the NextGenerationEU. Also, thanks to Edwin Lughofer, who passed away this year and significantly contributed to evolving fuzzy systems.

REFERENCES

[1] A. K. Lang and D. M. Kosak, "System and method employing individual user content-based data and user collaborative feedback data to evaluate the content of an information entity in a large information communication network," Nov. 9 1999, uS Patent 5,983,214.

[2] E. Lughofer, "On-line assurance of interpretability criteria in evolving fuzzy systems – achievements, new concepts and open issues," *Information Sciences*, vol. 251, pp. 22–46, 2013.

[3] I. Škrjanc, J. A. Iglesias, A. Sanchis, D. Leite, E. Lughofer, and F. Gomide, "Evolving fuzzy and neuro-fuzzy approaches in clustering, regression, identification, and classification: A survey," *Information Sciences*, vol. 490, pp. 344–368, 2019.

[4] E. Lughofer, "Evolving fuzzy systems — fundamentals, reliability, interpretability and useability," in *Handbook of Computational Intelligence*, P. Angelov, Ed. New York: World Scientific, 2016, pp. 67–135.

[5] P. V. de Campos Souza, "Fuzzy neural networks and neuro-fuzzy networks: A review the main techniques and applications used in the literature," *Applied Soft Computing*, vol. 92, p. 106275, 2020.

[6] P. V. de Campos Souza and E. Lughofer, "Identification of heart sounds with an interpretable evolving fuzzy neural network," *Sensors*, vol. 20, no. 22, 2020.

[7] P. V. de Campos Souza, E. Lughofer, and A. J. Guimaraes, "An interpretable evolving fuzzy neural network based on self-organized direction-aware data partitioning and fuzzy logic neurons," *Applied Soft Computing*, vol. 112, p. 107829, 2021.

[8] O. Shatalova, S. Filist, N. Korenevskiy, R. Taha Al-kasasbeh, A. Shaqadan, Z. Protasova, M. Ilyash, and A. Rybochkin, "Application of fuzzy neural network model and current-voltage analysis of biologically active points for prediction post-surgery risks," *Computer Methods in Biomechanics and Biomedical Engineering*, pp. 1–13, 2021.

[9] D. Leite, L. Decker, M. Santana, and P. Souza, "Egfc: Evolving gaussian fuzzy classifier from never-ending semi-supervised data streams – with application to power quality disturbance detection and classification," in *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2020, pp. 1–9.

[10] Z.-X. Yang, H.-J. Rong, P. P. Angelov, and Z.-X. Yang, "Statistically evolving fuzzy inference system for non-gaussian noises," *IEEE Transactions on Fuzzy Systems*, pp. 1–1, 2021.

[11] P. V. de Campos Souza and E. Lughofer, "Efnn-gen — a uni-nullneuron-based evolving fuzzy neural network with generalist rules," in *2022 IEEE International Conference on Evolving and Adaptive Intelligent Systems (EAIS)*, 2022, pp. 1–10.

[12] W. Pedrycz, "Fuzzy neural networks and neurocomputations," *Fuzzy Sets and Systems*, vol. 56, no. 1, pp. 1–28, 1993.

[13] P. Angelov, D. Filev, and N. Kasabov, *Evolving Intelligent Systems — Methodology and Applications*. New York: John Wiley & Sons, 2010.

[14] E. Lughofer and M. Sayed-Mouchaweh, *Predictive Maintenance in Dynamic Systems — Advanced Methods, Decision Support Tools and Real-World Applications*. New York: Springer, 2019.

[15] R. R. Yager, "Measures of specificity over continuous spaces under similarity relations," *Fuzzy Sets and Systems*, vol. 159, no. 17, pp. 2193–2210, 2008, theme: Fuzzy Relations.

[16] ——, "Measuring the quality of linguistic forecasts," *International journal of man-machine studies*, vol. 21, no. 3, pp. 253–257, 1984.

[17] D. Dubois and H. Prade, "The principle of minimum specificity as a basis for evidential reasoning," in *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*. Springer, 1986, pp. 75–84.

[18] D. Leite and I. Škrjanc, "Ensemble of evolving optimal granular experts, owa aggregation, and time series prediction," *Information Sciences*, vol. 504, pp. 95–112, 2019.

[19] D. Leite, G. Andonovski, I. Škrjanc, and F. Gomide, "Optimal rule-based granular systems from data streams," *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 3, pp. 583–596, 2020.

[20] P. V. de Campos Souza and E. Lughofer, "An evolving neuro-fuzzy system based on uni-nullneurons with advanced interpretability capabilities," *Neurocomputing*, vol. 451, pp. 231–251, 2021.

[21] X. Gu, P. P. Angelov, and J. C. Príncipe, "A method for autonomous data partitioning," *Information Sciences*, vol. 460, pp. 65–82, 2018.

[22] C. Borgelt, "Prototype-based classification and clustering," Ph.D. dissertation, Otto-von-Guericke-Universität Magdeburg, 2005, magdeburg, Germany.

[23] P. Angelov, X. Gu, and D. Kangin, "Empirical data analytics," *International Journal of Intelligent Systems*, vol. 32, no. 12, pp. 1261–1284, 2017.

[24] J. Dy and C. Brodley, "Feature selection for unsupervised learning," *Journal of Machine Learning Research*, vol. 5, pp. 845–889, 2004.

[25] E. Lughofer, "On-line incremental feature weighting in evolving fuzzy classifiers," *Fuzzy Sets and Systems*, vol. 163, no. 1, pp. 1–23, 2011, theme: Classification and Modelling.

[26] J. C. Fodor, R. R. Yager, and A. Rybalov, "Structure of uninorms," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 5, no. 04, pp. 411–427, 1997.

[27] R. R. Yager and A. Rybalov, "Uninorm aggregation operators," *Fuzzy sets and systems*, vol. 80, no. 1, pp. 111–120, 1996.

[28] P. V. de Campos Souza and E. Lughofer, "An advanced interpretable fuzzy neural network model based on uni-nullneuron constructed from n-uninorms," *Fuzzy Sets and Systems*, 2020.

[29] R. R. Yager, "Measures of specificity over continuous spaces under similarity relations," *Fuzzy Sets and Systems*, vol. 159, no. 17, pp. 2193–2210, 2008.

[30] V. I. Bogachev and M. A. S. Ruas, *Measure theory*. Springer, 2007, vol. 1.

[31] A. Albert, *Regression and the Moore-Penrose pseudoinverse*. Elsevier, 1972.

[32] G.-B. Huang, L. Chen, C. K. Siew *et al.*, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *IEEE Trans. Neural Networks*, vol. 17, no. 4, pp. 879–892, 2006.

[33] G. Castellano, C. Castiello, and A. M. Fanelli, "The fisdet software: Application to beer style classification," in *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2017, pp. 1–6.

[34] A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer, "MOA: Massive online analysis," *Journal of Machine Learning Research*, vol. 11, pp. 1601–1604, 2010.