Implication of Hamacher T-norm on Two Fuzzy-Rough Rule Induction Algorithms

A. Naumoski, G. Mirceva and K. Mitreski
Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University in Skopje, Skopje, North Macedonia
{andreja.naumoski,georgina.mirceva,kosta.mitreski}@finki.ukim.mk

***Abstract* - From the rule induction algorithms we can obtain models in If-Then form that are very easy to be interpreted by humans. To further improve this class of algorithms, in this paper we focus on QuickRules and Vaguely Quantified Rough fuzzy-rough rule induction algorithms, by introducing the novel Hamacher T-norm. It is important to know that T-norms as well as the fuzzy tolerance relationship metrics, implicators and vague quantifiers play an important role in model accuracy because they are used to calculate the lower and upper approximations. For this purpose, in our models' evaluation, we use five fuzzy tolerance relationship metrics to evaluate the performance of the models that are obtained with the new Hamacher T-norm. The AUC ROC metric was used to evaluate the performance, and later was used to evaluate the statistical significance. The results revealed that fuzzy tolerance relationship metrics have greater influence than the k-parameter from the Hamacher T-norm on models' performance, and this was also compared to the vaguely quantified algorithm that uses vague quantifiers. For future work, we plan to conduct further investigation of the influence of another T-norms and fuzzy tolerance relationship metrics on this type of algorithms.**

***Keywords* - Hamacher T-norm; Fuzzy tolerance relationship metrics; Fuzzy rough sets; Rule induction algorithms**

I. Introduction

The rule induction algorithms have a long-lasting impact on the machine learning research community. Various improvements have been made on this class of machine learning algorithms, ranging by modifying the algorithms parameters, heuristic functions, or introducing a different approach, which can result in a new algorithm. One direction that many machine learning algorithms have moved in is introduction of new or modified versions of machine learning algorithms, like deep learning, but there is also introduction of new mathematical elements in the previously well-established algorithms, like fuzzy-rough theory. Introducing fuzzy theory into machine learning algorithms produce models that can better handle noise in the data, as well as handling data instances that belong in many sets at the same time. As well, fuzzy-rough based algorithms can easily handle both nominal and numeric attribute values. As we noted here, there are some advantages of introducing fuzzy-rough theory into rule induction algorithm. Also, there are some drawbacks that need to be addressed, like curse of dimensionality or increased computational power needed to handle fuzzy and fuzzy-rough calculations. There are solutions like pre-processing or feature selection [1] that can reduce the number of input attributes, but still preserve the information needed to produce a high-performance model. More importantly, the fuzzy-rough theory can build models where in many cases no parameters are required in the process of feature selection, thus eliminating the subjective influence of the modeler.

Rough set theory as a scientific discipline in the last decade [2], has found many applications in different problem domains, mainly as an interesting mathematical theory. However, in the literature there are attempts to integrate the fuzzy-rough set theory and rule induction process to produce a method that can be used for rule induction. But, before there was a merger between fuzzy and rough set theory, some of the research work done in this field was regarding the fuzzy rule induction methods [3], [4], but no fuzzy-rough concepts were implemented, so therefore we don't know how much we don't know from the incomplete input dataset. Two special incomplete dataset cases were studied using rough set theory for rule induction: lost values [5] and "do not care" [6] condition. To cope with these problematic data elements, in fuzzy-rough theory, two concepts were introduced: lower and upper approximations, as well as boundary regions as stated in [7]. The researchers in [8] combine these reduct concepts and worked out a fuzzy-rough approach for the rule induction algorithm. They used the fuzzy-rough rule induction algorithm to build a tree and then convert the tree into rules. Other method exists for generating different types of rule sets, like certain and possible rules sets [9], different kinds of first-order fuzzy rule induction [10], fuzzy learning unordered fuzzy rule induction [11], semantics-preserving modelling [12]. The QuickRules algorithm [1] uses the previously stated advantages of the fuzzy-rough concepts and combines several other improvements. Another fuzzy-rough rule induction algorithm that uses as vaguely quantified rough set measure (VQRules) [13], replaces the traditional dependency measure to better handle the noise and uncertainty that exist inside the dataset. Many of the real-world datasets have noise and uncertainty due to measurements problems or investigated phenomena that don't offer possibility to be measured, so the need of such a method becomes more and more obvious. Changes in these algorithms and introducing different fuzzy tolerance relationship metrics (FTRMs), T-norms and implicators can be used to improve the model accuracy. Therefore, the aim of this paper is to investigate the influence of not only the FTRMs, but also the influence of the different T-norms on several well-known datasets [14], [15]. These experiments will include not only the VQRules algorithm, but also the QuickRules algorithm. To evaluate the models' performance, we use the AUC ROC metric, and later the obtained results for the models are used to calculate the statistical significance using a two-step procedure [16].

The rest of the paper is organized as follows. Section II gives a description of the concepts drawn from the fuzzy-rough set theory, and later we use them to describe the QuickRules and VQRules fuzzy-rough rule induction algorithms. In Section III we present the experimental setup used for evaluation and later we discuss the obtained results through statistical significance analysis of the results. Section IV concludes the paper and outline our future work.

II. Fuzzy Rough Rule Induction Algorithms

A. Concepts

In this section, we will present briefly the most important concepts of fuzzy-rough rule induction algorithm. Let's start with the rough set theory, that we need to define the first element, the information system. For that purpose, we define a couple (X, A) as a sets of finite objects (X) and attributes (A), either can have nominal or numeric values. When attribute a from the attribute set A is nominal, then it takes values from a finite set, and comparison is done with a-indiscernibility relationship R_a, defined with (1).

$$R_a = \{(x,y) \in X^2 \mid a(x) = a(y)\} \quad (1)$$

In the case, when a is real valued, then its values are drawn from closed interval and compared with fuzzy a-indiscernibility relationship R_a, defined with the equations (2), (3), (4), (5) or (6).

$$FTRM_1 = 1 - \frac{|a(x) - a(y)|}{|a_{\max} - a_{\min}|} \quad (2)$$

$$FTRM_2 = \exp\left(-\frac{(a(x) - a(y))^2}{2\sigma_a^2}\right) \quad (3)$$

$$FTRM_3 = \max\left(\min\left(\frac{(a(y) - (a(x) - \sigma_a))}{(a(x) - (a(x) - \sigma_a))}, \frac{((a(x) + \sigma_a) - a(y))}{((a(x) + \sigma_a) - a(x))}\right), 0\right) \quad (4)$$

$$FTRM_4 = \max\left(0, \min\left(1, \frac{\beta - \alpha * |a(x) - a(y)|}{a_{\max} - a_{\min}}\right)\right),\ \alpha = 0.5, \beta = 1.0 \quad (5)$$

$$FTRM_5 = \begin{Bmatrix} 1, & a = b \\ 0, & a \neq b \end{Bmatrix} \quad (6)$$

For each attribute (a) we denote in some of the metrics, a standard deviation (σ_a), while α and β symbols denote constants in (5). These relationships are known as fuzzy tolerance relationship metrics (FTRM). The FTRM metrics have the property of reflectiveness and symmetric relation, and therefore can be used to approximate element x as element member of set X. If we consider a subset of set A, for example subset B, the fuzzy B-indiscernibility relationship is defined with (7). In (7), the T-norm is represented with symbol T, and attribute a that is element in subset B can take both nominal and numeric type of elements.

$$R_B(x,y) = T(R_a(x,y)), a \in B \quad (7)$$

T-norm is a mathematical function T: $[0, 1]^2$ -> [0, 1], which satisfies the following conditions: $T(x, y) = T(y, x)$, $T(x, 1) = x$, as well as monotonicity and associativity. The Hamacher T-norm (T_H) is defined with (8) for $k \geq 0$.

$$T_H(x,y) = \begin{cases} 0, & \text{if } \ k = x = y = 0 \\ \dfrac{xy}{k + (1-k)(x + y - xy)} \end{cases} \quad (8)$$

Next, with equations (9) and (10) we set the definition of lower and upper approximations for a given fuzzy set A.

$$\underline{R_B}(y) = inf[I(R_B(x,y), A(x))], x \in X \quad (9)$$

$$\overline{R_B}(y) = sup[T(R_B(x,y), A(x))], x \in X \quad (10)$$

As we can note from eq. (9), there is another variable that need to be calculated and that is the implicator function (I). The mathematical definition of implicator is a function I: $[0, 1]^2$ -> [0, 1], which satisfies certain conditions, like decreases of the first argument and increases in second argument, and for boundary conditions it fulfils the following conditions: $I(1, 0) = 0$, $I(0,1) = 0$ and $I(1, 1) = 1$, according [17]. In the evaluation of the influence of T_H on two fuzzy rough rule induction algorithms, we use Kleene-Dienes implicator defined as $I(x, y) = \max(1 - x, y)$. By using these concepts and the concept of decision system, we can now construct the basic building blocks of process for fuzzy-rough knowledge discovery rule induction. The decision system in terms of fuzzy-rough terminology is defined as (X, A U $\{d\}$) and this forms an information system. Within this information system, d is an attribute that is excluded from set A, and can take different type of values, either numeric or nominal. Consequently, if d is numeric, then we are talking about regression task at hand, or when is nominal we talk about classification problem. By using the information regrading the d values, set X can be and is divided into several non-overlapping decision classes, and this is done by using the concept of decision reduct. According [18], decision reduct is defined from set B as a subset of A, for which B-positive region is a fuzzy set in X. The X set contains each object y that have equal values for d (decision) with approximately equal values for the attributes in B [18], calculated with (11).

$$POS_B(y) = \left(\bigcup_{i=1}^{p} \underline{R_B}\right)(y) = \max_{i=1}^{p} \inf_{x \in X} Ip(R_B(x,y), A_i(x)) \quad (11)$$

Now, we use these positive regions of sets B and A to distinguish the predictive ability d of the attributes in B by calculating the ratio between these sums. This degree of dependency of d on B is reflected with (12). It is important to note that subset B of A is preserving the decision-making power of A. If the ratio between these positive regions is equal to 1, it is called super reduct. If this super reduct cannot be further reduced, or both positive regions of A and B sets are equal, then it is called decision reduct. To find all the decision reducts for the input dataset, we must search the entire model space, and this is NP-hard problem.

$$\frac{|POS_B|}{|POS_A|} = \frac{\sum_{x \in X} POS_B(x)}{\sum_{x \in X} POS_A(x)} \quad (12)$$

Consequently, the implementation of this process in a form of algorithm is always difficult, so it will be sufficient that the implemented algorithm generates a single decision reduct. Following this practice, researchers have tried to follow this principle and have come out with many different algorithms. Two algorithms are considered in this paper: the Quick Reduct [1] and VQRules algorithm [13], whose main difference is the way how the lower and upper approximations are calculated.

B. *QuickRules Algorithm*

The first algorithm that we be discussed here is the QuickRules algorithm developed by [1], which embraces the concepts developed earlier with crisp rule induction algorithms and adds on the benefits from the fuzzy-rough set theory and its concepts. The QuickRules algorithm generates rules from training samples that resides in the input dataset, known as the decision system, and then using the fuzzy-rough concept of decision reduct tries to build an optimal model based on the data. Furthermore, the decision reduct concept ensures that each element from the input dataset will be judged when QuickRules algorithms generate the models. With QuickRules, the object from the dataset is placed in the set of upper approximation, while the objects that are related to all the elements in the dataset are set into the set of lower approximations. This is recursively repeated, while in the process, each class of the target attribute is related to each rule. In this way, as a part of the reduct process, each class is a subset of a decision concept. That's why it is possible to give a relevant prediction from the input dataset values according [1]. Beside all the advantages of the QuickRules algorithmic concepts, the algorithm is still not perfect, and it can generate rules that are too narrow or too wide. To avoid such pitfalls of the algorithm, some ideas of feature selection were introduced [1]. This is done by making fuzzy rules maximally cover the training objects only with minimum number of attributes. In this way, the rule induction process is optimized, but will also consume more processing power and it will take more time to build the model. The process of model building begins with setting the rule set to null, followed by creation of another set where the built rules will be grown and kept. Since we work in the fuzzy domain, the coverage degree for each entity is determined by set of rules. To get this, calculation metric is used in a form of function (see eq. (13)), whereas input argument is set to be the fuzzy set. As output, the covered function will obtain set of objects that are contained in the input fuzzy set.

$$\text{covered}(C) = \{x \in X \mid Cov(x) = POS_A(x)\} \quad (13)$$

According [1], the focus of the covered function is the relationship between the membership function and the positive region of the input dataset. If this is true, then the constructed rule will be included in the final dataset, and thus is true only if the rule is not previously covered. In the rule set, attributes that fulfill the condition that examined object of interest should belong maximally to POS_B U $\{a\}$, calculated according (5). And finally, we can produce a rule for the input objects, and that rule can be included in the final set, only when the tolerance class for the object fully included the decision concept [1]. To ensure that the final set of rules contains rules that have high information gain and they are compact, the algorithm checks the coverage value of the newly created rule, and this is added to the existing set, only if it has higher value of the known rule in the set. This means that if the already existing rule has lower coverage value compared to the newly found rule, then the existing rule is removed from the set. The rule adding continues if any new rule fulfills the given conditions, and stops when certain object is fully covered, in other words, terminated when all the objects from the input dataset are tested and cover all the positive region. The final output of the QuickRules algorithm is a rule set that covers all the objects of interest.

C. *VQRules Algorithm*

The VQRules algorithm described in details by [13] uses vague quantifiers [19] to define the previously defined two approximation concepts of a fuzzy-rough set. To accomplish this, the VQRules algorithm upgrades the upper approximation calculations with external quantifiers, and to calculate the lower approximations VQRules uses universal quantifiers. Using these modifications, the VQRules algorithm can better deal with some of the everyday problems of the datasets, like noise, uncertainty, and inconsistency. According to [13], where the definition of the vague quantified is given, this concept is used to alleviate the defined two approximations so that the object belongs to the upper approximation to the extent of some elements, and the object belongs to lower approximation to extend the most elements. In many cases the object can belong to two approximations, so we can use the VQRules quantifiers to determine which set it belongs to. Definition of the vague quantifiers by the VQRules algorithm is given with equations (14) and (15) by the authors of [13].

$$\underline{R_{Q1}}(y) = Q_1\left(\frac{|R_B \cap A|}{R_B}\right) = Q_1\left(\frac{\sum_{x \in X} \min(R_A(x,y), A(x))}{\sum_{x \in X} R_A(x,y)}\right) \quad (14)$$

$$\overline{R_{Q2}}(y) = Q_2\left(\frac{|R_B \cap A|}{R_B}\right) = Q_2\left(\frac{\sum_{x \in X} \min(R_B(x,y), A(x))}{\sum_{x \in X} R_B(x,y)}\right) \quad (15)$$

From the equations (14) and (15), the fuzzy set intersection can be calculated using min operations between the T-norm as first element and fuzzy cardinality as a second element, with sigma-count operation. To calculate R_A or R_B, we can use several different FTRMs, to better fit the model to the data and produce models that have better performance on unseen data. In this research paper we experimentally evaluate five FTRM metrics on the two fuzzy-rough rule induction algorithms with T_H.

III. Experiments and Results

A. Experimnetal Setup

Here we present the experimental setup of the experiments conducted to evaluate the influence of the T_H norm on the model accuracy. Furthermore, we briefly preset the VQRules parameters setup, as well as the 19 datasets that are used for evaluation [14], [15]. Dataset description with their respective number of instances and attributes are given in Table 1. In our experimental setup we used T_H with different values for the k parameter (1, 2, 3, 5 and 10), and for each of these values full evaluation was done with five FTRMs, noted as ^{5}FTRM$_3$ - represented the k = 5 and FTRM$_3$ metric with (4). The Kleene-Dienes Implicator was used for all the experiments. For the QuickRules algorithm, no additional setup was required, while for the VQRules algorithm two parameters were set at a = 0.2 and b = 1.0, according (16).

$$Q_{(a,b)}(x)=\begin{cases}0, & x\le a\\ \dfrac{2(x-a)^2}{(b-a)^2}, & a\le x\le \dfrac{a+b}{2}\\ 1-\dfrac{2(x-a)^2}{(b-a)^2}, & \dfrac{a+b}{2}\le x\le b\\ 1, & b\le x\end{cases} \tag{16}$$

Evaluation of the obtained models with the different parameters settings was done using the AUC ROC metric, and a two-step procedure was used to calculate the statistical significance [16]. First, the non-parametric Friedman [20] and Quade test [21] are conducted used to calculate the rank of each algorithm parameter settings. For each of the 19 datasets, the applied algorithm settings are sorted according to their Train/Test AUC ROC and a rank is assigned. Later, the obtained ranks are weighted considering the minimum and the maximum values obtained for each dataset, and thus the average weighted rank (AWR) is obtained. The model with highest AWR is assigned as control model for the next step of the procedure. In this stage, the Holm two-step rejection test [22] is used to examine the statistical significance difference between the control model and the rest of the models in a pairwise manner [16].

The result of the Holm procedure is the adjusted p-value (AVP) for each of the methods (Friedman and Quade). The adjusted p-value is obtained from the Holm two-step rejection procedure because, when p-value is considered in a multiple comparison, it reflects the probability of certain comparison, but it doesn't consider the remaining comparisons [16]. As a result, the obtained adjusted p-value will helps us to determinate if the null hypothesis is rejected or accepted, if AVP-values are equal or lower than 0.05. Separate experimental results are obtained for QuickRules (Table 2) and VQRules algorithm (Table 3).

B. Experimental Results

The experimental results are presented in this section as well as the discussion regarding the obtained performance results for each algorithm. We want to note that we evaluate not only the five different FTRMs, but also the values for the parameter k for the T_H.

TABLE I. Characteristics of the Datasets

Dataset Name	No. of Instances	No. of Attributes
Breast-cancer	569	32
Credit-a	690	15
Credit-g	1000	20
Dermatology	366	35
Glass	214	10
Heart-c	303	14
Heart-a	294	14
Heart-Statlog	270	14
Ionosphere	351	35
Iris	150	5
Lung-cancer	32	57
Vehicle	946	19
Vote	435	17
EcoData1	218	117
EcoData2	218	117
EcoData3	218	117
EcoData4	218	117
EcoData5	218	117
Wine	178	14

For that purpose, we index each FTRM in the following manner: ^{1}FTRM$_3$, where the index 1 represent the value of the k parameter for T_H, while 3 represents FTRM$_3$ given with (4). We use the AUC ROC metric for evaluation of both Train (Descriptive) and Test (Predictive) performance, and then using two-stage procedure we estimate the statistical significance of the obtained results.

The results from the experimental evaluation using the procedure to estimate the statistical significance of the results of the QuickRules algorithm over the 19 datasets are presented in Table 2 excluding FTRM$_2$ (since no models where obtained for several datasets), while the obtained results for the VQRules algorithm are presented in Table 3. Table 2 illustrates the results obtained with both Friedman and Quade tests. AWR values show that using k = 1 for T_H in combination with the FTRM$_5$ metric gives best descriptive performance over the 19 datasets. Same results are obtained with the Quade average weighted ranking test. Further analysis of the AWR results indicates that the FTRM$_5$ metric obtain better results than the rest of the metrics for all different values for k, followed by the FTRM$_3$ metric (except for k = 10). The statistical significance analysis showed that the obtained results didn't show any statistically significant difference (for both Friedman and Quade tests). Different discussion can be drawn from the prediction performance analysis. Here, the best AWR was obtained for the FTRM$_1$ metric (k = 2), and this was confirmed by the Friedman and Quade tests. The best AWR was closely followed by the FTRM$_3$ and FTRM$_4$ for several values for k. From Table 2, it is obvious that the lowest rank was obtained for the FTRM$_5$ metric, contrary from the descriptive performance analysis. This indicates that the models obtained with this metric are overfitted.

TABLE II. THE AVERAGE WEIGHTED WANK (AWR) AND AJUSTED *P*-VALUES (AVP) OVER THE 19 DATASETS OBTAINED BY THE FRIDMAN AND QUADE TESTS AND THEIR COORESPONDING P-VALUE OBTAINED BY THE HOLM STEP REJECTION PROCEDURE. THE HIGHEST AWR IS BOLDED, AND THE MODEL WITH HIGHEST AWR IS TAKEN AS CONTROL MODEL. THE AVP VALUES ≤ 0.05 ARE UNDERLINED FOR THE QUICKRULES ALGORTIHM.

Similarity metric	Description Performance				Prediction Performance			
	AWR^F	AWR^Q	AVP^F	AVP^Q	AWR^F	AWR^Q	AVP^F	AVP^Q
$^{1}FTRM_1$	12.55	13.02	0.873	1.000	11.61	11.52	0.350	1.000
$^{1}FTRM_3$	10.24	10.27	1.000	1.000	7.61	7.52	1.000	1.000
$^{1}FTRM_4$	12.55	13.02	0.873	1.000	11.61	11.52	0.350	1.000
$^{1}FTRM_5$	**9.42**	**9.22**	-	-	14.45	16.21	0.005	0.956
$^{2}FTRM_1$	10.39	10.27	1.000	1.000	**7.45**	**6.53**	-	-
$^{2}FTRM_3$	9.63	9.39	1.000	1.000	10.55	10.41	0.738	1.000
$^{2}FTRM_4$	10.39	10.27	1.000	1.000	7.45	6.53	1.000	1.000
$^{2}FTRM_5$	9.42	9.22	1.000	1.000	14.45	16.21	0.005	0.956
$^{3}FTRM_1$	12.08	12.32	0.954	1.000	10.29	9.31	0.797	1.000
$^{3}FTRM_3$	10.29	10.29	1.000	1.000	8.03	8.16	0.999	1.000
$^{3}FTRM_4$	12.08	12.32	0.954	1.000	10.29	9.31	0.797	1.000
$^{3}FTRM_5$	9.42	9.22	1.000	1.000	14.45	16.21	0.005	0.956
$^{5}FTRM_1$	10.82	11.05	1.000	1.000	10.32	8.86	0.797	1.000
$^{5}FTRM_3$	9.92	9.74	1.000	1.000	7.82	7.80	0.999	1.000
$^{5}FTRM_4$	10.82	11.05	1.000	1.000	10.32	8.86	0.797	1.000
$^{5}FTRM_5$	9.42	9.22	1.000	1.000	14.45	16.21	0.005	0.956
$^{10}FTRM_1$	10.03	9.85	1.000	1.000	8.45	7.31	0.998	1.000
$^{10}FTRM_3$	11.08	11.18	0.999	1.000	7.55	8.02	1.000	1.000
$^{10}FTRM_4$	10.03	9.85	1.000	1.000	8.45	7.31	0.998	1.000
$^{10}FTRM_5$	9.42	9.22	1.000	1.000	14.45	16.21	0.005	0.956

TABLE III. THE AVERAGE WEIGHTED WANK (AWR) AND AJUSTED *P*-VALUES (AVP) OVER THE 19 DATASETS OBTAINED BY THE FRIDMAN AND QUADE TESTS AND THEIR COORESPONDING P-VALUE OBTAINED BY THE HOLM STEP REJECTION PROCEDURE. THE HIGHEST AWR IS BOLDED, AND THE MODEL WITH HIGHEST AWR IS TAKEN AS CONTROL MODEL. THE AVP VALUES ≤ 0.05 ARE UNDERLINED FOR THE VQRULES ALGORTIHM.

Similarity metric	Description Performance				Prediction Performance			
	AWR^F	AWR^Q	AVP^F	AVP^Q	AWR^F	AWR^Q	AVP^F	AVP^Q
$^{1}FTRM_1$	18.82	18.81	0.008	1.000	12.89	13.67	0.794	1.000
$^{1}FTRM_2$	12.74	12.49	0.993	1.000	11.50	11.06	0.964	1.000
$^{1}FTRM_3$	11.53	11.25	1.000	1.000	12.79	12.81	0.794	1.000
$^{1}FTRM_4$	18.82	18.81	0.008	1.000	12.89	13.67	0.794	1.000
$^{1}FTRM_5$	11.39	11.28	1.000	1.000	19.87	21.65	7.13E-5	0.969
$^{2}FTRM_1$	14.61	15.42	0.715	1.000	9.71	9.15	0.975	1.000
$^{2}FTRM_2$	10.74	10.61	1.000	1.000	**8.71**	**7.53**	-	-
$^{2}FTRM_3$	11.00	10.67	1.000	1.000	11.97	12.12	0.914	1.000
$^{2}FTRM_4$	14.61	15.42	0.715	1.000	9.71	9.15	0.975	1.000
$^{2}FTRM_5$	11.39	11.28	1.000	1.000	19.87	21.65	7.13E-5	0.969
$^{3}FTRM_1$	15.08	15.75	0.581	1.000	11.32	11.11	0.964	1.000
$^{3}FTRM_2$	11.97	12.05	0.999	1.000	10.92	10.21	0.964	1.000
$^{3}FTRM_3$	12.24	11.99	0.999	1.000	12.76	12.50	0.794	1.000
$^{3}FTRM_4$	15.08	15.75	0.581	1.000	11.32	11.11	0.964	1.000
$^{3}FTRM_5$	11.39	11.28	1.000	1.000	19.87	21.65	7.13E-5	0.969
$^{5}FTRM_1$	15.21	15.65	0.566	1.000	11.47	10.62	0.964	1.000
$^{5}FTRM_2$	10.95	10.70	1.000	1.000	9.89	8.57	0.975	1.000
$^{5}FTRM_3$	11.61	11.29	1.000	1.000	12.84	12.65	0.794	1.000
$^{5}FTRM_4$	15.21	15.65	0.566	1.000	11.47	10.62	0.964	1.000
$^{5}FTRM_5$	11.39	11.28	1.000	1.000	19.87	21.65	7.13E-5	0.969
$^{10}FTRM_1$	13.21	12.94	0.978	1.000	10.47	9.94	0.975	1.000
$^{10}FTRM_2$	**10.24**	**9.71**	-	-	9.97	8.50	0.975	1.000
$^{10}FTRM_3$	11.18	10.70	1.000	1.000	12.55	11.82	0.797	1.000
$^{10}FTRM_4$	13.21	12.94	0.978	1.000	10.47	9.94	0.975	1.000
$^{10}FTRM_5$	11.39	11.28	1.000	1.000	19.87	21.65	7.13E-5	0.969

The statistical significance analysis clearly indicates that the models obtained with the $FTRM_5$ metric have statistically significantly worse results than the control model using the Friedman test. According to the Quade test, the models' result for $FTRM_5$ were not statistically significantly different. And this was obtained for all values of k for the T_H norm.

Next, our analysis continues with results from the VQRules algorithm presented in Table 3, from which we can note that (from the descriptive performance) the model obtained with the $FTRM_2$ metric for $k = 10$ achieved best results and thus obtained the highest AWR from both Friedman and Quade tests. Furthermore, models obtained with $FTRM_3$ and $FTRM_5$ metrics have also better results compared to the $FTRM_1$ and $FTRM_4$ across all studied case studies of k-parameter for T_H. However, the statistical significance procedures for both Friedman and Quade test reveal that the control model don't have statistically significantly different results compared to the rest of the models, except for the $^{1}FTRM_1$ and $^{1}FTRM_4$

models by the Friedman test. Little bit different is the situation when we look at the predictive performance of the obtained models, and we can note that the models with ${}^{2}FTRM_2$ metric have the highest AWR compared to the rest of the models. The results from this model are statistically significant only with the models obtained with the $FTRM_5$ metrics for all values of k for T_H. This was concluded from the results obtained by the Friedman test. However, as in Table 2, the Quade test didn't reveal any statistically significant difference of the control model compared to the rest of models. From both tables, we can note that variability of parameter k for T_H have little effect on the model performance. On contrary, the FTRM metrics have bigger effect on both descriptive and predictive performance of the models.

IV. Conclusion

The presented results in this research paper evaluated the new Hamacher T-norm (T_H) introduced to improve the performance of two fuzzy-rough rule induction algorithms: QuickRules and VQRules. In that direction, we analyzed the influence of the parameter k that defines T_H, as well as five fuzzy-rough tolerance metrics over 19 datasets. From the results of the experiments and the evaluation of the models for statistical significance we concluded that fuzzy tolerance relationship metrics have much greater influence than the parameter k. Furthermore, from the statistical significance analysis, we concluded that the control model in both algorithms have statistically significant difference in the results compared to the models obtained with the $FTRM_5$ metric. The overall analysis points to very interesting results, which evokes for more research to find fuzzy tolerance relationship metrics that could improve the model performance in combination with new T-norms.

Acknowledgment

This work was partially financed by the Faculty of Computer Science and Engineering at the Ss. Cyril and Methodius University in Skopje

References

[1] R. Jensen, C. Cornelis, and Q. Shen, 'Hybrid fuzzy-rough rule induction and feature selection', 2009. doi: 10.1109/FUZZY.2009.5277058.

[2] Z. Pawlak, 'Rough sets: Theoretical aspects of reasoning about data', Control Engineering Practice, vol. 4, no. 5. pp. 741–742, 1996. doi: 10.1016/S0967-0661(96)90021-0.

[3] N. C. Hsieh, 'Rule extraction with rough-fuzzy hybridization method', in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2008, vol. 5012 LNAI. doi: 10.1007/978-3-540-68125-0_89.

[4] Q. Shen and A. Chouchoulas, 'A rough-fuzzy approach for generating classification rules', Pattern Recognit., vol. 35, no. 11, pp. 2425–2438, 2002, doi: 10.1016/S0031-3203(01)00229-1.

[5] J. Stefanowski and A. Tsoukiàs, 'Incomplete information tables and rough classification', Comput. Intell., vol. 17, no. 3, 2001, doi: 10.1111/0824-7935.00162.

[6] M. Kryszkiewicz, 'Rules in incomplete information systems', Inf. Sci. (Ny)., vol. 113, no. 3–4, 1999, doi: 10.1016/S0020-0255(98)10065-8.

[7] S. Greco, M. Inuiguchi, and R. Slowinski, 'Fuzzy rough sets and multiple-premise gradual decision rules', in International Journal of Approximate Reasoning, 2006, vol. 41, no. 2. doi: 10.1016/j.ijar.2005.06.014.

[8] X. Wang, E. C. C. Tsang, S. Zhao, D. Chen, and D. S. Yeung, 'Learning fuzzy rules from fuzzy samples based on rough set technique', Inf. Sci. (Ny)., vol. 177, no. 20, 2007, doi: 10.1016/j.ins.2007.04.010.

[9] J. W. Grzymala-Busse and X. Zou, 'Classification strategies using certain and possible rules', in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 1998, vol. 1424. doi: 10.1007/3-540-69115-4_6.

[10] H. Prade, G. Richard, and M. Serrurier, 'On the induction of different kinds of first-order fuzzy rules', in Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science), 2003, vol. 2711. doi: 10.1007/978-3-540-45062-7_30.

[11] J. Hühn and E. Hüllermeier, 'FURIA: An algorithm for unordered fuzzy rule induction', Data Min. Knowl. Discov., vol. 19, no. 3, 2009, doi: 10.1007/s10618-009-0131-8.

[12] Z. Qin and J. Lawry, 'LFOIL: Linguistic rule induction in the label semantics framework', Fuzzy Sets Syst., vol. 159, no. 4, 2008, doi: 10.1016/j.fss.2007.10.008.

[13] R. Jensen and C. Cornelis, 'A new approach to fuzzy-rough nearest neighbour classification', in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2008, vol. 5306 LNAI. doi: 10.1007/978-3-540-88425-5_32.

[14] A. Naumoski, G. Mirceva, and K. Mitreski, Influence of Algebraic T-norm on Different Indiscernibility Relationships in Fuzzy-Rough Rule Induction Algorithms, vol. 778. 2017. doi: 10.1007/978-3-319-67597-8_12.

[15] C. L. Blake and C. J. Merz, 'UCI repository of machine learning databases'. niversity of California, Irvine, Dept. of Information and Computer Sciences, 1998.

[16] S. García, A. Fernández, J. Luengo, and F. Herrera, 'Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power', Inf. Sci. (Ny)., vol. 180, no. 10, 2010, doi: 10.1016/j.ins.2009.12.010.

[17] J. Fodor and M. Roubens, Fuzzy Preference Modelling and Multicriteria Decision Support. 1994. doi: 10.1007/978-94-017-1648-2.

[18] C. Cornelis, G. H. Martín, R. Jensen, and D. Ślęzak, 'Feature selection with fuzzy decision reducts', in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2008, vol. 5009 LNAI. doi: 10.1007/978-3-540-79721-0_41.

[19] C. Cornelis, M. De Cock, and A. M. Radzikowska, 'Vaguely quantified rough sets', in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2007, vol. 4482 LNAI. doi: 10.1007/978-3-540-72530-5_10.

[20] D. W. Wayne, 'Kruskal–Wallis one-way analysis of variance by ranks', in Nonparametric Statistics, 1990.

[21] D. Quade, 'Using weighted rankings in the analysis of complete blocks with additive block effects', J. Am. Stat. Assoc., vol. 74, no. 367, 1979, doi: 10.1080/01621459.1979.10481670.

[22] S. Holm, 'A simple sequentially rejective multiple test procedure', Scand. J. Stat., vol. 6, no. 2, 1979.