IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 28, 2021, accepted February 13, 2021, date of publication February 26, 2021, date of current version April 9, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3062763

ReG-Rules: An Explainable Rule-Based Ensemble Learner for Classification

MANAL ALMUTAIRI[1], FREDERIC STAHL[1,2], AND MAX BRAMER[3]

[1]Department of Computer Science, University of Reading, Reading RG6 6AY, U.K.
[2]German Research Center for Artificial Intelligence GmbH (DFKI), Laboratory Niedersachsen, Marine Perception, 26129 Oldenburg, Germany
[3]School of Computing, University of Portsmouth, Portsmouth PO1 3HE, U.K.

Corresponding author: Frederic Stahl (frederic_theodor.stahl@dfki.de)

ABSTRACT The learning of classification models to predict class labels of new and previously unseen data instances is one of the most essential tasks in data mining. A popular approach to classification is ensemble learning, where a combination of several diverse and independent classification models is used to predict class labels. Ensemble models are important as they tend to improve the average classification accuracy over any member of the ensemble. However, classification models are also often required to be explainable to reduce the risk of irreversible wrong classification. Explainability of classification models is needed in many critical applications such as stock market analysis, credit risk evaluation, intrusion detection, etc. Unfortunately, ensemble learning decreases the level of explainability of the classification, as the analyst would have to examine many decision models to gain insights about the causality of the prediction. The aim of the research presented in this paper is to create an ensemble method that is explainable in the sense that it presents the human analyst with a conditioned view of the most relevant model aspects involved in the prediction. To achieve this aim the authors developed a rule-based explainable ensemble classifier termed Ranked ensemble G-Rules (ReG-Rules) which gives the analyst an extract of the most relevant classification rules for each individual prediction. During the evaluation process ReG-Rules was evaluated in terms of its theoretical computational complexity, empirically on benchmark datasets and qualitatively with respect to the complexity and readability of the induced rule sets. The results show that ReG-Rules scales linearly, delivers a high accuracy and at the same time delivers a compact and manageable set of rules describing the predictions made.

INDEX TERMS Data mining, ensemble learning, explainable algorithms, rule-based classification.

I. INTRODUCTION

One of the most important tasks in Data Mining applications is predictive analytics, or, in other words, the classification of previously unseen data instances by learning models from training data with known groundtruth. Various algorithms exist to develop such predictive models, i.e. one popular predictive algorithm is the Top Down Induction of Decision Trees (TDIDT) such as ID3 [1] or C4.5 [2], also known as 'Divide and Conquer'. A more recent approach to predictive model generation is Deep Learning [3]. However, whereas Deep Learning has a reputation for developing highly accurate models in comparison to alternative approaches (such as decision trees), they are black box approaches, meaning they do not explain to the human analyst the causality of individual predictions. Such explainability has also been the motivation of rule-based algorithms for predictive analysis such as Ripper [4], CN2 [5], G-eRules [6], a set of related algorithms collectively termed the Prism family of algorithms with its first algorithm described in [7], etc. Rule-based algorithms also offer greater explainability compared with Decision Trees as tree-based classifiers tend to suffer from various problems, such as the 'replicated subtree problem' [7], [8]. Rule-base models offer a more concise explanation of how they arrive at a particular prediction.

However, standalone classifiers, such as the aforementioned rule based techniques, aim to create a perfect model during training and thus often overfit on the training data and do not perform well on the test data. There is no ideal learning algorithm that can avoid overfitting on all types of data sets [9]. An important technique to reduce overfitting of standalone classifiers is ensemble learning [10], [11].

The associate editor coordinating the review of this manuscript and approving it for publication was Massimo Cafaro.

Ensemble versions of standalone classifiers improve the classifier in terms of its stability and classification accuracy. An unstable classifier shows considerable variations to small changes in the training data.

Recent applications of such ensemble approaches have been for example to forecast demands in the electric energy sector [12], for fault diagnosis in refrigeration systems [13], in education to characterise at-risk students and improve retention [14], in banking systems to determine credit scoring [15], etc.

In ensemble learning various base classifiers are induced on various samples of the training data, typically using the same algorithm. The prediction is usually derived through a voting strategy, i.e. majority or weighted majority voting. Ensemble classification approaches tend to improve the average classification accuracy over any member of the ensemble. A notable representative of tree-based ensemble learning is the Random Forest classifier [16]. Also, rule-base ensemble learners have been developed, such as Random Prism [17]. However, the use of ensemble approaches with explainable base classifiers, such as trees or rule sets, defies the purpose of explainability, as the human analyst is presented with a large range of entire classification models, such as multiple decision trees. Random Prism builds an ensemble of rule sets using PrismTCS [18] as a base classifier to improve PrismTCS's classification accuracy. However, the ensemble votes on every prediction with the entire rule set and does not extract relevant rules for prediction. Hence many rules need to be considered for explaining a prediction which obscures the explainability of the approach.

The terms explainable and expressive are similar, but there is a subtle semantic difference how they are used in this paper. The term explainability refers to classification models that explain the outcome of a predicted label to the analyst. The less information is needed to explain the model the higher the degree of explainability. Similarly, the term expressive is used in this paper in the context of single rules. A rule is more expressive the more compact the information leading to a prediction is encoded in the rule. This paper focuses on the explainability aspect of ensemble classifiers by minimising the amount of rules needed to derive a prediction. However, on a rule level also the most expressive types of rules are utilised.

This paper's authors recent work has extended the aforementioned Prism family of rule-based classifiers by more expressive rule-terms in order to enhance explainability of Prism classifiers further [19]. Their recent development, G-Rules-IQR, has shown in empirical experimentation to outperform other members of the Prism family in terms of accuracy, F1 score, tentative accuracy and produces slimmer and thus easier to interpret rule sets [19]. This paper proposes a new rule-based ensemble learner that is different compared with its predecessors as it aims to maximise overall accuracy as well as maintaining a high level of explainability in terms of rule examinations needed for tracing individual predictions. It is based on the most recent G-Rules-IQR approach due to its more expressive rule term structure and proposes a method to merge local rule sets and thus in turn minimises the human analyst's number of rule examinations to explain a prediction. Furthermore, compared with standalone G-Rules-IQR, it increases accuracy and considerably reduces the abstaining rate. The abstaining rate for rule-based prediction is the percentage of data instances remaining unclassified due to no matching rules being available. This is sometimes seen as a drawback of rule-based classifiers, however, abstaining may be desirable in applications where a false classification is costly, such as in finance, health and safety, etc. E.g. one would want a self-driving car to abstain from decision-making and hand back control to the driver if it cannot classify a situation, rather than making an arbitrary decision. Nevertheless, for most applications a low abstaining rate is desired.

The contributions of this paper are (1) a new ensemble classification algorithm that produces expressive human-readable rules, (2) a local Rule Merging algorithm to reduce the overall number of rules induced by the classifier without loss of rule coverage and (3) a decision committee facility to reduce the overall number rules presented to the human analyst giving insights about the prediction.

Overall, an empirical evaluation presented in this paper shows that the proposed ensemble approach produces a higher classification accuracy than the original G-Rules-IQR classifier, offers a much lower abstaining rate and produces a moderate size prediction set of rules and thus maintains a high level of explainability for the human analyst.

The remainder of the paper is structured as follows: Section II describes related work on rule-based classifiers especially the Prism family of algorithms. Furthermore, this Section also gives a summary of ensemble learning approaches. Section III then examines the authors' previous work on G-Rules-IQR in more detail as this is a building block of the proposed ensemble approach. Then Section IV introduces the proposed explainable ensemble learner and Rule Merging strategy followed by a theoretical an empirical analysis in Section V. Section VI offers a final discussion of the presented ensemble approach and concluding remarks.

II. RELATED WORK

This Section distinguishes between two types of rule-base classification systems, (1) single rule-base systems and (2) ensemble rule-base systems.

A. SINGLE RULE-BASE SYSTEMS

Two common strategies to generate classification rules are the 'divide and conquer' and 'separate and conquer' approaches. *Divide and conquer* induces rules in the intermediate form of a decision tree by converting each branch of the tree into a rule. Despite its simplicity and popularity, the decision tree representation of rules suffers from several problems, most importantly, decision trees suffer from replicated subtrees. Rule learners based on *separate and conquer* approach, also called 'covering algorithms', do not suffer from the replicated

subtree problem [9]. They produce a set of IF…THEN classification rules directly from a training dataset. The general approach is as follows: rules are generated one at a time. Instances covered by that rule will be removed from the training data before the next rule is induced. Furthermore, each rule can be maintained independently of the remainder of the rule set, or even be removed without needing to rebuild the entire classifier [20], [21]. The aforementioned replicated subtree problem has been criticised by Cendrowska in [7] as a main reason for overfitting in decision trees. Although Cendrowska never uses the term replicated subtree problem, her study showed that the smallest tree representation for class x defined as:

$$\begin{aligned} &IF\ \ A_3\ \ AND\ \ B_3\ \ Then\ \ Class = x \\ &IF\ \ C_3\ \ AND\ \ D_3\ \ Then\ \ Class = x \end{aligned} \tag{1}$$

would result in 10 nodes and 21 branches in a decision tree, assuming that attributes (A, B, C, D) can each take one of three possible values and if a classification is not x, then it must be y. The reader is referred to Cendrowska's paper [7] for a detailed example of this problem. The Prism algorithm, which follows separate and conquer strategy, is introduced in the same study aiming to generate rules with many fewer redundant rules terms compared with those extracted from a tree-based classifier.

Apart from Prism algorithms, there are further rule-based separate and conquer algorithms such as AQ family, CN2 and RIPPER. AQ [22]–[24] uses a top-down beam search for discovering the best rule. CN2 algorithm [5] integrates ideas from AQ and ID3 algorithms. ID3 induces tree-based classification rules. CN2 produces a rule set based on AQ technique with ID3 capability of handling noisy data. RIPPER algorithm [4] considers the quality and length of generated rules by utilising an overall optimisation step.

As previously mentioned, the main purpose of Prism algorithm is to prevent the generated classification rule set from being redundant and unnecessarily complex. Redundant rule terms and complexity is a necessity in decision trees, but is also considered an unfavourable outcome of use of tree representations [25]. The original Prism pseudo code is described in Algorithm 1. The approach generates modular classification rules directly from training data by inducing one rule at a time. Each rule is specialised term-by-term by selecting the attribute-value pair that maximises the conditional probability of the rule's selected target class. The training stops once the rule only covers instances belonging to that pre-assigned target class. Instances covered by the induced rule will be removed from the training data before the induction of the next rule commences. The process is repeated until there are no instances left in the training data that match the target class. Then the same procedure is carried out for each of the remaining possible classification values.

However, the original Prism is unable to deal directly with continuous attributes. Also, it does not take clashes into account which may occur in the training phase when two or more instances are identical but belong to different classes. A rule encountering a clash during training is not able to specialise further and remains incomplete. Tie-breaking is another problem that can arise during the Prism rule induction process when there are rule-terms with equal highest conditional probability.

Consequently, several studies have been introduced to improve the performance of original Prism. Bramer's Inducer Software [18] implements an extended version of Prism that can handle continuous attributes using binary splitting or cut-point calculations as a local discretisation method. Also, the Inducer software deals with the clashes in training data by determining the majority class of the subset that caused the clash and if it matches the target class the rule is included in the rule set as it is. If the rule's target class is different than the majority class in the clash set, then the rule is discarded. In both cases instances that match the target class are removed. This strategy is illustrated further in [18]. However, this way of dealing with clashes could prompt underfitting if the discarded rule is covered by a large number of instances. In this case it would be likely during testing, that the rule set is not covering numerous data instances and thus abstains from classification. Regarding tie-breaking issue, the inducer implementation selects rule-terms with highest value of frequency [26].

PrismTCS [25] is another member in the Prism algorithm family that uses the minority classes in the training data first as target class. This may result in a lower number of unclassified examples. Compared with the original Prism, this algorithm is faster as it does not require to reset the training data back to it's original state before switching the induction process to a different target class [17]. However, it constructs a classifier with a similar accuracy level as original Prism.

B. ENSEMBLE RULE-BASE SYSTEMS

Generally speaking, ensemble methodology simulates our nature to look for several opinions / views before making any critical decision [27]. We mentally assess the individual views and combine them to attain our ultimate choice. Figure 1 shows the general concept of ensemble learning. It consists of a collection of n classifiers $(C1, C2, \ldots Cn)$, each trained on a different training subset $(S1, S2, \ldots Sn)$ using sampling with or without replacement and produces a single prediction (vote). Combining these individual votes (decisions) using a some kind of voting approach is likely to create an ensemble with a higher level of overall predictive accuracy than its base learners. Therefore, the ensemble methodology is considered to be one of the most effective strategies to improve prediction performance in data mining [28]. Such an ensemble classification system can be referred to as a system of systems. Generating an ensemble model can be done sequentially or in parallel.

The sequential paradigm uses the concept of dependence between the individual classifiers where the base learners are generated sequentially or hierarchically. *Boosting* is one of the well-known forms of this paradigm, *AdaBoost algorithms*

Algorithm 1: Pseudocode for Cendrowska's Original Prism Algorithm

```
foreach class C do
    Reset input Dataset D to its initial state ;
    while
     D does not contain only instances of class C do
        Create a rule R with an empty left hand side
         (LHS) that predicts class C ;
        repeat
            foreach attribute α not mentioned in R,
             and each value x do
                Consider adding the condition α = x
                 to the LHS of R ;
                Select α and x to maximise the
                 accuracy formula ;
            end
            Add α = x to R
        until R is perfect or there are no more
         attributes to use;
        Remove the instances covered by R form D
    end
end
```

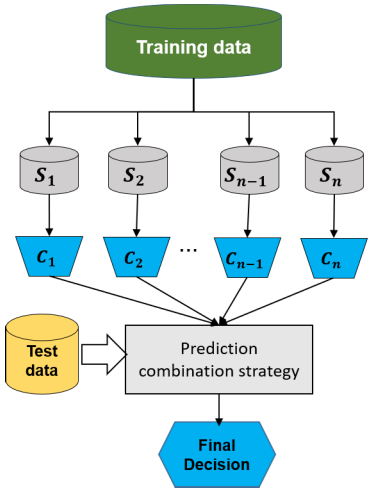

FIGURE 1. General ensemble classification.

in particular. Also, several sequential ensemble approaches have been recently proposed in the literature such as *Vote-boosting* algorithm [29], *SENF* approach [30] and *SEL* framework [31].

On the other hand, the parallel ensemble paradigm, which is more popular and easier to implement, draws on the independence and diversity between the base learners since combining their independent decisions can reduce the classification error effectively [32]. This study uses the parallel ensemble paradigm because of the beneficial usage of its independence advantage in parallel computing which can make the ensemble rule-based model more powerful in practical applications. Therefore, the following paragraphs briefly describe a number of parallel ensemble learning algorithms.

A widely used parallel method is *Bagging* which stands for **B**ootstrap **aggr**ega**ting**. The method introduced by Breiman in [33] aims to improve the stability and predictive performance of a composite classifier [28]. It involves sampling of data with replacement. Each sample is selected randomly with a size equal to that of the original data. This indicates that some of the training instances may appear more than once in the same sample set and some may not be included at all. Each classifier trains on a sample of instances which, statistically, is expected to contain 63.2% of the training data and provides one vote to its selected class. The final classification is typically decided by some form of voting, such as majority or weighted majority voting. The main advantage of Bagging is the ability to reduce bias and variance in the data [16], [33], [34].

Random Forest is also a popular independent ensemble method [16] based on decision trees. It can be considered as an extended version of Bagging and is inspired by the Random Decision Forest algorithm introduced by Ho in [35]. Random Forest essentially incorporates the basic Random Decision Forest approach with Breiman's Bagging method [17], [32]. The Random Decision Forest algorithm builds multiple decision trees. Each tree is constructed using the whole training dataset in sub-spaces selected randomly from the feature space. Ho argues that in high dimensional feature spaces, a considerable number of random subsets of that feature space can introduce differences in classifiers. Therefore, each individual tree generalises its classification. On the other hand, Random Forest evaluates the possible splits at each node before randomly selecting sub-space features. This increases (compared with Random Decision Forest) randomisation in the base classifier construction step and produces an ensemble classifier whose variance is lower than one produced by the individual learners [28].

Random Prism [17], is an ensemble learner not based on decision trees but on rule sets produced by PrismTCS algorithm [25]. It follows the parallel ensemble learning approach and takes a bootstrap sample by randomly selecting n instances with replacement from the training dataset. On average, each base classifier constructed in Random Prism will be trained on 63.2% of the total number of training instances. Thus, the remaining (about 36.8%) will be applied to Random Prism as a test dataset. It has been shown in [17], [36] that Random Prism outperforms its stand-alone base classifier with regard to accuracy and tolerance to noise.

There are also a number of new parallel ensemble algorithms. For example, a parallel deep rule-based ensemble classifier, called DRB [37], and a parallel fusing fuzzy rule-based decision tree via Map-Reduce called MR-FRBDT algorithm [38]. A further example for parallel ensemble classifiers is IP-kNN which integrates several parallel k-NN classifiers [39].

C. OBSERVATIONS ABOUT RELATED WORK

As previously described in Section II-A, practically, the original Prism algorithm can be adapted to work with continuous

attributes using binary splitting which is a local discretisation approach. However, this way of handling numeric values requires frequent cut-point calculations to calculate the conditional probabilities for each value in order to produce rule-terms in the form of $(x < \alpha)$ or $(\alpha \geq y)$ where α is the attribute's name and x and y are two current values of that attribute. Computationally, this is very inefficient as it is extremely costly in time and space complexity, especially for a large dataset. An alternative is to use a global discretisation approach, i.e. ChiMerge [40] in which the data is only discretised once prior to learning the rule set. That seems to be a computational advantage over cut-point calculations. However, ChiMerge suffers from a fundamental weakness as the method converts each attribute independently of the others, not considering that classifications are not determined by just the values of a single attribute. Nevertheless, both, local and global discretisation require sorting of the values of each attribute prior to the discretisation process, and the discretisation process itself can be a significant computational overhead. The interested reader is referred to [26] which gives further details supported by examples about both types of discretisation.

A new heuristic approach based on Gaussian Probability Density Distribution (GPDD) was proposed in [41] to develop an efficient way of handling continuous attributes in the Prism family of algorithms. The approach introduces a new rule-term structure in the form of $(x < \alpha < y)$ instead of two separate rule-terms combinations which greatly enhances readability of the individual rules. Also, the range of values between x and y are representing the most common values of α for a given target class. This would potentially reduce overfitting, a problem that most rule-based classification approaches suffer from. Three Prism based classifiers are integrating this approach in their numerical rule-term construction; *G-Prism-FB [41], G-Prism-DB [42] and G-Rules-IQR [19]*. Further explanations about making use of GPDD function in Prism family of algorithms are provided in Section III, as this method is used in the base learners of the ensemble learner introduced in this paper.

Concerning, **Ensemble rule-base System**, an extensive evaluation study conducted in [43] shows that Random Forest (RF) algorithm suffers from some weaknesses. Firstly, RF requires to construct a number of base learners (trees) in the range of 100 to 500 in order to significantly improve the predictive accuracy of the classification output. This might not be a practical solution in the real life applications where retrieving a fast classification decision is critical. Secondly, RF algorithms are likely to build highly-correlated complex trees from a high-dimensional datasets, which could considerably increase the complexity and the forests error rate. Thirdly, RF does not consider feature interaction (relationships) that might occur in the feature space. On the other hand, the Random Prism (RP) ensemble learner suffers from two essential drawbacks. The first weakness is highlighted in [17], [36] which is the high computational demand as RP makes use of all its base classifiers' votes to produce the final classification for every instance in the testing stage. Also, RP is an accuracy-oriented ensemble classifier because of its weighted majority voting system that uses each individual base classifier accuracy. However, several studies, such as [44], have found that the accuracy is unreliable as a measure for the quality of a classifier especially for unbalanced datasets.

III. PREVIOUS WORK

This section summarises some of the authors' preceding work on enriching the Prism family of algorithms with more expressive rule-based classifiers. One of the developed algorithms is modified as base learner for the presented ensemble classifier in Section IV. Section III-B gives a brief summary of the early versions of expressive rule-base classifiers: G-Prism-FB and G-Prism-DB, while Section III-C details the most recent G-Rule-IQR algorithm which is a cornerstone of the in this paper proposed ensemble approach. Next Section III-A describes the new numeric rule term structure used in previous and current work.

A. INDUCING RULE-TERMS DIRECTLY FROM NUMERICAL ATTRIBUTES

The idea of utilising GPDD function in the learning process is driven by the fact that Gauss or normal distribution is common in statistics in many natural phenomena [45]. As discussed in Section II-C, the GPDD based method can produce more expressive and computationally efficient numeric rule-terms compared with converting continuous attributes into categorical ones in the form of frequent discrete intervals [41]. The Gaussian distribution is calculated for each continuous attribute α_j with mean μ and variance σ^2 from all the values associated with classification, ω_i. The conditional probability for class ω_i is calculated using Equation 2.

$$\mathbb{P}(\alpha_j|\omega_i) = \mathbb{P}(\alpha_j|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} exp(-\frac{(\alpha_j - \mu)^2}{2\sigma^2}) \quad (2)$$

The value for $\mathbb{P}(\omega_i|\alpha_j)$ (or equivalently $log(\mathbb{P}(\omega_i|\alpha_j))$) is be calculated using Equation 3, and this value is then used to determine the probability of a given class label ω_i for a valid attribute value α_j.

$$\log(\mathbb{P}(\omega_i|\alpha_j)) = log(\mathbb{P}(\alpha_j|\omega_i)) + log(\mathbb{P}(\omega_i)) - log(\mathbb{P}(\alpha_j)) \quad (3)$$

The Gaussian distribution for each class label in the training data is then used to calculate the probability of α_j belonging to class label ω_i. This assumes that α_j lies between an upper and lower bound Ω_i. The assumption here is that values close to μ represent the most common values of numerical attribute α_j for ω_i [19], [41], [42].

B. G-PRISM ALGORITHMS

G-Prism-FB algorithm [41] and G-Prism-DB algorithm [42] are two recent Prism family members based on the new numeric rule-term structure where G stands for GPDD, FB and DB refer to the type of upper and lower bounds of the rule-terms, either fixed (FB) or dynamic (DB). The main

difference between these two algorithms is illustrated in Figure 2 as follows: **(a)** G-Prism-FB produces a rule-term in the form of $(x < \alpha \leq y)$ where x and y refer to the next adjacent attribute values left and right of the of the mean μ of attribute α; **(b)** G-Prism-DB has expanded the coverage of it's predecessor to include a user defined maximum number of values k left and right of μ. The algorithm then generates all possible candidate rule-terms within these maximum bounds and selects the one that maximises the conditional probability with which the rule-term covers the target class. The reader is referred to [19] for details about the advantages and disadvantages of these two algorithms. Loosely speaking, the advantages are an improved expressiveness of the rules induced, whereas the disadvantages are with G-Prism-FB that rule-term boundaries achieve low coverage of the target class and thus more rule-terms are induced compared with G-Prism-DB; and the disadvantage of G-Prism-DB is that the optimal rule-term boundaries may lie beyond the user defined range of boundaries.

C. G-RULES-IQR ALGORITHM

A recent study introduced G-Rules-IQR as a new algorithm of the Prism family with the aim of overcoming or mitigating some of the aforementioned limitations and drawbacks of both versions of G-Prism algorithm [19]. The approach is centered around two aspects: **(1)** a new rule-term induction method which is based on a combination of GPDD and Interquartile Range (IQR) to set boundaries; and **(2)** enabling/facilitating this rule-term induction on attributes that are not normally distributed. G-Rules-IQR is outlined in Algorithm 2. With respect to **(1)**, as highlighted in Algorithm 2, G-Rules-IQR algorithm utilises the quartiles that partition the probability density function into four quarters (each containing 25% of data points). Then the algorithm makes use of Gauss distribution on Z-Score scale to determine the third and the first quartiles as in Equation 4 in order to find the upper rule-term and the lower rule-term boundaries. σ is the standard deviation from the mean, z_1 is the standard score of the first quartile and is ≈ -0.67 while z_3 is the standard score of the third quartile and is ≈ 0.67. x usually represents the mean μ but in case of data that is normally distributed it represents the highest probability density of value of $\mathbb{P}(\alpha_j|\omega_i)$ as in lines 15 and 16 of Algorithm 2, where ω_i is the current target class.

$$\begin{aligned} Q_1 &= x = (\sigma * z_1) + \alpha_j \\ Q_3 &= y = (\sigma * z_3) + \alpha_j \\ IQR &= Q_3 - Q_1 \end{aligned} \tag{4}$$

Regarding **(2)**, G-Rules-IQR performs a test for normality for each attribute. If the values for an attribute are not normally distributed for a particular target class, then G-Rules-IQR transforms the attribute values with respect to that target class to approximate a normal distribution. Loosely speaking G-Rules-IQR reduces the skewness rate of attribute values from the normal distribution. A simple and common transformation for attribute values is to take the logarithm of the values [46]. This method to approximate normal distribution is used in this paper due to its simplicity. The normality of each attribute is individually tested against all possible classes in the dataset using Jarque-Bera test [47]. This is done before G-Rules-IQR is applied. If the values of an attribute are not normally distributed in regard to a target class, then the logarithmic approximation to normal distribution is applied.

D. EVALUATION SUMMARY OF G-PRISM AND G-RULES-IQR

G-Rules-IQR algorithm has been empirically evaluated in [19], comparing its performance with two different groups of Prism based approaches. The first group includes the original Prism with three different discretisation methods: cut-point calculations *(local discretisation)*, ChiMerge *(bottom-up global discretisation)*, and Caim *(top-down global discretisation)* [19]. The second group includes the two versions of G-Prism algorithms that were briefly described in Section III-B. The transformation to approximate normal distribution was implemented in both G-Prism versions and G-Rules-IQR and could be switched off. The study [19] concluded that G-Rules-IQR with transformation outperformed its competitors with respect to F1 score, accuracy, tentative accuracy and execution time.

IV. THE ReG-RULES ENSEMBLE LEARNER

The improved version of the G-Rules-IQR algorithm with approximation to normality component is the base inducer of the in this paper proposed ensemble classifier; therefore, it will be illustrated in detail in the current section. The reason for choosing this algorithm is because the stand-alone model of G-Rules-IQR approach shows a high performance in most cases comparing with several other Prism-based classifiers, while producing more expressive rules [19]. However, in general, single rule-base classifiers are not stable especially when they are applied on data containing noise and are also sensitive to the sampling techniques and consequently the level of predictive accuracy varies between different samples [48]. Ensemble learning is an effective approach that can address several single classifier limitations [48] and will be explained in Section IV-A.

A. STAND-ALONE CLASSIFICATION SYSTEM LIMITATIONS

According to [48], learning algorithms that produce only a single classification model suffer from three essential drawbacks that can be addressed by ensemble classification models: (i) the statistical problem, (ii) the computational problem, (iii) and the representation problem.

The statistical issue occurs when the learning algorithm is searching a large feature space for the amount of available training instances. In such cases, different classification models with similar predictive accuracy rates might be generated and hence selecting one of them is a difficult task. The risk of choosing an over-fitted model is rather high [21]. Therefore,

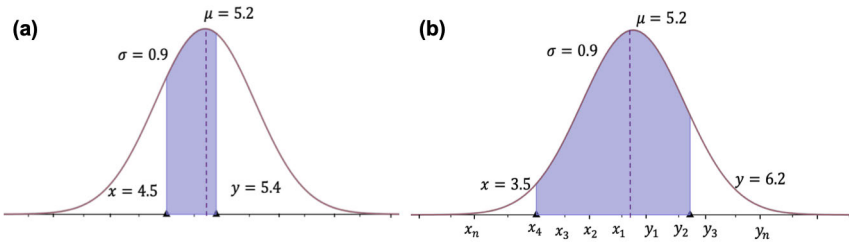

FIGURE 2. Example of finding rule-terms with G-Prism. The shaded area represents values of attributes α_j for class ω_i. Part (a) of the figure depicts finding a rule-term using G-Prism-FB and part (b) of the figure depicts finding a rule-term using G-Prism-DB.

combining the decisions (votes) of these models can lower this risk [48].

The *computational obstacle* relates to the size of the dataset. In real life datasets, considerable dependencies between different features are likely to exist especially among large datasets with high dimensionality in the feature space [49]. This makes the task of finding the best classification model in a computationally feasible time more challenging. Consequently, classification algorithms must utilise heuristic methods to deal with this problem. These heuristics might get trapped in 'local minima' and hence cannot guarantee identifying the best model. Therefore, like with the statistical issue, selecting several different classifiers rather than a single one reduces the risk of selecting a bad model, which might suffer from such a computational problem [48].

Lastly, the *representational problem* appears when there is no optimal classifier to be selected within the classification models spaces. In this case, constructing several weak classifiers might ensure better classification results than trying to choose the best representative.

In general, a learning model that suffers from statistical or computational problems is described as model with high '*variance*' while the one that experiences representational problems is said to have high '*bias*' [48]. Constructing an ensemble classification model by combining the predictions from several base classifiers can be an effective method to overcome these two problems as the main strength of ensemble learning is the ability to handle bias and variance in the data effectively.

B. FRAMEWORK FOR THE ENSEMBLE CLASSIFIER: ReG-RULES

This section proposes a new rule-based ensemble classification system named: **R**anked **e**nsemble **G-Rules**-IQR (ReG-Rules). Algorithm 3 details the pseudocode for this classifier. Figure 3 describes the general framework of this system which consists of 5 stages with several operations: (1) Diversity Generation, (2) Base Classifiers Inductions, (3) Models Selection, (4) Rule Merging, (5) Combination and Prediction. These stages will be illustrated in the following sections referring to lines of code in Algorithm 3.

C. ENSEMBLE DIVERSITY GENERATION

The performance of an ensemble classification model is highly dependent on the level of diversity among the group of classifiers that constitute the ensemble [27], [28], [32]. Clearly, combining individual classifiers with identical or even similar outputs leads to a do-nothing ensemble model. Therefore, if sufficient diversity is obtained, each classifier commits different errors at different times. Thus an appropriate combination strategy can result in reducing the total number of errors in the overall ensemble system.

Nevertheless, unlike regression, in classification context there is no explanatory theory that defines why and how diversity among individual classifiers contributes to overall ensemble accuracy [50], [51]. However, a widely used method to obtain classifiers diversity is: '*using different training datasets to train individual classifiers*' [32]. This method is also used in the ensemble classifier presented in this paper. In this approach all the subsets are drawn from a single data source, but they can just as well be entirely different datasets gathered from different data sources, capturing different aspects of data features if an appropriate randomness is introduced into their sampling technique.

Accordingly, as it can be seen in Figure 3, *diversity generation* part in particular, ReG-Rules utilises two types of sampling in order to maximise the level of base classifiers diversity: (1) sample a dataset randomly *without replacement* into train and test datasets. Please note that the test data is used only once as unseen data to assess the general performance of the ensemble classification model, not the individual base classifiers. (2) Bagging, which is a well-known sampling *with replacement* method [33] used to create multiple data samples. Each sample size is equal to the size of the trained dataset; hence, some instances may appear more than once in a sample set while some may not appear. Statistically, the bagging method produces a sample that is likely to contain 63.2% of the original training dataset. As a result, there are approximately 36.8% of the original training instances that are not used to train the model, these instances are called out-of-bag instances. This portion of the available instances is used as a validation dataset to measure the performance of a base classifier.

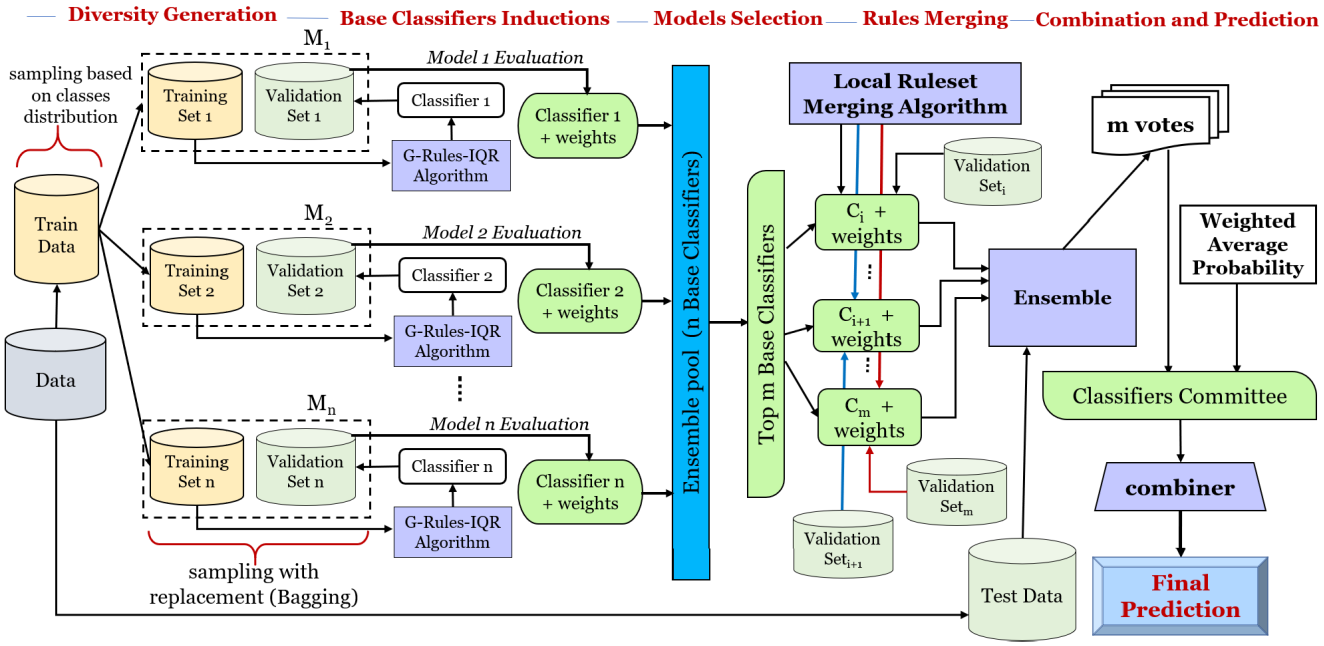

FIGURE 3. The general framework of the ensemble rule-based classifier ReG-Rules.

D. BASE CLASSIFIERS INDUCTIONS

Among the factors controlling the induction of any predictive ensemble model are (1) the total number of base classifiers induced which is represented by *ensemble pool* in Figure 3, and (2) the number of models selected from this pool to participate in the final ensemble decision [27], [28]. While the former is explained in this Section, the latter which is also known as the ensemble size, will be discussed in detail in the next Section (IV-E).

As it can be seen in Algorithm 3 (lines 2 to 5), ReG-Rules system utilises a user-defined parameter to induce M base classifiers from M bagged samples of the training dataset. An important aspect of ensemble learning is to determine how many (M) base learners should be induced. The impact of this on the ensemble efficiency in terms of runtime, memory consumption, diversity, and predictive accuracy make its determination difficult [50]. There is no ideal number of component classifiers within an ensemble. However, a major experimental study conducted in [52] suggested constructing between 64 and 128 base learners to ensure a balance between computational cost and accuracy. The same study has shown that there is no significant performance gain if a larger number of base models is induced. Therefore, a 100 base learners as a default number within this range has been chosen for ReG-Rules. Also, the experiments presented in this paper have been carried out with this default parameter. The induction of these base classifiers is invoked in line 5 of Algorithm 3. As mentioned in Section III-D, selecting this algorithm is based on its performance as a stand-alone model in [19] where it has been empirically evaluated and compared with other members of the Prism family of algorithms in terms of accuracy and expressiveness.

In ReG-Rules multiple models are constructed independently. Nevertheless, it is not possible to measure the quality of these models in order to choose the best learner that can lead to a smaller and more accurate ensemble, until the entire ensemble members contribute to deciding a final classification output. For this reason, as highlighted in Algorithm 3 (lines 6 to 8), a validation data subset is used during induction stage of base classifiers to perform what is called a *classifier performance weighting*.

The basic idea is to associate each individual classifier with a combination of measurements obtained during the validation phase, which assesses the performance of the individual learner. In other words, given M base classifiers are induced in the training phase, their metrics are organised as an M-dimensional vector which consists of: (1) rules set size, (2) average of a rule length, (3) CUR: stands for Correctly Used Rules, which is the number of times a rule was used during the validation phase and predicted the correct class label, (4) abstaining rate, (5) accuracy, and (6) tentative accuracy. Please note, metrics 1-3 are used in Rules Merging strategy, one of the contributions of this paper, which is described in Section IV-F while metrics number 3, 4 and 6 are used in the combination strategy which is described in Section IV-G. Definitions of all these metrics are given in Section V-A. The final step of this stage is represented by the term '*ensemble pool*' in Figure 3. The ensemble pool contains all the base classifiers that are independently evaluated, weighted and prepared for the models selection stage.

Algorithm 2: Learning Classification Rules Using G-Rules-IQR Algorithm

1 **for** $i = 1 \rightarrow C$ **do**
2 $D \leftarrow$ Training Dataset;
3 **while** D does not contain only instances of class ω_i **do**
4 **forall** attributes $\alpha_j \in D$ **do**
5 **if** attribute α_j is categorical **then**
6 Calculate the conditional probability, $\mathbb{P}(\omega_i|\alpha_j)$ for all possible attribute-value $(\alpha_j = x)$ from attribute α;
7 **else if** attribute α_j is continuous **then**
8 Calculate mean μ and variance σ^2 of continuous attribute α for class ω_i ;
9 **foreach** value α_j of attribute α **do**
10 Calculate the conditional probability $\mathbb{P}(\alpha_j|\omega_i)$ based on created Gaussian distribution created in line 8 ;
11 **end**
12 Select α_j of attribute α, which has highest value of $\mathbb{P}(\alpha_j|\omega_i)$;
13 Compute 1st and 3rd quartile using zscore values ;
14 $zScore = 0.67$;
15 $x = \sigma * (-zScore) + \alpha_j$;
16 $y = \sigma * (zScore) + \alpha_j$;
17 Create rule-term r_α in form of $(x < \alpha \leq y)$;
18 Calculate $\mathbb{P}(r_\alpha|\omega_i)$
19 **end**
20 **end**
21 Select $(\alpha_j = x)$ or $(x < \alpha_j \leq y)$ with the maximum conditional probability as a rule-term ;
22 Create a subset S from D containing all the instances covered by selected rule-term at line 21 ;
23 $D \leftarrow S$
24 **end**
25 The induced rule R is a conjunction of all selected rule-terms built at line 21 ;
26 Remove all instances covered by rule R from Training Dataset ;
27 **repeat**
28 lines 2 to 26 ;
29 **until** all instances of class ω_i have been removed form the training data;
30 **Reset** Training Data to its initial state ;
31 **end**
32 **return** induced Rules ;

Algorithm 3: Ensemble Rule-Based Classifier: ReG-Rules

1 initialise the ensemble model (ReG-Rules)
2 **for** $i = 1 \rightarrow M$ **do**
3 $s_i \leftarrow$ Random sample with replacement using Bagging method
4 $v_i \leftarrow$ out-of-bag set
5 Generate a base classifier BC_i by applying Algorithm 2 (G-Rules-IQR) on s_i dataset and learn a rules set R_i
6 Evaluate BC_i performance by applying R_i on v_i dataset
7 Calculate a weight for each rule induced in previous line
8 Send BC_i including its rules set weights to the ensemble pool E_{pool}
9 **end**
10 Rank all the base classifiers BC collected in E_{pool} according to the criteria described in Section IV-E
11 Eliminate weak BC by selecting the top models ($topBC$) ranked in the previous step according to the following if statement:
12 **if** ensemble size type = defualt **then**
13 Select top 20% BC models in line 10
14 **else**
15 user decide the ensemble size
16 Assign all the top BC ($topBC$) selected in line 11 to the ensemble model (ReG-Rules)
17 **for** $j = 1 \rightarrow$ topBC **do**
18 $w_1 \leftarrow$ R_j weight computed previously in line 6
19 Apply Algorithm 4 (Rule Merging) on current $topBc_j$ and update its rules set R_j
20 Re evaluate R_j on the same validation dataset used for weighting the rules in line 6
21 $w_2 \leftarrow$ Calculate the merged rules R_j weight returned from the previous line
22 **if** $w_2 > w_1$ **then**
23 replace rules set of the current $topBC_j$ by the new merged rules R_j
24 **end**
25 Sort the rules set R_j according to their correctly used times
26 **end**
27 **return** ReG-Rules Classifier

E. MODELS SELECTION

As stated in the previous section, how many component classifiers should be included in the final ensemble is an influential factor for building an efficient and accurate ensemble [28], [50]. A large ensemble explores different feature subspaces which might increase its general classification accuracy. However, it requires a higher computational overhead

than of a smaller one and decreases the ensemble's explainability. To overcome this trade-off, reducing the ensemble size should be considered but to what extent this reduction can be applied without causing significant accuracy loss to the whole model is difficult to determine. According to an empirical study presented in [53], a compact ensemble can be extracted from a large one without reducing the whole ensemble's predictive performance in terms of diversity and accuracy. Moreover, the theorem of 'many could be better than all' which was presented in [54] inspired researchers to introduce many ensemble selection methods such as *ranking-based* which is a popular approach for selecting the ensemble members. The reader is referred to [55] for additional models selection approaches.

The main concept of ranking-based approach is to separately rank each base classifier 'according to a certain criterion and chose the top ranked classifiers according to a threshold' [28]. The most commonly used criterion is the predictive accuracy which is in ReG-Rules computed for each individual base classifier using the separate validation dataset. However, accuracy alone might be an insufficient metric to evaluate the classifier especially in imbalanced domains [44]. Taking this into consideration, more measurements are considered in this study. Hence, as previously presented in Section IV-D, each individual base classifier induced in the proposed ensemble system (ReG-Rules) is associated with a combination of metrics that are acquired using different validation datasets. Three of these metrics namely (1) tentative accuracy, (2) CUR and (3) abstaining rate, are used as ensemble selection criteria by ranking all the base classifiers accordingly. Then, as highlighted in Algorithm 3 (lines 10 and 11), the weak base classifiers will be eliminated after selecting the top ranked models according to a predefined ensemble size. Please note that the number of base classifiers that are retained from the ensemble is determined using two types of threshold: (1) default or (2) user defined. There is no optimal ensemble size to be determined [50] but in this study, the default threshold is the top 20% of the ranked models and it was set in this way to ensure that only the strong base classifiers are selected. Thus from the 100 base learners induced in the experiments presented in this paper, only the top 20 ranked base classifiers are chosen to design the final ReG-Rules ensemble system and the remaining 80 models are discarded. Despite this big reduction in the ensemble size, the top 20 models were sufficient according to 'many could be better than all' theory [54] and this default threshold worked well in most cases investigated in this paper.

F. INTEGRATED RULE MERGING (RM) TECHNIQUE

Overlapping rules might occur within a rule set of a selected base classifier. Overlapping rules are generally unnecessary, need to be tested at prediction stage, thus incurring unnecessary computational cost of classification. The proposed integrated RM method aims to address locally and independently this problem for each selected base classifier in the ensemble model. The method is described in Algorithm 4 and represents a post-processing of the induced rules. First the rules are filtered according to their target class and attributes contained in their rule-terms. The Rule Merging is applied for the rules of each target class in turn. During this process some of the rules within the same target class will either be discarded or merged with other rules according to their similarities (overlap of features' ranges). This results in more concise and smaller base classifier rule sets, which are thus more easily read and understood by human analysts. The following passages describe RM technique using three exemplary scenarios.

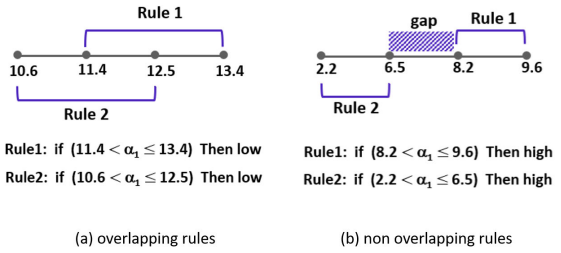

FIGURE 4. Rules sets with single term each rule sharing similar features and classes. In example (a) there is an overlap between rules and in example (b) the rules do not overlap.

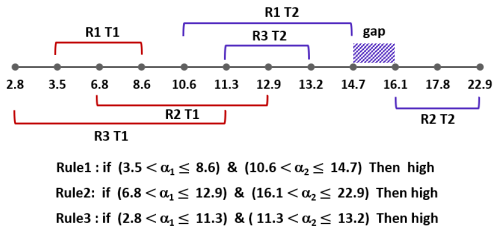

FIGURE 5. Rules sets with two rule-terms sharing similar features and classes.

Figure 4 shows a basic example of the process using two different rules having the same attributes and class where in (a) the two rules are overlapped and hence can be merged to the single rule; ($IF\ 10.6 < \alpha_1 \leq 13.4\ THEN\ low$). In case of (b) the figure shows a gap between the upper bound of the first rule and the lower bound of the second and thus the merging cannot be performed.

Figure 5 shows another example of three rules having the same attributes, α_1, α_2 and referring to the same class label. While the second rule cannot be incorporated in the merging process due to the gap existing between 14.7 and 17.8 in α_2, the first and third rules are overlapped and thus can be combined together to produce a single rule. The output of this approach is the following rule set:

$$\begin{aligned} &IF\ (2.8 < \alpha_1 \leq 11.3)\ and\ (10.6 < \alpha_2 \leq 14.7)\ Then\ high \\ &IF\ (6.8 < \alpha_1 \leq 12.9)\ and\ (16.1 < \alpha_2 \leq 22.9)\ Then\ high \end{aligned} \quad (5)$$

Algorithm 4: Local Rule Merging (RM) Algorithm

```
1  checkedRules → empty
2  for i = 1 → ℝ do
3      checkedRules ← ckeckedRules + R_i ;
4      Other ℝ ← ℝ [−checkedRules] ;
5      j = 1 ;
6      repeat
7          if ( class ω_i of R_i = class ω_j of OtherR_j) and
             ( all attributes α in R_i = all attributes α in
             otherR_j) then
8              OverlapExist ← True ;
9              foreach attribute α_r ∈ α do
10                 switch the type of attribute α_r do
11                     case Continuous do
12                         OverlapExist ← Range_i
                             Overlap Range_j
13                     case Categorical do
14                         OverlapExist ← value of
                             α_r(i) = value of α_r(j)
15                     end
16                     if OverlapExist = False then
17                         Exit for loop in line 9
18                     end
19                 end
20             end
21             if overlapExist then
22                 Compute new upper and lower bounds
                     for each rule-terms r_α ;
23                 Create merged rule in a form of
                     (x < α_r ≤ y) or (α_r = x) ;
24                 Replace R_i in ℝ rules list by the new
                     merged rule created in line 23
25             end
26         end
27         j ← j + 1 ;
28     until No more rules in Other ℝ list;
29 end
30 return new rules list ℝ
```

As previously stated, the main advantage of this approach is reducing the complexity and improving the interpretability of rules that might be generated from large datasets or high dimensional data. As a result, the number of rules for each selected base classifier in the ensemble model would be reduced by removing the overlap that might occur between rules and thus also reduce the computational cost of prediction. Following is another example to show how benefical this Rule Merging can be. Figure 6 includes four rules (Rule 1, Rule 2, Rule 3, Rule 4); each of which have four terms ($\alpha_1, \alpha_2, \alpha_3, \alpha_4$) and refer to the same class label (low). Assume that a given classifier is searching this rule set in the same order to find the first rule that covers an instance with the following attributes values: ($\alpha_1 = 8.1$ $\alpha_2 = 20.2$ $\alpha_3 = 27.5$ $\alpha_4 = 43.4$). In this case, the first rule that fires is the last one (Rule 4). Consequently, the classifier is required to check all 4 rules in order to find a match.

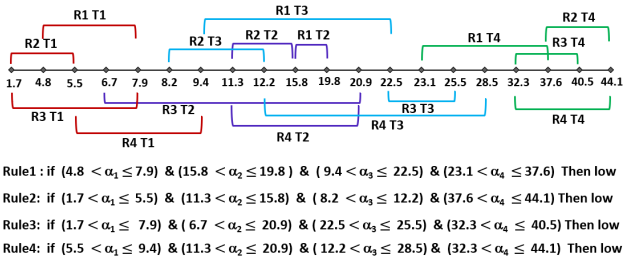

FIGURE 6. Rules set with multiple rule-terms sharing similar features and classes (before merging).

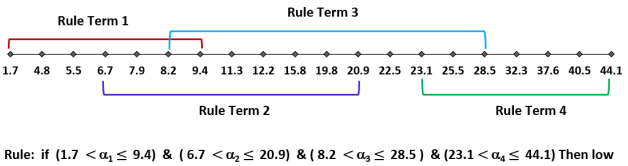

FIGURE 7. A rule with multiple rule-terms sharing similar features and classes (after merging).

As it can be seen from Figure 6, each rule-term in any of the rules in the example is either completely or partially overlapped with at least one rule that includes the same attribute. Applying the new merging method to this rules set, as shown in Figure 7, replaces the four rules with the single merged rule below and hence less effort is required to find a rule that matches the instance:

$$IF \ (1.7 < \alpha_1 \leq 9.4) \ and \ (6.7 < \alpha_2 \leq 20.9) \ and \ (8.2 < \alpha_3 \leq 28.5) \ and \ (23.1 < \alpha_4 \leq 44.1) \ Then \ low \tag{6}$$

G. COMBINATION STRATEGY

Instead of trying to determine the perfect single model, ensemble methods combine a diverse set of models to achieve accurate induction ability. Consequently, it is essential to an ensemble combiner to utilise the appropriate combination strategy in order to produce not only accurate but more robust classification results [28]. ReG-Rules adopts the parallel learning approach, meaning that the induction of each base learner is independent and can be built in parallel to other models without cooperation in the training phase. Instead collaborations between these models are taking place in the testing stage where their independent decisions are passed to a combiner using the combination strategy introduced in this section to generate the final classification decision [32].

A frequently used, simple and thus proven technique is the ***majority voting*** [28], [56]. In this type of voting, all the base models have the same weights [32]. Thus, in the testing stage, the ensemble classifier will assign an unlabelled instance to

TABLE 1. Example of metrics contained in a committee of 20 rules for the classification of one test instance.

Classifier No.	Rule ID	CUR times	Tentative Acc.	Vote	Classification Type
34	8	3	1.0	Class B	Rules
14	8	0	1.0	Class A	Rules
80	3	10	1.0	Class B	Rules
54	4	12	1.0	Class B	Rules
25	12	3	1.0	Class B	Rules
84	-	-	1.0	Class C	Majority class
20	3	12	1.0	Class B	Rules
59	10	0	1.0	Class A	Rules
77	4	7	1.0	Class B	Rules
12	3	12	1.0	Class B	Rules
38	-	-	1.0	Class C	Majority class
7	10	0	1.0	Class A	Rules
53	3	9	1.0	Class B	Rules
71	4	7	1.0	Class B	Rules
81	4	3	1.0	Class B	Rules
60	12	1	0.94	Class B	Rules
50	12	0	0.93	Class B	Rules
90	7	2	0.91	Class B	Rules
73	13	3	0.85	Class A	Rules
46	10	2	0.75	Class C	Rules

the class that has the highest number of votes. Several ensemble classifiers such as Random Forest adopt this equal voting. However, in classification tasks, it is favoured to use ***weighted voting*** instead to avoid a potential problem of reliability when some base classifiers are more reliable than others. Assigning higher weights to the decisions of those qualified models may further improve the overall predictive performance than can be achieved by the equal majority voting [27].

The combination method adopted in this research is based on the latter strategy, but not just on classifier level but also on individual rule level. For this, ReG-Rules builds a committee of rules, termed Classification Committee. The process is described in Algorithm 5. In the algorithm, i refers to the unseen instance, T denotes the test data and *topBC* is the subset of top ranked base classifiers build according to the selection method described in Section IV-E and represented by the *model selection* stage in the general framework of the system (Figure 3). Essentially for each unseen instance, i, the combiner creates a committee of rules, which comprises the first rule that fired from each base classifier contained in *topBC*. As previously explained in Section IV-F, please note that these rules are already improved locally within each base classifier contained in the topBC. The improvement involves applying the Rule Merging techniques to the rules of each target class in turn and then sorting the resulting merged rules according to their performance during validation phase (see lines 20 to 28 in Algorithm 3).

Table 1 shows this committee of rules on an example, how it has been computed by lines 1 to 6 in Algorithm 5. Each prediction received by the committee from the *topBC* is associated with the following components:

1) Tentative accuracy of the base classifier from which the rule comes from. The tentative accuracy is computed only on classification attempts.
2) The number of times a rule was used during the validation phase and predicted the correct class label (CUR).
3) The predicted class label of the rule.
4) The classification type, i.e. did the base classifier use a rule or was it just a majority vote.

Algorithm 5: Combiner: ReG-Rules Committees

```
1  for i = 1 → T do
2      Generate new classifier committee com
3      for n = 1 → topBC do
4          vote_n ← predict class C_i for instance t_i
5          Add vote_n to com_i including the weight of the
             model topBC_n and the weight of its rules set
             R_n that has been used for the prediction
6      end
7      Eliminate the abstaining classifiers whose Rules
         set does not cover the instance t_i
8      Compute the score w_i for each class in com_i
9      return committe decision com_i that has highest
         weighted average probability Evaluate com_i final
         prediction
10 end
```

Next in lines 7 to 10 in Algorithm 5 the votes are combined. First all votes that are based on majority class as classification type are not considered for computing the weight. The reason is because no rule has fired for these base classifiers, thus they have abstained and their votes are considered unreliable. In this example this is concerning classifiers 84 and 38. Next the score for each class label in Table 1 is calculated, in this case there are 3 class labels namely A, B and C.

The computed scores in this example are shown in Table 2. For each class, the score contains the following components: vote frequency, sum tentative accuracy, and total CUR. *Vote*

frequency is simply how often there is a base classifier in the classification committee that voted for a particular class. *Sum tentative accuracy* is simply the sum of tentative accuracies of the rules' base classifiers that have voted for that class. The *total CUR* is the sum of all CUR values of the rules' base classifier that voted for that class. Thus, as it can be seen in Table 2:

$$TotalCURforclassA = 3$$
$$TotalCURforclassB = 81$$
$$TotalCURforclassC = 2$$

Accordingly, CUR value is used to assign a class to the test instance for which the committee of rules was build for. A higher CUR indicates a better class label discrimination and thus is selected as the final prediction of the committee.

If there is a tie break, meaning two or more classes have achieved the same highest CUR, then the highest sum of tentative accuracies per class is used to discriminate further. If tie break issue still exist, then vote frequency per class label will be considered.

V. EVALUATION

This section first introduces the experimental setup in Section V-A and the datasets used in Section V-B. The evaluation comprises four investigations. The first investigation in Section V-C explores ReG-Rules runtime complexity. The second investigation, which is explored in Section V-D, aims to empirically evaluate the overall performance of the new rule-based ensemble learner ReG-Rules compared with the stand-alone G-Rules-IQR. The third investigation explained in Section V-E empirically evaluates the ranking-based [55] approach for selecting an ensemble subset. The approach, which is previously described in Section IV-E, is compared with another method for selecting an ensemble subset without ranking its members. Lastly, Section V-F describes the fourth investigation which qualitatively evaluates the performance of new proposed Rule Merging technique in terms of rules complexity and quantity.

A. EXPERIMENTAL SETUP

All the experiments were performed on a 2.9 GHz Quad-Core Intel Core *i*7 machine with 16*GB* 2133 MHz LPDDR3, running macOS Catalina version 10.15.1. All 24 datasets used in the experiments were picked randomly from the UCI repository [57], the only condition being that they contain continuous attributes and involve classification tasks. All algorithms have been implemented in the statistical programming language R [58] and reuse the same code base differing only in the methodological aspects described in this paper.

The algorithms were evaluated against 5 metrics for classifiers which are described below:

- *Number of Rules:* This is the total number of rules generated for G-Rules-IQR classifier and the average number of rules generated by the ensemble base classifiers.

TABLE 2. Predicted classes' scores.

Predicted Class	Vote Frequency	Total CUR per class	Sum Ten. Acc. per class
Class A	4	3	3.85
Class B	13	**81**	12.78
Class C	1	2	0.75

- *F1 Score:* This is also known as the harmonic mean of precision and recall. A high F1 score is desired. This is a number between 0 and 1.
- *Accuracy:* This is the ratio of data instances that have been correctly classified. Unclassified instances are classified using the majority class strategy. A high classification accuracy is desired. This is a number between 0 and 1.
- *Tentative Accuracy:* This is the ratio of correctly classified instances based only on the number of instances that have been assigned a classification. A high tentative accuracy is desired. This is a number between 0 and 1.
- *Abstaining Rate:* The proportion of cases a classifier abstains from classification, i.e. the proportion of examples not covered in the rule set. Tentative accuracy is based only on the number of instances that have been classified and does not count the ones the classifier abstained of, while accuracy considers the abstained instances as misclassification. Hence, the higher the abstaining rate, the higher the tentative accuracy and the lower the accuracy. This is a number between 0 and 1.

B. DATASETS

The characteristics of the datasets used in the experiments are highlighted in Table 3 in terms of number of instances, attributes (including type of attributes) and class labels. Datasets 15, 16 and 24 included few missing values. A common strategy to estimate each of the missing values using the values that are occur in the dataset is called: *replace by most frequent / average value* [26]. This approach is adopted in this research by replacing a missing categorical value with the most frequently occurring value and estimating a missing numerical value with the average value for the concerning attribute.

Two evaluation methods have been applied to all experimental datasets in the present paper: (1) train and test method in which each dataset was randomly sampled without replacement into train and test datasets. While the test set consists of 30% the data instances, the remaining 70% were used to build the classifiers. The test data is used only once to assess the general performance of the classification models. (2) five-fold cross validation method, in which each dataset was shuffled and randomly divided into 5 partitions (folds) of equal size. Then for each fold, a learning algorithm was trained on the remaining four folds and then tested on the current fold. The two evaluation methods were used to comparatively evaluate both classification systems; the presented ensem-

TABLE 3. Characteristics of the datasets used in the experiments.

No.	Dataset	No. Attributes	No. Classes	No. Instances
1.	iris	5 (4 cont)	3	150
2.	seeds	8 (7 cont)	3	210
3.	wine	14 (13 cont)	3	178
4.	blood transfusion	6 (5 cont)	2	748
5.	banknote	6 (5 cont)	2	1,372
6.	ecoli	9 (7 cont, 1 name)	8	336
7.	yeast	10 (8 cont, 1 name)	10	1,484
8.	page blocks	11 (10 cont)	5	5,473
9.	user modelling	6 (5 cont)	4	403
10.	breast tissue	11 (10 cont)	6	106
11.	glass	11 (10 cont, 1 id)	7	214
12.	HTRU2	10 (9 cont)	2	17,898
13.	magic gamma	12 (11 cont)	2	19,020
14.	wine quality-white	13 (12 cont)	11	4,898
15.	breast cancer	12 (10 cont, 1 id)	2	699
16.	post operative	10 (1 cont, 9 categ)	3	90
17.	wifi localization	8 (7 cont)	4	2,000
18.	indian liver patient	12 (10 cont, 1 categ)	2	583
19.	sonar	62 (61 cont)	2	208
20.	leaf	17 (15 cont, 1 name)	40	340
21.	internet firewall	12 (cont)	4	65,532
22.	bank marketing	17 (6 cont, 10 categ)	2	45,211
23.	avila	11 (10 cont)	12	20,867
24.	shuttle	10 (9 cont)	7	58,000

ble ReG-Rules versus the stand-alone rule based classifier G-Rules-IQR.

C. RUNTIME COMPLEXITY

The induction process of ReG-Rules involves 4 components that need to be considered for estimating the runtime complexity with respect to the number of instances N and the number of features M. These components are (1) Diversity Generation, (2) Base Classifier Inductions, (3) Models Selection and (4) Rules Merging (see Figure 3). These components are executed sequentially, hence the complexity of ReG-Rules is determined by the component with the highest complexity.

With respect to component (1) Bagging is used. Bagging has a complexity of $O(N)$ since N sample instances are taken. Bagging is not dependent on the number of features M. Hence, the complexity of Diversity Generation can be described with $O(N)$. With respect to component (2), G-Rules-IQR base classifiers have a theoretical worst case complexity of $O(N^2M)$ [19], in which case each data instance would be covered by a single rule. However, according to [36] the complexity of algorithms of the Prism family (to which G-Rules-IQR belongs) is estimated to be linear on average. Thus, the runtime complexity of ReG-Rules' Base Classifier Inductions component can be estimated as $d \cdot O(N^2 \cdot M)$, where d is the number of base classifiers induced. d can be neglected here as it is not dependent on the size of the training data. With respect to component (3) the Models Selection is not dependent on the training data but represents a summand m added to the runtime dependent only on the number of models generated. With respect to component (4) the merging of rules is not dependent on the training data but represents a summand r depending only on the number of rules generated. Thus, the total runtime complexity can be described as $O(N) + O(N \cdot M) + m + r$, which can be simplified to a runtime complexity of $O(NM)$. Thus it can be said that ReG-Rules is likely to scale linearly with respect to the number of data instances and features in the training data.

D. EMPIRICAL EVALUATIONS OF THE ENSEMBLE ReG-RULES CLASSIFIER

The experimental results presented in Tables 4 and 5 were obtained using the train and test evaluation method. Similarly, Tables 6 and 7 present the experimental results acquired using the five-fold cross validation method. Please note that each evaluation method's results will be discussed separately in the following sub-sections. In each table the # symbol refers to the number of the dataset in Table 3. The best result(s) in the tables for each dataset are highlighted in bold letters. The tables show the results with respect to the 5 evaluation metrics previously described in Section V-A.

1) EVALUATION USING SEPARATE TRAINING AND TEST DATASETS STRATEGY

Table 4 compares three types of induced rules sets for each dataset: (1) number of rules generated by G-Rules-IQR classifier, (2) average number of rules induced by ReG-Rules classifier before utilising the local RM algorithm, and (3) average number of rules generated by ReG-Rules after integrating the local RM algorithm in its selected base classifiers' rules sets. As it can be seen in Table 4, on average a ReG-Rules base classifier produces fewer rules than G-Rules-IQR for all the 24 datasets. However, further minimising in the number of induced rules without reducing the performance of the classifier is desired and beneficial to the human analyst. For this reason, ReG-Rules integrates the local RM approach in its construction. As it can be observed from Table 4 and Figure 8, in 18 out of 24 datasets a reduction in the number of rules was achieved after applying the local RM algorithm. In some cases the reduction was more than 45% and only in 6 out of 24 datasets ReG-Rules classifier produced the same number of rules sets before and after utilising the RM method. Due to this significance, the remaining experimental results in this section will consider only the version of ReG-Rules which employs the local RM technique in its construction. Table 4 also shows that the ensemble ReG-Rules is lowering the abstaining rate to zero in 21 out of the 24 datasets. On the remaining 3 cases, ReG-Rules's abstaining rate was very close to zero, while in terms of the single base classifier, G-Rules-IQR, the abstaining rates were higher than 10% on several datasets. In three datasets (9, 10, and 20) G-Rules-IQR's abstaining rate reaches 30%, 19% and 40% respectively.

Table 5 uses the evaluation method of separate Training and Test datasets to compare ReG-Rules and G-Rules-IQR in terms of F1 score, accuracy and tentative accuracy metrics. With respect to F1 score, which is the harmonic mean of precision and recall, the results show that the proposed ReG-Rules outperforms G-Rules-IQR on 12 out of 24 datasets.

TABLE 4. Number of rules and abstaining rates using separate training and testing sets method.

#	Number of Rules			Abstaining Rate	
	G-Rules-IQR	ReG-Rules		G-Rules IQR	ReG-Rules
		before merging	after merging		
1	18	17	**13**	0.07	**0.00**
2	22	19	**15**	0.03	**0.00**
3	13	13	**11**	0.06	**0.00**
4	20	16	**11**	**0.00**	**0.00**
5	89	**82**	**82**	0.02	**0.00**
6	37	**32**	**32**	0.02	**0.00**
7	99	**82**	**82**	0.04	**0.00**
8	131	115	**104**	0.02	**0.00**
9	57	45	**42**	0.30	**0.00**
10	28	24	**23**	0.19	**0.00**
11	30	25	**22**	0.11	**0.02**
12	31	26	**17**	**0.00**	**0.00**
13	79	69	**50**	**0.00**	**0.00**
14	126	115	**62**	0.02	**0.00**
15	11	9	**8**	**0.00**	**0.00**
16	29	**23**	**23**	0.11	**0.00**
17	59	48	**38**	0.01	**0.00**
18	47	**50**	**50**	0.17	**0.00**
19	16	13	**12**	0.13	**0.00**
20	124	101	**98**	0.40	**0.01**
21	21	20	**16**	**0.00**	**0.00**
22	40	**38**	**38**	0.01	**0.00**
23	158	164	**146**	0.06	**0.01**
24	27	21	**19**	**0.00**	**0.00**

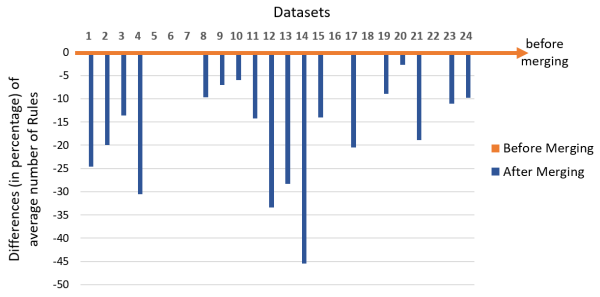

FIGURE 8. Difference (in percentage) of average number of rules of ReG-Rules classifier after integrating RM approach compared with before the merging process.

TABLE 5. F1 score, general accuracy and tentative accuracy using separate training and testing sets method.

#	F1 Score		Accuracy		Tentative Accuracy	
	G-Rules-IQR	ReG-Rules	G-Rules-IQR	ReG-Rules	G-Rules-IQR	ReG-Rules
1	**0.96**	0.93	0.91	**0.93**	**0.95**	0.93
2	**1.00**	**1.00**	0.97	**1.00**	**1.00**	**1.00**
3	0.98	**1.00**	0.94	**1.00**	0.98	**1.00**
4	0.98	**1.00**	0.97	**1.00**	0.97	**1.00**
5	0.98	**0.99**	0.98	**0.99**	**0.99**	**0.99**
6	**0.99**	0.91	**0.96**	0.95	**0.97**	0.95
7	0.87	**0.91**	0.93	**0.97**	0.96	**0.97**
8	**0.93**	0.87	**0.98**	**0.98**	**0.99**	0.98
9	**0.96**	0.95	0.72	**0.94**	**0.95**	0.94
10	0.81	**0.97**	0.66	**0.97**	0.81	**0.97**
11	0.86	**0.93**	0.86	**0.95**	**0.97**	**0.97**
12	0.99	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
13	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
14	**0.96**	0.87	0.97	**1.00**	0.99	**1.00**
15	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
16	0.52	**0.77**	**0.67**	0.63	**0.67**	0.63
17	**1.00**	**1.00**	0.99	**1.00**	**1.00**	**1.00**
18	0.83	**0.84**	0.71	**0.73**	0.71	**0.73**
19	0.95	**0.97**	0.87	**0.97**	0.94	**0.97**
20	**0.75**	0.67	0.39	**0.56**	**0.65**	0.57
21	**0.90**	**0.90**	**1.00**	**1.00**	**1.00**	**1.00**
22	**0.99**	**0.99**	**0.98**	**0.98**	**0.98**	**0.98**
23	0.89	**0.93**	0.85	**0.92**	0.88	**0.92**
24	0.99	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**

Also, in 6 out of the remaining 12 cases where ReG-Rules did not outperform its competitor, it still performs at the same level of score as G-Rules-IQR. On two datasets (1 and 9), ReG-Rules algorithm was not the best method, but was still very close within 3% difference to the best F1 score. With respect to accuracy, in almost all cases ReG-Rules achieved the highest accuracy. In particular, it outperforms the G-Rules-IQR algorithm in 15 out of 24 datasets and performs at the same level as its competitor in 7 out of the 9 remaining cases. On one of the two datasets (16) where ReG-Rules did not outperform G-Rules-IQR, the accuracy of ReG-Rules was lower by only 4% compared with G-Rules-IQR. With respect to tentative accuracy, the ReG-Rules algorithm performs better or equal than G-Rules-IQR in 18 out of 24 datasets. In these 18 cases, the proposed ReG-Rules classifier outperforms G-Rules-IQR in 8 cases. On 5 out of the remaining 6 cases, where ReG-Rules was not the best method, it only underperformed by a maximum difference of 3%.

2) EVALUATION USING CROSS VALIDATION STRATEGY

Table 6 compares the number of rules generated by the G-Rules-IQR classifier with the average number of rules generated by the ReG-Rules classifier. As it can be seen

TABLE 6. Number of rules and abstaining rates using cross validation method.

#	Number of Rules		Abstaining Rate	
	G-Rules-IQR	ReG-Rules	G-Rules IQR	ReG-Rules
1	20	**14**	0.10	**0.00**
2	26	**17**	0.08	**0.00**
3	13	**11**	0.07	**0.00**
4	21	**11**	0.01	**0.00**
5	94	**81**	0.01	**0.00**
6	43	**36**	0.08	**0.00**
7	111	**92**	0.05	**0.00**
8	135	**107**	0.02	**0.00**
9	72	**50**	0.09	**0.00**
10	30	**24**	0.29	**0.01**
11	30	**22**	0.12	**0.00**
12	31	**17**	**0.00**	**0.00**
13	83	**57**	**0.00**	**0.00**
14	132	**66**	0.02	**0.00**
15	10	**8**	0.01	**0.00**
16	34	**25**	0.09	**0.00**
17	59	**39**	0.02	**0.00**
18	51	**51**	0.18	**0.00**
19	17	**13**	0.12	**0.00**
20	138	**107**	0.42	**0.00**
21	23	**19**	0.01	**0.00**
22	41	**39**	0.02	**0.00**
23	160	**143**	0.11	**0.01**
24	26	**20**	**0.00**	**0.00**

TABLE 7. F1 score, general accuracy and tentative accuracy using cross validation method.

#	F1 Score		Accuracy		Tentative Accuracy	
	G-Rules-IQR	ReG-Rules	G-Rules-IQR	ReG-Rules	G-Rules-IQR	ReG-Rules
1	0.95	**0.96**	0.88	**0.97**	0.96	**0.97**
2	**1.00**	**1.00**	0.93	**1.00**	**1.00**	**1.00**
3	0.99	**1.00**	0.92	**1.00**	0.99	**1.00**
4	0.98	**1.00**	0.96	**1.00**	0.97	**1.00**
5	**0.97**	**0.97**	0.96	**0.97**	0.96	**0.97**
6	0.92	**0.93**	0.87	**0.95**	0.93	**0.95**
7	**0.90**	0.89	0.93	**0.97**	**0.98**	0.97
8	**0.90**	0.86	0.98	**0.98**	**0.99**	0.98
9	0.95	**0.96**	0.89	**0.96**	0.95	**0.96**
10	**0.94**	0.90	0.70	**0.88**	**0.93**	0.88
11	**0.93**	0.86	0.86	**0.93**	**0.96**	0.93
12	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
13	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
14	**0.96**	**0.96**	0.98	**1.00**	**1.00**	**1.00**
15	**1.00**	**1.00**	0.99	**1.00**	**1.00**	**1.00**
16	0.69	**0.76**	0.66	**0.69**	0.67	**0.69**
17	0.99	**1.00**	0.98	**1.00**	0.99	**1.00**
18	**0.83**	**0.83**	0.70	**0.71**	**0.72**	0.71
19	0.96	**0.97**	0.89	**0.97**	0.96	**0.97**
20	**0.80**	0.70	0.37	**0.55**	**0.61**	0.55
21	0.84	**0.85**	0.99	**1.00**	**1.00**	**1.00**
22	**0.99**	**0.99**	0.97	**0.98**	**0.98**	**0.98**
23	0.93	**0.94**	0.87	**0.93**	**0.92**	**0.92**
24	**0.99**	0.98	**1.00**	**1.00**	**1.00**	**1.00**

in the table, on average a ReG-Rules base classifier that integrated local RM in its construction produces fewer rules than G-Rules-IQR on all the 24 datasets. The table also shows that compared with its standalone G-Rules-IQR, the problem of abstaining in ReG-Rules was almost non-existent on all the 24 datasets. Only on two datasets (10 and 23) was ReG-Rules's abstaining rate slightly above zero.

Table 7 compares the proposed ensemble ReG-Rules and G-Rules-IQR in terms of F1 score, accuracy and tentative accuracy using the evaluation method of 5-fold cross validation. Regarding F1 score, the results illustrate that ReG-Rules achieves best score on 18 out of 24 datasets. On these 18 datasets, ReG-Rules was the best classifier in 10 cases and performs at the same level of scores as its competitor in the remaining 8 cases. Also, on 4 out of the remaining 6 datasets (7, 8, 10, 24), ReG-Rules classifier only underperformed by a maximum difference of 4%. With respect to accuracy, ReG-Rules classifier outperforms G-Rules-IQR in 21 out of 24 datasets and performs at the same level on the remaining 3 datasets. Regarding tentative accuracy, ReG-Rules performs equal or better than G-Rules-IQR on 18 out of 24 datasets. Among these 18 cases, ReG-Rules outperforms G-Rules-IQR on 9 datasets.

Generally speaking, the results of all the experiments conducted in this research using cross validation are consistent with the previous results obtained from the evaluation strategy of separate training and testing datasets. Both strategies show that ReG-Rules base classifiers are not only producing fewer rules on all the 24 datasets but also almost never abstain from making a classification decision compared with G-Rules-IQR which suffers from high abstaining rate on multiple datasets. In terms of F1 score, accuracy and tentative accuracy, both evaluation approaches demonstrate that ReG-Rules outperforms G-Rules-IQR in most cases.

E. EMPIRICAL EVALUATION OF RANKING CUR APPROACH

As explained in Section IV-E, the idea behind this approach is to rank once the individual ensemble members according to a certain criterion which is based on their rule sets quality and not just the overall accuracy of these base classifiers, and then select the top base classifiers whose rank is above a given threshold (a fixed user-specified amount or percentage of models). A rule's quality is measured using a track record of its performance during the validation stage and this track record is associated with the general performance of the base classifier that has generated this rule. In this part of the experimental study, ranking approach is empirically evaluated in order to show not only its performance but also to what extent this strategy contributes towards the improvement of overall accuracy of the ensemble classification. For evaluation purposes, another version of ensemble ReG-Rules is implemented using the same code base differing in the ensemble selection method. In other words, the second version of ReG-Rules algorithm will not rank the available composite classifiers before selecting a sub-ensemble according to the same user defined ensemble size that has been chosen in the first version. Detailed results of the experiments are depicted in Tables 8 and 9. The best result(s) in these tables for each dataset are highlighted in bold letters.

For simplicity, the bar chart shown in Figure 9 summarises the performance comparisons between the two different implemented versions of ReG-Rules algorithm. ***Version 1:*** ReG-Rules classifier incorporates a prior ranking to its base classifiers according to their tentative accuracies and

TABLE 8. Comparison between two types of ensemble selection models applied to ReG-Rules classifier in terms of number of rules and abstaining rate.

Datasets	Number of Rules		Abstaining Rate	
	No Ranking	Ranking	No Ranking	Ranking
1	**11**	13	**0.00**	**0.00**
2	16	**15**	**0.00**	**0.00**
3	**11**	**11**	**0.00**	**0.00**
4	16	**11**	**0.00**	**0.00**
5	**80**	82	**0.00**	**0.00**
6	**31**	32	**0.00**	**0.00**
7	**82**	**82**	**0.00**	**0.00**
8	**101**	104	**0.00**	**0.00**
9	**42**	**42**	**0.00**	**0.00**
10	**22**	23	**0.00**	**0.00**
11	**21**	22	**0.00**	0.02
12	20	**17**	**0.00**	**0.00**
13	90	**50**	**0.00**	**0.00**
14	64	**62**	**0.00**	**0.00**
15	**8**	**8**	**0.00**	**0.00**
16	**21**	23	**0.00**	**0.00**
17	**36**	38	**0.00**	**0.00**
18	51	**50**	**0.00**	**0.00**
19	**12**	**12**	**0.00**	**0.00**
20	**97**	98	**0.00**	0.01

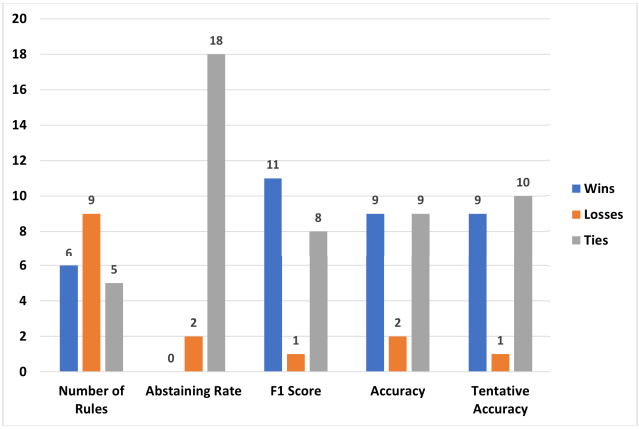

FIGURE 9. Performance of ReG-Rules classifier with ranking-based approach over ReG-Rules classifier without ranking-based approach.

the average of CUR numbers of these models' rules sets before selecting the top ranked members. ***Version 2:*** ReG-Rules classifier that does not involve any ranking process to its composite classifiers before selecting the same subset size of ensemble as for the first version. The figure reports the number of wins, losses and ties. These numbers refer to the number of datasets where Ranked ReG-Rules algorithm (version 1) outperformed, underperformed or performed the same compared with version 2. With regards to number of rules measure, the results demonstrated in Table 8 and in Figure 9 suggest that the ranked version of ReG-Rules achieves the best results in 11 out of 20 datasets with 6 wins, 5 ties. Hence the results indicates that there is no clear winner in terms of number of rules. Concerning the abstaining rates metric, both versions performed almost equally well with 18 ties out of 20 datasets.

However, the results detailed in Table 9 and summarised in Figure 9 show that integrating ranking method into the proposed ensemble algorithm improves the classification performance in most cases in terms of F1 score, accuracy and tentative accuracy. With regards to F1 score, Figure 9 reflects that ReG-Rules (version 1) outperforms (version 2) in 9 out of 20 datasets. Also, among the remaining 11 datasets where it is not surpassing, the ranked version of ReG-Rules algorithm achieves similar scores in 8 datasets compared with the other version. Concerning accuracy, ReG-Rules (version 1) achieves the highest results in 18 out of 20 datasets with 9 wins and 9 ties. Only on two datasets (11 and 20) where the proposed ReG-Rules algorithm was at most 2% lower in accuracy than the results accomplished by ReG-Rules (version 2). In terms of tentative accuracy, ReG-Rules (version 1) performs equal or better than the other version of ReG-Rules on all 20 datasets with 9 wins and 11 ties. It is important to note that the similarity in accuracy and tentative accuracy results highlighted in Table 9 are caused by having abstaining rates of nearly zero as can be seen in Table 8, this is due to the relationships between these metrics which were explained previously in Section V-A.

F. QUALITATIVE EVALUATION OF RULES MERGING (RM) ALGORITHM

The RM method developed in this paper is detailed in Algorithm 4 and aimed to mitigate the complexity of rules set for the individual classifier by reducing the number of rules/terms. The RM approach has been empirically evaluated with respect to the ReG-Rules ensemble in Section V-D. This Section evaluates the RM method qualitatively on two case studies where the rule sets produced by a G-Rules-IQR classifier without RM and one with RM are examined.

The two case studies are the blood transfusion and the wine datasets from the UCI repository [57]. The descriptions of the two datasets can be found in Table 3 in terms of number of instances, attributes (including type of attributes) and classes. Both datasets are used previously among other datasets to evaluate the original G-Rules-IQR algorithm in a published work [19]. Also they are used in the current study to evaluate the ensemble classifier (ReG-Rules). The datasets have been randomly sampled without replacement into train and test datasets; whereas the test sets consist of 30% the data instances and the remaining 70% were used to learn the rule set.

1) CASE STUDY 1: EXPERIMENTS CONDUCTED ON BLOOD TRANSFUSION DATASET

The same 524 training instances were used to learn the classifier and induce the rules sets illustrated below. The 20 original rules were induced by G-Rules-IQR algorithm before applying the merging approach while the 12 merged rules are the ones generated using RM approach. Both, the original and the merged rule sets are validated on the same test data examples which consists of the remaining 224 instances. The results can be seen in Table 10.

Original Rules:

$$R1 : 18.59 \ < Time \ \leq 51.41 \ \rightarrow 0$$
$$R2 : 30.8 \ \ < Time \ \leq 73.2 \ \rightarrow \ 0$$

TABLE 9. Comparison between two types of Ensemble selection models applied to ReG-Rules classifier in terms of F1 score, accuracy and tentative accuracy.

#	F1 Score		Accuracy		Tentative Accuracy	
	No Ranking	Ranking	No Ranking	Ranking	No Ranking	Ranking
1	0.91	**0.93**	0.91	**0.93**	0.91	**0.93**
2	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
3	0.98	**1.00**	0.98	**1.00**	0.98	**1.00**
4	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
5	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**
6	0.91	0.91	0.94	**0.95**	0.94	**0.95**
7	0.89	**0.91**	**0.97**	**0.97**	**0.97**	**0.97**
8	0.85	**0.87**	0.97	**0.98**	0.97	**0.98**
9	0.93	**0.95**	0.93	**0.94**	0.93	**0.94**
10	0.87	**0.97**	0.88	**0.97**	0.88	**0.97**
11	0.90	**0.93**	**0.97**	0.95	**0.97**	**0.97**
12	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
13	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
14	**0.87**	**0.87**	**1.00**	**1.00**	**1.00**	**1.00**
15	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
16	0.74	**0.77**	0.59	**0.63**	0.59	**0.63**
17	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
18	0.83	**0.84**	0.71	**0.73**	0.71	**0.73**
19	0.96	**0.97**	0.95	**0.97**	0.95	**0.97**
20	0.66	**0.67**	**0.57**	0.56	**0.57**	**0.57**

$R3: 2.41 < Monetary \leq 2.98 \;\&$
$-1.06 < Time \leq 33.06 \;\&$
$0.44 < Frequency \leq 0.51 \;\&$
$0.70 < Recency \leq 1.01 \rightarrow 0$

$R4: 2.41 < Monetary \leq 2.98 \;\&$
$-2.44 < Time \leq 34.44 \;\&$
$0.44 < Frequency \leq 0.51 \;\&$
$0.50 < Recency \leq 0.90 \rightarrow 0$

$R5: 2.41 < Monetary \leq 2.98 \;\&$
$0.44 < Frequency \leq 0.51 \;\&$
$1.45 < Time \leq 26.55 \rightarrow 0$

$R6: 0.69 < Recency \leq 1.12 \;\&$
$-3.45 < Time \leq 39.45 \;\&$
$0.17 < Frequency \leq 1.44 \;\&$
$2.39 < Monetary \leq 2.40 \rightarrow 0$

$R7: -6.43 < Time \leq 42.43 \;\&$
$0.17 < Frequency \leq 0.43 \;\&$
$0.93 < Recency \leq 1.42 \;\&$
$2.39 < Monetary \leq 2.40 \rightarrow 0$

$R8: 48.54 < Time \leq 99.46 \rightarrow 0$

$R9: 6.60 < Time \leq 15.41 \rightarrow 0$

$R10: 0.32 < Recency \leq 0.63 \;\&$
$0.21 < Frequency \leq 0.39 \;\&$
$1.99 < Time \leq 2.0 \rightarrow 0$

$R11: 12.43 < Time \leq 19.57 \rightarrow 0$

$R12: 3.99 < Time \leq 4.0 \rightarrow 0$

$R13: 1.12 < Time \leq 1.64 \rightarrow 1$

$R14: 0.75 < Time \leq 1.48 \rightarrow 1$

$R15: 1.25 < Time \leq 2.03 \;\&$
$0.87 < Frequency \leq 1.29 \rightarrow 1$

$R16: 0.29 < Time \leq 1.11 \rightarrow 1$

$R17: 1.76 < Time \leq 1.93 \rightarrow 1$

$R18: 1.61 < Time \leq 1.82 \rightarrow 1$

$R19: 1.60 < Frequency \leq 1.69 \rightarrow 1$

$R20: 1.95 < Time \leq 1.97 \rightarrow 1$

Merged Rules:

$R1: 18.59 < Time \leq 99.46 \rightarrow 0$

$R2: 2.41 < Monetary \leq 2.99 \;\&$
$-2.44 < Time \leq 34.44 \;\&$
$0.44 < Frequency \leq 0.51 \;\&$
$0.50 < Recency \leq 1.10 \rightarrow 0$

$R3: 0.69 < Recency \leq 1.42 \;\&$
$-6.43 < Time \leq 42.43 \;\&$
$0.17 < Frequency \leq 0.44 \;\&$
$2.39 < Monetary \leq 2.40 \rightarrow 0$

$R4: 6.6 < Time \leq 19.57 \rightarrow 0$

$R5: 0.29 < Time \leq 1.64 \rightarrow 1$

$R6: 1.61 < Time \leq 1.93 \rightarrow 1$
$0.21 < Frequency \leq 0.39 \;\&$
$1.99 < Time \leq 2.0 \rightarrow 0$

$R7: 2.41 < Monetary \leq 2.98 \;\&$
$0.44 < Frequency \leq 0.51 \;\&$
$1.45 < Time \leq 26.55 \rightarrow 0$

$R8: 0.69 < Recency \leq 1.12 \;\&$
$-3.45 < Time \leq 39.45 \;\&$
$0.17 < Frequency \leq 1.44 \;\&$
$2.39 < Monetary \leq 2.40 \rightarrow 0$

$R9: -6.43 < Time \leq 42.43 \;\&$

TABLE 10. Experimental results of case study 1.

Metrics	Original Rules set	Merged Rules set
Number of Rules	20	**12**
Abstaining Rate	0	0
Recall	1	1
Precision	0.966	0.971
F1 Score	0.982	0.985
Accuracy	0.973	0.977
Tentative Accuracy	0.973	0.977

$$
\begin{aligned}
0.17 < \ & \mathit{Frequency} \ \leq 0.43 \ \& \\
0.93 \ < \ & \mathit{Recency} \ \leq 1.42 \ \& \\
2.39 < \ & \mathit{Monetary} \leq 2.40 \rightarrow 0 \\
R10 : 48.54 \ < \ & \mathit{Time} \ \leq 99.46 \rightarrow 0 \\
R11 : 6.60 \ < \ & \mathit{Time} \ \leq 15.41 \rightarrow 0 \\
R12 : 0.32 \ < \ & \mathit{Recency} \ \leq 0.63 \ \& \\
0.21 \ < \ & \mathit{Frequency} \ \leq 0.39 \ \& \\
1.99 < \ & \mathit{Time} \leq 2.0 \rightarrow 0
\end{aligned}
$$

It can be seen that the number or rules and rule terms is considerably reduced, making it easier for the analyst to understand the rule model. In this case the number of rules were reduced from 20 to 12. The RM method merges without loss of information, thus instances covered by a rule before merging should still be covered either by the same rule or the resulting merged rule (leading to the same classification) after RM was applied. Nevertheless, what can also be seen in Table 10 is that there are very small variations in precision, F1 score, accuracy and tentative accuracy. A closer examination of the results on the test data revealed that the variation are a result of the order in which the rules are applied. Before merging a data instance may have been covered by two or more rules each leading to a different class label and the first rule applied and matching the data instance would determine the class label. The same effects are still true after the RM, if two rules are merged they are not anymore listed consecutively and the rule order may change slightly.

2) CASE STUDY 2: EXPERIMENTS CONDUCTED ON WINE DATASET

The same 125 training instances were used to learn the classifier and induce the rules sets illustrated below. The 13 original rules were induced by G-Rules-IQR algorithm before applying the merging approach while the 9 merged rules are the ones generated using RM approach. Both, the original and the merged rule sets are validated on the same test data examples which consists of the remaining 53 instances. The results can be seen in Table 11.

Original Rules:

$$
\begin{aligned}
R1 &: 0.09 < \mathit{Noflavan\ phenols} \leq 0.12 \rightarrow 1 \\
R2 &: 0.58 < \mathit{Total\ phenols} \leq 0.62 \rightarrow 1 \\
R3 &: 0.59 < \mathit{Total\ phenols} \leq 0.65 \rightarrow 1 \\
R4 &: 0.05 < \mathit{Noflavan\ phenols} \leq 0.11 \rightarrow 1 \\
R5 &: 13.68 < \mathit{Alcohol} \leq 13.70 \rightarrow 1
\end{aligned}
$$

TABLE 11. Experimental results of case study 2.

Metrics	Original Rules set	Merged Rules set
Number of Rules	13	**9**
Abstaining Rate	0.06	0.06
Recall	0.98	0.98
Precision	0.98	0.98
F1 Score	0.98	0.98
Accuracy	0.94	0.94
Tentative Accuracy	0.98	0.98

$$
\begin{aligned}
R6 &: 1.93 < \mathit{Magnesium} \leq 2.02 \rightarrow 2 \\
R7 &: 1.85 < \mathit{Magnesium} \leq 2.01 \rightarrow 2 \\
R8 &: 2.01 < \mathit{Magnesium} \leq 2.15 \rightarrow 2 \\
R9 &: 2.77 < \mathit{Proline} \leq 2.89 \rightarrow 2 \\
R10 &: 0.39 < \mathit{Total\ phenols} \leq 0.46 \rightarrow 3 \\
R11 &: 509.6 < \mathit{Proline} \leq 670.4 \rightarrow 3 \\
R12 &: 0.34 < \mathit{Total_p henols} \leq 0.43 \rightarrow 3 \\
R13 &: 0.57 < \mathit{Hue} \leq 0.62 \rightarrow 3
\end{aligned}
$$

Merged Rules:

$$
\begin{aligned}
R1 &: 0.05 < \mathit{Noflavan\ phenols} \leq 0.12 \rightarrow 1 \\
R2 &: 0.58 < \mathit{Total\ phenols} \leq 0.65 \rightarrow 1 \\
R3 &: 1.85 < \mathit{Magnesium} \leq 0.65 \rightarrow 1 \\
R4 &: 0.34 < \mathit{Total\ phenols} \leq 0.46 \rightarrow 1 \\
R5 &: 13.68 < \mathit{Alcohol} \leq 2.02 \rightarrow 1 \\
R6 &: 2.01 < \mathit{Magnesium} \leq 2.15 \rightarrow 2 \\
R7 &: 2.77 < \mathit{Proline} \leq 2.89 \rightarrow 2 \\
R8 &: 509.6 < \mathit{Proline} \leq 670.4 \rightarrow 3 \\
R9 &: 0.57 < \mathit{Hue} \leq 0.62 \rightarrow 3
\end{aligned}
$$

Here it can be seen as well that the number or rules and rule terms is considerably reduced, again, making it easier for the analyst to understand the rule model. In this case the number of rules were reduced from 13 to 9. As discussed for Case Study 1, the merging does not cause loss of information, merely the rule order may be influenced. In this case no effects of the rule order can be observed with respect to the performance metrics listed in Table 11.

VI. CONCLUSION

The paper presents the development of a new predictive ensemble learner termed ReG-Rules. ReG-Rules aims to improve the predictive performance of expressive and explainable rule-based predictive learners, while presenting the human analyst with an explainable model for predictions. ReG-Rules merges classification rules from the base classifiers and offers the analyst a human readable and compact rule set for a prediction. ReG-Rules uses a validation set to measure the base classifier performance which is a composite measure composed of various metrics. Out of these best ranked base models a classification committee of rules is being built for each classification attempt.

ReG-Rules was evaluated empirically and qualitatively and compared with the standalone G-Rules-IQR classifier it

aims to improve upon. With respect to *empirical evaluation* train and test as well as cross validation where used on real datasets. It was found that both empirical evaluation approaches achieved similar results for all performance metrics. The performance metrics considered were accuracy, tentative accuracy, F1 score and abstaining rate. It was found that for all empirical evaluation metrics, ReG-Rules outperformed G-Rules-IQR on average and in most cases. Abstaining from classification, a typical problem of rule-based classifiers, was almost non-existent in ReG-Rules. The potential changes of the classification due to the Rule Merging component of ReG-Rules were also examined empirically and it was found that there are many fewer rules in the model per base learner than using original unmerged G-Rules-IQR, while exhibiting only minor differences in the classification performance. The Rule Merging component was also evaluated *qualitatively* and it was found that merged rule sets are more compact and easier to read.

However, ReG-Rules requires a human analyst to examine a small set of rules (classification committee) per classification attempt, although this rule set is much smaller than the entirety of the rules induced by ReG-Rules. Thus ongoing work aims to extend ReG-Rules by a global Rule Merging facility to generate a consolidated rule set from all relevant base learners. This is expected to reduce the number of rules presented to the analyst per classification attempt and thus is expected to enhance ReG-Rules' expressive power further, while maintaining ReG-Rules' predictive power.

Overall, it can be said that rule-based predictive models are among the most expressive classification techniques in data mining. Ensemble learners aim to improve classification performance but generally often at the expense of explainability. ReG-Rules successfully provides an approach to harvest the predictive power of an ensemble learner, while maintaining explainable aspects of rule-based predictive models.

REFERENCES

[1] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, pp. 81–106, Mar. 1986.
[2] J. R. Quinlan, *C4. 5: Programs for Machine Learning*. Amsterdam, The Netherlands: Elsevier, 2014.
[3] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.
[4] W. W. Cohen, "Fast effective rule induction," in *Machine Learning Proceedings 1995*. Amsterdam, The Netherlands: Elsevier, 1995, pp. 115–123.
[5] P. Clark and T. Niblett, "The CN2 induction algorithm," *Mach. Learn.*, vol. 3, no. 4, pp. 261–283, Mar. 1989.
[6] T. Le, F. Stahl, J. B. Gomes, M. M. Gaber, and G. Di Fatta, "Computationally efficient rule-based classification for continuous streaming data," in *Proc. Int. Conf. Innov. Techn. Appl. Artif. Intell.* Cham, Switzerland: Springer, 2014, pp. 21–34.
[7] J. Cendrowska, "PRISM: An algorithm for inducing modular rules," *Int. J. Man-Mach. Stud.*, vol. 27, no. 4, pp. 349–370, Oct. 1987.
[8] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques With Java Implementations Morgan Kaufmann*. San Francisco, CA, USA: Morgan Kaufmann, 1999.
[9] J. Fürnkranz, D. Gamberger, and N. Lavrač, *Foundations of Rule Learning*. Springer, 2012.
[10] G. Dong and J. Bailey, *Contrast Data Mining: Concepts, Algorithms, and Applications* (Chapman & Hall/CRC Data Mining and Knowledge Discovery Series). Boca Raton, FL, USA: CRC Press, 2016.
[11] B. Johnston and I. Mathur, *Applied Supervised Learning With Python: Use Scikit-Learn to Build Predictive Models From Real-World Datasets and Prepare Yourself for the Future of Machine Learning*. Birmingham, U.K.: Packt, 2019.
[12] E. M. de Oliveira and F. L. C. Oliveira, "Forecasting mid-long term electric energy consumption through bagging ARIMA and exponential smoothing methods," *Energy*, vol. 144, pp. 776–788, Feb. 2018.
[13] Z. Zhang, H. Han, X. Cui, and Y. Fan, "Novel application of multi-model ensemble learning for fault diagnosis in refrigeration systems," *Appl. Thermal Eng.*, vol. 164, Jan. 2020, Art. no. 114516.
[14] J. Beemer, K. Spoon, L. He, J. Fan, and R. A. Levine, "Ensemble learning for estimating individualized treatment effects in student success studies," *Int. J. Artif. Intell. Educ.*, vol. 28, no. 3, pp. 315–335, Sep. 2018.
[15] P. Pławiak, M. Abdar, and U. R. Acharya, "Application of new deep genetic cascade ensemble of SVM classifiers to predict the Australian credit scoring," *Appl. Soft Comput.*, vol. 84, Nov. 2019, Art. no. 105740.
[16] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
[17] F. Stahl and M. Bramer, "Random prism: A noise-tolerant alternative to random forests," *Expert Syst.*, vol. 31, no. 5, pp. 411–420, Nov. 2014.
[18] M. Bramer, "Inducer: A public domain workbench for data mining," *Int. J. Syst. Sci.*, vol. 36, no. 14, pp. 909–919, Nov. 2005.
[19] M. Almutairi, F. Stahl, and M. Bramer, "A rule-based classifier with accurate and fast rule term induction for continuous attributes," in *Proc. 17th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2018, pp. 413–420.
[20] J. Han, J. Pei, and M. Kamber, *Data Mining: Concepts and Techniques*. Amsterdam, The Netherlands: Elsevier, 2011.
[21] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical Machine Learning Tools and Techniques*. San Mateo, CA, USA: Morgan Kaufmann, 2016.
[22] R. S. Michalski, "On the quasi-minimal solution of the general covering problem," in *Proc. 5th Int. Symp. Inf. Process. (FCIP)*, Yugoslavia, vol. A3, Oct. 1969, pp. 125–128.
[23] R. S. Michalski, "Learning by being told and learning from examples: An experimental comparison of the two methods of knowledge acquisition in the context of development an expert system for soybean disease diagnosis," *Int. J. Policy Anal. Inf. Syst.*, vol. 4, no. 2, pp. 125–161, 1980.
[24] R. S. Michalski, I. Mozetic, J. Hong, and N. Lavrac, "The multi-purpose incremental learning system AQ15 and its testing application to three medical domains," in *Proc. AAAI*, 1986, pp. 1–41.
[25] M. Bramer, "An information-theoretic approach to the pre-pruning of classification rules," in *Proc. Int. Conf. Intell. Inf. Process.* Boston, MA, USA: Springer, 2002, pp. 201–212.
[26] M. Bramer, *Principles of Data Mining*, vol. 530. London, U.K.: Springer-Verlag, 2016.
[27] L. Rokach, *Pattern Classification Using Ensemble Methods*, vol. 75. Singapore: World Scientific, 2010.
[28] L. Rokach, "Ensemble-based classifiers," *Artif. Intell. Rev.*, vol. 33, nos. 1–2, pp. 1–39, 2010.
[29] M. Sabzevari, G. Martínez-Muñoz, and A. Suárez, "Vote-boosting ensembles," *Pattern Recognit.*, vol. 83, pp. 119–133, Nov. 2018.
[30] K. Chen, D. Guan, W. Yuan, B. Li, A. M. Khattak, and O. Alfandi, "A novel feature selection-based sequential ensemble learning method for class noise detection in high-dimensional data," in *Proc. Int. Conf. Adv. Data Mining Appl.* Cham, Switzerland: Springer, 2018, pp. 55–65.
[31] C.-M. Vong and J. Du, "Accurate and efficient sequential ensemble learning for highly imbalanced multi-class data," *Neural Netw.*, vol. 128, pp. 268–278, Aug. 2020.
[32] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*. London, U.K.: Chapman & Hall, 2012.
[33] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, Aug. 1996.
[34] H. Pham and S. Olafsson, "Bagged ensembles with tunable parameters," *Comput. Intell.*, vol. 35, no. 1, pp. 184–203, Feb. 2019.
[35] T. K. Ho, "Random decision forests," in *Proc. 3rd Int. Conf. Document Anal. Recognit.*, vol. 1, 1995, pp. 278–282.
[36] F. Stahl, D. May, H. Mills, M. Bramer, and M. M. Gaber, "A scalable expressive ensemble learning using random prism: A mapreduce approach," in *Transactions on Large-Scale Data- and Knowledge-Centered Systems XX*. Berlin, Germany: Springer, 2015, pp. 90–107.
[37] X. Gu, P. P. Angelov, C. Zhang, and P. M. Atkinson, "A massively parallel deep rule-based ensemble classifier for remote sensing scenes," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 3, pp. 345–349, Mar. 2018.
[38] Y. Mu, X. Liu, L. Wang, and J. Zhou, "A parallel fuzzy rule-base based decision tree in the framework of map-reduce," *Pattern Recognit.*, vol. 103, Jul. 2020, Art. no. 107326.

[39] R. Agrawal, "Integrated parallel k-nearest neighbor algorithm," in *Smart Intelligent Computing and Applications*. Singapore: Springer, 2019, pp. 479–486.
[40] R. Kerber, "Chimerge: Discretization of numeric attributes," in *Proc. 10th Nat. Conf. Artif. Intell.*, 1992, pp. 123–128.
[41] M. Almutairi, F. Stahl, M. Jennings, T. Le, and M. Bramer, "Towards expressive modular rule induction for numerical attributes," in *Research and Development in Intelligent Systems XXXIII: Incorporating Applications and Innovations in Intelligent Systems XXIV*. Cham, Switzerland: Springer, 2016, pp. 229–235.
[42] M. Almutairi, F. Stahl, and M. Bramer, "Improving modular classification rule induction with g-prism using dynamic rule term boundaries," in *Research and Development in Intelligent Systems XXXIII: Incorporating Applications and Innovations in Intelligent Systems XXIV*. Cham, Switzerland: Springer, 2017.
[43] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we need hundreds of classifiers to solve real world classification problems?" *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3133–3181, 2014.
[44] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: Bagging-, Boosting-, and hybrid-based approaches," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 4, pp. 463–484, Jul. 2012.
[45] C. Walck, "Hand-book on statistical distributions for experimentalists," Univ. Stockholm, Stockholm, Sweden, Tech. Rep., 1996.
[46] H. C. Thode, *Testing for Normality*, vol. 164. Boca Raton, FL, USA: CRC Press, 2002.
[47] C. M. Jarque and A. K. Bera, "Efficient tests for normality, homoscedasticity and serial independence of regression residuals," *Econ. Lett.*, vol. 6, no. 3, pp. 255–259, Jan. 1980.
[48] S. Amari *et al.*, *The Handbook of Brain Theory and Neural Networks*, M. A. Arbib, Ed. Cambridge, MA, USA: MIT Press, 2003.
[49] C. C. Aggarwal, *Data Mining: The Textbook*. Cham, Switzerland: Springer, 2015.
[50] H. Bonab and F. Can, "Less is more: A comprehensive framework for the number of components of ensemble classifiers," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 9, pp. 2735–2745, Sep. 2019.
[51] G. Brown, J. Wyatt, R. Harris, and X. Yao, "Diversity creation methods: A survey and categorisation," *Inf. Fusion*, vol. 6, no. 1, pp. 5–20, Mar. 2005.
[52] T. M. Oshiro, P. S. Perez, and J. A. Baranauskas, "How many trees in a random forest?" in *Proc. Int. Workshop Mach. Learn. Data Mining Pattern Recognit.* Berlin, Germany: Springer, 2012, pp. 154–168.
[53] H. Liu, A. Mandvikar, and J. Mody, "An empirical study of building compact ensembles," in *Proc. Int. Conf. Web-Age Inf. Manage.* Berlin, Germany: Springer, 2004, pp. 622–627.
[54] Z.-H. Zhou, J. Wu, and W. Tang, "Ensembling neural networks: Many could be better than all," *Artif. Intell.*, vol. 137, nos. 1–2, pp. 239–263, May 2002.
[55] G. Tsoumakas, I. Partalas, and I. Vlahavas, "A taxonomy and short review of ensemble selection," in *Proc. Workshop Supervised Unsupervised Ensemble Methods Their Appl.*, 2008, pp. 1–6.
[56] L. Lam and S. Y. Suen, "Application of majority voting to pattern recognition: An analysis of its behavior and performance," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 27, no. 5, pp. 553–568, Sep. 1997.
[57] M. Lichman, "UCI machine learning repository," School Inf. Comput. Sci., Univ. California, Irvine, Irvine, CA, USA, Tech. Rep., 2013.
[58] *R: A Language and Environment for Statistical Computing*, R Develop. Core Team, R Found. Stat. Comput., Vienna, Austria, 2008.

MANAL ALMUTAIRI received the B.Sc. degree in computer and information sciences from King Saud University, Riyadh, Saudi Arabia, and the M.Sc. degree in advanced computer science from the University of Reading, U.K., where she is currently pursuing the Ph.D. degree with the Department of Computer Science.

Since 2006, she has been employed as a Programmer and Software Designer in the Information Centre of Saudi Customs, Ministry of Finance, Riyadh. She is also working on a research project 'Development of an Expressive Rule-Based Ensemble Classifier System.' During her Ph.D. research, she has published three peer-reviewed articles, including articles published through the IEEE.

FREDERIC STAHL received the Dipl.-Ing. (FH) degree in bioinformatics from the University of Applied Science, Weihenstephan, Germany, in 2006, and the Ph.D. degree in computer science from the University of Portsmouth, U.K., in 2010.

From 2010 to 2012, he was a Senior Research Associate with the Department of Computer Science, University of Portsmouth, U.K. In 2012, he worked as a Lecturer with the Department of Design Engineering and Computing, Bournemouth University, U.K. From 2012 to 2019, he was a Lecturer and Associate Professor with the University of Reading, U.K. Since 2019, he has been the Deputy Head, Team Leader, and Senior Researcher for Subject Area Marine Perception at the German Research Centre for Artificial Intelligence (DFKI GmbH). He has published over 60 papers in peer-reviewed conferences, journals, and book chapters. He has been working in the field of data mining for more than ten years focusing on the research domain of big data analytics. His particular research interests include developing scalable algorithms for building adaptive models for real-time streaming data, developing scalable parallel data mining algorithms and workflows, and applications in big data analytics.

Dr. Stahl is a member of the British Computer Society (BCS) and has been elected three times as a committee member of the BCS's Specialist Group on Artificial Intelligence (SGAI), servicing on the committee since 2013.

MAX BRAMER received the Ph.D. degree in artificial intelligence from the Open University, in 1977, for his research in knowledge representations.

He is currently an Emeritus Professor of Information Technology at the University of Portsmouth, having previously served as a Digital Professor of IT since 1989. His previous appointments include Knowledge Engineering Programme Manager at Hewlett-Packard Labs, Bristol, and currently the Head of the School of Computing and Information Technology, University of Greenwich. He has been actively involved in AI since becoming a part-time research student while a member of staff at the Open University, in 1972. He has published approximately 200 peer-reviewed publications, and has edited several collections of papers on AI topics. His other publications include a popular textbook *Principles of Data Mining* (Springer, 2020) which has now reached its fourth edition. His research interests include, among many others, the development of data mining algorithms, especially rule-based algorithms.

Prof. Bramer has served as the Chair of the British Computer Society Specialist Group on Artificial Intelligence for many years and has acted as the Conference Chair or Program Chair for many of its annual international conferences. He has also served for six years as the Chair of the Technical Committee on Artificial Intelligence of the International Federation for Information Processing (IFIP), followed by six years as its Vice-Chair. He was also the Chair of the IFIP working group on AI Applications for eight years, launching two annual series of international conferences. He is currently the Honorary Secretary of IFIP, having served as the Vice-President for six years. He was a member of the original International Steering Committee for the IEEE International Conference on Data Mining (ICDM) and was the Conference Chair for one of the earliest conferences, which was held in Brighton, U.K.