IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 13 November 2022, accepted 24 November 2022, date of publication 28 November 2022,
date of current version 5 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3225435

RESEARCH ARTICLE

Facing Up Fare War: Generating Competitive Price Models With Gene Expression Programming

MARCO ANTONIO BARRÓN[1], JOSÉ MARÍA LUNA[2], AND SEBASTIÁN VENTURA[2], (Senior Member, IEEE)

[1]Department of Computer Science and Numerical Analysis, University of Cordoba, 14071 Cordoba, Spain
[2]Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI), University of Cordoba, 14071 Cordoba, Spain

Corresponding author: Sebastián Ventura (sventura@uco.es)

This work was supported in part by the Spanish Ministry of Science and Innovation under Project PID2020-115832GB-I00, and in part by the European Fund of Regional Development.

ABSTRACT In the airline industry, the Revenue and Pricing teams generally spend a considerable amount of time analysing and interpreting the actions of their competitors. Most of the time the analysts have to use their analytical skills to create ad-hoc methods to interpret or find patterns in the fares. In this field, it is key to automate the process, avoid human errors, and add new features that provide accurate fares. Considering this, a gene expression programming algorithm is proposed to carry out this process, returning an interpretable rule set which acts as a recommender system to ease the daunting process done by the pricing teams manually. The proposal was applied to a real scenario with the information provided by the Air Canada airline for five months in Canadian markets. The experimental analysis revealed, by means of non-parametric statistical tests, that the proposed gene expression programming algorithm was key to getting the appropriate features and, hence, accurate and highly interpretable results. The proposal obtained extremely accurate results (around 96% in both accuracy and F1 measure) with a reduction of around 50% in the rule set in many cases.

INDEX TERMS Gene expression, airline fare war, classification, recommender systems.

I. INTRODUCTION

In recent years, there is a price war not only in airlines but also in other kind of industries and well-known companies such as Amazon [4] and Walmart [16]. In this regard, data mining (DM) techniques have been considered to provide the right prices according to their competitors, what is called pricing intelligence or competitive monitoring, which refers to the awareness of market-level pricing intricacies and the impact on business [4]. Pricing intelligence and the use of DM techniques are of special importance in the airline industry since highly interpretable rules can be used to implement automatic processes that monitor and obtain the right description of what is happening on the market landscape daily. Such descriptions may allow to improve any pricing strategy and minimize the loss of revenue due to the actions of their competitors [42].

When talking about the airline industry, pricing refers to the process of determining the fare classes, along with different products, services and restrictions in an origin and a destination (O-D) market. Each price point released to the public is attached to a specific fare class which is identified by one-letter fare codes depending on the airline. To earn money, the airlines offer tickets in different fare classes for every flight, but what is not known by passengers is the fact that every class (we usually consider it as a service class: economy, premium, business, and first) has also a subdivision, and this subdivision varies from airline to airline. This unknown structure of the fare classes is based on both service amenities and fare restrictions [27].

The almost universal rapid growth of new low-fare airlines with less restricted fare classes in the early 2000s, along with the response of legacy carriers to match those

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

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

fare classes led to an intensive war of prices which has become the norm of the industry [27]. Several times a day, airlines must classify the fares of other airlines into their fare classes, and this process is the key to maintaining the market's competitiveness, and protecting their revenue [15]. Thus, the Revenue and Pricing teams expend a considerable amount of time analyzing and trying to interpret the actions of their competitors, trends and patterns. This process is extremely complex as the number of restrictions, price points, and fares per fare class on the market is high. As a result, a partial understanding of the real actions taken by other airlines might be obtained, meaning useless information.

Nowadays, there is an important lack of commercial frameworks that provide automatic methodologies for pricing tasks. It is therefore necessary a new methodology able to automate pricing analysis, avoid human errors, add new features that provide accurate fares, and return an interpretable rule set that acts as a recommender system to ease the labour of the Pricing teams. Considering this, a gene expression programming (GEP) algorithm [11], [47] is proposed. GEP has been applied to many real-world applications [20], [37], [47], and it presents simplicity in the encoding, and versatility to explore huge search spaces. The propsoed GEP algorithm works as a feature learning algorithm that produces features (metrics) usually considered by the Pricing teams in their analyses: mean, lowest fare file, mode, standard deviation. New datasets are therefore formed to improve the predictive power of well-known classification algorithms without increasing the number of rules in the models. The experimental results carried out on 18 algorithms revealed statistical differences in terms of accuracy, F1 measure and interpretability when the proposed gene expression programming algorithm was applied. Last but not least, it is important to highlight that the proposed methodology was specifically designed to the airline industry as a recommender for the Pricing teams. Real data provided by Air Canada were considered to obtain significant (real) rules.

The novelty of this work is described as follows:

- We propose a gene expression programming algorithm to mimic a feature learning process, and to modify the fares as pricing teams do in a daily basis to find significant rules.
- We propose an automated methodology for the extraction of rules to obtain high interpretability when the market landscape changes due to the release of public fares by an airline. Additionally, the proposed automation of existing methodologies to provide the right fares reduces the human errors.

This paper is organized as follows: Section II briefly reviews the related background works. Section III shows the methodology of our study. Section IV introduces the performance measures, presents our experimental study and the discovered rules. Finally, section V summarizes the main conclusions and future research.

II. BACKGROUND

This section briefly presents reviews of the most important works related to rules interpretation and some of the most notable pricing wars work in the airline industry in recent years.

A. HIGHLY INTERPRETABLE ALGORITHMS

Classification with rule-based systems comes with two contradictory requirements in the obtained model [7]: interpretability and accuracy. Obtaining high degrees of interpretability and accuracy is a contradictory purpose and, in practice, one of the two properties prevails over the other. The interpretability is usually denoted as the capability to represent the behaviour of the real system in an understandable way. The accuracy is the capability on how well the system can guess the value of the predicted attribute for new data. To find the best trade-off between them [22] is an optimization problem that is very difficult to solve efficiently. According to the applicability domain [35], interpretability will be preferred to accuracy when the goal is to reveal hidden patterns in the data and act as a part of positive feedback loop back to the domain experts. The experts can indeed learn new dependencies and correlations from those models that can be easily interpreted. High interpretable models have been considered for many different tasks. For example, authors in [39] proposed a model that revealed the reasons why the drug would or would not work in specific cases to be much more meaningful and enable the experts to design better therapeutic drugs in the future. Bhargava et al. [5] proposed a methodology to detect malicious executable programs by predicting the outliers of the threat datasets. School failure has also been a focus of attention of some researchers [25]. As for the medical domain, many different authors have considered the use of high interpretable models on different diseases: metabolic syndrome [46], rheumatoid arthritis [8], diabetes [44], and breast cancer [3], among others. Additionally, some other authors [43] worked on the problem of face detection, being able to process images extremely fast and achieving high detection rates through simple and interpretable models. Therefore, when the goal is to obtain high interpretability, rule-based algorithms (Ridor [2], JRip [9], PART [12], OneR [17]) and decision tree algorithms (JCDT [1], simpleCART [6], BFTree [13], Decision Stump [21], J48 Consolidated [30], C4.5 [36], LMT [38], J48 Graft [41]) and associative classification algorithms (CBA [23], FOIL [32], PRM [33], FOIL2 [34], CPAR [45]) are usually considered.

B. PRICING WAR INTERPRETATION MODELS

A price war can be defined as a situation in which different companies compete by reducing prices. Airlines live in a constant price war and, to compete with other carriers, each airline must develop its monitoring and strategic methodologies to understand the actions that are happening on the market landscape. In general, airlines focus their analysis on price movement rather than an understanding of the rules to interpret the other carrier actions.

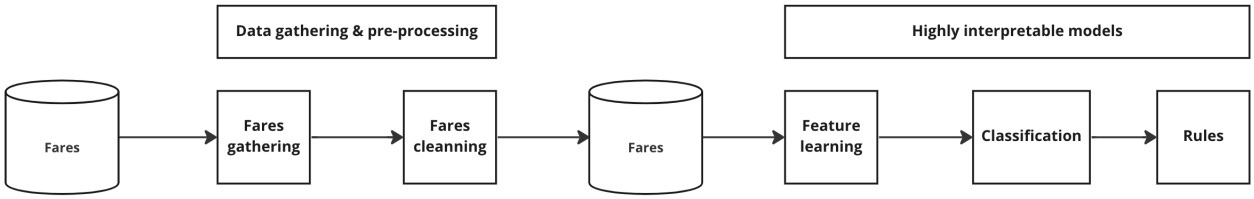

FIGURE 1. Proposed workflow for the extraction of highly interpretable rules.

Time series analysis has been the main technique used in this regard. Pitfield et al. [31] proposed an analysis of the pricing behaviour of competitive low-cost carriers in Europe. Pets and Rietveld [29] presented a methodology to analyze the pricing behaviour for the London-Paris market, considering Easy Jet and Ryan in the analysis. However, the use of time series analysis in pricing war may imply a low understanding of the insights, mainly due to the number of restrictions that can appear on the fares. In fact, time series analysis is a specific technique to show the price movement through a time period, but this technique lacks the ability to identify other important characteristics in the fares such as the routing number, advance purchase, between others. Hence, some authors have considered the use of different data mining techniques that improve the usefulness of the analysis. In this regard, Wolfhart et al. [42] developed a methodology to group flights for which the price series present similar behaviour. Haris et al. [15] proposed the use of different methods to predict trends to perform accurate decisions.

With all the above into consideration, it is important to highlight that, to the best of our knowledge, no approach provides a set of rules to understand the market behaviour. All the existing proposals act as black-box methodologies, and their final goal is to decide the action to take on prices accurately.

III. METHOD

The section describes the workflow proposed in our methodology (see Figure 1). This workflow includes five different steps that are described in depth below: gathering and data preprocessing, feature learning, classification and rules interpretation. All these steps are described below except for the rule interpretation, which is considered as a study case on real data in Section IV.

A. DATA GATHERING & PREPROCESSING

This first stage automatically applies a fare quotes process to gather the fares from a data base for all available historical public fares. In this stage, data preprocessing is required as most of the airlines present their fares in text format. This process might be different from company to company so this is considered as an open step which result should be a dataset including attributes and each line denoting travels.

As a matter of clarification, we show a sample data gathering and preprocessing step carried out for a specific airline (Air Canada). The original rraw data is a text file (see Figure 2) with the following attributes:

- Origin (ORG): airport or city where the trip starts.
- Destination (DES): airport or city where the trip ends.
- Fare basis code: alpha numeric code which summarizes the restrictions of the fares.
- Fare: the amount of the fare.
- Travel-ticket: it is the last day that the fare can be booked at the price shown and with the same restrictions.
- Advance purchased (AP): the number of day prior the commence of the trip in which the fare applies with the same price points and restrictions.
- Min and max stay: this restriction for round trip fares, which in this work we retrieved just one-way fares, thus, these restrictions do not apply.
- Routing (RTG): routing number.
- Travel Date: which is the date that applies to a specific fare.

Due to the attributes mentioned above are received as text a data preprocessing is required to produce data in the right input format. Thus, some specific data cleaning and preprocessing tasks are carried out in this stage to prepare all the previously described data so the next step in the proposed workflow could be performed correctly. The fare cleaning stage applies an intensive cleanup process to remove several characters and convert the text data into a table format. It removes some unusable attributes to determine the values of the attributes FARE and AP as numeric values. Thus, our method automatically extracts five attributes from the text

```
YTO-NYC       CXR-AC       THU 14DEC19                        CAD
THE FOLLOWING CARRIERS ALSO PUBLISH FARES YTO-NYC:
AA AS CO DL JJ JU LA NW PD PK UA US WS
//SEE FQHELP FOR INFORMATION ABOUT THE NEW FARE DISPLAYS//
ALL FEES/TAXES/SVC CHARGES INCLUDED WHEN ITINERARY PRICED
SURCHARGE FOR PAPER TICKET MAY BE ADDED WHEN ITIN PRICED
AC     YTONYC          14DEC19
    V FARE BASIS     BK    FARE   TRAVEL-TICKET AP  MINMAX  RTG
  1   WNA7A0TG       W X   192.00        T12JA  7/1  -/  -  636
  2   VNA7A0TG       V X   219.00        T12JA  7/1  -/  -  636
  3   WNA7A0FL       W X   242.00        T12JA  7/1  -/  -  636
  4   QNA3A0TG       Q X   251.00        T12JA  3/1  -/  -  636
  5   VNA7A0FL       V X   269.00        T12JA  7/1  -/  -  636
  6   HNA3A0TG       H X   292.00        T12JA  3/1  -/  -  636
  7   QNA3A0FL       Q X   301.00        T12JA  3/1  -/  -  636
  8   WNA7A0CO       W X   302.00        T12JA  7/1  -/  -  636
  9   VNA7A0CO       V X   329.00        T12JA  7/1  -/  -  636
 10   HNA3A0FL       H X   342.00        T12JA  3/1  -/  -  636
 11   UNA0A0TG       U X   346.00        T12JA  -/‡  -/  -  636‡
```

FIGURE 2. Fares retrieved in text format.

TABLE 1. Sample data subset of the transformed input data.

DMKT	FARE	ORG	DES	AP	RTG	TrDay	TrMonth	TrYear	DOW	Season	Class
YHZYZZ	98	YHZ	YYZ	30	R-900	27	Feb	2020	Thu	A	K
YULYYZ	166	YUL	YYZ	14	R-905	7	Jan	2020	Tue	L	S
YEGYZZ	98	YEG	YYZ	30	R-900	27	Feb	2020	Thu	Q	L
YVRYZZ	221	YVR	YYZ	21	R-900	7	Jan	2020	Tue	L	T
YVRYYC	129	YVR	YYC	30	R-900	5	Apr	2020	Sun	H	L
YYCYYZ	187	YYC	YYZ	14	R-5	20	Jan	2020	Wed	A	A
YYCYZZ	226	YYC	YYZ	21	R-900	15	Apr	2020	Wed	Q	T
YYCYYZ	187	YYC	YYZ	14	R-5	24	Jan	2020	Fri	A	A

files: ORG, DES, FARE, AP and RTG. During this stage additional features are also integrated. The Travel Date is split into three additional attributes, Travel Day (TrDay), Travel Month (TrMonth) and Travel Year (TrYear). A fourth attribute is created as a join of the ORG and DES attributes extracted from the text files which is called Direct Market (DMKT). Also, a fifth attribute denominated Day of Week (DOW) is created from the Travel Date. In order to add the final attribute called Season, we previously did a fare basic code analysis for Air Canada to identify the season in which the fares apply; therefore, the automated process reads the results of the analysis and assigns the season label to each fare class, which are L for low season, H for high season and Q for super peak season. After that, the attribute Class is filtered to consider the five fare classes provided by Air Canada (K, A, L, T, S), which belong to economy cabin. The K value, also known as coach discounted, represents the most economical fare class that include just the seat in flight with no ancillary. The A value represents a fare class that is exclusive for Air Canada, and it is applied to specific flights. Finally, the values L, T, and S belong to what is known economy coach fares, and they include some kind of ancillaries like a documented bag, certain food in the flight, among others. Small differences can be described for these fare classes as they depend on advanced purchases. Last but not least, it is important to highlight that every airline identifies its own fare classes based on its fare structure (the fare names vary). As a common standard in the airline industry, the fare classes are ordered by price points and after certain levels, they stop competing with the other airlines based on the market performance.

At the end of this stage, a new dataset is formed as shown in Table 1. The attribute DMKT is a joint of the attributes ORG and DES; we decided to keep these three attributes to find rules that might apply by each of these attributes. In fact, in the practice, there are markets or routes which have the same fare structures which are known as common rate routes. An example of these cases could be that a route between Toronto and Montreal could have the same fare structure as Toronto and Ottawa; therefore, keeping the three attributes the algorithms could find significant rules either by just DMKT, ORG or DES and they can provide or identify useful rules for these type of markets.

B. HIGHLY INTERPRETABLE MODELS

This section introduces the proposed gene expression programming (GEP) algorithm, specifically designed to mimic the feature learning process that the pricing teams must do on a daily basis. Here, it selects, groups and modifies some of the attributes in order to adapt the datasets before it is used by classification models. The final aim is to provide the best features to the classification algorithm and to find relevant rules (a small set of rules), allowing better interpretation and understanding of the market landscape. The following are the main processes of the proposed algorithm:

1) ENCODING

The proposed GEP algorithm encodes individuals as symbolic strings (fixed length), which are then expressed as non-linear entities of different sizes and shapes (expression trees) [26]. The algorithm considers a function set formed by attributes from the dataset, and a set of terminals (mmetrics) usually managed by the pricing teams while analyzing the fares T = {Mean, Lowest Fare File (LFF), Mode, Standard Deviation (SD)}. This set of terminals, is really useful to imitate the actions done by the Pricing teams. Here, the use of the mean fare, or its standard deviation, can be useful to find special sales or strategies applied in some of the markets. The lowest fare filed, which is the lowest fare available to the public by fare class, might identify specific discounts on certain dates. Last but not least, the mode fare could identify similar or equal strategies in different sales that are published constantly.

In order to improve the understanding of this encoding, we consider a real scenario (real data provided by Air Canada as we explained in the first step of the methodology: data gathering and preprocessing). In this example, the function set is formed by nine attributes F = {ORG, DES, AP, RTG, TrDay, TrMonth, TrYear, DOW, Season}. At this point, it should be remarked that attributes DMKT, FARE and Class were excluded from the function set since they remain as constant elements of each solution candidate. The FARE attribute is excluded for the function set because every solution candidate needs this attribute in order to calculate the selected measure from the set of terminals. The Class attribute is also excluded to preserve the class of every instance on the new datasets and the DMKT to maintain interpretability when the classification results are obtained. The fare class values

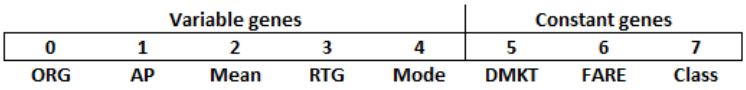

FIGURE 3. Sample chromosome including terminal variables and functions.

provided by Air Canada (K, A, L, T, S) were used in our study as the attribute Class to be classified.

Figure 3 represents an encoding sample of a chromosome created by our proposal on the sample dataset. The variable genes are the attributes and metrics which have been selected by a random process, and which will be used to modify the original dataset and obtain a new one. The constant genes are the ones we previously mentioned, which will be part of every of the chromosomes generated by the proposed algorithm. This algorithm aims to integrate a feature learning process by reading and decoding all the variable genes in the chromosome. Back to the sample encoding shown in Figure 3, it defines the first and second genes as elements from the function set. Therefore, the algorithm will act as a feature selector for the first gene (attribute ORG) which will be part of the new dataset without any modification on its values. After that, the algorithm will review the next two genes which are the attribute AP and the metric Mean. Thus, the algorithm will group the original dataset by using the attributes AP and two of the constant attributes: DMKT and Class to calculate the Mean over the attribute FARE creating a new attribute. This process will be explained in detail in the next procedure. The process will continue by checking again the next two genes RTG and Mode, because the first gene is an element of the set of functions and the second gene is an element of the terminal set, the transformation process is applied again by grouping the dataset using the attributes RTG, DMKT and Class, and it calculates the mode fare using the FARE attribute to create a new feature. Again, this process will be explained in detail in the next procedure. This feature learning process will run until all the variable genes are reviewed. At the end we obtain a new solution candidate (dataset) which it is also encoded in its chromosome form (see Figure 4).

It is important to highlight that some constraints are required to be satisfied. For example, the length of the solution candidates is set up to the formula $n = F + T = 13$, F being the number of elements in the terminal set, and T is the number of elements in the function set. It represents a total of 51,895,935 combinations.

2) INITIAL SET OF SOLUTION CANDIDATES

In order to obtain the initial set of solution candidates our algorithm runs two sequential processes: the initial encoding process and the feature learning (FL) process of the solution candidates. The first process randomly initializes the initial set of solution candidates by encoding and selecting the attributes from the set of functions and the terminal set until the size of the chromosome is reached, as it was previously shown in Figure 3. After the first process is executed, the feature learning process is run to transform and create the new solution candidates (datasets).

Algorithm 1 Generation of the Initial Population

Input pop-size, chromosome-size, F, T
Output solution candidates

```
1:  S ← F ∪ T
2:  solution candidates ← ∅
3:  for i = 1 to pop-size do
4:      chromosome ← ∅
5:      for j = 1 to chromosome-size do
6:          g ← get.random.element(S)
7:          if g ∈ chromosome and g ∈ F then
8:              g = null
9:          end if
10:         chromosome ← chromosome ∪ g
11:     end for
12:     chromosome ← remove.null.values(chromosome)
13:     solution candidates ← solution candidates ∪ chromo-
        some
14: end for
15: return solution candidates
```

Algorithm 1 shows the pseudo-code for the generation of the initial set of solution candidates in their encoding form. The algorithm starts with the creation of two variables; the variable s, which contains the elements of the set of functions and terminals set, and the variable to keep the solution candidates obtained. The values in s are the genes that will form the chromosomes after a random process is executed to select them. Lines 5 to 11 denote the random process to select a value from s and which it is assigned to the variable g; this variable will determine the gene in the j-th position of the chromosome. In line 7 if the chromosome already contains the value of g, and if this value is an element from the set of functions, the line 8 is executed so the g variable will be changed as *null*. This condition is set to avoid repeated attributes in the same solution candidate. Finally, g is included into the chromosome. Once the chromosome is formed, any null value is removed and it is included in the solution candidates list. When the number of chromosomes created is equal to the predefined pop-size, the process ends and it returns the solution candidates in their encoding form.

Algorithm 2 defines the pseudo-code of the feature learning process. The algorithm receives the initial set of solution candidates in their encoding form which were previously

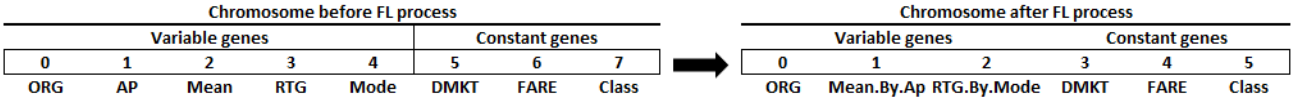

FIGURE 4. Feature learning process of a solution candidate.

Algorithm 2 Feature Learning

Input solution candidates, F
Output t-solutions, e-solutions

```
1:  for ∀ solution candidate ∈ solution candidates do
2:    chromosome ← solution candidate
3:    t-chromosome ← Ø
4:    t-dataset ← Ø
5:    while j = 1 to length(chromosome) - 3 do
6:      if chromosome[j] ∈ F and chromosome[j + 1] ∈ F
        then
7:        apply feature selection to chromosome[j]
8:        t-chromosome[j] ← new encoded attribute
9:        t-dataset[j] ← new attribute
10:     else
11:       create a new feature using chromosome[j] and
          chromosome[j+1]
12:     end if
13:     j ← j + 1
14:   end while
15:   e-solutions ← e-solutions ∪ t-chromosome
16:   t-solutions ← t-solutions ∪ t-dataset
17: end for
18: return t-solutions, e-solutions
```

created by Algorithm 1, and the set of functions (F). Line 1 represents the general loop to read every chromosome that is going to be decoded and transformed to obtain the datasets. Lines 2 to 13 include such decoding and transformation processes. For each chromosome in the set of solution candidates, the algorithm analyses every of its genes (see Lines 5 to 14). Checking j-th gene, it also takes the $j+1$-th gene to determine which action the FL algorithm should apply. If both genes are elements of F, then the algorithm acts as a feature selector leaving the attribute in j-th position without any modification to be part of the transformed dataset. If the j-th gene belongs to F, but the $j+1$-th gene does not (it belongs to the set of terminals), then the algorithm creates a new feature by grouping the attribute in the j-th position, the element in the $j+1$-th position, and the constant genes. As we previously explained, the FARE is a numeric attribute which represents the amount of the fare being the only suitable attribute to calculate the measures, allowing to create the modified chromosomes by integrating new attributes. To understand it better, let us go back again to the sample individual shown in Figure 4, which is an encoded solution candidate created by the initial Algorithm 1 and its transformation by Algorithm 2. In this case, we have ORG in the j-th position and the next gene is the attribute AP, both being part of the function set. The algorithm selects ORG attribute as part of the transformed dataset without modifying its values. When the next iteration is run, AP is in the j-th position of the chromosome, and the next gene is the Mean. In this case, the algorithm groups the dataset by using the attributes Class and AP and calculate the Average fare using the attribute FARE creating a new attribute that is called Mean.By.AP, and which is integrated to the new dataset. As we previously explained the process continues until all the genes of the chromosome have been decoded (see line 12). Continuing with the example illustrated in Figure 4, the Algorithm checks the next genes obtaining RTG and Mode. Therefore, the algorithm applies again the FL process and creates a new attribute called RTG.By.Mode. This process stops when all the variable genes are analysed. The new encoded chromosome (e-solutions in Algorithm 2) and a transformed dataset are also created (t-dataset Algorithm 2). The proposed algorithm finally returns the t-solutions as datasets, and the new chromosomes in their encoded form e-solutions. Note that solution candidates have not been evaluated yet.

3) EVALUATION

This procedure is responsible for assigning a fitness value to each solution candidate (see Equation 1). In the proposed methodology we have to evaluate how good the resulting datasets obtained through the FL process are. To do this, a classification model is applied to each resulting dataset to provide a fitness function that is based on a combination of two measures: F-score (F_1) and interpretability (Nr or number of rules). F-score rate represents the harmonic mean between recall and precision values [19], thus, it is calculated from the precision and recall of the test, where the precision is the number of true positive results divided by the number of all positive results, including those not identified correctly, and the recall is the number of true positive results divided by the number of all samples that should have been identified as positive. It is formally defined as $F_1 = \frac{T_P}{T_P+1/2(F_P+F_N)}$. Interpretability [14] is a measurement of simplicity of the resulting model. In a rule-based model, it is quantified in terms of the number of rules that forms the model (model size) [14], so the lower this number, the more interpretable the model is.

$$Fitness = \frac{F_1 * Nr}{2} \tag{1}$$

Algorithm 3 shows the pseudo-code of the evaluation process. The fitness function optimizes the solution candidates according to the transformed attributes given by the GEP

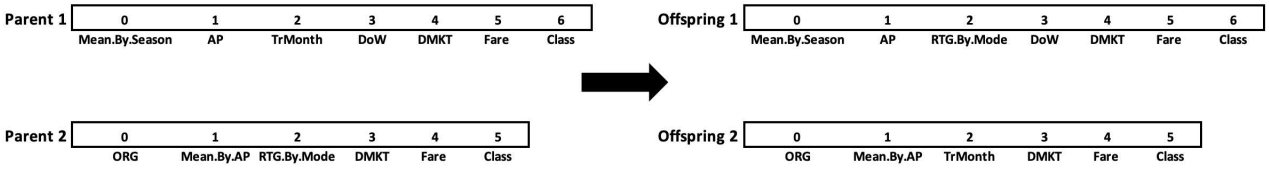

FIGURE 5. Sample of the generation of two offspring by applying the crossover genetic operator.

Algorithm 3 Evaluation of the Solution Candidates

Input t-solutions, e-solutions
Output Evaluated solution candidates

```
Evaluated solution candidates ← ∅
for i = 1 to length(t-solutions) do
    if length(get.original.features(solution candidate[i])) < 4 then
        fitness ← 0
    else
        model ← create.classification.model(t-solution[i])
        fitness ← get.fitness(model)
    end if
    Evaluated solution candidates ← Evaluated solution candidates ∪ [e-solution[i], fitness]
end for
return Evaluated solution candidates
```

(e.g. the average flights' fare in a specific month), looking for good attributes and combination of them in terms of accuracy and interpretability. This algorithm receives a set of solution candidates as datasets (t-solutions), and the set of solution candidates in the encoded form (e-solutions). The Algorithm works as a main loop to evaluate every t-solution created in Algorithm 2. In each iteration, the proposal verifies if the solution candidate contains at least four of the original attributes from the original dataset, evaluating the solution candidate through a classification algorithm. On the contrary, if the solution candidate does not contain at least four of the original attributes (excluding the attribute Class), then a fitness value of 0 is assigned. The requirement of including at least four or more of the original attributes is to maintain interpretability. Finally, the proposed procedure returns the solution candidates in their new encoded form together with their respective fitness value. Back to the example shown in Figure 3, the chromosome contains just three of the original attributes after the FL process: ORG, DMKT and FARE. Therefore, a 0 fitness value is assigned to this solution candidate.

In order to obtain the rules and calculate the fitness value, we propose the use of white box classification models. In this regard, the evaluation process is carried out by considering classical classification algorithms available in the well-known Rweka tool [18] and the LAC library [28]:

- Four induction algorithms: JRip [9], which is a propositional rule learner; OneR [17], which uses the minimum-error attribute for class prediction; PART [12], which uses separate- and-conquer method; and Ridor [2], which is an implementation of the RIpple-DOwnRule learner.
- Nine decision trees: JCDT [1]; simpleCART [6]; BFTree [13]; Decision Stump [21]; J48 Consolidated [30]; algorithms for generating a pruned or unpruned trees J48 and C4.5 [36]; LMT [38]; J48 Graft [41].
- Five associative classification algorithms: CBA [23] discovers a subset of association class association rules and produce a classification model on the extracted rules. FOIL [32] and FOIL2 [34] greedy algorithms that learns rules to distinguish positive from negative examples. CPAR [45] inherits the basic idea of FOIL in rule generation and integrates the features of associative classification in predictive rule analysis. PRM [33] selects the best rule among a set of rules generated.

4) GENETIC OPERATORS

Here every genetic operator used in the proposed algorithm is described. First, as for the selection, a roulette-wheel selection [47] is taken, which consists of mapping the fitness of each solution candidate to a slice of the roulette wheel proportional to its fitness. Then, the roulette is spun as many times as there are solution candidates in the population in order to keep the population size constant. Thus, with roulette-wheel selection the solution candidates are selected both according to fitness and the luck of the draw, which means that some times the best traits might be lost. However, by combining roulette-wheel selection with the cloning of the best solution candidates of each generation, we guarantee that at least the very best traits are not lost. This technique of cloning the best-of-generation solution candidates is known as elitism and is used by most stochastic selection schemes.

Two key genetic operators are proposed to obtain new solution candidates in every generation. The crossover genetic operator was proposed to intensify the population diversity. This genetic operator randomly chooses a cut point among the variable genes of one parent's chromosome. The same process is repeated for another parent. Two offspring are formed by combining the genes of the split chromosomes.

FIGURE 6. Sample of the generation of a new individual applying the mutation genetic operator.

These offspring include genes from both parents. As a matter of clarification, let us consider two sample parents (see Figure 5 encoded by the Air Canada dataset we have considered along this paper for the explanations). Parent 1 represents a dataset including the mean by season for each of the direct markets (DMKT). It also includes the advanced purchase (AP) attribute as the number of days in which the fare is analysed, and also the month and the day of the week in which the fare is applied. Parent 2 represents a dataset including the origin (airport or city where the trip starts), the mean by AP, and the mode for each routing of each DMKT. In this sample, the cut point is randomly chosen at gene 2, returning two offspring with information from both parents: offpsring 1 represents a dataset including the mean by season for each of the direct markets (DMKT), the advanced purchase (AP) attribute as the number of days in which the fare is analysed, and the mode for each routing of each DMKT; offspring 2 represents a dataset including the origin, the mean by AP, and the month in which the fare is applied.

Additionally, the mutation genetic operator was proposed to diversify the population. This genetic operator randomly chooses a gene from the variable genes of one parent's chromosome, and this gene is replaced with a new random value (could be a blank space or removal). The new individual is similar to the previous one (just a small variation is added). As a matter of clarification, let us consider a sample parent (see Figure 6 encoded by the Air Canada dataset we have considered along this paper for the explanations), which represents a dataset including the mean by season for each of the direct markets (DMKT), the advanced purchase (AP) attribute as the number of days in which the fare is analysed, the mode for each routing, and the day of the week in which the fare is applied. A removal of the attribute representing the mean by season is applied, so the new individual represents a dataset with the advanced purchase (AP) attribute as the number of days in which the fare is analysed, the mode for each routing, and the day of the week in which the fare is applied.

Last but not least, the general workflow of the proposed GEP-FL algorithm (see Algorithm 4) is described. The first three steps are needed to produce the initial set of solution candidates. Then an iterative process is carried out over a number of iterations (generations) returning the best solution candidates found after the loop. No additional explanation is required for this algorithm since all the processes carried out by it were previously described.

Algorithm 4 GEP-FL Algorithm

Input n-iterations
Output best n solution candidates

1: F = function set
2: T = terminal set
3: Generation of the initial population (see Algorithm 1)
4: Feature Learning (see Algorithm 2)
5: Evaluation of solution candidates (see Algorithm 3)
6: **for** i = 1 to n-iterations **do**
7: Roulette-wheel selection
8: Apply genetic operators
9: Evaluation of solution candidates
10: Update population and keep best n solution candidates
11: **end for**
12: **return** best n solution candidates

IV. PERFORMANCE MEASURES AND EXPERIMENTAL STUDY

The aim of this section is to firstly analyse whether there is a clear improvement when the proposed GEP-FL algorithm is used. Then, we analyse which classification algorithm is better for the problem at hand. As we previously described, the evaluation process could be carried out by any classification algorithm proposing a total of eighteen algorithms (see Section III-B). Finally, we apply the proposed methodology to obtain highly interpretable rules on a real scenario. Air Canada gave us a dataset including travels along 5 months.

A. EXPERIMENTAL RESULTS

This first subsection carries out several experiments to demonstrate the usefulness of the proposed GEP-FL approach. In this regard, eighteen different classification algorithms are used in the evaluation procedure. Ten different executions were done on each algorithm and the average results of these ten runs are shown in Table 2. The final aim is to demonstrate that the use of feature learning is appropriate for this problem, so we compare how the algorithms behave with and without the use of the GEP-FL approach on three metrics: accuracy, f-score and interpretability. We also compare this methodology with a classic random search approach (no crossover and mutation genetic operators) to demonstrate that the proposed evolutionary approach is appropriate and produces a clear improvement. In this regard, we first compare the proposed GEP-FL approach to two different

TABLE 2. Experimental results obtained by considering different classification algorithms in the evaluation phase.

	Accuracy (Acc)			F-score (F1)			Interpretability (Nr)		
Algorithm	Org. Data	Rand-FL	GEP-FL	Org. Data	Rand-FL	GEP-FL	Org. Data	Rand-FL	GEP-FL
BFTree	99.6	99.8	**99.9**	99.5	99.7	**99.9**	289	119	**64**
C50	99.7	99.7	**99.9**	99.6	99.8	**99.9**	166	102	**74**
CBA	97.9	98.7	**99.9**	97.9	98.6	**99.9**	238	60	**39**
CPAR	98.4	98.6	**99.6**	98.4	98.3	**99.6**	502	93	**84**
DecisionStump	45.7	**47.2**	45.1	60.9	**62.8**	61.7	**3**	**3**	**3**
FOIL	99.3	99.8	**99.9**	99.3	99.8	**99.9**	217	61	**37**
FOIL2	99.3	99.8	**99.9**	99.3	99.8	**99.9**	217	73	**45**
J48	99.7	99.8	**99.9**	99.6	99.7	**99.9**	273	114	**99**
J48Consolidated	99.6	99.7	**99.9**	99.6	99.7	**99.9**	298	**89**	119
J48graft	99.7	99.7	**99.9**	99.6	99.8	**99.9**	919	349	**232**
JCDT	99.6	**99.9**	99.6	99.6	**99.9**	99.6	238	167	**112**
JRip	99.7	99.8	**99.9**	99.6	99.8	**99.9**	89	80	**37**
LMT	97.8	98.9	**99.1**	97.9	98.8	**99.1**	113	**10**	**10**
OneR	86.4	**98.4**	94.5	85.3	**98.1**	94.0	**2**	**2**	**2**
PART	99.6	99.2	**99.9**	99.6	99.3	**99.9**	146	**48**	65
PRM	98.5	**99.8**	99.6	98.5	**99.8**	99.7	344	**64**	75
Ridor	99.6	97.4	**99.9**	99.6	97.2	**99.9**	**6**	85	142
simpleCART	99.7	**99.9**	**99.9**	99.6	99.8	**99.9**	146	91	**65**

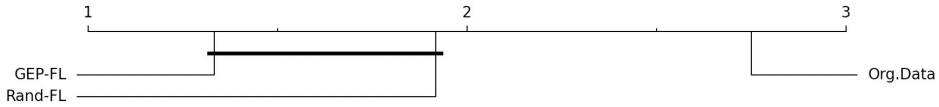

FIGURE 7. Critical difference diagram showing a statistical comparison of the Accuracy using the Shaffer's test.

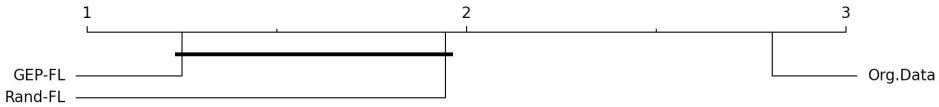

FIGURE 8. Critical difference diagram showing a statistical comparison of the F1 measure using the Shaffer's test.

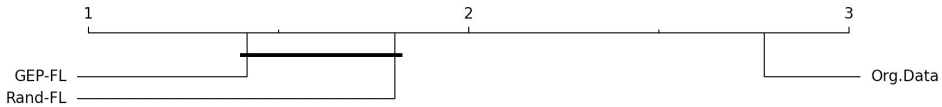

FIGURE 9. Critical difference diagram showing a statistical comparison of the number of rules using the Shaffer's test.

methodologies: *a*) considering the original data; *b*) using a random search approach. In this analysis, all the classification algorithms were executed using a ten fold cross-validation, and all the available information after the data gathering a pre-processing phase has been completed.

Comparing the accuracy results obtained by these three methodologies (see Table 2), it can be observed that no huge differences were obtained, so a non-parametric Friedman [10] test was conducted to statistically determine whether there are differences among these three methodologies (Original data, random search, and GEP approach). The p-value computed, that is, $p = 0.000063$, through the statistic of the test rejected the null hypothesis that all the methodologies equally perform in terms of Accuracy with an α value of 0.01. A post-hoc test in therefore applied to obtain significant differences among the methodologies. The Shaffer's test (see Figure 7) demonstrated that GEP-FL is the methodology that best performs in terms of Accuracy with an α value of 0.01, existing significant differences with regard to the original data, and huge differences with regard to the random search methodology. In terms of the F1 measure, the Friedman test returned p-value of $p = 0.000015$, rejecting the null hypothesis that all the methodologies equally perform with an α value of 0.01. A post-hoc test in then applied to obtain significant differences among the methodologies. The Shaffer's test (see Figure 8) demonstrated that GEP-FL is the methodology that best performs in terms of F1 measure with an α value of 0.01, existing significant differences with regard to the original data, and huge differences with regard to the random search methodology.

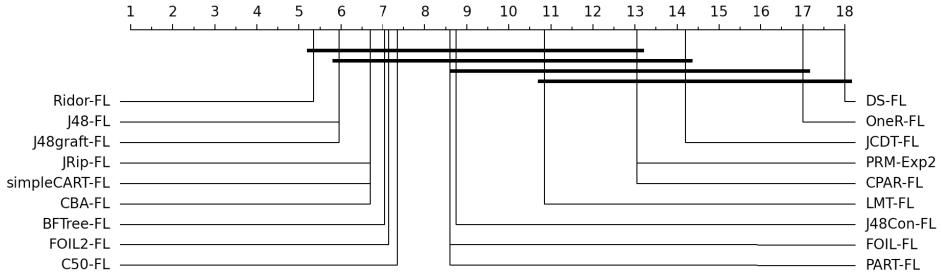

FIGURE 10. Critical difference diagram showing a statistical comparison of the Accuracy using the Shaffer's test.

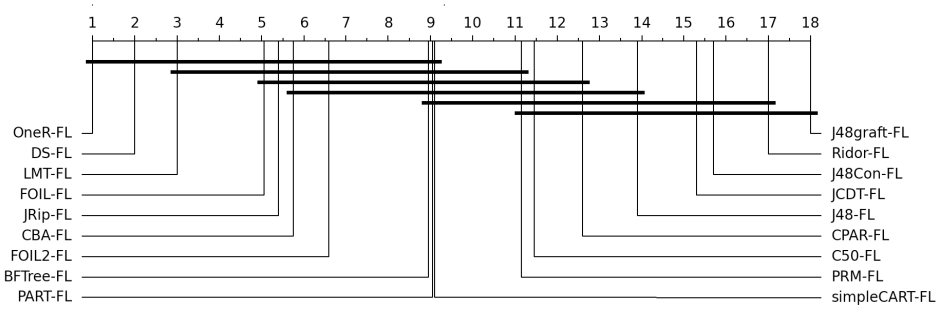

FIGURE 11. Critical difference diagram showing a statistical comparison of the interpretability using the Shaffer's test.

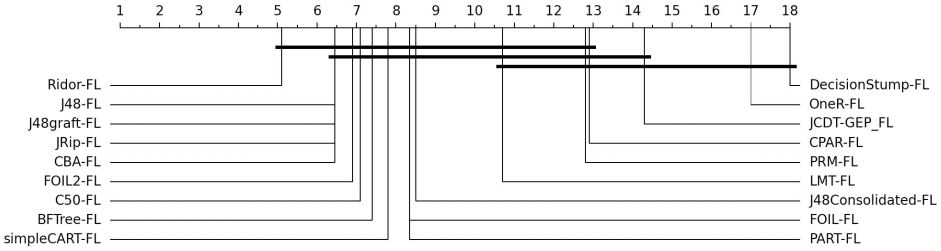

FIGURE 12. Critical difference diagram showing a statistical comparison of the F-score measure using the Shaffer's test.

TABLE 3. Friedman results for different GEP-FL algorithms in the evaluation phase.

Measure	Statistic	p-Value
Accuracy (Acc)	111.00	7.822e-16
F-score (F_1)	108.29	2.531e-15
Size (Nr)	163.81	2.2e-16

Finally, as of the number of rules, the Friedman test returned p-value of $p = 0.000040$, rejecting the null hypothesis that all the methodologies equally perform with an α value of 0.01. The Shaffer's post-hoc test (see Figure 9) showed that GEP-FL is the best methodology in terms of number of rules. The random search methodology also performs really well.

To sum up, the statistical results showed that the proposed GEP-FL approach improves the classification performance and the interpretability of the models. Therefore, in order to determine which classification algorithm is the most suitable to be applied to our methodology we conducted a second statistical analysis based on the differences among the algorithms (see Table 3). First, the p-values computed through the statistic of Accuracy, F-score and interpretability (Nr) tests rejected the null hypothesis that all algorithms equally behave considering $\alpha = 0.01$. Then, the Shaffer's post-hoc test was performed to detect where these significant differences were located, considering a significance level of $\alpha = 0.01$. The results are summarized through the critical difference diagrams shown in Figures 10, 11 and 12. As it is shown, JRip is in fourth position in Accuracy and F-score, whereas it is in fifth position in terms of interpretability. In none of these three analyses the JRip algorithm is statistically worse than the best algorithm (see Figures 10, 11 and 12). With all the above into considerantion, we take JRip as the algorithm that produces an overall best performance.

TABLE 4. Rules discovered when the proposed methodology is executed on a real scenario.

No.	Resulting rules:
1	(Season = A) and (AP.Mode <= 63) => Class=K (512.0/0.0)
2	(AP <= 7) and (AP.Mode = 148.45) => Class=K (114.0/0.0)
3	(AP <= 7) and (AP.Mode <= 124.88) and (ORG = YWG) => Class=K (119.0/0.0)
4	(DMKT = YYCYYZ) and (AP.Mode <= 129.4) => Class=K (96.0/0.0)
5	(AP.Mode <= 145) and (DES = YYZ) and (ORG = YVR) => Class=A (641.0/0.0)
6	(AP.Mode <= 145) and (ORG = YWG) => Class=A (842.0/0.0)
7	(RTG = R-5) and (AP <= 7) => Class=A (451.0/0.0)
8	(AP <= 21) and (AP >= 18) and (AP.Mode <= 141) => Class=T (1294.0/0.0)
9	(Season = A) and (AP.Mode <= 188) and (AP.Mode >= 177) => Class=T (580.0/0.0)
10	(ORG = YUL) and (Season = H) and (AP.Mode <= 151) => Class=T (144.0/0.0)
11	(AP >= 21) => Class=L (7030.0/0.0)
12	(AP.Mode = 151) => Class=L (1298.0/0.0)

B. DISCOVERED RULES

The aim of this study is to apply the proposed methodology to a real scenario. In this study, we will apply JRip as the classification algorithm used in the evaluation phase since it was the algorithm that statistically outperforms the rest in the previous section. Table 4 presents the results obtained by the proposed methodology, denoting by parentheses the number of cases satisfied and not satisfied by the rule. The attributes that appear the most are *Season* and *AP.Mode*. The latter is one of the features learned by the GEP-FL algorithm, which provides interesting rules and easy to be interpreted. Below we provide an interpretation of twelve most significant discovered rules:

- Rule number 1 is able to show that the amount boundary for fare class K is \$63 for all markets and all seasons.
- Rule number 2 shows a clear pricing strategy for published fares with an advance purchase of 7 days and fare class K the mode fare is \$148.45. Due to the number of cases applied we can observed that they are tactical fares which apply just for a restricted number of flights in which their performance is not optimal; therefore, in this case a low price with a low advance purchase means that the carrier is trying to to fill some of these flights. This type of tactical fares is difficult to detect in a regular manual process.
- Rule 3 shows that the mode fares with an advance purchase of 7 days for flights from Winnipeg to other destinations are at \$124.88 for class K.
- Rule 4 shows an interesting pricing strategy in which the most common price published for trips between Calgary and Toronto are \$129.4 for class K. Due to the number of cases that this rule applies we can deduce that this the structural amount for this fare class in this specific market.
- We can observed from Rule 5 that the structural price point for the fare class K in a flight between Toronto and Vancouver with and advance purchase of at least 45 days is at \$145.
- Rule 6 show structural pricing strategy for flights which origin is Winnipeg and with an advance purchase of 45 days for class A at \$145.
- It can be observed in Rule 7 an interesting discovery from our methodology that fares that belong to class A and with the restriction of 7 days of advance purchase in most of the cases are for non-stop. This is a significant discovery because we can observe a clear use of the fare class as that is being used for tactical fares which are difficult to be discovered from manual processes.
- Rule 8 shows that fares with an advance purchase between 18 and 21 days the most common price point for class T is \$141.
- It can be observed that Rule 9 that fares belonging to class T for fares that apply all year agnostic of the advance purchase the price points fluctuates between \$117 and \$188. This a a very interesting discovery because the rule is showing the fare band for this class, which it is basically the lowest and the highest price point in which a T fare class can be priced.
- Rule 10 is showing that flights from Montreal for high season the mode fare agnostic of the advance purchase for class T is \$151.
- Rule 11 shows that the L fare class is available for just people trying to book flights with at least 21 days from the departure date.
- Rule 12 shows that most common price point for L fare class is \$151.

At this point, it is key to demonstrate whether the returned rules are good enough, so these rules are also compared to those obtained by the well-known FP-Growth [40] algorithm for mining association rules on the best solution candidates (datasets) that were obtained from GEP-FL. Comparison based on the number of rules (see Table 5) clearly demonstrates that the proposed approach returns a small set of rules that can be understood by the end-user in an easier way than a set of thousands of rules as FP-Growth returns. Additionally, we have analysed the small set of rules returned by our approach in terms of the Lift quality measure [24], which measures the importance of the rule. This is a measure of the performance of a targeting model at classifying cases with an enhanced response compared to a random choice targeting model. The obtained rules have a Lift value greater than 4, which clearly denotes the importance of the rules discovered. Additionally, the returned rules are not general rules that can

TABLE 5. Number of rules obtained by GEP-FL JRip and FP-growth.

Data	GEP-FL	FP-growth
1	37	1,692
2	35	1,897
3	41	1,857
4	33	1,757
5	37	1,787
6	44	1,716
7	29	964
8	37	1,997
9	37	1,985
10	45	2,039

be easily obtained. These rules appear in less than 5% of the transactions, so it is computationally complex to discover them, and they are not easily obtained by analysing the data.

V. CONCLUSION

The extraction of interpretable rules in airline fares can be a difficult task not only because it is a multifactor problem, but also because the numbers of fares published by a carrier can contain similar restrictions and characteristics in the fare classes that becomes a very complex task to find relevant information. To solve this issue, this paper proposed a pricing methodology to create special datasets by transforming the attributes through the implementation of a GEP feature learning algorithm. The proposal is able to obtain high interpretable models with enough number of rules and smaller number of antecedents per rules without affecting the classification performance.

The proposed methodology was applied to a real scenario using data from the biggest airline in Canada, that is, Air Canada. Based on the results obtained, different conclusions could be described:

1) The use of an automated methodology that allows to produce interesting information to pricing teams is feasible, avoiding the human interaction and, therefore, human errors.
2) The proposed methodology is able to reduce the number of rules extracted, and decrease the number of conditions on the rules, which is crucial to increase the interpretability of the models.
3) The proposed methodology is able to mimic the feature learning process usually carries out by the pricing teams.

REFERENCES

[1] J. Abellán and S. Moral, "Building classification trees using the total uncertainty criterion," *Int. J. Intell. Syst.*, vol. 18, no. 12, pp. 1215–1225, Dec. 2003.

[2] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Mach. Learn.*, vol. 6, no. 1, pp. 37–66, 1991.

[3] M. Alshammari and M. Mezher, "A comparative analysis of data mining techniques on breast cancer diagnosis data using WEKA toolbox," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 8, pp. 224–229, 2020.

[4] N. Archak, A. Ghose, and P. G. Ipeirotis, "Deriving the pricing power of product features by mining consumer reviews," *Manage. Sci.*, vol. 57, no. 8, pp. 1485–1509, 2011.

[5] N. Bhargava, A. Jain, A. Kumar, and D.-N. Le, "Detection of malicious executables using rule based classification algorithms," in *Proc. 1st Int. Conf. Inf. Technol. Knowl. Manage.*, Jan. 2018, pp. 35–38.

[6] L. Breiman, J. Friedman, C. Stone, and R. Olshen, *Classification and Regression Trees*. New York, NY, USA: Chapman & Hall, 1984.

[7] A. Cano, A. Zafra, and S. Ventura, "An interpretable classification rule mining algorithm," *Inf. Sci.*, vol. 240, pp. 1–20, Aug. 2013.

[8] S. P. Chokkalingam and K. Komathy, "Comparison of different classifier in WEKA for rheumatoid arthritis," in *Proc. Int. Conf. Hum. Comput. Interact. (ICHCI)*, Aug. 2013, pp. 1–6.

[9] W. W. Cohen, "Fast effective rule induction," in *Proc. 12th Int. Conf. Mach. Learn.*, Tahoe City, CA, USA, Jul. 1995, pp. 115–123.

[10] T. Eftimov and P. Korošec, "A novel statistical approach for comparing meta-heuristic stochastic optimization algorithms according to the distribution of solutions in the search space," *Inf. Sci.*, vol. 489, pp. 255–273, Jul. 2019.

[11] C. Ferreira, *Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence* (Studies in Computational Intelligence), vol. 21. Cham, Switzerland: Springer, 2006.

[12] E. Frank and I. H. Witten, "Generating accurate rule sets without global optimization," in *Proc. 15th Int. Conf. Mach. Learn. (ICML)* San Francisco, CA, USA: Morgan Kaufmann, 1998, pp. 144–151.

[13] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: A statistical view of boosting (with discussion and a rejoinder by the authors)," *Ann. Statist.*, vol. 28, no. 2, pp. 337–407, 2000.

[14] S. García, A. Fernández, J. Luengo, and F. Herrera, "A study of statistical techniques and performance measures for genetics-based machine learning: Accuracy and interpretability," *Soft Comput.*, vol. 13, no. 10, p. 959, 2009.

[15] N. A. Haris, M. Abdullah, A. T. Othman, and F. A. Rahman, "Optimization and data mining for decision making," in *Proc. World Congr. Comput. Appl. Inf. Syst. (WCCAIS)*, vol. 10, Jan. 2014, pp. 1–4.

[16] A. S. Harsoor and A. Patil, "Forecast of sales of Walmart store using big data applications," *Int. J. Res. Eng. Technol.*, vol. 4, no. 6, pp. 51–59, Jun. 2015.

[17] R. C. Holte, "Very simple classification rules perform well on most commonly used datasets," *Mach. Learn.*, vol. 11, no. 1, pp. 63–90, Apr. 1993.

[18] K. Hornik, C. Buchta, and A. Zeileis, "Open-source machine learning: R meets weka," *Comput. Statist.*, vol. 24, no. 2, pp. 225–232, May 2009.

[19] H. M and S. M. N, "A review on evaluation metrics for data classification evaluations," *Int. J. Data Mining Knowl. Manage. Process*, vol. 5, no. 2, pp. 1–11, Mar. 2015.

[20] J. Hu and W. Guo, "Flexibility analysis in waste-to-energy systems based on decision rules and gene expression programming," in *Proc. IEEE Int. Conf. Syst., Man Cybern. (SMC)*, Oct. 2019, pp. 988–993.

[21] W. Iba and P. Langley, "Induction of one-level decision trees," in *Machine Learning Proceedings*. Amsterdam, The Netherlands: Elsevier, 1992, pp. 233–240.

[22] U. Johansson, C. Sönströd, T. Löfström, and H. Boström, "Obtaining accurate and comprehensible classifiers using Oracle coaching," *Intell. Data Anal.*, vol. 16, no. 2, pp. 247–263, Mar. 2012.

[23] B. Liu, W. Hsu, Y. Ma, and B. Ma, "Integrating classification and association rule mining," in *Proc. Kdd*, vol. 98, 1998, pp. 80–86.

[24] J. M. Luna, M. Ondra, H. M. Fardoun, and S. Ventura, "Optimization of quality measures in association rule mining: An empirical study," *Int. J. Comput. Intell. Syst.*, vol. 12, no. 1, pp. 59–78, Nov. 2018.

[25] C. Márquez-Vera, A. Cano, C. Romero, and S. Ventura, "Predicting student failure at school using genetic programming and different data mining approaches with high dimensional and imbalanced data," *Appl. Intell.*, vol. 38, no. 3, pp. 315–330, Apr. 2013.

[26] M. Mitchell, *An Introduction to Genetic Algorithms*. Cambridge, MA, USA: MIT Press, 1998.

[27] A. P. Ododni Belobaba and C. Barnhart, "The global airline industry," in *Soft Computing and Industry*, 2nd ed. Hoboken, NJ, USA: Wiley, 2016, p. 82.

[28] F. Padillo, J. M. Luna, and S. Ventura, "LAC: Library for associative classification," *Knowl.-Based Syst.*, vol. 193, Apr. 2020, Art. no. 105432.

[29] E. Pels, "Airline pricing behaviour in the london–Paris market," *J. Air Transp. Manage.*, vol. 10, no. 4, pp. 277–281, May 2004.

[30] J. M. Pérez, J. Muguerza, O. Arbelaitz, I. Gurrutxaga, and J. I. Martín, "Combining multiple class distribution modified subsamples in a single tree," *Pattern Recognit. Lett.*, vol. 28, no. 4, pp. 414–422, Mar. 2007.

[31] D. Pitfield, "A time series analysis of the pricing behaviour of directly competitive 'low-cost' airlines," *Int. J. Transp. Econ.*, vol. 32, no. 1, pp. 15–39, Feb. 2005.
[32] J. R. Quinlan and R. M. Cameron-Jones, "Induction of logic programs: FOIL and related systems," *New Gener. Comput.*, vol. 13, nos. 3–4, pp. 287–312, Dec. 1995.
[33] D. Rai, A. S. Thoke, and K. Verma, "Enhancement of associative rule based FOIL and PRM algorithms," in *Proc. Students Conf. Eng. Syst.*, Mar. 2012, pp. 1–4.
[34] K. D. Rajab, "New associative classification method based on rule pruning for classification of datasets," *IEEE Access*, vol. 7, pp. 157783–157795, 2019.
[35] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Mach. Intell.*, vol. 1, no. 5, pp. 206–215, 2019.
[36] S. L. Salzberg, "C4.5: Programs for machine learning by J. Ross Quinlan. Morgan Kaufmann publishers, Inc., 1993," *Mach. Learn.*, vol. 16, no. 3, pp. 235–240, Sep. 1994.
[37] J. Shiri, A. A. Sadraddini, A. H. Nazemi, O. Kisi, G. Landeras, A. F. Fard, and P. Marti, "Generalizability of gene expression programming-based approaches for estimating daily reference evapotranspiration in coastal stations of Iran," *J. Hydrol.*, vol. 508, pp. 1–11, Jan. 2014.
[38] M. Sumner, E. Frank, and M. Hall, "Speeding up logistic model tree induction," in *Proc. Eur. Conf. Princ. Data Mining Knowl. Discovery*. Cham, Switzerland: Springer, 2005, 675–683.
[39] J. Vamathevan, D. Clark, P. Czodrowski, I. Dunham, E. Ferran, G. Lee, B. Li, A. Madabhushi, P. Shah, M. Spitzer, and S. Zhao, "Applications of machine learning in drug discovery and development," *Nature Rev. Drug Discovery*, vol. 18, no. 6, pp. 463–477, 2019.
[40] K. Wang, L. Tang, J. Han, and J. Liu, "Top down FP-growth for association rule mining," in *Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining*. Cham, Switzerland: Springer, 2002, pp. 334–340.
[41] G. I. Webb, "Decision tree grafting from the all-tests-but-one partition," in *Proc. IJCAI*, vol. 2, 1999, pp. 702–707.
[42] T. Wohlfarth, S. Clémençon, F. Roueff, and X. Casellato, "A data-mining approach to travel price forecasting," in *Proc. 10th Int. Conf. Mach. Learn. Appl. Workshops*, Dec. 2011, pp. 84–89.
[43] R. Xiao, M.-J. Li, and H.-J. Zhang, "Robust multipose face detection in images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 1, pp. 31–41, Jan. 2004.
[44] P. Yasodha and N. R. Ananthanarayanan, "Comparative study of diabetic patient data's using classification algorithm in WEKA tool," *Int. J. Comput. Appl. Technol. Res.*, vol. 3, no. 9, pp. 554–558, Sep. 2014.
[45] X. Yin and J. Han, "CPAR: Classification based on predictive association rules," in *Proc. SIAM Int. Conf. Data Mining*, May 2003, pp. 331–335.
[46] C.-S. Yu, Y.-J. Lin, C.-H. Lin, S.-T. Wang, S.-Y. Lin, S. H. Lin, J. L. Wu, and S.-S. Chang, "Predicting metabolic syndrome with machine learning models using a decision tree algorithm: Retrospective cohort study," *JMIR Med. Informat.*, vol. 8, no. 3, Mar. 2020, Art. no. e17110.
[47] J. Zhong, L. Feng, and Y.-S. Ong, "Gene expression programming: A survey," *IEEE Comput. Intell. Mag.*, vol. 12, no. 3, pp. 54–72, Aug. 2017.

MARCO ANTONIO BARRÓN received the master's degree in corporate networks from the Polytechnic University of Valencia, Spain, in 2004. He is currently pursuing the Ph.D. degree with the KDIS Laboratory, Research Group, University of Cordoba. His research interests include interpretable models using evolutionary computation, base-rules, and subgroup discovery methods which can be applied to real case scenarios in the airline industry.

JOSÉ MARÍA LUNA received the Ph.D. degree in computer science from the University of Granada, Spain, in 2014. He is currently an Associate Professor with the Department of Computer Science and Numerical Analysis, University of Cordoba, Spain. He has also been engaged in four national and regional research projects. He has contributed to three international projects. He is the author of the two books related to pattern mining, published by Springer. He has published more than 35 papers in top ranked journals and international scientific conferences, and he is the author of two book chapters. His research interests include evolutionary computation and pattern mining.

SEBASTIÁN VENTURA (Senior Member, IEEE) received the B.Sc. and Ph.D. degrees in sciences from the University of Córdoba, Spain, in 1989 and 1996, respectively. He is currently a Full Professor with the Department of Computer Science and Numerical Analysis, University of Córdoba, where he heads the Knowledge Discovery and Intelligent Systems Research Laboratory. He has also been engaged in 15 research projects (being the coordinator of seven of them) supported by the Spanish and Andalusian governments and the European Union. He has published three books and about 300 papers in journals and scientific conferences, and he has edited three books and several special issues in international journals. His main research interests include data science, computational intelligence, and their applications. He is a Senior Member of the IEEE Computer Society, the IEEE Computational Intelligence Society, and the IEEE Systems, Man and Cybernetics Society, and also the Association of Computing Machinery (ACM).