Applied Intelligence (2022) 52:16900–16915
https://doi.org/10.1007/s10489-022-03328-3

Driving maneuver classification from time series data: a rule based machine learning approach

Md. Mokammel Haque[1] · Supriya Sarker[1] · M. Ali Akber Dewan[2]

Accepted: 31 January 2022 / Published online: 28 March 2022

Abstract

Drivers' improper driving behavior plays a vital role in road accidents. Different approaches have been proposed to classify and evaluate driving performance to ensure road safety. However, most of the techniques are based on neural networks which work like a black box and make the logical reasoning behind the classification decision unclear. In this paper, we propose a rule-based machine learning technique using a sequential covering algorithm to classify the driving maneuvers from time-series data. In the sequential covering algorithm, the impact of each rule is measured as the metrics of coverage and accuracy, where the coverage and accuracy indicate the amount of covered and correctly identified instances in a maneuver class, respectively. The final ruleset for each maneuver class is formed with only the significant rules. In this way, the rules are learned in an unsupervised manner and only the best performance of the rules are included in the ruleset. The set of rules is also optimized by pruning based on the performance of the test data. Application of the proposed system is beneficial compared to the traditional machine learning and deep learning approaches which typically require a larger dataset and higher computational time and complexity.

Keywords Rule-based machine learning · Driving maneuver · Driving behavior classification · Sequential covering · Rule learning · Explainable AI · Interpretable machine learning

1 Introduction

With the immense technological advancements in the field of automobile, companies are developing vehicles with increased efficiency and performance. However, traffic safety could not be ensured entirely yet. Drivers' aggressive driving behavior is one of the main reasons for road accidents [1, 2]. A predictive driving assistant system serves to notify about aggressive driving behavior. So, the classification of aggressive and non-aggressive driving maneuvers is important. A good number of research studies have been conducted to classify drivers' behavior in the last several years. Among those, fuzzy inference, machine learning (ML), and deep learning (DL) approach [3]-[7] are the most commonly proposed methods by other researchers.

Typically, the fuzzy inference systems [8–10] calculate scores and notify about the dangerous events. However, these systems have some limitations. A particular dangerous event is very circumstantial and can take place for many reasons. A driver can perform an aggressive maneuver from a positive intention too; for instance, saving a passer-by from hitting. If a system takes an instant decision about the drivers' driving behavior and notifies, it should not be considered drivers' unintentional driving behavior just based on this single observation. For this reason, data science is concerned about the analysis of drivers' driving maneuver time-series data to understand their regular driving practice before reaching out to a decision. Thus, it is important to gather a good amount of driving data over time and analyze to know about someone's driving behavior.

✉ Supriya Sarker
sarkersupriya7@gmail.com

Md. Mokammel Haque
mokammel@cuet.ac.bd

M. Ali Akber Dewan
adewan@athabascau.ca

1 Department of Computer Science and Engineering, Chittagong University of Engineering and Technology, Chattogram, Bangladesh

2 School of Computing and Information Systems, Faculty of Science and Technology, Athabasca University, Athabasca, AB, Canada

Besides, with the availability of rich and larger datasets, several approaches have become unclear due to the recent highly influential data-driven deep neural techniques [11]. The challenges lie in the acceptance of the trusted output of the predictive model to the end-user. It emphasizes the significance of research to ensure the clarity of the output of the predictive deep learning models. Besides, the performances of the deep learning models are highly dependent on the tuning of the hyperparameters and sufficient training, however, optimizing the hyperparameters is a crucial but challenging and time-consuming task. It also requires high performance computing system to implement.

Although Artificial Neural Networks (ANN) [12] or Random Forests (RF) [13] are popular due to their high predictive performance, this scenario is changing as the attention towards explainability, interpretability, and understandability of machine learning models are increasing [14, 15]. Rule-Based Machine Learning (RBML) models promote understanding along with logically exploring the reasons behind decision making and predictions in comparison to other black box type techniques [16, 17]. RBML requires less dataset and less time to process the dataset in comparison to the ML and DL. RBML aims to express the knowledge base in the form of the IF-THEN rule; hence, it can be trustworthily used for data analysis tasks in machine learning and data mining [16, 18]. Since typically people express their knowledge in the form of rules, domain experts extract significant input during the learning process of historical data through rule learning [19]. In prior studies, various rule learning approaches have been explained [20, 21]. However, rule learning approaches may be vary depending on the problem domain.

Therefore, the research question addressed in this paper is - "How to develop an interpretable RBML model to classify driving maneuvers from sensor fusion time series data?". To address this research question, we develop a *rule-based machine learning framework using a sequential covering algorithm* for the classification of driving maneuvers from the sensor fusion time series dataset.

The three main contributions made in this paper can be summarized as the following:

- We develop a rule-based machine learning model for the classification of driving maneuvers from time-series data.
- We utilize an unsupervised sequential covering approach for rule learning.
- We ensure the compactness of the learned ruleset by pruning redundant and insignificant rules and evaluating the performance of the learned rules.

The rest of the paper is organized as follows. The state-of-the-art methods for driving maneuver classifications are discussed in Section 2. The theoretical background of the framework is discussed in Section 3. The methodology of the proposed RBML techniques is discussed in Section 4. Section 5 presents the and preparation of the dataset, experimental results and analyses. Section 6 draws conclusions with some future recommendations.

2 Literature review

Driving maneuver classification involves classifying various types of maneuvers a driver typically performs while driving, such as turning, passing, parking, and yielding to others on the road. In the recent years, automatic driving maneuver classification has received increasing attention for detecting aggressive driving patterns on the road [22]. A good number of research studies on driving maneuver/ behavior classification have been published in the last several years, which we discussed below.

Castignani et al. [8] developed an application that used Fuzzy Inference in the form of IF-THEN rules to analyze driving behavior and classify the behaviors into normal, moderate, and aggressive that corresponds to a score between 0 and 100. The authors used 18 IF-THEN rules for this classification. The limitations of this work are that the rules set have not been extensively evaluated using test data and some other useful rules might have been missed. Saiprasert et al. [9] proposed a driver's profiling algorithm, which considered road condition and Safety Index (SI). The four ranges of SI categorized driving behaviors into four safety levels which include very safe, safe, aggressive, and very aggressive. However, setting up the threshold values for the SI is challenging in this research study.

Van Ly et al. [10] proposed a system to provide timely feedback to drivers about aggressive driving maneuvers. To develop individual driver's profiles the authors collected Controller Area Network (CAN) bus data from a vehicle equipped with sensors and vision systems such as front side radar and CAN signal of the vehicle. Finally, SVM and K-mean clustering have been used for the driving maneuver classification. Due to high availability the authors utilize smartphones; however, the author utilized inertial sensor data to capture lateral acceleration (left and right), longitudinal acceleration (forward and backward) acceleration, and yaw angular velocity. A ground truth label has been associated with the data captured to train and test the classifier model. The authors also considered 951 events data and labeled the data of the events based on some threshold values. For example, starting of a braking event was defined by the brake light indicator that turns from off (0) to on (1). Similarly, the end of a braking event was defined by the brake light indicator is 0 or when the vehicle speed is 0. However, the manual process of the association of data labeling can be better done with a rule-based system more efficiently.

Cervantes-Villanueva et al. [3] applied random forest (RF), support vector machines (SVM), and fuzzy rule-based classifier (FRC) to develop a speed-based breakout detection agent for the classification of driving maneuvers from accelerometer data. RF is a combination of decision trees [13] that randomly selects a small group of input features at each tree's node. The authors have developed an android application to read, store and label accelerometer sensor measurements. While a person drove the subject vehicle, another person used the application to label the accelerometer sensor data. Time statistical features were collected from a labeled dataset with 110200 timestamped. However, the labeling process of the accelerometer data by the smartphone application and the parameter settings for the fuzzy rule-based classifiers are challenging, which have not been explained well in this research study.

A multi-label supervised learning classification problem has been modeled by Ferreira et al. [5] to identify the best combination from 35 combinations of inertial sensors, machine learning models, different parameters, and varying sliding window frames to detect individual driving event types from 69 events. Authors evaluated the quantitative performance of four machine learning models- Multi-Layer Perceptron (MLP), SVM, RF and Bayesian Networks in the classification of driving maneuvers from smartphone sensors, and found that the best result has been achieved by the RF.

Carvalho et al. [4] investigated the performance of three variants of Recurrent Neural Network (RNN) which are RNN, Long Short Term Memory (LSTM), and Gated Recurrent Unit (GRU) in order to classify seven driving maneuvers using data collected by smartphone accelerometer. The authors examined the performance of the classifier with a varying number of units in the hidden layers; however, the other hyperparameters tuning such as batch size, optimizer, learning rate, and dropout rate were not explained well. Authors indicate that the SimpleRNN and LSTM are more susceptible to the number of hidden units.

Alvarez-Coello et al. [6] considered the problem of classifying driving style into aggressive or non-aggressive as a binary classification problem and applied RF in a time window. This method also applied RNN for classifying seven driving maneuvers. In this study, statistical features (*e.g.,* mean, median, standard deviation, and trend) are extracted with varying window size (*e.g.,* $2, 4, \ldots, 10$) to use with the RF classifier while the RNN was used with varying hyperparameters settings, such as the number of hidden layers, recurrent units, optimizer, and dropout. However, the authors concluded that the collection of sample data for aggressive driving is time-consuming and heavily dependent on some other external factors (*e.g.,* dedicated track, qualified drivers, and correct labeling).

Wang et al. [23] proposed a semi-supervised support vector machine (S3VM) approach to classify aggressive and non-aggressive driving styles. In order to label a few data instances, the authors used rule-based k-means clustering. For optimization, a differentiable surrogate of a loss function was proposed, quasi-Newton algorithm was used to assign a label. The authors manually set certain thresholds for some of the features, such as vehicle speed and the throttle opening, in a simulated dataset.

Mammeri et al. [24] proposed a coarse thresholding technique to label some CAN bus data, removed labeling error and trained a simple Convolutional Neural Network (CNN) to classify ten driving maneuvers which are stop, move, acceleration, deceleration, constant speed, left/right turning, left/right curving, and constant direction.

Martinelli et al. [25] proposed a supervised approach based on the nearest neighbor classifier and a set of rules that differentiate between an actual owner and a fake owner of a vehicle data. In this study, six features were extracted from on board diagnostic (OBD) [26] data which include the amount of CO_2, percentage of the engine load, angular rotation of engine, fuel flow rate, amount of fuel remaining, turbo boost and vacuum gauge. Analyzing the extracted features set, this method differentiated aggressive and non-aggressive driving behavior to identify the actual owner and fake owner. However, a major limitation of this study is that an actual owner can also have aggressive driving behavior. Moreover, the vehicle might have more than one user and each user might have different driving patterns.

Sarker et al. [7] discussed the significance of domain-specific features and proposed a supervised learning approach using LSTM model. The model was trained with both domain-specific features and statistical features and found that the proposed model outperformed the other supervised deep neural models. This work was extended in [27] where a transfer learning approach had been used that combined an unsupervised LSTM autoencoder with a supervised model similar to [7]. The compressed latent representation learning of the LSTM autoencoder was transferred to train the LSTM model. This approach also developed seven maneuver class functions that contain the maneuvers' time series changing patterns [28]. In another work, Sarker et al. [29] proposed an algorithm for labeling unlabeled driving maneuvers from sensor fusion time series data. Similar to some other rule based models of [25, 45] and [46], the rules were extracted from domain knowledge of driving events. However, in this approach, the decision is made based on a fixed ruleset. The time-series data patterns described in [7] and [27] can be expressed as the IF-THEN rule and can be used in RBML based driving maneuver classification system as done in this research study.

3 Theoretical background

Moving vehicle time series data changing rules and rule-based machine learning techniques are two important concepts of this study. We discussed these two concepts in detail in the below subsections. We specifically elaborated more on some important domain knowledge and its relation with the time series data and various common driving maneuvers. We interpret all the important symbols and their definitions used throughout the paper in Table 1.

3.1 Moving vehicle time series data changing rules

ML and DL models can be used to classify the time series data; however, the intermediate processing steps of these models are unclear. On the other hand, the RBML model learns rules from domain expert knowledge of a problem area and expresses rules with logical reasoning. Hence, understanding the underlying issues of a problem domain is important while developing the RBML model. However, classifying time series data in the form of rules is very challenging.

When a vehicle in concern moves, the mounted sensors capture the movement information as a form of time series data. In this work, we consider accelerometer and gyroscope data for the rule-based classification of seven driving maneuvers. It is observed that when a vehicle moves, the time series data maintains a specific pattern for a particular driving maneuver [29]. Figure 1 shows the non-aggressive range of all the maneuvers. The changing pattern of the time series data during longitudinal, lateral and angular movement of a vehicle is presented by a_x, a_y and w_z, respectively. The underlying rules are demonstrated along with the maneuvers in Fig. 1. During non-aggressive acceleration of the vehicle, the a_x lies between 0 and 2 and while in the non-aggressive braking, it lies between 0 and -2. On the other hand, the left turn (LT), right turn (RT), left lane change (LLC) and right lane change (RLC) maneuvers are related to the lateral and angular changes of a motion. Therefore, while performing LT and RT, a_y increases more than 1.5 and decreases more than -1.5, respectively. At the same time, w_z increases more than 0.4 and decreases more than -0.4 for LT and RT, respectively. During LLC and RLC, the amount of angular velocity is less than that of LT and RT and lies between 0.2 and 0.4 for LLC and -0.2 and -0.4 for RLC. These patterns can be expressed using some IF-THEN rules. Besides, statistical features of time series data such as slope, energy, variance can improve the accuracy of the rules. In this paper, the proposed approach learns the rules and measure the performance of the rules

Table 1 Interpretation of symbols

Symbol	Interpretation
$\vec{a}$	Acceleration
$\vec{w}$	Angular velocity
a_x, a_y, a_z	Accelerometer data in x, y and z axis
w_x, w_y, w_z	Gyroscope data in x, y and z axis
$a_{x_{slp}}, a_{y_{slp}}$	Slope of accelerometer in x and y axis
$w_{z_{slp}}$	Slope of gyroscope in z axis
$a_{x_{eng}}, a_{y_{eng}}$	Energy of x and y axis of accelerometer
$w_{z_{eng}}$	Energy of z axis of gyroscope
TS_{a_x}	Set of time series data from the a_x
TS_{a_y}	Set of time series data from the a_y
TS_{w_z}	Set of time series data from the w_z
C_{non}	Non-aggressive driving maneuver class
C_{acl}	Aggressive Acceleration driving maneuver class
C_{brk}	Aggressive Brake driving maneuver class
C_{lt}	Aggressive LT driving maneuver class
C_{rt}	Aggressive RT driving maneuver class
C_{llc}	Aggressive LLC driving maneuver class
C_{rlc}	Aggressive RLC driving maneuver class
R	Ruleset
C_{prior}	Class priority
R_{prior}	Rule priority
A	Set of attributes of time series

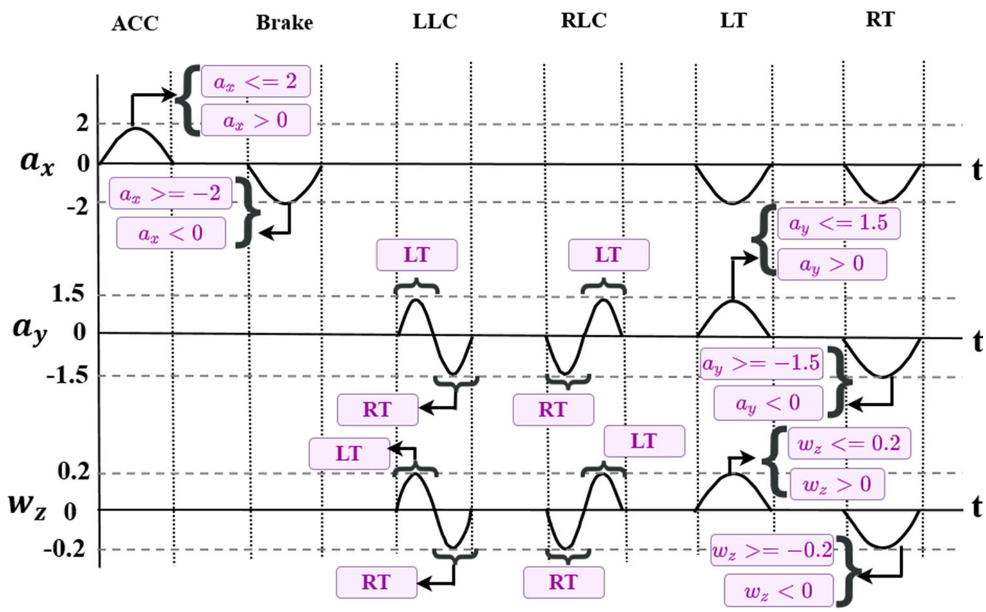

Fig. 1 Illustration of time series with simple rules

over the dataset. The pattern of LLC and RLC consists of both LT-RT (RT after LT) and RT-LT (LT after RT) patterns. So, the rules for LLC and RLC are a bit critical.

3.2 Rule based machine learning techniques

The automatic induction of rules for classification is drawing attention in machine learning and data mining decision support and decision-making applications [30]. Rule-based machine learning approaches learn and make interpretable classification decisions based on some simple but effective if-else rules. The characteristic of a rule-based machine learning approach is the identification of a set of rules that contain the knowledge base of a specific domain and utilize the knowledge to predict or classify a new data set of rules that have not been seen before by the deployed system. There are various rule-based classification techniques such as Zero-R [31], One-R [32], decision trees [33], Ripper Down Rule Learner, Repeated Incremental Pruning to Produce Error Reduction (RIPPER) [31]. A commonly used rule-based technique is the decision tree where each branch of the tree corresponds to a rule and based on the highest weighted rules, the dataset is repeatedly split into the smallest portion.

Since, we are dealing with the classification of driving maneuvers by analyzing time-series dataset, in this paper, we focus on rule learning approaches from time-series data. Time series data contains a collection of precise and distinct values gained from subsequent calculations over time. Time series rule mining intends to discover significant patterns of sequential data holding quantitative temporal values [20]. Earlier enormous studies of time series rule discovery have been conducted to various problems such as sales performance prediction in business [34], epidemic detection in healthcare [35], tourist demand forecasting in tourism [36], cattle movement analysis in agriculture [37], climate prediction [38], and so on.

A symbolic aggregate approximation (SAX) and matrix profile method are proposed by Schwarz [39] to classify the time-series pattern of naturalistic driving data. Zhang et al. [40], proposed an unsupervised approach to learn salient time series subsequences by regularized least-squares technique and spectral analysis to learn shapelets. A shapelet similarity minimization has been applied to disregard learning similar Shapelet and a descent algorithm has been applied to get the label concurrently. Similarly, in [41] a novel mathematical formulation has been proposed for shapelet learning through an objective function and a learning algorithm to learn optimal shapelets. Wang et al. [42] proposed a shapelet mining technique where a shapelet layer, a distance layer, and a softmin layer have been used to learn parameters to determine incubation period, trends of transmission, and predict the upcoming new COVID-19 cases in geographically adjacent locations. In [43] a combination of extreme learning machine and decision rule has been proposed for ECG heart beat PQRST time-series unsupervised clustering. The author calculated Poffset and Tonset by applying Chan Slope method, Ponset and Toffset by applying a combination of Chan method and ecgpuwave software. In [44] a method has been proposed to discover time series classification rules that are similar to fuzzy rules and the situation where uncertainty associated with motif has been also considered.

Mehtab et al. [45] proposed a rule-based machine learning framework incorporating Adaboost with Decision tree [33] for classifying benign and malicious behavior of the android applications. Jamian et al. [46] proposed a classification method using Random Forest (RF) and regression tree (CART) to categorize systemic sclerosis patients from electronic health record databases. Vojíř et al. [19] focused on rule-based machine learning models that are interpretable and editable. They designed a web-based rule editing software to generate an association rule list from an uploaded dataset that is editable and users can evaluate the predictive performance of the ruleset on a test dataset through a scoring procedure to ensure the explainability of the rule model. This work was an extension of EasyMiner [47] that is a web-based machine learning system for anomaly detection in Foreign Portfolio Investment (FPI) and its variation. The system worked based on association rule learning [48] and Classification Based on Association Rules Algorithm (CBA) [49].

The prior research of time series pattern discovery or classification techniques have utilized the temporal features of time series for segmentation and to discover significant patterns; however, have lack of interpretability and explainability. On the other hand, in rule mining research, rules were extracted not learned based on the underlying domain. Hence, there is a possibility of redundant and missing promising rules in the ruleset. Moreover, a fixed ruleset may not perform as good on the test dataset as in the training dataset. Therefore, optimization of the ruleset needs to be focused on.

In this paper, we apply a sequential covering algorithm which is an unsupervised rule-based machine learning technique. It learns one attribute or rule at a time and uncovers data points that are matched with the learned one rule and remove those points from the sample dataset. This process continues until all the data is classified. Each new rule serves as a cover hypothesis [50]. Rather than splitting the dataset like decision trees, a sequential covering algorithm find the most promising rule one at a time, apply to the dataset and remove the positive data points. This procedure continues for the next promising rules. At a point when the discovered rule is not much significant, the process stops. Therefore, in this paper, we focused on unsupervised non-redundant rule learning and optimization for the classification of driving maneuvers from time-series data.

4 Methodology

4.1 Definitions and notations

This section defines the major symbols and notions that are used to describe the various concepts throughout the paper concerning the rule-based machine learning method to discover rules of individual driving maneuvers from sensor fusion time series data.

At a time instance, t both the accelerometer and the gyroscope sensor produces three time series data in x, y and z axis. The set of time series data from the x and y-axis of accelerometer sensor is $TS_{a_x} = \{a_{x_1}, a_{x_2}, \ldots, a_{x_n}\}$ and $TS_{a_y} = \{a_{y_1}, a_{y_2}, \ldots, a_{y_n}\}$, respectively and the set of Time Series data from the z-axis gyroscope sensor is defined as $TS_{w_z} = \{w_{z_1}, w_{z_2}, \ldots, w_{z_n}\}$. In these 3 time series data n represents the total number of instances data captured by the sensors.

Time series contains certain attributes and depending on attributes, rules are learned. A set of attributes, $A = A_1, A_2, \ldots, A_j$ where j is the number of attributes or aspects of the time series data. Each time series data point represents an individual maneuver class. The maneuver classes are defined as, $C = \{C_{non}, C_{acl}, C_{brk}, C_{lt}, C_{rt} C_{llc}, C_{rlc}\}$ where C_{non}, C_{acl}, C_{brk}, C_{lt}, C_{rt}, C_{llc} and C_{rlc} are non-aggressive, aggressive acceleration, aggressive braking, aggressive LT, aggressive RT, aggressive LLC and aggressive RLC, respectively.

A rule, R covers a set of data points at time, t if the characteristics of the data points satisfy the condition of the rule, R. Let, $R = R_0, R_0 \ldots, R_n$ is applied to the data points of time series TS_{a_x}, TS_{a_y} and TS_{w_z}. A rule that covers an instance of a time series data for a particular maneuver class is denoted as $[R_i \Rightarrow C_i]$ where R is antecedent or the body of the rule (*i.e.,* condition) and C is consequent (*i.e.,* driving maneuver class).

Class priority, C_{prior} is presented as $C_{prior} = \{1, 2, \ldots c_p\}$ where 1 and c_p represents the most and least frequent class priority value.

Since we follow a rule based ordering, depending on the quality of the rules an priority is associated to each learned rule for maneuver classes and denoted by $R_{prior} = 1, 2, \ldots, r_p$ where 1 and and r_p represents the most and least promising rule that can classify highest number of time series data points.

The knowledge base is an essential element of a knowledge-based system that holds the underlying facts of a problem domain [16]. In a rule-based system, the rules can be learned from the knowledge of the domain. Therefore, as our problem domain deals with time series data, therefore, during preprocessing temporal features are extracted. Let the set of temporal feature set, $f = \{mean_{(a_x,a_y,w_z)}, \mathrm{var}_{(a_x,a_y,w_z)}, \mathrm{std}_{(a_x,a_y,w_z)}, \mathrm{slp}_{(a_x,a_y,w_z)}, \mathrm{eng}_{(a_x,a_y,w_z)}, \max_{(a_x,a_y,w_z)}, \min_{(a_x,a_y,w_z)}\}$

Hence, the proposed Rule Base Driving Maneuver System classifies a time series data points $t \in (TS_{a_x}, TS_{a_y}, \ldots TS_{w_z})$ into a class $C_m \in C$ where m is a particular driving maneuver.

4.2 Proposed rule based driving maneuver classifier

In this paper, we develop a rule-based machine learning technique using a sequential covering algorithm. In sequential covering algorithm, the learning approach can be of two types: general to strategic and strategic to general. Typically, in the general to strategic approach, the rules are learned from the most frequent class to the least frequent class. In the strategic to general approach, the rules are learned from the least frequent class to the most frequent class. We start with the most general hypothesis and then go through the specialized steps. We conjunct attributes to a rule to improve the performance of rules on the dataset. For a particular class, multiple rules are learned by the algorithm and a rule can be satisfied by the instances of the other classes. Hence, there is a possibility of conflict among classes. Considering this challenge, the rules can be ordered in two ways: rule-based ordering and class-based ordering. We use rule-based ordering for simplification of the ruleset and class-based ordering for the avoidance of conflicts among the classes.

In order to learn rules, the proposed algorithm consists of four major modules which are class rule growing, rule evaluation, instance elimination, and rule pruning. An abstract view of the proposed rule-based driving maneuvers classifier model is shown in Fig. 2. For each class, these four modules continue to learn the rules until the stopping criteria are met for a considering class.

The rule learning starts from the most frequent class. For our dataset, the non-aggressive class is the most frequent class and typically drivers do non-aggressive maneuvers more frequently than the aggressive maneuvers. We calculated the number of data instances in each class and set priority for the classes from the most frequent to less frequent. This rule learning process is called general-to-strategic rule learning in RBML.

Class rule growing, rule evaluation, instance elimination, and rule pruning need to be done for each class. However, we discussed each of the steps only for the non-aggressive maneuver class to avoid redundancy. Only the learned rule, measured accuracy, coverage and gain will vary for the other classes.

4.2.1 Rule growing

We use an unsupervised process for rule growing in rule learning for each class. The pseudo code for rule learning is presented in Algorithm 1. The input of the algorithm is a set of time series data and a set of attributes, A. Initially, the ruleset is initialized as an empty set, $R = \{\varnothing\}$ and over time the new rules, R_j are learned and added to this set from the attribute set. The new rules are added to the rule depending on its performance on the dataset for the considering class. For instance, a few simple attributes are observed in the data change pattern of time series driving maneuvers and demonstrated in Fig. 1. For non-aggressive class, as stated earlier in this paper, the value of a_x data increases upto 2 (*i.e.*, $a_x \leq 2$) while non-aggressive acceleration and decreases upto -2 (*i.e.*, $a_x \geq -2$) during non-aggressive braking, so these phenomena can be considered as two individual simple rule for non-aggressive maneuver.

In Fig. 3, a general-to-strategic rule growing process is shown where rule growing starts from the empty ruleset that is indicated as the root node in the tree. Each simple single rule is being applied to the dataset during the rule learning process for non-aggressive class and either added to the ruleset or discarded. Sometimes, multiple rules are conjunct or disjunct to improve the performance.

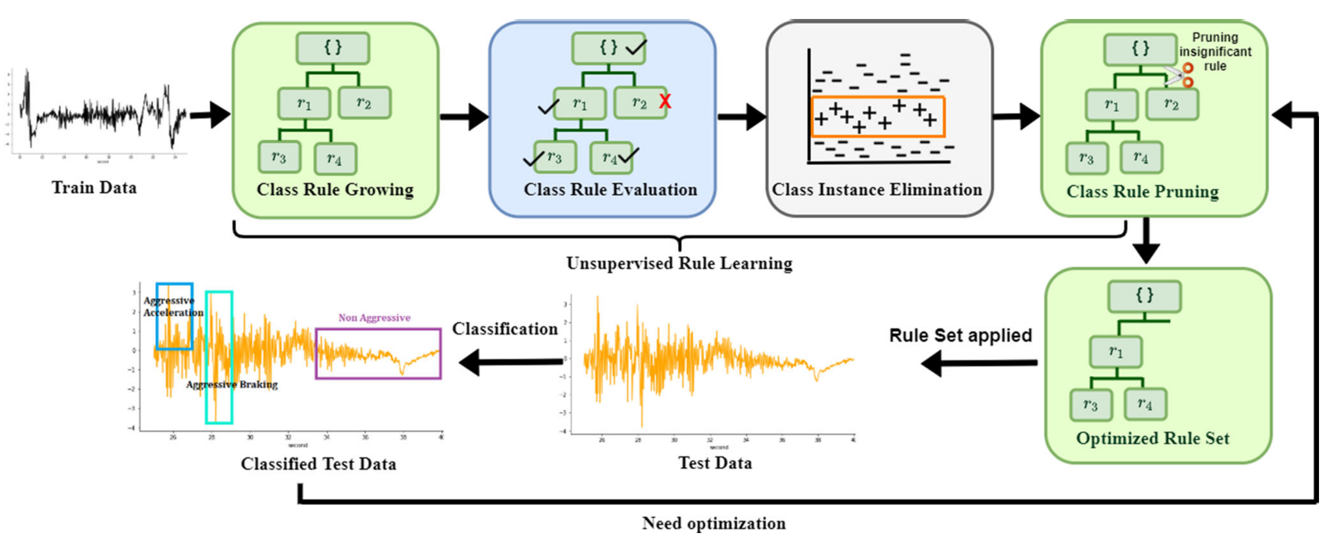

Fig. 2 Rule-based driving maneuvers classifier model

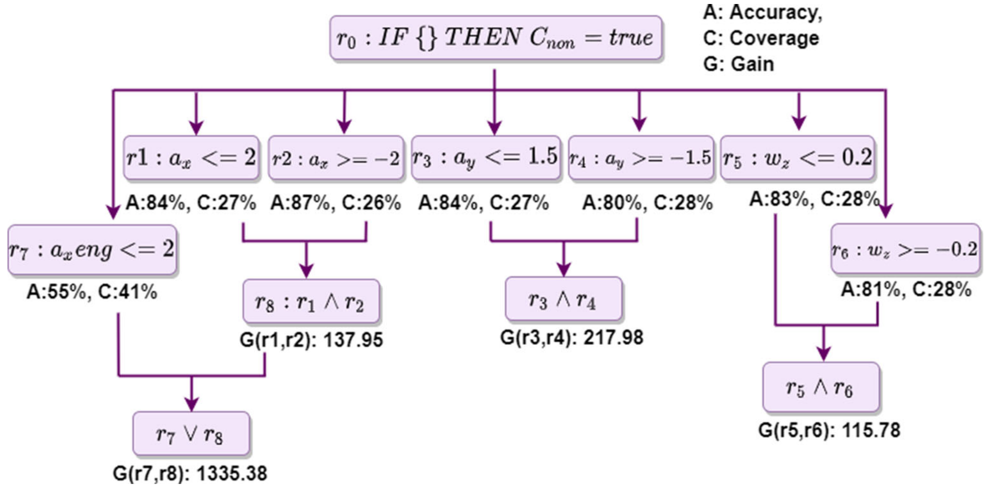

Fig. 3 Rule growing for non-aggressive class

4.2.2 Rule evaluation

In the rule evaluation process, the performance of each learned rule on the training dataset is measured using a function evaluate() before including it in the ruleset, R. The function calculates the coverage and accuracy of a rule on the training dataset. The coverage of a rule is the fraction of records that are covered by the antecedent of the rule. The accuracy of a rule is the fraction of records that can be satisfied by both the antecedent and consequent. The coverage and accuracy are defined by (1) and (2), respectively.

$$Coverage(r) = \frac{n_{cover}}{D} \tag{1}$$

$$Accuracy(r) = \frac{n_{correct}}{n_{cover}} \tag{2}$$

where, n_{cover} and $n_{correct}$ are the number of records that are triggered (*i.e.*, satisfy only the antecedent) and the number of correct records that satisfy both the antecedent and the consequent by the rule, r. D is the number of records in the dataset. Coverage and Accuracy can measure the performance of a single rule on dataset. Gain is another significant performance measuring metric that is used to measure the performance of multiple rules which are combined with conjunction or disjunction by propositional logic. Gain of two or more rules on a dataset can be measured by FOIL(First Order Inductive Learner)'s information gain measure. FOIL's information gain is defined by (3).

$$Gain(r_0, r_1) = t[log(\frac{pos_1}{pos_1 + neg_1}) - log(\frac{pos_0}{pos_0 + neg_0})] \tag{3}$$

In (3), t is the number of positive instances triggered by both r_0 and r_1, pos_0 and neg_0 is the number of the positive and negative instances triggered by r_0, respectively. pos_1 and neg_1 is the number of the positive and negative instances triggered by r_1, respectively.

Based on the performance of the rules, the highest promising rule is used to positively classify the class. In the next iteration, another adjunct simple rule is added with logical propositional relation and the performance of the newly formed rule is measured. The gain of two simple rules is calculated by the algorithm. If the gain is significant (*i.e.*, above the threshold) means that the combination of both rules improves the performance. Otherwise, the added rule is discarded. A list of performance evaluations of a few rules is shown in Table 2. In Fig. 3, rules, r_1, r_2 and r_9 have the (accuracy, coverage) on the training dataset is (84%, 27%), (87%, 26%) and (55%, 41%), respectively. After combining the rules r_1 and r_2, a new rule, r_8 is created. The gain of r_1 and r_2 is calculated as 137.95 by (3). In order to increase the gain, another single rule r_7 is disjunct with r_8 by logical OR operation and the calculated gain is 1335.38 which is greater than other gain calculated from all other rules. Therefore, the non-aggressive class is classified by r_1, r_2 and r_7; hence, the combined rule can be expressed by $[(r_1 \wedge r_2) \vee r_7] \Rightarrow C_{non}$ where C_{non} is the non-aggressive class.

In this rule, $(r_1 \wedge r_2)$ implies that the value of any data instances of x axis of accelerometer *i.e.,* a_x is greater than / equal to -2 but a_x less than 2, jointly means data instance of a_x lies between -2 and $+2$. Similarly, r_7 can be expressed as $(eng_{a_x} <= 2)$ means that the values of energy of data instances of a_x those are less than / equal to 2. Collectively, $(r_1 \wedge r_2) \vee r_7$ is satisfied by the data instance of a_x when a non-aggressive maneuver is performed.

4.2.3 Instance elimination

Instance elimination is the process of removing the positive instance covered by a rule on a dataset. A data point or instance that adapts to both the antecedent and consequent

Table 2 List of performance evaluation of few rules

Rule	Attribute	Coverage(%)	Accuracy(%)	Gain(%)
R1	$a_x >= -2$	27.40	83.97	—
R2	$a_x <= 2$	25.99	87.05	—
R3	$a_y <= 1.5$	27.86	83.47	—
R4	$a_y >= -1.5$	28.37	80.91	—
R5	$w_z <= 0.2$	27.66	82.99	—
R6	$w_z >= -0.2$	28.36	80.95	—
R7	$eng_{a_x} = 2$	28.47	55.36	—
R8	$R_1 \wedge R_2$	26.88	84.02	137.95
R9	$R_3 \wedge R_4$	40.59	55.39	217.98
R10	$R_5 \wedge R_6$	35.90	63.94	115.78
R11	$R_1 \wedge R_2 \vee R_7$	38.24	66.78	1335.38

of a rule is called a positive data point for that rule. Otherwise, it is called a negative instance for that class covered by the rule. When considering the highest priority class, all other classes are considered as a negative class by the rule. The positive instances are being removed from the dataset and the rule learning process iterates until all the negative points of a class are covered. Figure 4. illustrated the positive and negative instances by rule $[(r_1 \wedge r_2) \vee r_7] \Rightarrow C_{non}$ *i.e.,* for the non-aggressive maneuver. The positive instances are removed from the dataset and negative instances will participate in the next rule learning process. However, in Fig. 4, few data points are falsely classified as non-aggressive by the rule. To overcome the problem of conflicts with classes, we utilize class priority value.

4.2.4 Stopping criteria and rule pruning

The next step we applied is the rule pruning. Rule pruning is the process to discard any rule that does not improve the performance of the classifier. The discard of any rule can be done based on a stopping criterion. We have iteratively experimented with several percentages of accuracy for each rule and monitored the performance of each rule separately on the dataset. In Fig. 6 we plotted the number of rules associated with and the number of data instances positively satisfied by the rules against the accuracy of those rules. It is clear that selecting a lower threshold values, the number of positively predicted instances is high, but it increases the number of redundant rules in the ruleset. On the other hand, selecting accuracy threshold value of 60% to 85% the number of positively predicted data instances is the highest and redundancy of rules is the least. Hence, we chose accuracy>60% as the threshold value of accuracy. Besides, we also noticed whether any simple rule can get the highest gain value when combined with other rules. If a simple rule with below threshold accuracy can bring the highest gain after combining with other rules, then we consider that simple rule as significant. For example, the accuracy of $r7$ is lower than the accuracy threshold, but combining it with $r8$ *i.e.,* $((r_1 \wedge r_2))$ we get the highest gain (*i.e.,* 1335.38). So, finally, we consider $[(r_1 \wedge r_2) \vee r_7] \Rightarrow C_{non}$ and prune the other rules during the optimization process.

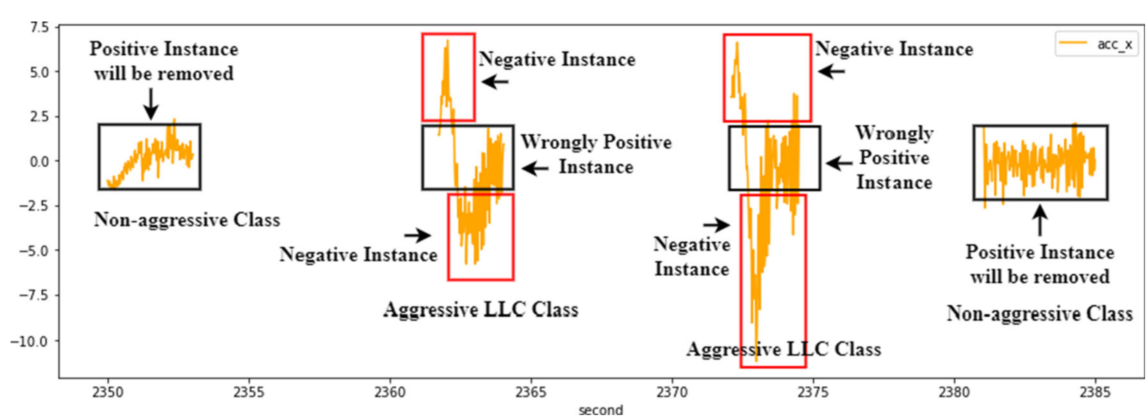

Fig. 4 Instance elimination for Non aggressive class

In Fig. 3. simple rules of a_x, a_y and w_z cover above 25% of the dataset and among the covered dataset, above 80% of data are accurately covered. If any rule covers less than 20% and accuracy less than 60% then the rule can be discarded. For the non-aggressive class, when the most gain is calculated the other rules are pruned from the tree.

4.2.5 Optimize ruleset

The final ruleset is applied to the test data for classifying each class. If any ruleset provides less satisfactory performance to classify a class on test the data, the ruleset is optimized by using rule pruning.

4.3 RBML driving maneuver classifier

The RBML driving maneuver classifier determines the class of driving maneuvers from new time series test data. In order to classify, new unlabeled data is fed to the classifier and the pre-learned ruleset is applied to the test data. The performance of the ruleset on test data is measured. If performance is not satisfactory then the ruleset is being optimized by pruning to improve the accuracy of the classifier.

Algorithm 1 Rule-based driving maneuver classifier.

```
Input: Time Series, attributes, class
Output: Ruleset, Time Series class label
/* set priority for particular class */
1. Sort maneuver classes according to maximum data points
2. Grow a rule using learn_rule(TS, attributes, class)
3. Remove training instances covered by the rule
4. Repeat (2) and (3) until stopping criterion is met
5. Rule Pruning
Function learn_rule (TS, attributes, class):
    2.1. Start from an empty rule: R = {∅} ⇒ C
    2.2. Add conjunct as a rule
    2.3. Measure the rule that maximizes coverage, accuracy and gain
    2.4.
    if measure>threshold then
        Add rule to ruleset, R
    else
        Discard the rule
    end
    2.5 return R
end
```

5 Result and analysis

In Section 5, we provide the description of the dataset, experimental setup, and results and analyse of the work. We measure the performance of the proposed model both quantitatively and qualitatively.

5.1 Dataset description

In this research study, we used two datasets for our experiments: Dataset 1 [51] and Dataset 2 [52].

– Dataset 1 [51] We use a sensor fusion time series driver behavior dataset collected from four road trips by two drivers. The dataset contains 156512 records where only 11077 records have class label information. Time series data is collected from accelerometer, gyroscope and magnetometer sensor. These sensors are previously installed on a smartphone and the smartphone was placed steadily on the windshield of a moving vehicle. However, we only utilize accelerometer, gyroscope and ground truth data. While moving, the driving events were recorded by a camera, the ground truth data was included with the time-series data later on. We use the labeled data because we need to measure the performance of the classifier on the dataset. The data contains seven driving maneuvers which are aggressive acceleration, aggressive braking, aggressive LT, aggressive RT, aggressive LLC, aggressive RLC and non-aggressive.
– Dataset 2 [52] Driving behavior dataset [52] was collected by three male drivers for two weeks and their age was 27, 28 and 37 years. The dataset contains 1114 records and all are labeled. Time series data is collected from Raspberry Pi Model B with MPU6050 (3-axis accelerometer and 3-axis gyroscope) sensor. During data collection, the device was mounted in front of the dashboard. The road was dry asphalt and the weather was sunny. The data contains four driving maneuvers which are aggressive acceleration, aggressive braking, aggressive LT and aggressive RT. The vehicle speed was 30-40 kmph and 70-80 kmph for aggressive turn and aggressive acceleration, deceleration, respectively.

The instance distributions of maneuver classes of the two datasets (dataset 1 and dataset 2) that we use in our experiments, are shown in Fig. 5 The classes are sorted in ascending order of the prevalence and assigned a value from class priority set, C_{prior} for each class. In dataset 1, non-aggressive maneuver class, C_{non} is the most frequent class and rules are learned for this class first. The algorithm assigns class priory for the dataset 1 in the following order: non-aggressive, aggressive acceleration, aggressive RT, aggressive LT, aggressive LLC, aggressive RLC, aggressive braking as shown in Fig. 5a. For the dataset 2, the order is as follows: aggressive LT, aggressive RT, aggressive acceleration, aggressive braking as depicted in Fig. 5b.

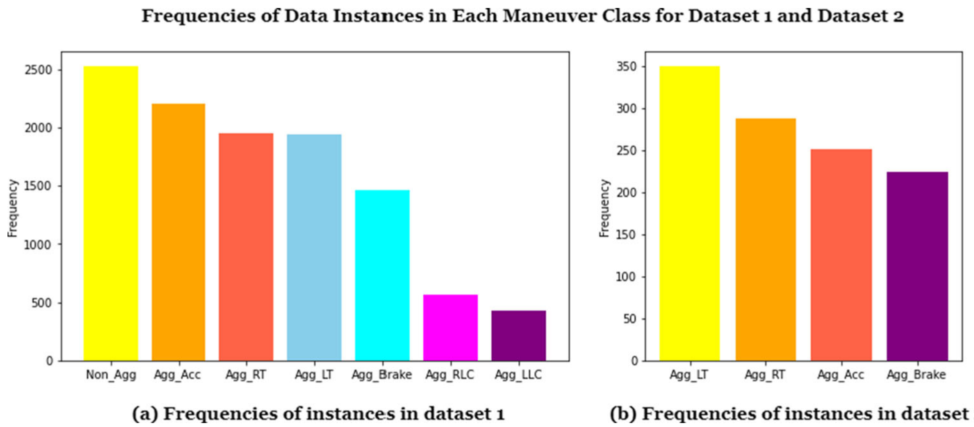

Fig. 5 Frequencies of data instances in each maneuver class for (**a**) dataset 1 and (**b**) dataset 2

5.2 Evaluation metrics

For quantitative performance, we measure the accuracy, precision, recall, and F-1 score of the model. For evaluation of the performance of each rule, we calculate the coverage, accuracy, and gain of the rule. Tables 3 and 4 demonstrate the evaluation scores for each class of the classifier on the dataset [51] and [52], respectively. Accuracy, recall, precision and F1-score can be defined by (4)-(7), respectively.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$F1 - score = \frac{Recall \times Precision}{Recall + Precision} \tag{7}$$

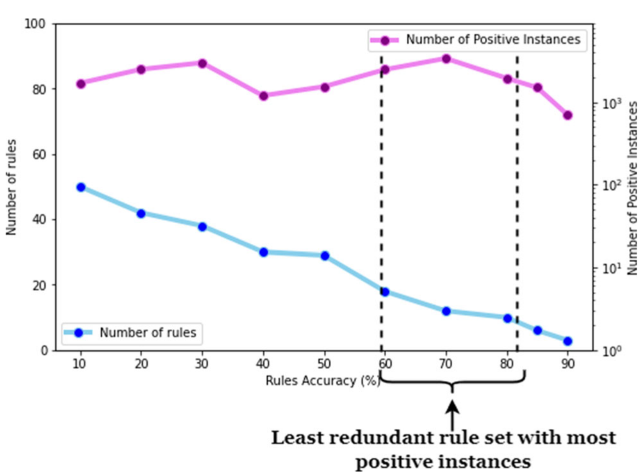

Fig. 6 Number of rules and positively predicted instances for rules accuracy (%)

where TP is the True Positive *i.e.*, the number of instances predicted positive is actually positive, FP is the False Positive *i.e.*, the number of instances predicted positive is actually negative, TN is the True Negative, *i.e.*, the number of instances predicted negative is actually negative and FN is the False Negative *i.e.*, the number of instances predicted negative is actually positive.

5.3 Experimental setup

We use the Python programming language (python == 3.8.3) and Jupyter notebook IDE to implement the proposed work. The model is implemented in a CPU instance with RAM 8GB in a core i−5 processor.

We use 10-fold cross-validation which randomly selects 9 fold as train set and the remaining fold as testing set in each iteration to deal with class imbalance. On the contrary, if we split the data in a certain percentage (*e.g.,* 70% training data and 30% testing data) the class imbalance could negatively impact the overall performance of the classification model.

In our experimental setup, we select threshold accuracy> 60%, threshold coverage> 20% and highest gain value as stopping criteria after iterating with various accuracy, coverage and gain values and prune the rule during the optimization process. We plot the number of positive instances covered by a rule and the number of rules for a corresponding accuracy value in Fig. 6. The graph indicates that when there is more number of rules combined and we select a low accuracy threshold value (*i.e.* 10%) the number of covered positive instances for a class is high. When we gradually increase the threshold value, the number of combined rules decreases as not all the rules have a high accuracy, which means they are less significant to cover data instances positively. We get the optimal value of the threshold from 60% to 80% where the number of positive instances is high and the number of rules is also nominal.

Table 3 Evaluation scores for each maneuver class for dataset [51]

Class	Evaluation Metrics			
	Accuracy	Precision	Recall	F1-score
Non-aggressive	0.8388	0.5321	0.8884	0.6656
Aggressive Acceleration	0.8508	0.6467	0.5197	0.5763
Aggressive Braking	0.9089	0.7473	0.4595	0.5690
Aggressive LT	0.8419	0.5327	0.5561	0.5442
Aggressive RT	0.8411	0.5111	0.4742	0.4919
Aggressive LLC	0.9625	0.5217	0.2791	0.3636
Aggressive RLC	0.9406	0.2963	0.1429	0.1928

5.4 Result and discussion

In section 5.4 we present and discuss the experimental results including the performance of the ruleset. We apply 10-fold cross-validation on our dataset to measure the performance of the proposed approach. The proposed algorithm learns rules from the train set and the performance of the learned rules is evaluated on the validation set. Based on the performance, the ruleset is being optimized.

This section aims to answer the following research questions to justify the proposed classifier on test data.

- Does the proposed classifier able to learn rules from the sensor fusion time-series driving dataset?
- Does the learning approach able to remove redundancy while learning rules?
- How efficient the proposed rule-based machine learning model is compared to any other relative model.

In order to estimate the quantitative efficiency of the proposed rule-based machine learning approach, we compute the efficiency of learned rules, the predictive accuracy of the developed method for unseen test data. The predictive performance indicates rules' quality as well.

Table 2 shows the list of the most significant learned rules and their corresponding accuracy, coverage and gain on the trained data for non-aggressive class. Single rules do not have gain; therefore, from R1 to r7, empty gain values are indicated as dashed line. Rule $R1$ covers 27.40% of the training dataset and the amount the covered data 83.97% belongs to the non-aggressive class. Similarly, the accuracy and coverage of each rule are calculated. Rules that have accuracy $< 60\%$ are not considered significant for our classifier. Rule $R7$ has 55.36% accuracy which is less than the threshold accuracy value, but it increases the gain (1335.38) while combined with r_1 and r_2. After learning all the rules for a class, rules are being sorted according to ascending order of their performance.

Table 3 presents the evaluation results of each maneuver class for the dataset [51]. From this Table 3, we can see that the highest accuracy has been achieved for the class aggressive LLC (*i.e.,* 0.9625). For aggressive braking, a few negative data is covered as positive (*i.e.,* FP is low) so, the precision is better than other maneuvers. On the other hand, aggressive RLC contains the most wrongly positively covered data instance (*i.e.,* FP is high); hence, precision is lowest (*i.e.,* 0.2963) than other maneuver classes. As aggressive RLC contains a combination of aggressive acceleration and sharp braking, the rules create conflicts between two different classes.

Table 4 presents the evaluation scores of the proposed model with the dataset [52]. We can see that the accuracy for almost all of the maneuvers is above 80% and precision, recall and F1-score is also better than the dataset [51]. As the dataset [52] contains less amount of data than the dataset [51], there are less variety in the dataset and less conflicts among the rules for the classes.

Table 5 listed a qualitative comparison of the proposed work with ML, DL and fuzzy rule-based approaches. In

Table 4 Evaluation scores for each maneuver class for dataset [52]

Class	Evaluation Metrics			
	Accuracy	Precision	Recall	F1-score
Aggressive Acceleration	0.8408	0.6140	0.6731	0.6422
Aggressive Braking	0.8952	0.8125	0.5909	0.6842
Aggressive LT	0.8945	0.8133	0.8472	0.8299
Aggressive RT	0.8987	0.7627	0.8182	0.7895

Table 5 Comparison of the proposed work with previous related work

Metric	Techniques	Features	Processing Time	System	Hyperparameters Tuning	Explainability
Proposed Work	RBML	Domain knowledge, Statistical features	High	No specification	Not required	High
[3]	RF, SVM, Fuzzy Rule	Speed, Statis-, tical features,	Moderate	HPC	Required	Moderate
[4]	RNN, LSTM, GRU	Not mentioned	High	HPC	Required	Low
[5]	ANN, SVM, RF, BN	Statistical features, tendency	High	HPC	Required	Low
[6]	RF+ RNN	Statistical features, trend	High	HPC	Required	Low
[7]	LSTM	Domain-knowledge, Statistical Features	High	HPC	Required	Low
[8]	Fuzzy Inference Rules	Jerk, orientation rate, speed and bearing variation, time, weather data	Low	SQLite database, smart-phone	Not required	High
[9]	Pattern matching	Acceleration, GPS data	Moderate	HPC	Not required	Moderate
[10]	SVM, K-means Clustering	Acceleration, yaw angular velocity	High	HPC	Required	Low
[23]	Rule-based k-means clustering, S3VM	Vehicle Speed, Throttle Opening	High	HPC	Required	Moderate
[27]	AE+LSTM	Domain-knowledge, Statistical Features	High	HPC	Required	Low

Table 5, the comparison is done based on the applied techniques, considered features, required time and system for processing, the necessity of hyperparameters tuning and the ability to explain. Typically, deep learning based models *i.e.,* [4–7, 27] need high performance computer (HPC) and the sufficient amount of time to train the model after tuning several optimum hyperparameters and models' structure such as the number of layers, the number of units in each hidden layers, activation function, learning rate, drop out rate, and so on. In machine learning based techniques *i.e.,* [3, 10, 23] are also required to set parameters, for example, C, gamma value, etc. So, processing time and system cost are high; however, explainability is low. On the other hand, fuzzy inference rule-based [8] and pattern matching [9] techniques required comparatively low processing time and cost.

6 Conclusion

This paper proposed an unsupervised learning technique using the sequential covering algorithm and classifying driving maneuvers from time-series data. In this method, we evaluated the performance of the learned ruleset and eliminated the redundant and low significant rule. The ruleset was also optimized based on the performance of test data using rule pruning. To handle the conflict between classes for the same time series instance, we chose the highest priority class for that instance. We evaluated the performance of the proposed rule-based classifier with some related works. As time-series holds many ambiguous patterns, it is difficult to classify a complete event altogether. Many events are a combination of multiple separate events. In this scenario, our proposed technique classified a whole event as multiple separate events. Though machine learning and deep learning based models gain the highest accuracy, the drawbacks of these methods have been compared and alleviated by the proposed method. Hence, our proposed model can be adopted for driving maneuver classification while labeled data is insufficient and required to reduce system cost and processing time. This work can further be extended as follows: implementing the classification task with Explainable AI tools to understand which features are contributing to get the output class without compromising the accuracy; collecting more datasets from varieties of situations; and using ensemble classifiers and features, such as moods of drivers' from different signals (e.g., EEG signal, eye tracking, activity signal etc.).

References

1. Martinez C. M., Heucke M., Wang F. Y., Gao B., Cao D. (2017) Driving style recognition for intelligent vehicle control and advanced driver assistance: a survey. IEEE Trans Intell Transp Sys 19(3):666–676
2. Paefgen J., Kehr F., Zhai Y., Michahelles F. (2012) Driving behavior analysis with smartphones: insights from a controlled field study. In: Proceedings of the 11th International Conference on mobile and ubiquitous multimedia, pp 1–8
3. Cervantes-Villanueva J, Carrillo-Zapata D, Terroso-Saenz F, Valdes-Vela M, Skarmeta AF (2016) Vehicle maneuver detection with accelerometer-based classification. Sensors 16(10):1618
4. Carvalho E, Ferreira BV, Ferreira J, De Souza C, Carvalho HV, Suhara Y, Pessin G (2017) Exploiting the use of recurrent neural networks for driver behavior profiling. In: IJCNN'17. IEEE, pp 3016–3021
5. Ferreira J, Carvalho E, Ferreira BV, de Souza C, Suhara Y, Pentland A, Pessin G (2017) Driver behavior profiling: An investigation with different smartphone sensors and machine learning. PLoS one 12(4):e0174959
6. Alvarez-Coello D, Klotz B, Wilms D, Fejji S, Gómez JM, Troncy R (2019) Modeling dangerous driving events based on in-vehicle data using Random Forest and Recurrent Neural Network. In: IV'19. IEEE, pp 165–170
7. Sarker S, Haque MM (2020) An approach towards domain knowledge based classification of driving maneuvers with LSTM network. In: IJCACI'20
8. Castignani G, Derrmann T, Frank R, Engel T (2015) Driver behavior profiling using smartphones: a low-cost platform for driver monitoring. IEEE Intell Transp Syst Mag 7(1):91–102
9. Saiprasert C, Thajchayapong S, Pholprasit T, Tanprasert C (2014) Driver behaviour profiling using smartphone sensory data in a V2I environment. In: ICCVE. IEEE, pp 552–557
10. Van Ly M, Martin S, Trivedi MM (2013) Driver classification and driving style recognition using inertial sensors. In: 2013 IEEE Intelligent Vehicles Symposium (IV). IEEE, pp 1040–1045
11. Raschka S, Patterson J, Nolet C (2020) Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence. Information 11(4):193
12. McCulloch WS, Pitts W (1943) A logical calculus of the ideas immanent in nervous activity. Bullet Math Biophys 5(4):115–133
13. Kam HT (1995) Random decision forest. In: International Conference on Document Analysis and Recognition, vol 1416, pp 278282
14. Preece A (2018) Asking 'Why'in AI: Explainability of intelligent systems–perspectives and challenges. Intell Syst Account Financ Manag 25(2):63–72
15. Fernández-Delgado M, Cernadas E, Barro S, Amorim D (2014) Do we need hundreds of classifiers to solve real world classification problems? J Mach Learn Res 15(1):3133–3181
16. Wei H, Jia H, Li Y, Xu Y (2020) Verify and measure the quality of rule based machine leaning. Knowl-Based Syst 205:106300
17. Fürnkranz J, Kliegr T, Paulheim H (2020) On cognitive preferences and the plausibility of rule-based models. Mach Learn 109(4):853–898
18. Gudyś A, Sikora M (2020) Wróbel, Lstrok RuleKit: A comprehensive suite for rule-based learning. Knowl-Based Syst 194:105480
19. Vojíř S, Kliegr T (2020) Editable machine learning models? a rule-based framework for user studies of explainability. ADAC 14(4):785–799
20. Rashid R. A., Nohuddin P. N, Zainol Z (2017) Association rule mining using time series data for Malaysia climate variability prediction. In International Visual Informatics Conference. Springer, Cham, pp 120–130
21. Sarma H. K. D, Mishra S (2016) Mining time series data with Apriori tid algorithm. In 2016 International Conference on Information Technology (ICIT). IEEE, pp 160–164
22. Xie J., Hilal A. R., Kulić D (2017) Driving maneuver classification: A comparison of feature extraction methods. IEEE Sensors J 18(12):4777–4784
23. Wang W, Xi J, Chong A, Li L (2017) Driving style classification using a semisupervised support vector machine. IEEE Trans Human-Mach Syst 47(5):650–660
24. Mammeri A, Zhao Y, Boukerche A, Siddiqui AJ, Pekilis B (2019) Design of a semi-supervised learning strategy based on convolutional neural network for vehicle maneuver classification. In: 2019 IEEE international conference on wireless for space and extreme environments (WiSEE)IEEE, pp 65–70

25. Martinelli F, Mercaldo F, Nardone V, Santone A, Vaglini G (2018, September) Real-time driver behaviour characterization through rule-based machine learning. In: International conference on computer safety, reliability, and security. Springer, Cham, pp 374–386
26. On Board Diagnostics. [Online]. Available: https://en.wikipedia.org/wiki/On-board_diagnostics
27. Sarker S, Haque MM, Dewan MAA (2021) Driving Maneuver Classification Using Domain Specific Knowledge and Transfer Learning. IEEE Access
28. Wu M., Zhang S., Dong Y. (2016) A novel model-based driving behavior recognition system using motion sensors. Sensors 16(10):1746
29. Sarker S, Haque MM (2020) A Data Change Rule Based Empirical Framework for Labeling Unlabeled Time Series Driving Data. In: 2020 2nd International Conference on Advanced Information and Communication Technology (ICAICT). IEEE, pp 476–479
30. Liu H (2015) Rule based systems for classification in machine learning context. Doctoral dissertation, University of Portsmouth
31. Witten IH, Frank E, Hall MA, Pal CJ, DATA M (2005) Practical machine learning tools and techniques. In: Data Mining, vol 2, p 4
32. Holte RC (1993) Very simple classification rules perform well on most commonly used datasets. Mach Learn 11(1):63–90
33. Quinlan JR (1986) Induction Of decision trees. Mach Learn 1(1):81–106
34. Yu X, Liu Y, Huang X, An A (2010) Mining online reviews for predicting sales performance: a case study in the movie domain. IEEE Trans Knowl Data Eng 24(4):720–734
35. Chan EH, Sahai V, Conrad C, Brownstein JS (2011) Using web search query data to monitor dengue epidemics: a new model for neglected tropical disease surveillance. PLoS Neglected Trop Diseas 5(5):e1206
36. Claveria O, Torra S (2014) Forecasting tourism demand to Catalonia: Neural networks vs. time series models. Econ Model 36:220–228
37. Nohuddin P, Coenen F, Christley R (2016) The application of social network mining to cattle movement analysis: introducing the predictive trend mining framework. Soc Netw Anal Min 6(1):1–17
38. Esling P, Agon C (2012) Time-series data mining. ACM Comput Surv (CSUR) 45(1):1–34
39. Schwarz C (2017) Time Series Categorization of Driving Maneuvers Using Acceleration Signals
40. Zhang Q, Wu J, Zhang P, Long G, Zhang C (2018) Salient subsequence learning for time series clustering. IEEE Trans Pattern Anal Mach Intell 41(9):2193–2207
41. Grabocka J, Schilling N, Wistuba M, Schmidt-Thieme L (2014) Learning time-series shapelets. In: SIGKDD'14, pp 392–401
42. Wang Z, Cai B (2021) COVID-19 cases prediction in multiple areas via shapelet learning. Appl Intell. Available: https://doi.org/10.1007/s10489-021-02391-6
43. Annam JR, Surampudi BR (2016) AAMI based ECG heart-beat time-series clustering using unsupervised elm and decision rule. In: 2016 international conference on information technology (ICIT). IEEE, pp 137–141
44. Fisch D, Gruber T, Sick B (2010) Swiftrule: Mining comprehensible classification rules for time series analysis. IEEE Trans Knowl Data Eng 23(5):774–787
45. Mehtab A, Shahid WB, Yaqoob T, Amjad MF, Abbas H, Afzal H, Saqib M. N. (2020) Addroid: rule-based machine learning framework for android malware analysis. Mob Netw Appl 25(1):180–192
46. Jamian L, Wheless L, Crofford LJ, Barnado A (2019) Rule-based and machine learning algorithms identify patients with systemic sclerosis accurately in the electronic health record. Arthritis Res Therapy 21(1):1–9
47. Vojíř S, Zeman V, Kuchař J, Kliegr T (2018) Easyminer. eu: Web framework for interpretable machine learning based on rules and frequent itemsets. Knowl-Based Syst 150:111–115
48. Agrawal R, Imieliński T, Swami A (1993) Mining association rules between sets of items in large databases. In: Proceedings of the 1993 ACM SIGMOD international conference on Management of data, pp 207–216
49. Liu B, Hsu W, Ma Y (1998, August) Integrating classification and association rule mining. In: Kdd, vol 98, pp 80-86
50. Sequential Covering Rule Building. Accessed on: May 07, 2021. [Online]. Available: https://deepai.org/machine-learning-glossary-and-terms/sequential-covering-rule-building
51. Driver Behavior Dataset, Sep 4, (2019). Accessed on: May 7, 2021. [Online]. Available: https://github.com/jair-jr/driverBehaviorDataset
52. Yuksel A, Atmaca S Driving Behavior Dataset. Mendeley Data (2020). Accessed on: May 7, 2021. [Online]. Available: https://org/10.17632/jj3tw8kj6h.2

Md. Mokammel Haque received the B.Sc.Eng. degree in computer science and engineering from the Chittagong University of Engineering and Technology (CUET), Bangladesh, the M.Sc. degree in computer engineering from Kyung Hee University, South Korea, and the Ph.D. degree in computing from Macquarie University, Australia. He is currently working as a full-time Professor with the Department of Computer Science and Engineering, CUET. His research interests include cybersecurity, computer networks, cryptography, machine learning, and algorithms.

Supriya Sarker received the B.S. degree and the M.S. degree in computer science and engineering from the Chittagong University of Engineering and Technology, Bangladesh. Since 2018, she has been a full-time Lecturer with the Department of Computer Science and Engineering, World University of Bangladesh, Dhaka, Bangladesh. Her research interests include artificial intelligence, machine learning, deep learning, explainable AI, big data, and intelligent transportation systems. She regularly serves as a reviewer in IEEE journals and conferences.

M. Ali Akber Dewan is an associate professor with the School of Computing and Information Systems, Faculty of Science and Technology, Athabasca University, Canada. Before joining the University, he was a postdoctoral researcher at École de Technologie Supérieure (Montreal, Canada) from 2012 to 2014 and at Concordia University (Montreal, Canada) from 2009 to 2012. He was a lecturer and assistant professor at Chittagong University of Engineering and Technology, Bangladesh from 2003 to 2009. He received the PhD degree in Computer Engineering from Kyung Hee University, South Korea, in 2009 and the BSc degree in Computer Science and Engineering from Khulna University, Bangladesh, in 2003. His research interests include artificial intelligence in education, educational data mining, machine learning, affective computing, computer vision, information visualization and medical image analysis. He received the Dean's Award and the Excellent Research Achievement Award for his academic performance and research achievements during his PhD studies in South Korea. He served as chair/co-chair, program committee member, and reviewer in many prestigious conferences and journals. He is an Associate Editor for Circuit Systems and Signal Processing and Computers Journals. He is a member of IEEE.