Modelling Driver Behaviour at Urban Signalised Intersections Using Logistic Regression and Machine Learning

Ahmad H. ALOMARI¹, Bara’ W. AL-MISTAREHI², Areen A. AL-JAMMAL³, Taqwa I. ALHADIDI⁴, Motasem S. OBEIDAT⁵

¹ Corresponding author, alomarish@yu.edu.jo, Department of Civil Engineering, Yarmouk University
² bwmistarehi@just.edu.jo, Department of Civil Engineering, Jordan University of Science & Technology
³ aaaljammal@eng.just.edu.jo, Department of Civil Engineering, Jordan University of Science & Technology
⁴ t.alhadidi@ammanu.edu.jo, Department of Civil Engineering, Al-Ahliyya Amman University
⁵ 2019751019@alumni.yu.edu.jo, Department of Computer Science, Yarmouk University

ABSTRACT
This study investigated several factors that may influence driver actions throughout the yellow interval at urban signalised intersections. The selected samples include 2,168 observations. Almost 33% of drivers stopped ahead of the stop line, 60% passed the intersection through the yellow interval, and 7% passed after the yellow interval was complete (red light running, RLR violations). Binary logistic regression models showed that the chance of passing went up as vehicle speed went up and down as the gap between the vehicle and the traffic light and green interval went up. The movement type and vehicle position influenced the passing probability, but the vehicle type did not. Moreover, multinomial logistic regression models showed that the legal passing probability declined with the growth in the green time and vehicle distance to the traffic signal. It also increased with the growth in the speed of approaching vehicles. Also, movement type directly affected the chance of legally passing, but vehicle position and type did not. Furthermore, the driver’s performance during the yellow phase was studied using the k-nearest neighbours algorithm (KNN), support vector machines (SVM), random forest (RF) and AdaBoost machine learning techniques. The driver’s action run prediction was the most accurate, and the run-on-red camera was the least accurate.

KEYWORDS
traffic signal; traffic safety; logistic regression; machine learning; yellow phase; red light running.

1. INTRODUCTION
At signalised intersections, the yellow phase plays a significant transitional role between the green and red intervals. When a signal light changes from green to yellow, drivers must decide whether to cross the intersection safely or stop before the stop line. Also, drivers at the onset of the yellow phase need to interact with other drivers in front and back to prevent unsafe decisions [1]. Wrong driver decisions during the yellow interval may lead to right-angle crashes, left-turn crashes, rear-end crashes, or red-light running (RLR) violations. RLR and inconsistent stopping behaviour are considered risky causes of traffic crashes at signalised intersections [1]. Several RLR violations happened due to drivers’ presence in the dilemma zones during the yellow period. At the onset of the yellow phase, dilemma zones occur upstream of the intersection approach [2]. A dilemma zone is formed when the driver approaches the intersection at a speed greater than the speed limit; conversely, an “option zone” is formed when the driver traverses slower than the speed limit [3]. Li and Wei [4] showed that dynamic dilemma zone models could predict the dilemma zone more accurately than the traditional dilemma zone and the type II dilemma zone models, which is the area where more than 10% but less than 90% of drivers would choose to stop at the start of the yellow interval. Driver behaviour during the yellow interval can be categorised as aggressive, normal or conservative based on stop/go decisions and distance to the stop line at the beginning of the yellow interval [5].
Several studies developed linear, non-linear and logistic regression models to represent and predict driver behaviour in dilemma zones based on several influencing factors at urban, suburban and rural signalised intersections. Results found that RLR enforcement cameras increase driver-stopping decisions and decrease RLR violations [6–9]. Rakha et al. [10] found that the dilemma zone for the older drivers was larger and closer to the intersection than for the younger drivers. Also, results showed that the most significant factors affecting driver behaviour through the yellow phase, stop/go decisions and RLR violations were delay [8], approaching speed [11–15], acceleration and deceleration of vehicles [9, 13, 16], traffic conditions [8, 9], weather conditions [9], perception-reaction times [16], time to stop line at the beginning of yellow interval [10, 14, 17, 18], vehicle position and distance to stop line [9, 11–15, 17, 19], vehicle type [9, 11, 12, 14, 15, 17], intersection type [11, 12], roadway grade [18], duration of yellow interval [11, 12] presence of interruptions such as vehicles, bicycles or pedestrians on the side street [17], driver age and gender [14, 20].

Driving simulators were also used in several studies to investigate driver behaviour in dilemma zones. According to Swake et al. [21], driver decisions, deceleration rates and brake response duration all influence driver behaviour. Also, they state that driving simulators are a good way to predict how drivers will act in certain situations. Chaudhary and Velaga [22] revealed that phones and music players’ distractions decrease the probability of yellow signal crossing, where the crossing possibility was positively associated with driving speed and negatively associated with time to stop line, type of manoeuvre and the presence of the distractions. Bryant et al. [23] concluded that the clearance interval at signalised intersections should consider the truck’s characteristics and how the driver acts. With the right design, trucks can get through the intersection before the green light changes to let other cars go. Hussain et al. [1] suggested that RLR violations were significantly decreased by installing red-LED earth lights combined with the regular traffic signal and RLR camera warning support. Also, it was seen that when the green signal was set to flash, people stopped in different ways. Banerjee et al. [24] investigated how red-light violation warning (RLWV) systems affect the way drivers act. Results showed that the tested system slowed down approaching vehicles by a lot, giving drivers more time to come to a safe stop at the red-light intersection.

Many countermeasures were tested to reduce RLR violations in the field and a simulated environment. Najmi et al. [25] showed that dilemma zones at signalised roundabouts were shorter and closer to the stop-line than regular signalised intersections because drivers move more conservatively at the roundabouts with a safer stopping ability. Also, Wang et al. [16] concluded that implementing five seconds of yellow interval proposed the best results for reducing risky behaviour at high-speed intersections. Moreover, Sun et al. [26] recommended an exclusive heavy vehicle lane as a future safety countermeasure to reduce vehicle conflict and intersection delay. Furthermore, Zhang et al. [9] focused on reducing RLR by installing red-light cameras and countdown timers to increase stopping behaviour at the onset of the yellow interval and reduce risky driving behaviour. Finally, Ni et al. [27] stated that a mandatory stop during a solid yellow light could control aggressive drivers efficiently, which reduces the approaching speeds significantly and enhances the acceptance of more significant headways between vehicles. However, it increased the rear-end collision probability, raising the demand for more drivers’ educational programs for traffic safety and conservative behaviours.

Other useful modelling techniques were used to represent and predict motorist performance in dilemma zones at traffic signals, such as artificial neural networks [28], fuzzy logic and decision tree modelling [15, 28, 29, 30], hidden Markov modelling [31] and other machine learning (ML) algorithms [32–40]. Results showed that these techniques could produce a high accuracy level similar to the linear, non-linear and logistic regression models. Elhenawy et al. [32–34] specified that driver aggressiveness at signalised intersections significantly affected the driver’s stop/go decisions and positively increased the models’ accuracy. They also verified that all modelling approaches generated similar prediction accuracy. Khanfar et al. [35] studied the driving behaviour at signalised intersections using unsupervised ML and a driving simulator dataset. The approach confirmed that driving behaviour reflects drivers’ habits and character rather than the signal condition; however, it still represents the nature of the intersection, which requires drivers to be more careful. Tawfeek [37] modelled the speed of unassisted drivers using ML as the yellow light turned on to improve connected and autonomous vehicle implementations at signalised intersections and enhance driver comfort. The findings suggest that the speed at the yellow light can be estimated using observations that account for the perceptual ability of drivers. Karri et al. [38, 39] examined driving behaviour (safe and unsafe stopping) at signalised intersections using ML.
based on the driving features. Findings showed that the suggested method could assist in developing a system to alert drivers reaching signalised intersections, thereby reducing rear-end collisions and crashes.

The primary goal of this study is to investigate the main factors that may influence driver actions throughout the yellow interval of the traffic signal at urban intersections in Jordan. This paper classified driver actions during the yellow interval at traffic signals into: “stopping before the stop line,” “crossing the intersection before the end of the yellow phase” and “crossing the intersection after the end of the yellow phase.” This paper developed logistic regression and ML models to investigate the relationships between motorist actions through the yellow interval and influencing factors. The rest of this article is organised as follows: Section 2 introduces the methodology used for modelling driver actions using logistic regression and ML techniques, including binary logistic regression, multinomial logistic regression, k-nearest neighbours algorithm (KNN), support vector machines (SVM), random forest (RF) and AdaBoost. It also defines the study area and the data collected in this work. Section 3 shows the modelling results by analysing the methods employed. Finally, Section 4 introduces the essential conclusions of this paper.

2. METHODOLOGY

Eight intersections controlled by traffic signals with channelised right-turn movements (Figure 1) were chosen in Irbid City, Jordan [41]. Four of them were fully actuated with RLR cameras, and RLR cameras did not control the rest. Intersection characteristics were also gathered, including the speed limit (60 km/h), lanes on the studied approach, lanes crossed, approach width (meters), width of lane (meters), flow (vehicles/hour/lane), number of approaches on the intersection and pavement marking conditions. An approach operating vehicle speed was measured using a laser radar gun. Three-legged and four-legged intersections were considered. Data were gathered during peak hours in fine weather and dry road conditions. Table 1 presents the summary of data collection.

<table>
<thead>
<tr>
<th>Intersection</th>
<th>Studied approach</th>
<th>No. of lanes crossed</th>
<th>No. of lanes</th>
<th>Width of lane [m]</th>
<th>Width of intersection [m]</th>
<th>Traffic flow [veh/h]</th>
<th>No. of phases</th>
<th>No. of legs</th>
<th>Operating speed [km/h]</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>NB</td>
<td>8</td>
<td>3</td>
<td>3</td>
<td>23.2</td>
<td>476</td>
<td>4</td>
<td>4</td>
<td>34</td>
<td>Level</td>
</tr>
<tr>
<td>T2</td>
<td>SB</td>
<td>9</td>
<td>2</td>
<td>3.5</td>
<td>37</td>
<td>382</td>
<td>4</td>
<td>4</td>
<td>39</td>
<td>Upgrade</td>
</tr>
<tr>
<td>T3</td>
<td>WB</td>
<td>9</td>
<td>4</td>
<td>3</td>
<td>31</td>
<td>394</td>
<td>4</td>
<td>4</td>
<td>47</td>
<td>Level</td>
</tr>
<tr>
<td>T4</td>
<td>NB</td>
<td>8</td>
<td>3</td>
<td>3.57</td>
<td>39</td>
<td>502</td>
<td>4</td>
<td>4</td>
<td>43</td>
<td>Level</td>
</tr>
<tr>
<td>T5</td>
<td>EB</td>
<td>5</td>
<td>4</td>
<td>2.925</td>
<td>32</td>
<td>342</td>
<td>3</td>
<td>3</td>
<td>37</td>
<td>Level</td>
</tr>
<tr>
<td>T6</td>
<td>EB</td>
<td>5</td>
<td>4</td>
<td>3.12</td>
<td>33.5</td>
<td>359</td>
<td>3</td>
<td>3</td>
<td>41</td>
<td>Level</td>
</tr>
<tr>
<td>T7</td>
<td>WB</td>
<td>6</td>
<td>3</td>
<td>2.933</td>
<td>27.5</td>
<td>221</td>
<td>4</td>
<td>4</td>
<td>29</td>
<td>Level</td>
</tr>
<tr>
<td>T8</td>
<td>EB</td>
<td>10</td>
<td>3</td>
<td>3</td>
<td>43.7</td>
<td>272</td>
<td>4</td>
<td>4</td>
<td>35</td>
<td>Level</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Intersection</th>
<th>Cycle range [s]</th>
<th>Red period [s]</th>
<th>Yellow period [s]</th>
<th>Green period [s]</th>
<th>Green split</th>
<th>All red period [s]</th>
<th>RLR</th>
<th>Green flash</th>
<th>Pavement markings</th>
<th>Pedestrians</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>131</td>
<td>95</td>
<td>4</td>
<td>30</td>
<td>0.229</td>
<td>2</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Low</td>
</tr>
<tr>
<td>T2</td>
<td>139</td>
<td>102</td>
<td>3</td>
<td>32</td>
<td>0.23</td>
<td>2</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Medium</td>
</tr>
<tr>
<td>T3</td>
<td>109</td>
<td>75</td>
<td>2</td>
<td>30</td>
<td>0.275</td>
<td>2</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Low</td>
</tr>
<tr>
<td>T4</td>
<td>146</td>
<td>104</td>
<td>3</td>
<td>37</td>
<td>0.253</td>
<td>2</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Low</td>
</tr>
<tr>
<td>T5</td>
<td>64</td>
<td>44</td>
<td>5</td>
<td>15</td>
<td>0.234</td>
<td>0</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Low</td>
</tr>
<tr>
<td>T6</td>
<td>82</td>
<td>56</td>
<td>3</td>
<td>21</td>
<td>0.256</td>
<td>2</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Low</td>
</tr>
<tr>
<td>T7</td>
<td>112</td>
<td>82</td>
<td>2</td>
<td>26</td>
<td>0.232</td>
<td>2</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Heavy</td>
</tr>
<tr>
<td>T8</td>
<td>126</td>
<td>95</td>
<td>3</td>
<td>26</td>
<td>0.206</td>
<td>2</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Low</td>
</tr>
</tbody>
</table>

Figure 1 – Bird’s-eye view of the studied intersections
Binary and multinomial logistic regression models were developed to predict motorist actions throughout the yellow interval of the traffic signals at urban intersections, whether or not they have RLR cameras. Logistic regression can be represented as the following formula (Equation 1) [42]:

\[
\text{Logit}(P) = \ln \left( \frac{P}{1-P} \right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n
\]

where \( P \) is the probability of a decision to pass (\( Y = 1 \)), \( \beta_0 \) is the model constant, \( \beta_i \) is the coefficient of variable and \( X_i \) represents the predictor variable.

In this paper, the dependent variable was categorical. So, binary and multinomial logistic regression models are the best choices for predicting the probability of a categorical dependent variable [42]. Also, they were selected to overcome the problem of violating the linearity assumption. The proposed models involved two types of variables: categorical and continuous. Table 2 describes all the variables involved in the proposed models.

Previous research has widely used several ML classification algorithms. To predict driver behaviour, the commonly used KNN, SVMs, RF and AdaBoost are used in this paper. The same training dataset was employed to train the different ML techniques, and the models' performance was reported using the same testing dataset.

The KNN algorithm is a straightforward non-parametric modelling technique [43]. It is based on the probability that similar data points belong to the same cluster. KNN begins by locating the K nearest neighbourhoods of the training dataset and then predicts the major class within the K nearest neighbours. Due to its simplicity and ability to predict in less time, it has been chosen as one of the best data mining algorithms [44].

Table 2 – Description of variables in binary and multinomial regression models

<table>
<thead>
<tr>
<th>Variable</th>
<th>Variable Type</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speed</td>
<td>Continuous</td>
<td>km/h</td>
</tr>
<tr>
<td>Vehicle distance to stop line</td>
<td>Continuous</td>
<td>Meter</td>
</tr>
<tr>
<td>Volume in selected approach</td>
<td>Continuous</td>
<td>veh/h/lane</td>
</tr>
<tr>
<td>Yellow period</td>
<td>Continuous</td>
<td>Second</td>
</tr>
<tr>
<td>Green period</td>
<td>Continuous</td>
<td>Second</td>
</tr>
<tr>
<td>Red period</td>
<td>Continuous</td>
<td>Second</td>
</tr>
<tr>
<td>Cycle period</td>
<td>Continuous</td>
<td>Second</td>
</tr>
<tr>
<td>Green split</td>
<td>Continuous</td>
<td>Unitless</td>
</tr>
<tr>
<td>Vehicle type</td>
<td>Categorical</td>
<td>0 pc (regular cars), 1 taxi, 2 pickup, 3 truck, 4 van, 5 bus</td>
</tr>
<tr>
<td>Intersection type</td>
<td>Categorical</td>
<td>0 3 legs, 1 4 legs</td>
</tr>
<tr>
<td>Type of movement</td>
<td>Categorical</td>
<td>0 left, 1 U-turn, 2 through</td>
</tr>
<tr>
<td>Vehicle position</td>
<td>Categorical</td>
<td>0 platoon, 1 not platoon</td>
</tr>
<tr>
<td>Lane width</td>
<td>Continuous</td>
<td>Meter</td>
</tr>
<tr>
<td>Intersection width</td>
<td>Continuous</td>
<td>Meter</td>
</tr>
<tr>
<td>No. of lanes in</td>
<td>Categorical</td>
<td>2 two-lanes, 3 three-lanes, 4 four-lanes</td>
</tr>
<tr>
<td>No. of lanes crossed</td>
<td>Categorical</td>
<td>0 5-lanes, 1 6-lanes, 2 8-lanes, 3 9-lanes, 4 10-lanes</td>
</tr>
<tr>
<td>Grade</td>
<td>Categorical</td>
<td>0 level, 1 upgrade, 2 downgrade</td>
</tr>
<tr>
<td>RLR cameras</td>
<td>Categorical</td>
<td>0 yes, 1 no</td>
</tr>
<tr>
<td>Pavement marking</td>
<td>Categorical</td>
<td>0 yes, 1 no</td>
</tr>
<tr>
<td>Green flash</td>
<td>Categorical</td>
<td>0 yes, 1 no</td>
</tr>
<tr>
<td>Pedestrians</td>
<td>Categorical</td>
<td>0 low, 1 medium, 2 heavy</td>
</tr>
</tbody>
</table>
The KNN accuracy depends on choosing the best cluster size; the optimum K was selected based on prediction accuracy. Afterwards, the response (i.e. driver action in our problem) is classified by considering the majority vote of the K closest points within the class as shown in Equation 2; where $R$ is the number of assigned classes based on checking the model accuracy for each value, $y_{test}^R$ is the test observation which is assigned to class R based on the majority of class R voting after training the model using $x_{train}^R$ as input variables and $y_{train}^R$ as response variable [45].

$$y_{j}^{test} = \frac{1}{R} \sum_{x_j \in C_k} y_{j}^{train}$$  \hspace{1cm} (2)

The SVM algorithm is a supervised learning method that sorts data into groups based on how different the groups are. Equation 3 shows that the algorithm looks for the hyperplane (also called a “splitter”) that is the closest to the training data. The SVM looks for the weight ($w$) with the most significant margin near the hyperplane and meets the two constraints (see Equations 4 and 5 [46]).

$$\min_{w,b,\xi} \left( \frac{1}{2} w^T w + c \sum_{n=1}^{N} \xi_n \right)$$  \hspace{1cm} (3)

subjected to:

$$y_n (w^T \phi(X_n) + b) \geq 1 - \xi_n , n = 1, \ldots , N$$  \hspace{1cm} (4)

$$\xi_n \geq 0 , n = 1, \ldots N$$  \hspace{1cm} (5)

$w$ – set of parameters used to define class boundaries

c – penalty parameter

$\xi_n$ – parameter to express the margin error

$b$ – intercept is linked with the hyperplane functions to change data from X space

$\phi(X)$ – transform data from X space to Z space

$y_n$ – target value

The objective function is simplified by adding the two terms in Equation 2. Primarily, the first term aims to clarify the difference between classes. Reducing its length is identical to enlarging the gap between classes. The other term aims to reduce the penalty (regularisation) parameter times the error term. The penalty term is intended to address overfitting, whereas the term $c$ is intended to optimise the performance of the model. Therefore, $n$ represents the index of the data observation, $w$ denotes the decision border between classes, $c$ denotes the regularisation (or penalty) parameter and $\xi_n$ represents the margin violation error parameter. $K$ is the number of observations in the X space that the $\phi(X_n)$ function moved to another space. The transformation is done to make a Z space that can be used to make class boundaries easier to define. On the other hand, certain functions (i.e. kernels) can be used directly to create transformations more easily, as demonstrated in this paper. Meanwhile, Equation 2 can be solved using kernels or $\phi(X_n)$ to transform data to the Z plane. Before model construction, the kernel type should be determined (i.e. linear). One kernel could work better than another. Some realistic recommendations propose using different kernels at various data sizes and problems [46].

Random forest (RF) is a successful ensemble prediction technique. Breiman [47] used the strong law of large numbers (SLLN) to demonstrate that there is no overfitting of RFs as more trees are established. The fundamental concept underlying ensemble approaches is that creating many simple models will improve overall performance. An RF is a collection of unpruned decision trees with random feature selection at each split. Classification and regression tree (CART), a well-known ML technique, is a frequently used decision tree in RFs [48]. In ensemble terms, RFs start with the CART, which refers to the weak model.

CART partitions the feature space into two regions to optimise its objective function locally (children). This procedure is repeated for every child until the termination criteria are met. Cases from each region have (nearly) identical outcomes. Using the assumption that the training dataset consists of H cases, P predictors and M trees to generate for each of the M iterations, the RF classification algorithm is as follows:

- Build a bootstrap trial from the first dataset by randomly selecting H cases and replacing them. The subset must comprise around 66 percent of the initial training set; the left cases should be duplicated.
- For certain numbers, $\sqrt{P}$ predictor variables are chosen randomly from all predictor variables at each node.
- From the $\sqrt{P}$ predictor variables, the best predictor variable is employed to generate a binary split on that node.

843
Avoid value-complexity pruning and keep the tree in its current state, along with other constructed trees from prior iterations. During the testing phase, the recently delivered case is moved down each tree. By supplying a class label, each tree votes for one class. RF determines the class with the most votes. This method will be evaluated as part of this research effort because it can improve driver stop/run behaviour modelling.

The adaptive boosting (AdaBoost) algorithm is an incremental contribution-based ML algorithm [48]. AdaBoost was developed in response to whether it was possible to combine a cluster of “weak” learner algorithms with low accuracy to generate a learning algorithm with a high one. Prior to the running of AdaBoost, the conventional ML technique consisted of selecting the highest-discriminating class of features. In other terms, algorithms should be classified. AdaBoost employs a collection of weak classifiers, each of which is trained on the same training dataset but has a different weight distribution. Every learner concentrates on the instances where the previous learner failed. AdaBoost’s output is the weighted average of all weak learners’ outputs. It has a minor misclassification than the sum of weak learners and a generalisation error limit [48, 49].

In a classification problem, the output could be a true positive prediction (TP), a true negative prediction (TN), a false positive prediction (FP) or a false negative prediction (FN). These distinct outcomes were utilised to compute the various evaluation metrics. Precision, recall, F1-score and support are the evaluative metrics (Equations 6–8). Precision and recall are two methods for evaluating the performance of a classifier in binary and multiclass classification problems. Precision is determined by dividing the number of accurate positives by the accurate and false positives summation. Recall is the proportion of correctly classified instances (true positives) to the total instances that should have been classified as positive (true positives plus false negatives). The F1-score is utilised to evaluate the accuracy of a model on a dataset. It assesses classification systems that categorise instances as “positive” or “negative”. The F-score combines the model’s precision and recall [40]. Support refers to the number of actual class occurrences in the dataset. It is the count of true instances for each class. These indices are calculated as follows:

\[
\text{Precision} = \frac{TP}{TP + NP} \tag{6}
\]

\[
\text{Recall} = \frac{TP}{TP + FN} \tag{7}
\]

\[
\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{8}
\]

These metrics are essential for evaluating the performance of classification models, and they help in understanding how well a model is doing in correctly identifying positive and negative cases.

3. RESULTS AND DISCUSSION

The data extraction process yielded a total of 2,168 samples, including stop, pass and RLR violations. Only 721 (33%) drivers stopped ahead of the stop line, 1,296 (60%) passed the intersection through the yellow interval and 151 (7%) passed after the yellow interval was complete (RLR violations). According to the findings, drivers had more potential to stop ahead of the stop line during the yellow interval on intersections with RLR cameras, green light flashing, heavy pedestrian activity, pavement markings and intersections with four approaches. In addition, platoon-positioned vehicles had more pass actions (69.8%) than non-platoon-positioned vehicles (46.6%). Moreover, the van carried the most significant percentage of pass action among all vehicle types (68.1%), while the taxi was the lowest (54.5%). In contrast, the pass rates for trucks and pickups were 64% and 65.9%, respectively. The percentages for straight, left and U-turn manoeuvres were 58.4%, 61.3% and 57.8%, respectively. However, straight movement had the highest RLR violation rate (8.6%). Table 3 displays the frequencies and percentages of driver actions for the studied intersections.

3.1 Binary logistic regression models

In the Statistical Package for the Social Sciences (SPSS), sequential logistic regression models were made to predict how drivers would do during the yellow phase. Two models were considered. Model one top-level logit, including a stop or go action, and model two bottom-level logit models, including only legal passes through the yellow phase or RLR violations. Figure 2 shows a two-step decision process for binary logistic regression.
Model-I (stop and pass action) looked at 2,168 observations, including people stopping before the stop line and passing through the intersection. Also, Model-II (legal pass and RLR violations), which looked at 1,450 observations, only looked at legal passes through the intersection during the same phase and RLR violations. Model-I and Model-II have different sample sizes because the driver takes different actions during the same phase in each model. Table 4 presents the binary logistic regression analysis for Model-I and Model-II.
For Model-I, the negative sign of the variable’s vehicle distance to the stop line and the green interval indicates that the probability of passing action increases with the raising of these variables. The positive sign of the variable operating vehicle speed suggests that the likelihood of passing action increases with the raising of this variable. Moreover, the passing probability is found to be safely affected by the presence or absence of RLR cameras, movement type and vehicle position. The reference movement type and vehicle position were taken through movement, not platoon position. The platoon vehicle was more likely to pass than not, given the platoon vehicle’s position. Also, the left movement was more likely to pass than the U-turn movement. Finally, drivers at locations with no RLR cameras had a greater chance of passing than locations with RLR cameras.

The odds ratio of the operating vehicle speed means that for each unit raised in the variable operating vehicle speed, the odds of passing probability increase by 1.081 times. Also, for each unit raised in the variable “vehicle distance”, the odds of passing probability decrease by 0.913 times.

Table 5 shows the classification predicted for Models I and II. The overall prediction accuracy for Model-I and Model-II was 76.7% and 94.4%, respectively, indicating that the prediction results are close to reality. The Negelkerke R-squared for Model-I and Model-II was found to be 0.364 and 0.645.

### 3.2 Multinomial logistic regression models

MLR (multinomial logistic regression) models were made to predict how drivers would do during the yellow phase. In the proposed model, driver actions, including stopping before the stop line, passing through the intersection, and breaking RLR rules, were considered. Figure 3 shows the step-decision process for multinomial logistic regression.
The proposed model describes the driver actions as “stop” (Y=0), “pass through the yellow phase” (Y=1) and “RLR violations” (Y=2). Also, two types of variables were included in the proposed models: categorical and continuous variables. Multinomial logistic regression analysis for a stop to RLR violations and a legal pass to RLR violation models are presented in Table 6.

For the action model, the negative sign of the “vehicle distance” indicates that the probability of a stop action for RLR violations decreases with the increase in this variable. Moreover, for the legal-pass action model, the stopping possibility was discovered to be safely affected by vehicle position. Also, for a legal-pass action, the negative sign of the variable’s “vehicle distance” and green interval indicates that the probability of a legally passing action decreases with the increase in these variables. The positive sign of the variable operating vehicle speed suggests that the likelihood of legally passing action increased with this variable’s increase. In addition, movement type had an impact on the passing likelihood.

The odds ratio of the operating vehicle speed means that for each unit increase in the variable operating vehicle speed, the odds of a legal passing probability increase by 1.957 times. Also, for each unit raised in the...
variable “vehicle distance”, the odds of a legal passing probability decrease by 0.715 times. Table 7 shows the classification of predicted stop-action and legal-pass action models.

**Table 7** – Classification results, R-squared and models summary

<table>
<thead>
<tr>
<th>Action</th>
<th>Percentage correct</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stop</td>
<td>60.2%</td>
</tr>
<tr>
<td>Legal-pass</td>
<td>88.6%</td>
</tr>
<tr>
<td>RLR violations</td>
<td>49%</td>
</tr>
<tr>
<td>Overall percentage</td>
<td>76.4%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model</th>
<th>-2 log likelihood</th>
<th>Cox and Snell R-squared</th>
<th>Nagelkerke R-squared</th>
<th>Mc Fadden</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1745.320</td>
<td>0.483</td>
<td>0.589</td>
<td>0.384</td>
</tr>
</tbody>
</table>

Model-I stop- and pass-action

Logit (P) = \( \ln \left( \frac{P}{1-P} \right) = 1.175 + 0.078 \times S - 0.091 \times D - 0.084 \times G.1 - 0.698 \times M1 - 0.832 \times M2 + 0.694 \times V.P1 + 1.284 \times PoF RLR \)

Model-II legal-pass and RLR violations

Logit (P) = \( \ln \left( \frac{P}{1-P} \right) = 0.20 \times S - 0.291 \times D + 0.746 \times Y.I - 0.798 \times M1 - 2.001 \times M2 \)

Multinomial logistic regression

Stop action model

\( P(\text{Stop}/\text{RLR}) = -0.192 \times D - 0.794 \times V.P1 \)

Legal-pass action model

\( P(\text{Legal pass} / \text{RLR}) = 0.405 \times S - 0.380 \times D - 3.684 \times G.1 - 0.705 \times M1 - 1.410 \times M2 \)

The total prediction accuracy was 76.4%, indicating that the prediction results were close to reality. Table 7 also shows R-squared results for stop action and legal-pass action models. The Mc Fadden R-squared was found at 0.384, which indicates that it is effective enough to forecast driver performance through the yellow phase.

### 3.3 Machine learning (ML) models

This section discusses the outcomes of the Python-based ML methods applied in this paper. The first step in modelling data was feature engineering. It began with a data type check, followed by a report of the original data correlation matrix and a review of the problem’s most relevant variables. Figure 4 shows the correlation matrix of the original dataset for the different collected variables.

All the various variables have different correlation values. Red interval, no. of lanes crossed, cycle length, green interval, pavement markings, intersection type and presence of RLR cameras are highly inversely correlated variables. The number of lanes is related to driver behaviour. Nonetheless, the most relevant variables were chosen using the P-value and F-score. The selected characteristics were determined based on P-values and F-scores exceeding 0.05 and 5, respectively.

Table 8 presents the original variables along with their P-values and F-scores. There were 2,168 total instances in this dataset, out of which 1,734 (80%) random data were used for training, and the remaining 434 (20%) were used for testing and validating the model. According to Table 8, the chosen variables are the intersection type, green interval, cycle length, number of lanes crossed, number of lanes in, location, volume in selected approaches, lane width, grade, vehicle position, and the existence of RLR cameras, pavement marking, green flash, yellow interval and pedestrians. These variables were selected as the X matrix, while driver actions were chosen as their output. This study used four methods: KNN, SVM, RF and AdaBoost.
Table 8 – Variables with their P-value and F-score

<table>
<thead>
<tr>
<th>Input variable</th>
<th>P-value</th>
<th>F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Presence of RLR cameras</td>
<td>1.00E-62</td>
<td>298.3186</td>
</tr>
<tr>
<td>Intersection type</td>
<td>1.06E-57</td>
<td>272.2201</td>
</tr>
<tr>
<td>Presence of pavement marking</td>
<td>1.06E-57</td>
<td>272.2201</td>
</tr>
<tr>
<td>Green interval</td>
<td>1.78E-62</td>
<td>154.6135</td>
</tr>
<tr>
<td>Cycle length</td>
<td>1.98E-54</td>
<td>241.7396</td>
</tr>
<tr>
<td>No. of lanes crossed</td>
<td>1.31E-52</td>
<td>244.9839</td>
</tr>
<tr>
<td>Red interval</td>
<td>2.67E-52</td>
<td>241.7396</td>
</tr>
<tr>
<td>Location</td>
<td>2.45E-34</td>
<td>154.6135</td>
</tr>
<tr>
<td>Volume in selected approach</td>
<td>2.64E-20</td>
<td>85.13198</td>
</tr>
<tr>
<td>Lane width</td>
<td>1.54E-14</td>
<td>59.86941</td>
</tr>
</tbody>
</table>

**Table 9** presents the overall classification report and confusion matrix of the test data. This table demonstrates four metrics: precision, recall, F1-score and support.

The KNN was used to evaluate which of three possible driver actions occurred. A 10-fold cross-validation was used to select the best model for each value of K, and the 10-fold with the highest average accuracy was selected. The optimal value of K was determined by comparing different values of K to overall classification accuracy. Figure 5 illustrates the classification accuracy of KNN with varying K neighbours.

As illustrated in Figure 3, using pooled features in the proposed hierarchical framework yielded higher classification accuracy than just time-domain features. The optimal K was determined to be nine, with an accuracy of 67.5%.
Regarding SVM, a 10-fold cross-validation was used to get the optimal model for each value of K. The model with the highest average accuracy across all 10 folds was chosen. In addition, different kernels were used to train the model, and the best kernel was RBF with a gamma of 0.001, with an accuracy of 68.2%. In RF, many trees were trained, with the best accuracy of 69.58% coming from 400 trees. Finally, the AdaBoost method achieved an accuracy of 69.58%. The driver action run prediction had the highest accuracy for all methods (KNN, SVM, RF and Adaboost). In contrast, the run-on-red camera had the lowest precision because the number of its samples in the train and test data is low, and that is normal since not many people run on a red camera. The overall accuracy of all models was 68.7%.

The correct configuration of optimal ML models is crucial for practitioners implementing them. For example, practitioners can replicate these results with SVM by employing a 10-fold cross-validation approach and choosing the best kernel (RBF with a gamma of 0.001). This method suits scenarios where a balance between precision and recall is essential. On the other hand, with the knowledge that 400 trees yield the best accuracy, practitioners can set up their RF classifier accordingly. RF is known for its robustness and is suitable for handling large datasets. Also, the AdaBoost ensemble method achieved a competitive accuracy score. It can be applied when emphasis needs to be placed on the classification of harder-to-detect instances. Moreover,
the selection of K is vital for KNN and impacts the model’s performance. The experiments found K=9 to be optimal for this specific dataset. Practitioners should consider a similar tuning process when applying KNN to their data.

Overall, the investigated models, binary logistic regression, multinomial logistic regression and the ML models, can be applied in practice in three major approaches.

Classification of driver actions. The primary application of the investigated models is in classifying driver actions based on data obtained from various sensors or sources. For instance, these models can be deployed in a real-time setting within a vehicle to predict and classify driver actions such as “run” actions. This prediction can be utilised for several practical purposes, including:
- driver assistance systems - these models can be integrated into driver assistance systems, providing real-time feedback to the driver. For example, if the model predicts a “run” action, it can trigger warnings or corrective actions, such as automatic braking or steering assistance, to prevent accidents.
- traffic safety - the models’ ability to predict driver actions can contribute to improving overall traffic safety. Law enforcement or traffic management authorities can use this information to identify and address risky behaviour patterns, making roads safer for all users.
- insurance industry - insurance companies could leverage this model to assess driver behaviour and risk. It could be used to offer more accurate and personalised insurance premiums based on individual driving habits, ultimately promoting safer driving practices.
- fleet management - companies with large vehicle fleets can use these models to monitor driver behaviour and enhance the efficiency and safety of their operations. It can help identify drivers who consistently exhibit risky behaviour and may require additional training or supervision.

Optimal model selection. Detailed information about the optimal models and their configurations is crucial for practitioners who want to implement these techniques.

Data considerations. It is essential to mention that the models’ performance may vary depending on the quantity and quality of training data. In cases like “run-on-red camera”, where the sample size is limited, practitioners should be cautious about the model’s reliability for such specific scenarios.

In summary, the investigated models’ practical applications extend to driver assistance systems, traffic safety, insurance, fleet management and more. The detailed model configurations and optimal parameters provided in the paper can serve as a valuable starting point for practitioners looking to implement similar systems in their respective domains.

4. CONCLUSIONS

The objective of this study was to construct statistical models representing the relationships between various parameters and driver actions during the yellow interval at urban intersections controlled by traffic signals, whether or not they have red-light running (RLR) cameras. A video camera was utilised and positioned at an appropriate height ahead of the intersection to observe traffic signals, driver actions and parameters that may influence driver behaviour. Around 2,168 observations of motorist behaviour have been gathered from the data. Results showed that only 33% of drivers stopped ahead of the line, 60% passed the intersection in the yellow interval, and 7% passed after the yellow interval was complete (RLR violations). The following are the main findings:
- The likelihood of vehicles stopping before the line through the yellow interval with RLR cameras, the green flash tool, multiple pedestrians, pavement markings and intersections with four legs.
- At 68.1%, vans had the most significant proportion of pass actions among all vehicle types. In comparison, the taxis experienced the lowest pass rate, at 54.5%, although trucks and pickups had comparable pass rates, at 64% and 65.9%, respectively.
- The pass rates for through, left and U-turn manoeuvres were 58.4%, 61.3% and 57.8%, respectively. Nevertheless, the through direction had the most significant percentage of RLR violations. In addition, platoon-positioned vehicles had more pass actions (69.8%) than non-platoon-positioned vehicles (46.6%).
- The prediction accuracy of binary logistic regression Model-I was 76.7%, and Model-II’s was 94.4%. Model-I (stop and pass action) indicated that the probability of a pass action increased with the rise in speed and dropped with the growth in the green interval and the length to the stop line. Also, the presence of RLR
cameras, movement type, and vehicle position significantly influenced the passing probability, but vehicle type did not.

- The binary logistic regression Model-II (legal pass and RLR violations) showed that the likelihood of legally passing rose with the increase in vehicle speed and yellow interval and dropped as the distance from the stop line increased. Also, movement type had a meaningful impact on the passing probability, but vehicle type and vehicle position did not.

- The prediction accuracy of the proposed multinomial logistic regression model was 76.4%, and McFadden’s R-square was 0.384. The proposed models showed that the likelihood of stopping before the stop line declined with the increase in vehicle distance to the stop line. Also, vehicle position had an essential effect on the stopping probability, but movement and vehicle types did not. The likelihood of passing in the yellow interval decreased with the increase in vehicle distance to the stop line and green interval and increased with the increase in speed. Moreover, movement type had a meaningful impact on the passing probability, but vehicle position and vehicle type did not.

- This paper also used the commonly utilised KNN, SVM, RF and AdaBoost ML techniques to predict driver behaviour. The same training dataset was employed to train the different ML methods, and the models’ performance was reported using the same testing dataset. As a result, the driver action run prediction had the highest accuracy, while the run-on-red camera had the lowest precision. The overall accuracy of all models was 68.7%.

Additional research is suggested to explore the influence of geometric design features, asphalt conditions, the characteristics of drivers, whether there are any passengers in the vehicle and the usage of mobile phones throughout the day.

REFERENCES


Elhenawy M, Jahangiri A, Rakha HA, El-Shawarby I. Modeling driver stop/run behavior at the onset of a yellow indication considering driver run tendency and roadway surface conditions. Accident Analysis & Prevention. 2015;83:90-100. DOI: 10.1016/j.aap.2015.06.016.


