Received: 17 May 2021 | Revised: 8 September 2021 | Accepted: 20 October 2021

DOI: 10.1049/itr2.12132

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH PAPER

Using genetic programming on GPS trajectories for travel mode detection

Farnoosh Namdarpour[1] | **Mahmoud Mesbah**[1,2] | **Amir H. Gandomi**[3] | **Behrang Assemi**[4]

[1] Department of Civil and Environmental Engineering, Amirkabir University of Technology, Tehran, Iran

[2] School of Civil Engineering, The University of Queensland, Brisbane, Australia

[3] Faculty of Engineering & Information Technology, University of Technology Sydney, Ultimo, Australia

[4] School of Built Environment, Queensland University of Technology (QUT), Brisbane, Australia

Correspondence
Mahmoud Mesbah, Department of Civil and Environmental Engineering, Amirkabir University of Technology, Tehran 15916–34311, Iran.
Email: mmesbah@aut.ac.ir

Abstract

The widespread and increased use of smartphones, equipped with the global positioning system (GPS), has facilitated the automation of travel data collection. Most studies on travel mode detection that used GPS data have employed hand-crafted features that may not have the capabilities to detect all complex travel behaviours since their performance is highly dependent on the skills of domain experts and may limit the performance of classifiers. In this study, a genetic programming (GP) approach is proposed to select and construct features for GPS trajectories. GP increased the macro-average of the F1-score from 77.3 to 80.0 in feature construction when applied to the GeoLife dataset. It could transform the decision tree into a competitive classifier with support vector machines (SVMs) and neural networks that are both able to extract high-level features. Simplicity, interpretability, and a relatively lower risk of overfitting allow the proposed model to be readily used for passive travel data collection even on smartphones with limited computational capacities. The model is validated by a second dataset from Australia and New Zealand, which indicated that a decision tree with the GP constructed features as its input has a considerably higher transferability than SVMs and neural networks.

1 | INTRODUCTION

Individuals use different modes of transportation for travelling, such as walking, biking, cars, buses and trains, which are referred to as their travel modes. Travel mode is a key attribute of travel behaviour, always considered for effective travel demand management and transportation planning. Transportation agencies can adopt mode-specific strategies to reduce travel time, traffic congestion and air pollution [1]. Providing users with personalized information, and incentivizing them toward sustainable travel behaviour can also be achieved when individuals' travel modes are known. Data on travel attributes, including mode, can be collected by traditional survey methods; however, these methods have major drawbacks, such as low response rates, high involvement of respondents, and misreporting of travel details [2, 3]. Automatic methods of collecting data, mostly based on global positioning systems (GPS), can provide accurate spatiotemporal attributes of travel and can be used as a supplementary or an alternative method of data collection.

GPS has been used for travel data collection since the mid-1990s [4]. The increase in and widespread use of smartphones, which are equipped with GPS technology, has enabled collecting of such data at a much lower cost, and the smartphones' access to the internet has facilitated the transmission of collected data. As smartphones have turned into an inseparable part of people's lives, and are carried by users most of the times, they can contain invaluable information about individuals' travels. Although temporal and spatial attributes of travel can be extracted directly from GPS data, attributes such as travel mode need to be questioned or inferred. Different methods have been applied to automatically detect travel modes; nonetheless, similarities between the characteristics of different travel modes, such as private cars and buses, make it difficult to distinguish them from one another. Hand-crafted features, such as maximum velocity and mean acceleration, have been used for travel mode detection, which are usually defined by domain experts. Therefore, the ability of experts in defining effective features has a fundamental role in the performance of travel mode detection algorithms [1]; such features may be ineffective for detecting complex travel attributes [5]. For instance, a private car in traffic congestion may have a similar maximum velocity to a pedestrian. This issue especially limits the performance of algorithms—such as decision trees (DTs)—that are not capable of extracting features from inputs intrinsically [6]. One way to address this issue is by constructing features automatically. Although complex models, such as neural networks, can be used for this purpose, the created features cannot usually be identified explicitly and there is a high potential of overfitting [7]. Therefore, in this study, genetic programming (GP) is used for feature construction that can lead to the development of simpler models and provide a better understanding of the learned concepts [8].

GP is an evolutionary computation technique that can evolve a mathematical model without the need to fully specify the model in advance [9]. The flexibility of GP allows adapting to the particular needs of each problem, and makes it a potentially suitable tool for addressing different classification tasks [10, 11]. GP can be used to extract classifiers using different kinds of representations, such as DTs and classification rules. It can also be used for pre-processing tasks, such as feature selection and construction, and post-processing purposes, such as constructing ensemble classifiers [10]. In this study, the feature selection and construction capabilities of GP are investigated for travel mode detection, using the GeoLife GPS dataset collected by Microsoft Research Asia [12]. After cleaning the data and extracting the initial features defined based on the literature, four different approaches are investigated to construct features using GP. Then, their performance in mode detection is evaluated using a DT. The best results are compared with the results of other classification algorithms and other relevant studies. The selected approach is then implemented on another dataset, which is collected in New Zealand and Australia [13, 14] to validate the obtained results.

Improving the performance of travel mode detection using only GPS data, while keeping the constructed models simple, is an important step toward the automation of travel data collection. This study proposes a methodology that uses GP to construct features that can improve travel mode detection. GP is a building block that can improve the performance of simple classifiers, such as DTs. DTs can be used to develop interpretable models with low computational costs [6]. Consequently, they can be readily used for passive travel data collection even on smartphones with limited computational capacities. GP has been used in many different fields for classification tasks but its applications in travel mode detection are unexplored.

The remainder of the paper is organized into several sections. The background of travel mode detection and genetic programming is reviewed in Section 2. The details of the proposed methodology, including the data preparation steps and settings of different GP approaches for feature construction, are explained in Section 3. The results are presented and analysed in Section 4, and the study is concluded in Section 5.

2 | BACKGROUND

2.1 | Travel mode detection

Automated travel mode detection generally includes three steps of data cleaning, identifying single modal trip segments, and inferring the travel mode of each segment, as explained next.

GPS data might include poor-quality points due to several error sources, such as receivers or satellites, atmospheric disturbances, and urban canyons [4]. Therefore, raw GPS data is usually passed through some rule-based filters, and erroneous points are detected using velocity, location, or other local features of GPS points [1, 14–19].

Then, single modal trip segments should be identified. A trip is the movement of a person for a specific purpose, and a single modal trip segment is a part of a trip which is traversed using a single mode [18]. Most researchers have adopted rule-based methods to detect trip segments [2], which often include considering a threshold of dwell time, such as 120, 150, 180, 200, or even 300 s with or without other conditions [4, 14]. The selected thresholds may vary based on the attributes of local activities [4]. If the dwell time exceeds the considered threshold, the segment ends. The identified trip segments are used for travel mode detection in the next step.

Methodologies for travel mode detection using GPS data can be classified into three main categories: (1) rule-based methods, (2) probabilistic methods, and (3) machine learning algorithms [4]. In rule-based methods, travel mode is detected based on some predetermined criteria [4, 16, 18]. In probabilistic methods, the probability of each mode is estimated based on characteristics of GPS data and respondents; probability matrix and fuzzy logic are examples of this method [4, 17]. Finally, different machine learning algorithms, such as Bayesian and neural networks, support vector machines (SVMs), DTs, and random forests have also been used for travel mode detection [1, 3, 5, 19–21].

Several studies have integrated GPS data with other sources of information to improve the performance of mode detection. For example, Stopher et al. [16], Tsui and Shalaby [17], and Gong et al. [18] combined GPS and Geographic Information Systems (GIS) data. Feng and Timmermans [20], and Ansari Lari and Golroo [19] integrated GPS and accelerometer data. Other sources of information, such as personal and socio-demographic characteristics of travellers, or data from other smartphone sensors, such as gyroscope and magnetometer, have also been used along with GPS data to detect travel mode [22, 23]. Deploying multiple sources of data can lead to better results, however, the application of such methods is often limited due to a lack of access to such information sources on a large scale. Accordingly, only GPS trajectories are used in this research to demonstrate the advantage of GP to detect travel modes.

Assemi et al. [14] applied multinomial logistic regression, nested logit, and multiple discriminant analysis models to detect travel modes using only GPS data. Seven major categories of features were used in this study, and different statistical variables were defined for each category. Zheng et al. [24] applied three classification methods, namely DT, SVM, and Bayesian network while using bootstrap aggregating as an ensemble meta-algorithm to detect travel mode by specifying a set of features. In a later study [25], Zheng et al. introduced three new features and proposed a graph-based post-processing algorithm to further improve their method's performance. However, these studies have all used hand-crafted features which limits the performance of proposed methods to the skills of experts in defining discriminative features.

Several studies have used the features proposed by Zheng et al. [24, 25] to construct high-level features which can improve the accuracy of travel mode detection [5, 26]. Endo et al. [26] represented raw GPS data of each segment as a two-dimensional image, which was fed into a deep learning model to extract high-level features. Then, they combined these model-generated features and hand-crafted features to create several feature sets, which were used to evaluate the performance of several classifiers.

Wang et al. [5] extracted deep point-level features from hand-crafted point-level features, such as velocity and distance of GPS points, by using a sparse auto-encoder. Deep trajectory-level features were then extracted from deep point-level features using a convolutional neural network. Both deep and hand-crafted trajectory-level features were used separately and then together as the feature set, and their performance was evaluated using a number of classifiers. However, the results of these studies could not outperform the results of Zheng et al. [25] which only used hand-crafted features.

Dabiri and Heaslip [1] constructed high-level features by applying a convolutional neural network (CNN). They used speed, acceleration, jerk (the rate of change in the acceleration), and bearing rate of GPS points as the input to their model. Trip segments were divided in such a way that all have the same number of GPS points, since the structure of input instances in CNN should be similar. As a result, 14,000 real segments were converted to about 32,000 segments. Consequently, these sub-segments might not include all characteristics needed to detect travel modes, and thus the performance of the model could adversely be affected.

Nawaz et al. [27] used the four features in [1], and added other features, including a regional index, time of day, day of the week, and weather features in a convolutional long short-term memory model to detect travel mode. They showed the performance could improve by adding the above-mentioned features. However, similar to [1], segments had to have similar lengths, and while using additional features in a more sophisticated model, results could not outperform those of [1].

To address the aforementioned limitations, this study investigates the possibility of using GP as a technique for automatic feature selections and construction. Although GP has been used for classification tasks in many fields, to the best of the authors' knowledge, it has not been explored for improving the performance of travel mode detection. In this study, multiple approaches are investigated for this purpose, and the best results are compared with those classifiers which are able to construct high-level features, such as SVM and neural networks.

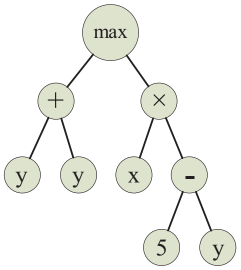

FIGURE 1 Example of a tree representation in GP

2.2 | Genetic programming (GP)

GP is an evolutionary computation technique in which a population of individuals (called a GP population here) is evolved stochastically to a population that is potentially better than the previous one, given a specific fitness function (performance measure) that determines the fitness (quality) of each individual [28]. It cannot guarantee optimality but is equipped with processes to avoid local minima/maxima in which deterministic methods potentially get trapped [9]. GP has been used successfully for developing novel solutions for many problems [9].

2.2.1 | Structure and algorithm

Individuals of a GP population can have different structures out of which the tree structure is the most common one [9]. For example, if equation max $(y + y, x \times (5 - y))$ is given, Figure 1 depicts its equivalent tree structure. The terminal set of $\{x, y, 5\}$ is the variables and constants that constitute the leaves (terminal nodes) of the tree. Functions of maximum, addition, multiplication, and subtraction are arithmetic operations that constitute the interior nodes of the tree and form the function set. Any possible combination of variables, constants, and arithmetic operations create an individual of the GP population. The depth of the tree is determined by the depth of its deepest node, which is equal to 3 in this example.

The evolution of populations in GP is illustrated in Figure 2. Six principal steps are involved in this process:

1. An initial and usually random population of individuals is generated using the defined function and terminal sets. Different approaches, such as full, grow or ramped half-and-half methods can be used for this purpose [9].
2. The fitness of each individual is determined using a predefined fitness function [29]. These functions are discussed in the next section.
3. One or two individuals of the population are selected to participate in genetic operations (see the next step). Individuals with better fitness are more likely to be chosen for this

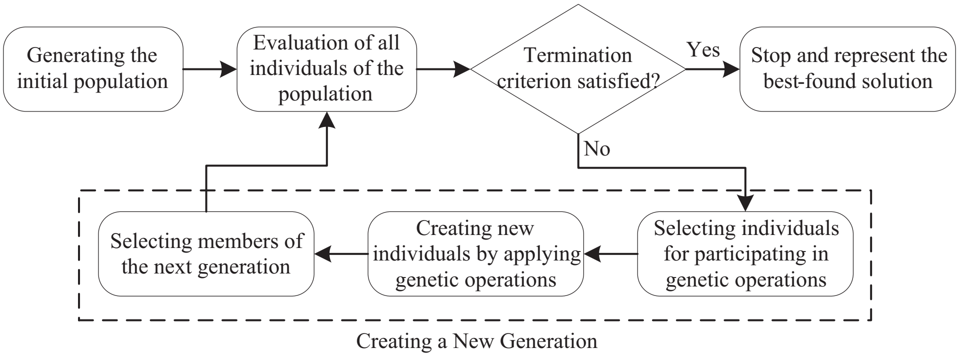
Generating the initial population
Evaluation of all individuals of the population
Termination criterion satisfied?
Yes
Stop and represent the best-found solution
No
Selecting individuals for participating in genetic operations
Creating new individuals by applying genetic operations
Selecting members of the next generation
Creating a New Generation

FIGURE 2 Flowchart of the genetic programming (GP) process

participation, and have a higher chance of transferring their genetic information to the next generation [10].

4. Genetic operations of crossover, mutation, and reproduction are applied to the selected individuals of the previous step to create the new individuals (offsprings). Generally, crossover swaps the randomly chosen parts of the two selected individuals; mutation randomly changes a small part of the selected individual; and reproduction copies the selected individual [10, 29, 30].
5. A specific number of the newly created offspring are selected based on their fitness to go to the next generation.
6. If a termination criterion is satisfied (such as reaching a specific number of generations or reaching a predefined fitness level), the evolutionary process stops, and the best-found individual is represented as the output of the process; otherwise, the process goes back to Step 2.

2.2.2 | Applications in classification

GP has successfully been applied to many different fields, including classification [10, 11, 31]. In classification problems, each instance is represented by a number of features. The role of classifiers is to predict the class of each instance by learning from the training data. The original features are directly entered into the classifier, and the process of training and prediction is performed based on these original features. Previous studies have indicated that the representation of features has a great impact on the performance of classifiers [8, 32]. Many classifiers (such as decision trees or decision rules) do not perform well due to their limited ability in transforming initial features into a suitable form. This issue is especially prevalent in symbolic classification [33]. In contrast, classifiers such as SVMs and neural networks can have a better performance in discriminating different classes by transforming the input features in their internal structure into high-level features. However, such high-level features are difficult to be explicitly expressed and the models have a high risk of over-fitting [7, 33].

When the original features do not have a desirable performance, it is essential to perform a feature construction or transformation step to improve the performance of symbolic classifiers [8]. In this step, the representation of the original features is changed, so that a set of new high-level features replaces the original ones. In this process, new features are derived from the original ones in a way that they would have a better performance than the original features. The process of creating a new representation of features is a non-deterministic polynomial-time (NP)-hard problem; therefore, finding an exact solution is not justifiable [33]. Metaheuristic methods, such as evolutionary computation techniques (e.g. GP), are used for the purpose of finding suboptimal solutions in a reasonable time [33]. GP has unique capabilities for feature transformation, including feature selection and construction, that can improve the performance of classifiers, specifically the symbolic ones, while maintaining their interpretability [6, 8, 11, 33]. Feature selection can be regarded as a special case of feature construction as there is usually no limitation in cloning original features in the feature construction process.

Two different approaches can be implemented for feature construction using GP from a functional (defining the fitness function) viewpoint [10]:

1. Filter or non-wrapper approach: In this approach, separate measures, such as statistical, logical, or information-content principles, are defined as the fitness function of GP [10]. The filter approach, compared to the wrapper approach, usually requires less computational time for evaluation of individuals' fitness and has usually the advantage of constructing very general features, since the classification is not dependent on any external classifier [6]. However, features are constructed based on a measure that may not be efficient for the classifier that is finally selected for classification [33].
2. Wrapper approach: The fitness function of GP in this approach is a performance measure of the same classifier that is eventually used for classification–in other words, a particular classifier is wrapped in the fitness function. Thus, for the evaluation of each individual, a model is developed by the classifier to measure its performance. Although this approach usually requires a higher computational effort, constructed features usually have a better performance compared to those of a filter approach [30].

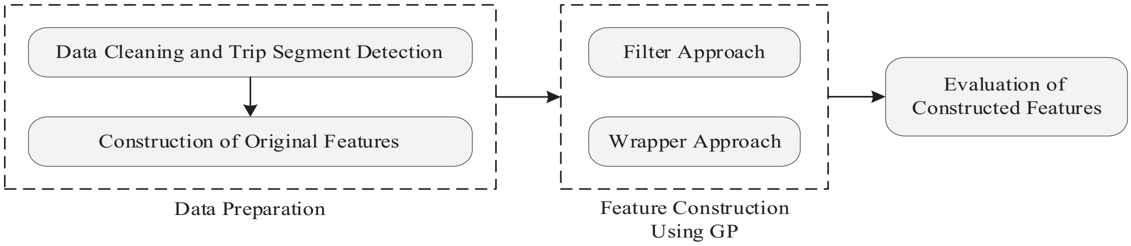
Data Cleaning and Trip Segment Detection
Construction of Original Features
Data Preparation
Filter Approach
Wrapper Approach
Feature Construction Using GP
Evaluation of Constructed Features

FIGURE 3 Overview of the methodology

Single or multiple features can be constructed as the output of the GP using both approaches. When constructing a single feature, each individual is a single high-level feature represented by a single tree [30]. Different methods have been used for multiple feature construction. For instance, each individual may consist of multiple trees, while each tree is representative of one feature. Therefore, the collection of these trees in one individual constitutes multiple features [32, 33]. Alternatively, each individual may represent a single feature, and multiple features are constructed by implementing GP multiple times [6]. As another option, after the completion of a GP run and construction of a single high-level feature, other feature sets are extracted from that single feature [30]. For example, in [34], each individual is considered as a single tree, while all possible subtrees of this tree are considered as constructed features. Hence, the output of this GP can be multiple features that are displayed in the form of a single tree by an individual. These methods are implemented and compared in the following sections. An expanded review of different feature construction methods using GP can be found in [8].

3 | METHODOLOGY

In this section, the steps of data preparation are presented first, followed by the details of the proposed approaches for feature construction. The overview of this process is illustrated in Figure 3.

3.1 | Data preparation

GPS data contain a set of GPS points that are sorted in chronological order. In the simplest case, each GPS point includes latitude, longitude, timestamp, and a trip mode label (e.g. car, train, or bus) given that the data is used for supervised learning.

3.1.1 | Data cleaning and trip segment detection

If more than one GPS point is recorded with the same timestamp, duplicate points are excluded. Reviewing the range of considered thresholds for dwell time in the literature [4, 14], if the time interval between two consecutive GPS points is more than 150 s, the trip is split into two segments at that point. A maximum length of 40 km is considered for segments; the first 40 km of a long segment is taken as an independent segment and breaking stops when the remaining part is less than 40 km itself.

To decrease the impact of random variations caused by sources, such as urban canyons or weak signal strength, smoothing techniques are applied. Data smoothing is essential to increase mode detection accuracy in future steps. In this study, latitudes and longitudes are smoothed separately using a modified version of the moving average method as follows: for each point, the average of the three preceding and the three succeeding points is computed, if their temporal distance from the point of interest is less than 60 s. When less than three points exist in the 60 s interval, equal numbers of preceding and succeeding points are used for averaging. The average of the value at the point of interest and the calculated mean, substitute for the current value. If none of the adjacent points meet the temporal condition, the point's value remains unchanged. This process is presented in Algorithm 1.

Algorithm 1 Data Smoothing

```
$n$:    number of GPS points of a segment
        $t_i$: timestamp of point $i$
        $P_i$: unsmoothed coordinate (latitude or longitude) of point $i$
        $S_i$: smoothed coordinate (latitude or longitude) of point $i$
for $i \leftarrow 0$ to $n - 1$ do
    $sum = 0$
    $count = 0$
    for $j \leftarrow 1$ to $3$ do
        if $(i - j \geq 0)$ & $(i + j < n)$ then
            $t_{before} = t_i - t_{i-j}$
            $t_{after} = t_{i+j} - t_i$
            if $(t_{before} < 60)$ & $(t_{after} < 60)$ then
                $sum = sum + P_{i-j} + P_{i+j}$
                $count = count + 2$
            endif
        endif
    endfor
    if $count > 0$ then
        $S_i = 0.5(\frac{sum}{count}) + 0.5(P_i)$
    else
        $S_i = P_i$
    endif
endfor
```

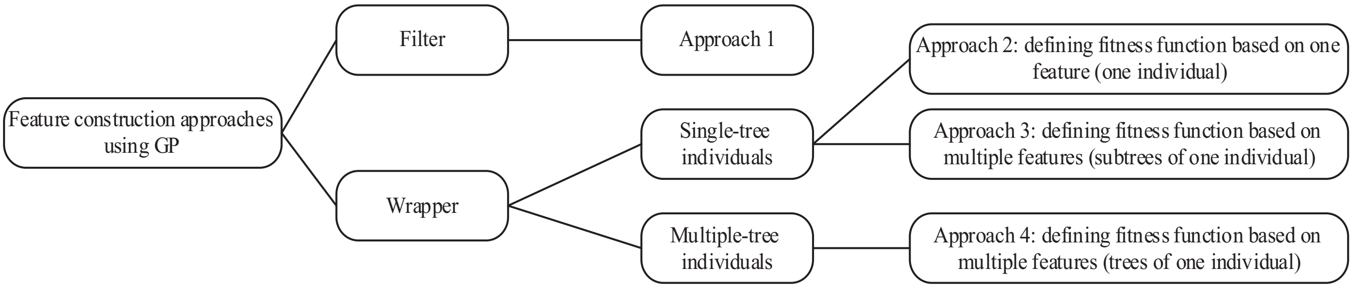
Feature construction approaches using GP
Filter
Wrapper
Approach 1
Single-tree individuals
Multiple-tree individuals
Approach 2: defining fitness function based on one feature (one individual)
Approach 3: defining fitness function based on multiple features (subtrees of one individual)
Approach 4: defining fitness function based on multiple features (trees of one individual)

FIGURE 4 Feature construction approaches

TABLE 1 Maximum velocity and acceleration of different transportation modes

Travel mode	Maximum velocity (m/s)	Maximum acceleration (m/s^2)
Walk	7	3
Bike	10	3
Bus	35	10
Car	50	10
Train	75	3

After data smoothing, points with a velocity or acceleration above an acceptable threshold are removed. The maximum velocities and accelerations of different transportation modes considered for this purpose are extracted from different sources, such as [1] and [19], and presented in Table 1. Segments with less than 10 GPS points or 30 m are also removed.

3.1.2 | Construction of original features

After data cleaning, the original features are calculated. The features proposed by Zheng et al. [24, 25]–used frequently by previous research [1, 5, 26, 27]–are used as the original features in this study. These original features are later used for the construction of high-level features by the GP. These 14 features, all calculated at the segment level, include length, time duration, mean velocity (length of the segment divided by its time duration), expectation of velocity (average of velocities of the GPS points in a segment), velocity variance (variance of velocities of the GPS points in a segment), top three velocities, top three accelerations, heading change rate (number of GPS points of a segment with a heading change of more than a specific threshold divided by the segment length), stop rate (number of GPS points of a segment with a velocity of less than a specific threshold divided by the segment length), and velocity change rate (number of GPS points of a segment with a velocity rate of more than a specific threshold divided by the segment length).

The distance between two consecutive GPS points is calculated using Vincenty's formula [35] in this study, and the sum of consecutive calculated distances is considered as the segment length. Expanded definitions and details of other features can be found in [24] and [25].

3.2 | Feature construction using GP

As depicted in Figure 4, a filter approach and three wrapper approaches are evaluated in this study. The wrapper approaches are divided into two different groups based on the structure of each individual in the GP population. In the first group, each individual is a single tree, while in the second group, an individual consists of multiple trees. The details of each approach are explained next.

3.2.1 | Approach 1

The method proposed by Neshatian et al. [6] is used in Approach 1. In this method, the output of every GP run is one high-level feature constructed for a particular class of the classification problem. Thus, for a mode detection problem with n distinct modes, n high-level features will be constructed.

Each individual in this approach is a tree, which represents a high-level feature. Each tree (feature) that is constructed during the GP process is a function that its value is determined based on the original features. Thus, the values of the high-level features can be calculated for the training instances. The fitness function in Approach 1 is defined based on the concept of information entropy; it minimizes the impurity of the desired class (the class for which the GP is run to construct a high-level feature). To determine the impurity, the "class interval" over the high-level feature should be calculated first. The class interval is the smallest interval that covers a given portion of all high-level feature values for that class (i.e., excluding extremely high or low values). The impurity of an interval (individual's fitness) is equal to the number of instances of undesired classes that are in the desired class interval. The lower this number, the better and purer the desired class interval over the high-level feature will be. More details of the algorithm can be found in [6].

The GP settings used for the implementation of this approach are presented in Table 2, which were determined based on [28] and [6]. The function set includes four basic arithmetic operators. The division operator is protected, which means it returns zero if the denominator is zero. All operators

TABLE 2 GP settings

Function set	$+, -, \times, \div$(protected division)
Variable terminals	Original features ($x_1, x_2, \ldots, x_{14}$)
Constant terminals	Random float values ranging from -10 to $+10$
Population size (mu)	1000
Number of generated offsprings for each generation (lambda)	$2 \times mu = 2000$
Number of generations	50
Initialization	Ramped half-and-half
Selection method	Tournament (size = 7)
Crossover method	One-point
Mutation method	Uniform (random subtree creation)
Maximum tree depth	10
Mutation probability	30%
Crossover probability	60%
Reproduction probability	10%

have two arities/arguments. The terminal set contains the 14 original features and a random constant float value. To provide a suitable search space, the population size and the number of features should be proportionate [30]. Since the number of features of this study is in the same order as Neshatian et al. [6], the population size is set to 1000. The number of generated offspring for each generation is assumed to be twice the size of the population (mu). To prevent bloating [28, 36], the maximum tree depth is set to 10. The GP process is terminated when a maximum number of 50 generations is reached, which always converged in the conducted experiments. Mutation and crossover probabilities were selected by comparing the results of different combinations of multipliers of 10 for these probabilities. The best results were obtained when mutation and crossover probabilities were set to 30% and 60%, respectively.

Based on the constructed high-level features, two feature sets are created: (1) the first one includes n high-level features for n travel modes and (2) the second one includes n high-level features and the 14 original features.

3.2.2 | Approach 2

In the wrapper approach, individuals are evaluated using an external classifier. A decision tree is selected for this purpose. In Approach 2, each individual is a GP tree that represents a high-level feature (F'_0). The GP tree is used to construct a high-level feature, which is then used in the decision tree for classification. Thus, the two trees (one in the structure of GP and one as the external classifier) have different functionalities. For each individual, the value of the high-level feature is calculated. Thus, the number of features is reduced from 14 to 1 in the projected data. For evaluation of each GP individual, the classification accuracy is determined by a decision tree using five-fold cross-validation. This process is illustrated in Figure 5.

To choose a suitable measure for evaluation of GP individuals, macro- and micro-averages of both $F1$-score and area under the receiver operating curve (ROC), which is denoted by AUC (Area Under Curve) [37], are investigated, all of which can be determined using a decision tree. In the macro-average, the desired measure is computed for each class and then averaged regardless of the number of instances in each class, while the micro-average is determined by calculating the average over all instances. The macro- and micro-averages can result in different values if one class is dominant (i.e. one travel mode). After the selection of a suitable measure, the impact of maximum tree depth in GP settings is studied by reducing it from 10 to four. Other GP settings are similar to those in Table 2.

A significant reduction is achieved in the complexity of the problem by reducing the number of features from 14 (original) to one (high-level), although the one high-level feature may not have the same performance as the original features. Therefore, after the completion of the GP process (termination of 50 generations), six different feature sets introduced by Tran et al. [30] are constructed. Figure 6 shows an example of a constructed high-level feature; original features of F_1, F_3, and F_4 were used to construct this feature. The six feature sets for this example are specified next.

Set 1: The single high-level feature, which is $F'_0 = (F_1 - F_4) \times (F_3 + F_1)$.

Set 2: Original features and the high-level features, which are $\{F_1, F_2, \ldots, F_{14}, F'_0\}$.

Set 3: Original features that appeared in the terminal nodes of the high-level feature, which are $\{F_1, F_3, F_4\}$.

Set 4: Combination of sets 1 and 3, which are $\{F'_0, F_1, F_3, F_4\}$.

Set 5: All possible subtrees of the high-level feature, which are $\{F'_0, F'_1, F'_2\}$, where $F'_1 = (F_1 - F_4)$, and $F'_2 = (F_3 + F_1)$.

Set 6: Combination of sets 3 and 5, which are $\{F_1, F_3, F_4, F'_0, F'_1, F'_2\}$.

After constructing these feature sets, their performance is evaluated using a decision tree. The advantage of Approach 2 is its simplicity and lower execution time, compared to Approach 3 and 4 since only a single feature is constructed and evaluated during the GP process, and feature sets are constructed after the completion of the GP process.

3.2.3 | Approach 3

Similar to the previous approaches, each individual is a tree in Approach 3. However, for the evaluation of each individual, as proposed by Ahmed et al. [34], all possible subtrees of that individual are specified and included in the feature set of the decision tree, as shown in Figure 5. Approach 3 has a higher execution time compared to Approach 2 because of its higher computational complexity. To compare the approaches at a relatively similar execution time, the population size and the number of generations are reduced to 100 and 30, respectively. The maximum tree depth in GP settings is also set to four. Other settings are similar to those in Table 2.

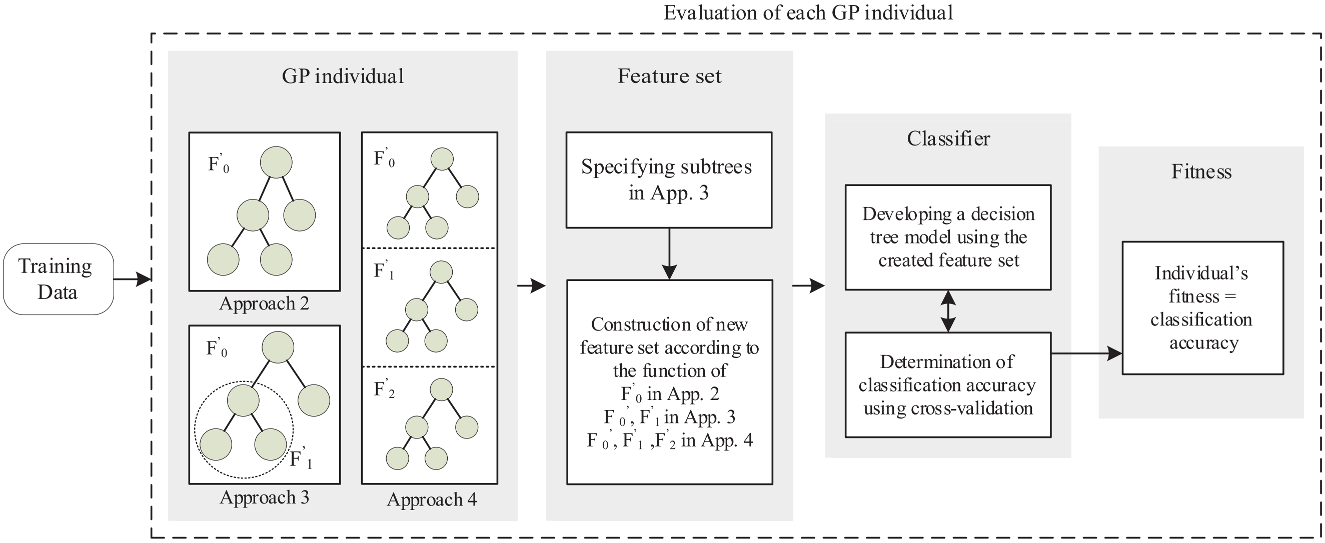
Evaluation of each GP individual
Training Data
GP individual
F`0
Approach 2
F`0
F`1
Approach 3
F`0
F`1
F`2
Approach 4
Feature set
Specifying subtrees in App. 3
Construction of new feature set according to the function of
F`0 in App. 2
F`0, F`1 in App. 3
F`0, F`1, F`2 in App. 4
Classifier
Developing a decision tree model using the created feature set
Determination of classification accuracy using cross-validation
Fitness
Individual's fitness = classification accuracy

FIGURE 5 GP individual evaluation process in approaches 2 to 4

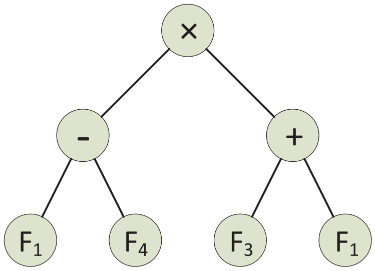

FIGURE 6 An example of a high-level feature

The output of Approach 3 is a high-level feature in which all of its subtrees were evaluated as a feature set of a decision tree. After the completion of the GP process, the six feature sets introduced in the previous section are constructed based on the best-generated individual. While in Approach 2, feature set 1 of these six sets has already been evaluated during the GP process, in Approach 3, set 5 has already been evaluated. The advantage of Approach 3 compared to Approaches 1 and 2 is that the performance of a group of features (e.g., F'_0, F'_1) is evaluated in the GP process. A single feature may have a good performance to identify a class in Approach 1; however, their collation (a set of separately developed features for each class) may not perform well. Approach 3 finds a group of features that have a good performance in tandem.

3.2.4 | Approach 4

In this approach, each individual consists of multiple trees, and each tree represents a single high-level feature. Therefore, each individual constitutes multiple high-level features. The number of trees of each individual is predetermined. For mutation, one of the trees of a given individual is chosen randomly; then a random subtree of this tree is replaced by another randomly generated subtree. In the crossover, two individuals are selected, and one subtree of a randomly chosen tree is swapped between individuals. This process is shown in Figure 7.

The evaluation of individuals is shown in Figure 5. Similar to Approach 3, the performance of a group of features is evaluated in this approach; however, features of each individual have a higher degree of freedom to evolve since these features are independent of each other and can evolve separately specifically to improve their collective performance. The execution time is longer for Approach 4 because of its higher complexity compared to Approaches 1 and 2. To enable comparability of the approaches at a relatively similar execution time, the maximum tree depth in the GP settings and the population size are decreased to 4 and 500, respectively. The number of trees of each individual can also be adjusted. It is set once to the number of classes of the classification problem and is set to twice this value at another time for sensitivity analysis.

4 | RESULTS

The data preparation steps were coded in Java. GP was implemented using the DEAP library in Python [38], and the Scikit-learn library in Python [39] was used for developing classification models. Eighty per cent of the data was used for training to construct features using GP, and the remaining was later used for testing. The results reported next are based on the test data. A decision tree classifier was used for the final evaluation of constructed feature sets in all approaches. The GP processes were run on a system with a Core i7-4790 CPU @ 3.60 GHz and 16 GB of memory.

4.1 | Data preparation

To implement and evaluate the proposed methodology, the GeoLife dataset [12] collected by 64 users mostly in China was used. This dataset includes trips performed by different modes

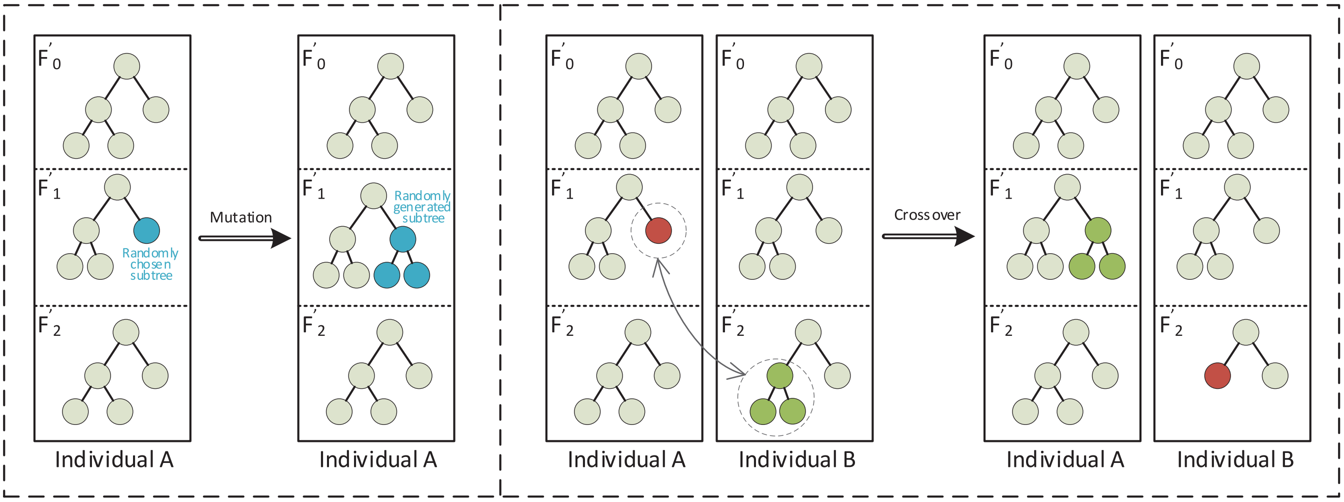

FIGURE 7 Mutation and crossover operations in Approach 4

TABLE 3 Distribution of segments in travel modes

Travel mode	Walk	Bike	Bus	Car	Train
Number of segments	6395	2142	2787	2165	1834

of transportation, among which the five land transportation modes of walk, bike, bus, car, and train are considered in this study. Both the car and taxi modes are considered as cars, and all rail-based modes (light rail, subway and train) are considered as trains in this research. After applying all the preparation steps described in Section 3.1, 15,323 segments were obtained as presented in Table 3.

4.2 | Feature construction using GP

The process of feature construction in each approach was repeated 30 times using a set of 30 different random seeds because of the stochastic behaviour of GP. The evaluation of GP individuals in the population was performed in parallel using six processor cores to minimize the execution time.

4.2.1 | Approach 1

The test results in terms of the macro-average of $F1$-score (which will be discussed in the next section) are reported in Table 4. The optimal value of maximum depth of the decision tree was determined by applying five-fold cross-validation to the training data. The number of features in each set is shown in column "#F". The best, average, and standard deviation of 30 sets of results is presented in Table 4, while each set was generated using a different random seed for feature construction. The results corresponding to the application of 14 original features as the input of the decision tree are used as the baseline for comparison. To compare the results of different GP approaches with the original features, the t-test has been conducted. The t values and their statistical significance according to one tail t-distribution are reported in the table.

As shown in Table 4, the performance of feature set 1 could not reach the performance of original features, which indicates that developing a number of features separately may not result in a good performance. In set 2 of Approach 1, the mean value was slightly improved compared to the original set, after adding five constructed high-level features to the original features. Given the small improvement in the $F1$ value, more approaches are investigated in the following to achieve better results.

4.2.2 | Approach 2

The results of using different performance measures as the fitness criterion are presented in Table 5. Regardless of the performance measure being tested, the values of other performance measures were also calculated and reported in this table.

The average values of all performance measures were less when the AUC (either macro- or micro-averages) was used as the GP fitness criterion rather than the $F1$-score. Therefore, the $F1$-score had a better performance than the AUC. Since no considerable differences between the results of the macro- and micro-average of $F1$ were found, the best fitness measure was selected based on the concept of these two measures. In many transportation systems, the mode share is unbalanced (in the GeoLife dataset, for example, walking constitutes 42% of the trip segments). Thus, the macro-average that assigns equal weights to different classes/modes can result in a model that can predict all travel modes with an acceptable level of accuracy and is not biased towards the dominant mode. Therefore, the macro-average of $F1$ was used as the GP fitness measure in Approaches 2–4.

The best and average values of different performance measures by a single feature were less than their respective values in the original features, which demonstrates that a single high-level

TABLE 4 GP settings and test results of different approaches

				F1 macro								
Approach	**Maximum tree depth**	**Set**	***#F***	**Best**	**Mean**	**SD**	***t***	**Sig.**	**Mu**	**Lambda**	**No. of generations**	**Execution time (min)**
		Original	**14**		**77.3**							
1	10	1	5	75.5	72.5	1.66	–		1000	2000	50	69
		2	19	78.6	77.9	0.37	8.88	***				
2	10	1	1	73.6	71.0	1.20	–		1000	2000	50	44
		2	15	78.9	77.9	0.43	7.64	***				
		3	9.6	78.8	77.4	1.02	0.54					
		4	10.6	79.0	77.6	0.85	1.93	*				
		5	25.1	77.8	75.6	1.16	–					
		6	34.8	79.1	77.4	0.90	0.61					
	4	1	1	72.6	70.1	0.99	–		1000	2000	50	40
		2	15	78.6	77.7	0.43	5.10	***				
		3	6.4	77.9	76.2	1.08	–					
		4	7.4	78.8	76.3	1.20	–					
		5	9.3	78.0	74.5	1.51	–					
		6	15.8	79.2	76.4	1.37	–					
3	4	1	1	62.0	46.1	11.03	–		100	200	30	172
		2	15	78.7	77.9	0.38	8.65	***				
		3	8.6	79.0	77.7	0.56	3.91	***				
		4	9.6	79.2	77.9	0.60	5.48	***				
		5	13.5	79.3	78.0	0.72	5.33	***				
		6	22.1	79.4	78.1	0.53	8.27	***				
4	4	1	5	79.3	77.5	0.72	1.52		500	1000	50	66
		1	10	**80.0**	**78.5**	**0.74**	8.88	***	500	1500	50	176

*Significance at 95% confidence level.

***Significance at 99.9% confidence level.

TABLE 5 Test results of approach 2 to determine GP fitness criterion

			AUC macro			AUC micro			F1 macro			F1 micro		
Performance measure used as the GP fitness criterion	**Set**	***#F***	**Best**	**Avg**	**SD**	**Best**	**Avg**	**SD**	**Best**	**Avg**	**SD**	**Best**	**Avg**	**SD**
	original	14		92.3			94.3			77.3			81.5	
AUC macro	set 1	1	89.7	88.4	0.56	92.7	91.8	0.48	67.9	64.6	1.62	74.3	72.1	1.07
AUC micro	set 1	1	88.9	88.1	0.58	92.4	91.9	0.27	65.2	62.9	2.34	72.8	71.3	0.93
F1 macro	set 1	1	90.0	89.2	0.75	93.0	92.2	0.70	73.6	71.0	1.20	78.6	76.4	0.99
F1 micro	set 1	1	90.3	89.5	0.41	93.1	92.6	0.31	73.0	71.2	1.21	78.2	76.9	0.95

feature could not have a similar performance to the 14 original ones.

To investigate the impact of maximum tree depth in GP settings on the performance of constructed features, particularly in feature selection, the maximum depth was set to ten and four. The results of the six feature sets for these maximum depths are shown in Table 4. Overall, set 2 had a better performance compared to the other sets. Although this set achieved similar results to set 2 of Approach 1, it contained a smaller number of features, which means it performed better than Approach 1. However, the improvement in performance in this set was still not considerable compared to the original set.

The average values of *F*1 decreased in all sets by reducing the maximum tree depth from 10 to four, and the standard deviations increased in almost all sets. Nevertheless, the average number of features in sets 3 to 6 was also significantly decreased, which demonstrates the effectiveness of reducing maximum tree depth in feature selection by choosing features that play a

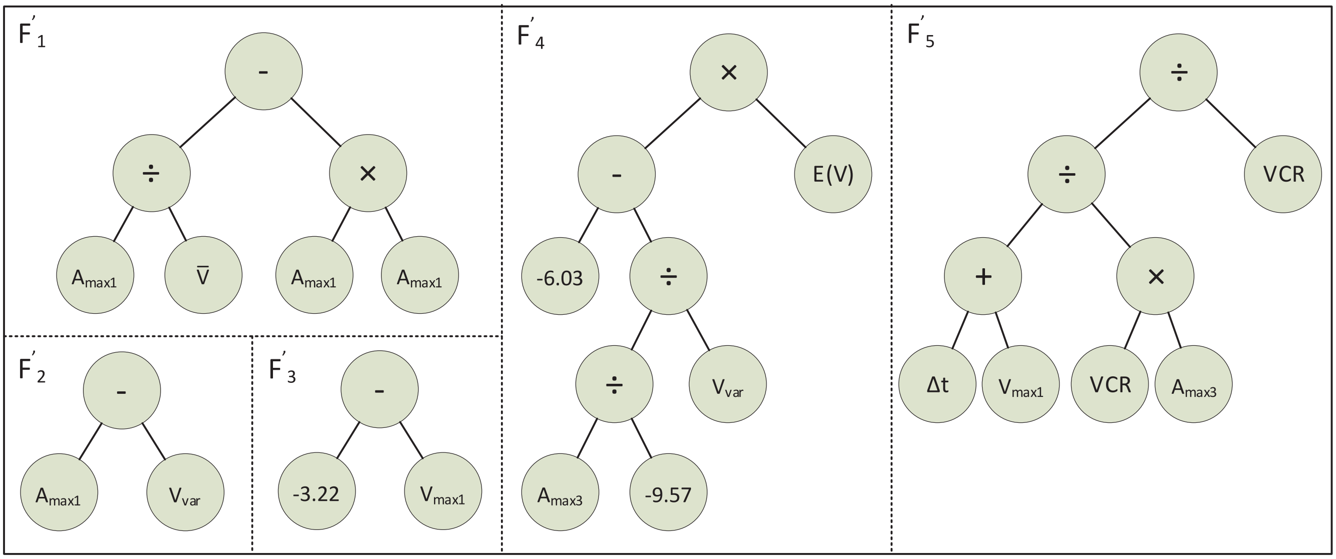

FIGURE 8 An example of constructed feature set using Approach 4

critical role in detecting travel mode. Therefore, the maximum tree depth can be determined based on the factor that has a higher priority (achieving a higher accuracy or constructing simpler models with less execution time).

4.2.3 | Approach 3

The results of Approach 3 are presented in Table 4. Except for set 1, all values of $F1$ were higher than those of the original set. Although the maximum tree depth was 4, the average and the best values of $F1$ scores (of sets 2 to 6) were increased and the standard deviations were decreased compared to Approach 2 at a maximum depth of both 10 and 4. The average number of features in sets 3 to 6 was slightly increased compared to Approach 2 at a maximum depth of 4; while these sets produced better results and had an acceptable performance in feature selection. In set 3, for instance, the six features of length, expectation of velocity, velocity variance, first maximum velocity, first maximum acceleration, and velocity change rate produced values of 78.4 and 82.4 for the macro- and micro-averages of $F1$, respectively.

4.2.4 | Approach 4

In Approach 4, only feature set 1, which contained the high-level features of an individual, was created. The number of high-level features was set to either five (equal to the number of travel modes) or ten for sensitivity analysis. The results are presented in Table 4.

In the first case, the five high-level features could achieve the performance of the 14 original features. The superiority of this approach is clearly evident over Approach 1 in which the five high-level features had a lower performance compared to the original features; while the maximum depth of each tree (feature) was set to ten in Approach 1 and four in Approach 4. The superiority of Approach 4 compared to Approach 1 is because the five features in the latter were constructed separately, while a group of five features were constructed simultaneously in the former. An example of evolved GP individual by this approach is presented in Figure 8. This individual consists of five trees representing five high-level features. This feature set produced the value of 79.3 for the macro average of $F1$, when used as the input of the decision tree. In this figure, A_{max1}, $\bar{V}$, V_{var}, V_{max1}, A_{max3}, $E(V)$,Δt, and VCR represent first maximum acceleration, mean velocity, velocity variance, first maximum velocity, third maximum acceleration, expectation of velocity, time duration, and velocity change rate, respectively.

The results of constructing ten features in Approach 4 were significantly better than the original set (based on the t-test at a significance level of 0.05) as well as the other approaches. The best high-level feature set yielded an increase in the value of macro-average of F1 by more than 2.5% compared to the original set. The mean execution time of constructing features for a single random seed in each approach is reported in Table 4. Approaches 3 and 4 were the slowest approaches despite the smaller population size and number of generations in Approach 3, reflecting the higher computational complexity of Approach 3 compared to the other approaches. Overall, Approach 4 performed better in constructing new features and Approach 3 in feature selection (see set 3 of Approach 3).

The best high-level feature set (the 'best' set of Approach 4 with 10 features) was used as the input of a decision tree, and the confusion matrix was calculated, as presented in Table 6. The optimal value of maximum depth of the decision tree was identified by performing a five-fold cross-validation on the training data, and the performance of the developed model was evaluated on the test data. As shown in Table 6, walk and bike modes had the highest $F1$ values, while the car had the lowest. The

TABLE 6 The confusion matrix, recall, precision, and F1-score for the best-constructed feature set of approach 4

		Predicted class						
		Walk	Bike	Bus	Car	Train	Sum	Recall (%)
Actual class	Walk	1207	22	5	4	6	1244	97.0
	Bike	48	381	11	6	1	447	85.2
	Bus	56	14	394	71	21	556	70.9
	Car	31	6	75	285	23	420	67.9
	Train	38	7	17	48	288	398	72.4
	Sum	1380	430	502	414	339	3065	
	Precision (%)	87.5	88.6	78.5	68.8	85.0		
	F1-score (%)	92.0	86.9	74.5	68.6	78.2		

TABLE 7 Comparison of GP system performance with SVM, MLP and DT

Model	HP range	Optimal HP	$F1$-score	Accuracy
DT	[5–20]	8	77.3	81.5
MLP	[1–10]	2	80.7	84.0
SVM	[0.1, 1, 10, 100, 1000]	100	81.0	83.9
DT[GP]	[5–20]	10	80.0	83.4

TABLE 8 Comparison with relevant studies

Study	Best model	DT	MLP	SVM
Endo et al. [26]	67.9	–	–	–
Wang et al. [5]	74.1	68.9	–	52.3
Dabiri and Heaslip [1]	84.8	75.2	59.4	65.4
Nawaz et al. [27]	83.8	–	57.1	66.2
This study	83.4	81.5	84.1	83.9

largest prediction error for cars corresponds to identifying segments as bus and vice versa. In addition to the typical similarities of bus and car segment characteristics, taxis compared to private cars have more stops to pick-up or drop-off passengers, which makes them even more similar to buses that have regular stops at stations, and can lead to a false prediction. Separating car and taxi labels by providing more instances of them can be an effective way to reduce this error.

4.3 | Comparison of GP with other classification algorithms

SVMs and neural networks are both capable of constructing high-level features, which makes them similar to the proposed GP feature construction approach. Therefore, the best results of the proposed GP approaches are compared with results of optimized SVM and neural networks to provide a better perspective of the GP performance. In addition to SVM and MLP, a standalone DT (i.e. without being combined with a GP) is applied. The 14 original features were used as the input of these models. The SVM was implemented with a radial basis function, and its regularization parameter (c) was optimized by five-fold cross-validation on the training set. For the gamma coefficient, the default value of 'scale' in the Scikit-learn library was used. For the neural network, a multilayer perceptron (MLP) was implemented in which the optimal number of hidden layers was determined by five-fold cross-validation. Following Dabiri and Heaslip [1], the number of nodes in each hidden layer was set to twice the number of nodes of its previous layer. For DT, the maximum tree depth was optimized. The range of hyperparameters and their optimal values are provided in Table 7. The test results are presented in terms of macro-average of $F1$-score, and the micro-average of $F1$ (which is equal to the measure of 'accuracy' in a multiclass classification with each instance assigned to exactly one class). The measures of recall and precision are not shown to keep the results concise. Since SVM and MLP models are not scale-invariant, data were first standardized using the z-score for these two models so that each feature would have a zero mean and a unit variance [40]. The last row of the table presents the results of using the best-constructed feature set by the GP as the input of a decision tree in the proposed approach (DT[GP]).

It is important to repeat that SVM and MLP parameters are optimized to produce their best results. The performance of optimized SVM, MLP, and DT[GP] are very similar to each other, while the performance improvement of the standalone DT using the constructed features by a GP is evident. These results indicate that GP could significantly improve the performance of a simple classifier, such as a DT, and while maintaining simplicity, it could produce competitive results with classifiers that are able to construct high-level features intrinsically.

4.4 | Comparison with relevant studies

The results of this research were also compared with a number of studies that used the GeoLife dataset. These studies include Nawaz et al. [27] that used a convolutional long short-term memory (CLSTM) and Dabiri and Heaslip [1] that employed a convolutional neural network (CNN) to infer travel mode, Wang et al. [5] and Endo et al. [26] that applied deep learning to construct high-level features. Although the definition of segments, and consequently, the level of aggregation differs in these studies, because of using the same dataset it is possible to have a general comparison of their performance. Since the measure of accuracy is common in all of these studies, their highest reported accuracy along with other provided classification algorithm results are presented in Table 8. These results are directly taken from the original papers.

According to Table 8, Dabiri and Heaslip [1] achieved the highest accuracy; however, similar to Nawaz et al. [27], there is a considerable difference among the results of different classification algorithms in their study. This bias could be the

TABLE 9 Distribution of segments among different travel modes in Australia and New Zealand dataset

Travel mode	Walk	Bike	Bus	Car
Number of segments	719	65	470	2578

TABLE 10 Test results of Australia and New Zealand dataset

		*F*1 macro			
Set	#F	Best	Mean	SD	GP execution time (min)
Original	14		70.9		–
1 (Approach 4)	8	**75.6**	**72.3**	**2.17**	36

consequence of limiting the segments to have equal sizes since equal sizes was a requirement for their utilized CNN architecture; thus, the performance of DT, MLP, and SVM was adversely affected. The results of the present research (the last row of Table 8) demonstrate that data preparation steps have a significant impact on the results, and similar outcomes can be obtained through different algorithms. Not applying data standardisation before feeding data into the SVM and MLP models could be another reason for the observed differences among outcomes of classical classification algorithms in Nawaz et al. [27], Dabiri and Heaslip [1] and Wang et al. [5].

By using GP to extract high-level features for a DT, the overall accuracy became similar to those of sophisticated models, such as CNN and CLSTM. Notedly, this competitive result is achieved using a classifier that is not only simpler, but also has a higher level of interpretability, and a lower risk of over-fitting. Therefore, the proposed model can be used even on smartphones with limited computational capacities in order to passively collect travel data with a high level of accuracy for mode detection.

4.5 | Validation of the proposed method

To validate the proposed approach and the best-constructed features, the approach is applied to another dataset collected in Australia and New Zealand [13,14]. The New Zealand data was collected in 2014 corresponding to 76 users, and the Australian data was collected in 2014 corresponding to 99 users. Both datasets contain four travel modes of walk, bike, bus, and car. After applying data preparation steps according to Section 3.1, 3832 segments were produced, which were distributed among different modes as illustrated in Table 9.

Approach 4 in constructing features using the GP, which produced the best results in the GeoLife dataset, was applied to this dataset by setting the number of trees in each individual to twice the number of travel modes (i.e. eight). Because of the strongly unbalanced distribution of data among different transportation modes, the macro-average of *F*1 was again used as the fitness measure of individuals. Other settings were similar to the previous sections. Table 10 compares the test results of using the constructed features by GP through 30 different random seeds as the input of decision tree with results of using the 14 original features. The results are again significantly better than the original set according to the t-test at a significance level of 0.05.

TABLE 11 Comparison of the performance of the developed models for the Geolife dataset on the Australia and New Zealand dataset

Model	#F	*F*1-score	Accuracy
DT	14	59.1	57.1
MLP	14	31.0	24.1
SVM	14	43.4	47.1
DT[GP]	10	61.9	62.9

To investigate the transferability of different models, Table 11 compares the performance of the developed models for the GeoLife dataset on the Australia and New Zealand data. The 14 original features were used as the input of DT, MLP, and SVM, and the best-constructed features for the GeoLife dataset were used as the input of DT[GP]. In this validation stage, the developed models of Table 7 are applied to a totally new dataset (the Australia and New Zealand data). Consequently, the results of these models on the new dataset are not expected to be comparable with the best-found results in Table 7 or Table 10, where features and models were exclusively constructed and developed for the datasets. According to Table 11, the results of DT and DT[GP] were considerably better than MLP and SVM, which indicates that a DT is much less prone to over-fitting. DT[GP] outperformed the other models and achieved the best results in terms of both measures of *F*1-score and accuracy.

5 | CONCLUSIONS

In this paper, the possibility of using GP as a method for feature selection and construction of high-level features was investigated for travel mode detection. Feature selection and construction can be used to improve the performance of classification algorithms, such as a DT, which are not capable of changing the representation of inputs in the model development process. The GeoLife dataset [12] and the 14 original features proposed by Zheng et al. [24, 25] were employed for this purpose.

After applying data preparation steps, four different approaches for feature construction using GP, including one filter and three wrapper approaches were examined. A DT was used to evaluate GP individuals in wrapper approaches, while different fitness measures including macro- and micro-averages of both F1-score and the area under the ROC curve were investigated. Feature selection as a special case of feature construction was also investigated through these approaches. The performance of high-level feature sets was also evaluated by a decision tree.

Overall, the wrapper approaches (Approaches 2–4) had a better performance compared to the filter approach. Based on the results of Approach 2, the macro-average of F1 was selected as the fitness measure. Approach 3 was successful in feature selection; it needed less computational effort by reducing 14 original features to six and had a higher classification accuracy. Approach 4 was the most efficient for feature construction.

The results of the best high-level features were compared to SVM and MLP, both of which were capable of changing the representation of features, and then compared to a number of relevant studies in the literature. Results of these comparisons indicate that a simple classifier, such as a DT, can demonstrate a competitive performance to more sophisticated algorithms by extracting high-level features using GP, while being less prone to over-fitting.

The proposed approach was validated by another dataset collected in Australia and New Zealand, which indicated that a DT with the GP constructed features as its input has a higher transferability than SVM and MLP. As for future research directions, different techniques for controlling bloat in GP individuals could be investigated; bloat problem in GP is an increase in average tree size and depth without any significant increase in fitness. In this study, the maximum depth of trees was adjusted for this purpose. The generated numeric constants in GP individuals could also be optimized to improve the performance of mode detection. In addition, the impact of using other classification algorithms such as random forests of a DT in wrapper approaches can be studied.

ACKNOWLEDGEMENTS

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

CONFLICT OF INTEREST

The authors have declared no conflict of interest.

ORCID

Mahmoud Mesbah https://orcid.org/0000-0002-3344-1350
Behrang Assemi https://orcid.org/0000-0003-2043-8782

REFERENCES

1. Dabiri, S., Heaslip, K.: Inferring transportation modes from GPS trajectories using a convolutional neural network. Transp. Res. Part C Emerging Technol. 86, 360–371 (2018)
2. Shen, L., Stopher, P.R.: Review of GPS travel survey and GPS data-processing methods. Transp. Rev. 34(3), 316–334 (2014)
3. Gonzalez, P.A., Weinstein, J.S., Barbeau, S.J., Labrador, M.A., Winters, P.L., Georggi, N.L., et al.: Automating mode detection for travel behaviour analysis by using global positioning systems-enabled mobile phones and neural networks. IET Intell. Transp. Syst. 4(1), 37–49 (2010)
4. Gong, L., Morikawa, T., Yamamoto, T., Sato, H.: Deriving personal trip data from GPS data: A literature review on the existing methodologies. Procedia-Social Behav. Sci. 138(0), 557–565 (2014)
5. Wang, H., Liu, G., Duan, J., Zhang, L.: Detecting transportation modes using deep neural network. IEICE Trans Inf Syst. E100.D(5), 1132–1135 (2017)
6. Neshatian, K., Zhang, M., Andreae, P.: A filter approach to multiple feature construction for symbolic learning classifiers using genetic programming. IEEE Trans. Evol. Comput. 16(5), 645–661 (2012)
7. Gandomi, A.H., Roke, D.A.: Assessment of artificial neural network and genetic programming as predictive tools. Adv. Eng. Software 88, 63–72 (2015)
8. Muñoz, L., Trujillo, L., Silva, S.: Transfer learning in constructive induction with genetic programming. Genet. Program. Evolvable Mach. 21, 529–569 (2020)
9. Poli, R., Langdon, W., McPhee, N., Koza, J.: A Field Guide to Genetic Programming. Lulu. com, Raleigh, NC, US (2008)
10. Espejo, P.G., Ventura, S., Herrera, F.: A survey on the application of genetic programming to classification. IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. 40(2), 121–144 (2009)
11. Bi, Y., Xue, B., Zhang, M.: Genetic Programming for Image Classification: An Automated Approach to Feature Learning. Springer Nature, Switzerland AG (2021)
12. Zheng, Y., Chen, Y., Xie, X., Ma, W.Y.: GeoLife2.0: A location-based social networking service. **In**: 2009 10th International Conference on Mobile Data Management: Systems, Services and Middleware, Taipei, Taiwan, 18–20 May (2009)
13. Safi, H., Assemi, B., Mesbah, M., Ferreira, L.: Trip Detection with Smartphone-Assisted Collection of Travel Data. Transp. Res. Rec. J. Transp. Res. Board. 2594(1), 18–26 (2016). https://doi.org/10.3141/2594-03
14. Assemi, B., Safi, H., Mesbah, M., Ferreira, L.: Developing and validating a statistical model for travel mode identification on smartphones. IEEE Trans. Intell. Transp. Syst. 17(7), 1920–1931 (2016)
15. Rojas, M.B., Sadeghvaziri, E., Jin, X.: Comprehensive review of travel behavior and mobility pattern studies that used mobile phone data. Transp. Res. Rec. J. Transp. Res. Board. 2563(1), 71–79 (2016)
16. Stopher, P., Jiang, Q., FitzGerald, C.: Processing GPS data from travel surveys. 28th Australas Transp. Res. Forum, vol. 28, pp. 1–21. Curtin University, Australia (2005)
17. Tsui, S., Shalaby, A.: Enhanced system for link and mode identification for personal travel surveys based on global positioning systems. Transp. Res. Rec. 1972(1), 38–45 (2006)
18. Gong, H., Chen, C., Bialostozky, E., Lawson, C.T.: A GPS/GIS method for travel mode detection in New York City. Comput. Environ. Urban. Syst. 36(2), 131–139 (2012)
19. Ansari Lari, Z., Golroo, A.: Automated transportation mode detection using smart phone applications via machine learning: Case study mega city of tehran. Transportation Research Board 94th Annual Meeting, Washington DC, USA, 11–15 January 2015
20. Feng, T., Timmermans, H.J.P.: Transportation mode recognition using GPS and accelerometer data. Transp. Res. Part C Emerging Technol. 37, 118–130 (2013)
21. Yu, J.J.Q.: Travel mode identification with GPS trajectories using wavelet transform and deep learning. IEEE Trans. Intell. Transp. Syst. 22(2), 1093–1103 (2020)
22. Bantis, T., Haworth, J.: Who you are is how you travel : A framework for transportation mode detection using individual and environmental characteristics. Transp. Res. Part C Emerging Technol. 80, 286–309 (2017)
23. Eftekhari, H.R., Ghatee, M.: An inference engine for smartphones to preprocess data and detect stationary and transportation modes. Transp. Res. Part C 69, 313-327 (2016)
24. Zheng, Y., Liu, L., Wang, L., Xie, X.: Learning transportation mode from raw GPS data for geographic applications on the web. In: Proceeding of the 17th International Conference World Wide Web - WWW. April 2008, 247–256 (2008). https://dl.acm.org/doi/10.1145/1367497.136753
25. Zheng, Y., Li, Q., Chen, Y., Xie, X., Ma, W.-Y.: Understanding mobility based on GPS data. In: Proceeding of the 10th International Conference Ubiquitous Comput.– UbiComp '08, (49), 312–321 (2008)
26. Endo, Y., Toda, H., Nishida, K., Kawanobe, A.: Deep feature extraction from trajectories for transportation mode estimation. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining pp. 54–66. Cham: Springer(2016)

27. Nawaz, A., Zhiqiu, H., Senzhang, W., Hussain, Y., Khan, I., Khan, Z.: Convolutional LSTM based transportation mode learning from raw GPS trajectories. IET Intell. Transp. Syst. 14(6), 570–577 (2020)
28. Koza, J.R.: Genetic Programming: On the Programming of Computers By Means of Natural Selection Complex Adaptive Systems. MIT Press, Cambridge, MA 1992
29. Sette, S., Boullart, L.: Genetic programming: Principles and applications. Eng. Appl. Artif. Intell. 14(6), 727–736 (2001)
30. Tran, B., Xue, B., Zhang, M.: Genetic programming for feature construction and selection in classification on high-dimensional data. Memetic Comput 8(1), 3–15 (2016)
31. In: Gandomi, A.H., Alavi, A.H., Ryan, C., (eds.): Handbook of Genetic Programming Applications. Editors, editor. Charm, Switzerland: Springer (2015)
32. Tran, B., Xue, B., Zhang, M.: Genetic programming for multiple-feature construction on high-dimensional classification. Pattern Recognit. 93, 404-417 (2019)
33. Krawiec, K.: Genet. Program. Evolvable Mach. 3(4), 329–343 (2002). https://doi.org/10.1023/a:1020984725014
34. Ahmed, S., Zhang, M., Peng, L., Xue, B.: Multiple feature construction for effective biomarker identification and classification using genetic programming. In: Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation. Vancouver BC Canada, 12–14 July 2014
35. Vincenty, T.: Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations. Surv. Rev. 23(176), 88–93 (1975)
36. Haeri, M.A., Ebadzadeh, M.M., Folino, G.: Statistical genetic programming for symbolic regression. Appl. Soft. Comput. 60, 447–469 (2017)
37. Powers, D.M.W.: Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. Int. J. Mach. Learn. Technol. 2(1), 37–63, (2020) arXiv Prepr arXiv201016061
38. Fortin, F.-A., De Rainville, F.-M., Gardner, M.-A., Parizeau, M., Gagné, C.: DEAP: Evolutionary algorithms made easy. J. Mach. Learn. Res. 13(1), 2171–2175 (2012)
39. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. 12, 2825–2830 (2011)
40. Han, J., Pei, J., Kamber, M.: Data Mining: Concepts and Techniques, Elsevier, Amsterdam, Boston (2011)
41. Safi, H., Assemi, B., Mesbah, M., Ferreira, L.: An empirical comparison of four technology-mediated travel survey methods. Journal of Traffic and Transportation Engineering (English Edition). 4(1), 80–87 (2017). https://doi.org/10.1016/j.jtte.2015.12.003
42. Safi, H., Assemi, B., Mesbah, M., Ferreira, L., Hickman, M.: Design and Implementation of a Smartphone-Based Travel Survey. Transp. Res. Rec. J. Transp. Res. Board. 2526(1), 99–107 (2015). https://doi.org/10.3141/2526-11

How to cite this article: Namdarpour, F., Mesbah, M., Gandomi, A.H., Assemi, B.: Using Genetic Programming on GPS Trajectories for Travel Mode Detection. IET Intell. Transp. Syst. 16, 99–113 (2022). https://doi.org/10.1049/itr2.12132