A Hierarchical Ensemble Learning Framework for Energy-Efficient Automatic Train Driving

Guohua Xi  
thecrrc corporation limited, beijing 100078, china.

Xibin Zhao  
the school of software and key laboratory for information system security, ministry of education (kliis)/beijing national research center for information science and technology (bnrist), tsinghua university, beijing 100084, china.

Yan Liu  
the school of software and key laboratory for information system security, ministry of education (kliis)/beijing national research center for information science and technology (bnrist), tsinghua university, beijing 100084, china.

Jin Huang  
the school of software and key laboratory for information system security, ministry of education (kliis)/beijing national research center for information science and technology (bnrist), tsinghua university, beijing 100084, china.

Yangdong Deng  
the school of software and key laboratory for information system security, ministry of education (kliis)/beijing national research center for information science and technology (bnrist), tsinghua university, beijing 100084, china.

Recommended Citation
Guohua Xi, Xibin Zhao, Yan Liu et al. A Hierarchical Ensemble Learning Framework for Energy-Efficient Automatic Train Driving. Tsinghua Science and Technology 2019, 24(2): 226-237.
A Hierarchical Ensemble Learning Framework for Energy-Efficient Automatic Train Driving

Guohua Xi, Xibin Zhao, Yan Liu, Jin Huang*, and Yangdong Deng

Abstract: Railway transportation plays an important role in modern society. As China's massive railway transportation network continues to grow in total mileage and operation density, the energy consumption of trains becomes a serious concern. For any given route, the geographic characteristics are known a priori, but the parameters (e.g., loading and marshaling) of trains vary from one trip to another. An extensive analysis of the train operation data suggests that the control gear operation of trains is the most important factor that affects the energy consumption. Such an observation determines that the problem of energy-efficient train driving has to be addressed by considering both the geographic information and the trip parameters. However, the problem is difficult to solve due to its high dimension, nonlinearity, complex constraints, and time-varying characteristics. Faced with these difficulties, we propose an energy-efficient train control framework based on a hierarchical ensemble learning approach. Through hierarchical refinement, we learn prediction models of speed and gear. The learned models can be used to derive optimized driving operations under real-time requirements. This study uses random forest and bagging–REPTree as classification algorithm and regression algorithm, respectively. We conduct an extensive study on the potential of bagging, decision trees, random forest, and feature selection to design an effective hierarchical ensemble learning framework. The proposed framework was testified through simulation. The average energy consumption of the proposed method is over 7% lower than that of human drivers.

Key words: machine learning; energy efficiency; train driving system; feature selection; ensemble learning

1 Introduction

Railway transportation is the backbone of industrialized countries. Owing to the large scale and passenger/cargo capacity of railway networks, locomotives consume a large amount of energy. A recent report of Chinese Railway stated that the annual energy consumed by trains in China amounted to 142 billion kWh, which is approximately 0.4% of the total energy consumption of the country[1, 2]. This level of energy consumption suggests that even a 2% energy saving can support the residential power usage of a major metropolitan city, such as Shanghai, and has a huge overall impact on carbon dioxide emissions. With the rapid deployment of smart railway transportation technology[3, 4], energy-efficient train driving solutions are necessary to alleviate the pressure on energy consumption and environmental pollution. The unique characteristics of railway transportation present opportunities for energy optimization. Given a certain railway route, the geographic characteristics are known a priori and only the parameters (e.g., loading) of a train vary in each run. Therefore, an energy-optimal driving profile for each trip can be obtained by considering the geographic and inherent conditions. However, the train gear operation...
The energy-efficient train driving problem has been widely studied for decades. Many advanced numerical and heuristic techniques have been proposed to calculate an optimal trajectory, i.e., trip profile, of the train operation. Han et al. [5] and Li and Hou [6] used genetic algorithms to construct a reference trajectory that optimizes train control for energy efficiency. Wang et al. [7] proposed two approaches to solve this optimal control problem under various constraints, such as fixed arrival time, by treating energy consumption and riding comfort as tradeoffs in calculating the cost function. A commercial system for operating trains was developed by Kumar et al. [8]. Many researchers have extended the optimization problem to treat railway systems as a whole. Caprara et al. [9] formulated the major optimization problems for the planning of a passenger railway system. The problems range from the definition of the routes and frequencies of the trains in the railway network to the construction of the duties and schedules of drivers and conductors. Minimizing the total energy consumption of railway systems was discussed by Miyatake and Ko [10] by applying numerical methods such as dynamic programming, gradient method, and sequential quadratic programming. An evolutionary algorithm-based Pareto optimization approach for speed tuning in a railway system was presented by optimizing travel duration and energy saving; this approach proposed a set of diversified non-dominated solutions to decision makers [11]. Su et al. [12] developed the concept of optimized integrated timetable, which includes the timetable and speed profiles. The aforementioned approaches have superior solution qualities but tend to be computationally intensive. As a result, they are more aligned to offline processing and extremely time-consuming for onboard real-time control.

As each trip has varying system parameters (e.g., loads), onboard optimization of the train trip profile is essential to improve energy efficiency. Online real-time control techniques [13, 14] according to system dynamic performances [15] for train operations have been studied extensively. Salmasi [16] classified and reviewed the state-of-the-art control strategies for hybrid electric vehicles. Ding et al. [17] proposed an optimal driving model for energy-efficient operation of trains under fixed block and mobile block conditions, and designed corresponding heuristic optimization algorithms. Ke et al. [18] presented a heuristic method to optimize the train speed trajectory and control sequence by considering track gradient, average speed, restriction of train speed, acceleration, and jerk. The problem of an automatic train operation system with multiple working conditions was investigated by Wang et al. [19]. Gao et al. [20] presented a neuro-adaptive robust control method for automatic train operation, which was subject to unknown systematic time-varying dynamics. The aforementioned real-time techniques met the requirements for online processing but suffered from lack of guarantee for solution quality.

The preceding analysis suggests that an energy saving solution to the train control optimization problem should address the issue of optimization quality and computation efficiency. In this paper, we propose a data-driven and hierarchical ensemble learning framework for energy-efficient train driving. Based on the fundamental idea of learning to drive by mining the driving log files of experienced train drivers, the proposed ensemble learning framework integrates various prediction models such as bagging, decision tree, random forest, and feature selection. We train the prediction models offline and an onboard system feeds the train parameters into the models to derive an optimized sequence of gear controls. The effectiveness of our framework is evaluated on a simulation platform. Results showed that on average, our framework can achieve 7.15% energy saving. The current study is among the first to explore machine learning techniques for intelligent train driving problems. The rest of this paper is organized as follows. Section 2 describes the problem of energy-efficient train driving. The hierarchical ensemble learning framework for energy-efficient train driving is elaborated in Section 3. Section 4 presents a practical application to demonstrate the effectiveness of the approach. Section 5 summarizes the contents of this study.

2 Problem Statement

This study focuses on the energy-efficient train driving problem, which can be formulated as a driving trip planning problem with the optimization object of minimizing the energy consumption and time deviation under various constraints. The output is a control sequence consisting of a series of discrete or continuous settings of the control throttle with predefined traction or braking forces.
2.1 Train model and railway environment

We adopt a mass-point model to compute the train dynamics in the solution of the train optimal driving problem. The motion of a train can be expressed by the following model[21]:

\[
\begin{align*}
\frac{d}{dt} & = f(s) - R_b(v) - R_l(s), \\
\frac{ds}{dt} & = v
\end{align*}
\]

(1)

where \( m, v, \) and \( s \) are the mass, velocity, and position (i.e., displacement) of the train, respectively; \( \rho \) is a factor that accounts for the rotating mass; \( f(s) \) is the traction or braking force bounded by the maximum traction force \( f_{\text{max}} \) (\( f_{\text{max}} > 0 \)) and the maximum braking force \( f_{\text{bmax}} \) (\( f_{\text{bmax}} > 0 \)); \( R_b(v) \) is the basic resistance including roll resistance and air resistance; and \( R_l(s) \) is the line resistance caused by track grade, curves, and tunnels.

The empirical equation of the basic resistance \( R_b(v) \) is defined as

\[
R_b(v) = m(a_1 + a_2 v + a_3 v^2)
\]

(2)

where the coefficients \( a_1, a_2, \) and \( a_3 \) depend on the train characteristics and can be obtained through experiments. The empirical equation of line resistance \( R_l(s) \) is defined as

\[
R_l(s) = m \cdot g \cdot \sin \alpha(s) + f_c(r(s)) + f_l(l_i(s), v)
\]

(3)

where \( g \) is the gravitational acceleration; \( \sin \alpha(s) \), \( r(s) \), and \( l_i(s) \) are the slope, radius of the curve, and length of the tunnel along the track, respectively. When passing through a tunnel, the train experiences a higher air resistance that depends on the shape of the tunnel, smoothness of the tunnel walls, exterior surface of the train, and other factors.

The curve resistance \( f_c() \) and tunnel resistance \( f_l() \) are given by empirical equations:

\[
\begin{align*}
f_c(r(s)) &= \begin{cases} 6.3/(r(s) - 55) & \text{for } r(s) \geq 300 \text{ m}, \\
4.91/(r(s) - 30) & \text{for } r(s) < 300 \text{ m}, \end{cases} \\
f_l(l_i(s), v) &= \frac{l_i v(s)^2}{10^7}
\end{align*}
\]

(4)

Different trains may exhibit varying resistances reflected by the values of the coefficients (Eq. (4)). In our locomotive model, selecting the position \( s \) as an independent variable is more convenient than selecting time \( t \). Such a treatment streamlines the consideration of track-related data, such as line resistance and speed limits. In addition, the analytical and numerical study of the optimal problem is simplified considerably. The choice of kinetic energy instead of speed \( v \) facilitates the solution of the optimal control problem by eliminating some of the nonlinearities. Thus, we set kinetic energy per mass unit as \( K = 0.5v^2 \). The train motion can then be captured with the following continuous-space model:

\[
\begin{align*}
\frac{d}{ds} & = f(s) - R_b(\sqrt{2K}) - R_l(s), \\
\frac{dt}{ds} & = \frac{1}{\sqrt{2K}}
\end{align*}
\]

(5)

with all terms as previously defined.

2.2 Energy-efficient train trip optimization problem

The energy-efficient train driving problem can be formulated as a general optimization problem. The traction or braking force \( f(s) \) is the control input determined by the discrete or continuous settings of the throttle for most of the railway locomotives. The state variables include the train position \( s \) and speed \( v \). The objective function to be minimized can be the trip time deviation from the schedule or the energy consumption for a given trip time, or a combination of both. In this paper, we consider both the energy consumption and time deviation in the objective function, with all other factors such as safety treated as constraints. By employing the train dynamics model (Eq. (5)), we can state the optimization objectives in the following position-dependent form:

\[
\begin{align*}
J_E &= \int_{s_0}^{s_{\text{end}}} \phi(f) \left( f(s) + \lambda \left| \frac{df(s)}{ds} \right| \right) ds, \\
J_T &= |T - \tilde{T}|
\end{align*}
\]

(6)

subject to the following constraints:

\[
- f_{\text{bmax}} < f(s) < f_{\text{max}}, \\
0 \leq T(s) \leq T_{\text{max}}(s), \\
v(s) \leq v_{\text{lim}}(s)
\]

(7)

and the following boundary conditions:

\[
\begin{align*}
s(0) &= s_{\text{start}}, & v(0) &= v_{\text{start}}, \\
s(T) &= s_{\text{end}}, & v(T) &= v_{\text{end}}
\end{align*}
\]

(8)

Here, \( J_E \) and \( J_T \) represent the optimization objective of energy consumption and time deviation, respectively. \( \phi(f) \) stands for the throttle depended coefficient. \( \tilde{T} \) is the scheduled time for a train trip and \( T \) as the real time cost for the trip. The maximum allowable velocity \( v_{\text{lim}}(s) \) depends on the train characteristics and line conditions; thus, it is usually a piecewise constant function of the coordinate \( s \). \( s_{\text{start}} \) and \( v_{\text{start}} \) are the position and velocity at the beginning of the route. \( s_{\text{end}} \) and \( v_{\text{end}} \) are the position and velocity at the end of the
route. The duration of the trip $\tilde{T}$ is usually given by the timetable.

We assume that the unit kinetic energy $\Delta E(s) > 0$, which means that the speed of the train is always strictly larger than 0, and the train travels in a non-stopping manner in the given trip.

2.3 Analysis of human driving records

A railway route has unique geographic distributions. A representative slope map of railways is shown in Fig. 1. During driving, train drivers use their experience to make decisions on the selection of throttling/braking operation to accelerate/decelerate the train. All these throttling/braking operations play a role in the final energy consumption and punctuality. By reviewing a large number of human driving records (i.e., with operating gear data obtained from the train data recorder), we find that common patterns exist in the driving behaviors of experienced drivers. Figure 2 shows a few representative driving patterns represented as velocity and throttling/braking. The trends of velocity variation in similar types of route slopes are similar. Meanwhile, the speed and gear changes under similar tendencies are also similar. Thus, we can conclude that the drivers’ behaviors are alike in the same type of route slope, while certain patterns can be easily recognized for most routes.

The hidden driving patterns provide an important clue for us to solve the problem. Using the driving data of experienced drivers to train our prediction model, we can determine the driving performance close to that of experienced drivers. On such a basis, a hierarchical framework was developed in this study to solve the energy-efficient train driving problem.

3 Hierarchical Ensemble Learning Framework for Energy-Efficient Train Driving

The energy-efficient train driving problem is a typical multi-constrained and nonlinear optimization problem. Although existing methods such as genetic algorithms, neural networks, artificial heuristic design operation strategy, and others can eventually obtain an optimized gear sequence, and their limitations are obvious. For example, the searching algorithms cannot guarantee the consistency of the results by considering all train driving situations and can hardly meet the allowed computing time for onboard real-time control systems. To overcome these limitations, we adopt a hierarchical refinement scheme. We design a two-layer prediction framework to predict the speed and gear changes with the speed obtained in the first layer, and the gear change in the second stage. Using layer-by-layer refinement predication, we can finally determine the control gear sequence. The proposed method is detailed in the following.

3.1 Solution framework

As discussed in the previous section, driving behaviors are very similar under the same kind of slope conditions. We learn the generic driving patterns from the driving data of experienced drivers and use these patterns as rules for generating driving solutions online. The velocity change trend in one segmentation is a coarse-grained rule, which contains a combination of velocity changes and proportion of the velocity change distance. As we need fine-grained rules for gear operations, in the learning stage, we cut a route into multiple sections so that the prediction of speed and gear can be predicted for each section. The collection of gear predictions along a route constitutes a gear operation sequence.

Figure 3 shows the solution framework. First, we pre-process the data on the route and the train. Then, we use railway domain knowledge and feature selection
algorithms to select proper features. Thereafter, an ensemble machine learning algorithm is adopted to train prediction models. With the trained model, we predict the detailed gear operations before each trip.

In the proposed solution framework, the hierarchical organization of the prediction models is the most important part. The first layer mainly deals with velocity prediction. It takes velocity changes and the proportion of distance associated with the changes as input. As shown in Fig. 4, one route is divided into multiple segments on which we define the rules for acceleration, deceleration, and uniformity of driving. By mining a large number of driving records of experienced drivers, we can find the rules for manipulating the speed of trains given a specific slope and loading. Figure 4 shows one segment of a route. The change of speed for this segment can be described as Speed up–Speed down–Uniform speed–Speed up.

We use “1, −1, 0” to represent “Speed up, Speed down, Uniform speed”, respectively. As a result, the speed option for the segment in Fig. 4 is “1, −1, 0, 1”. We train a regression model to predict the proportion of distance associated with the velocity change in the segment in Fig. 4, where the proportions are 10%, 30%, 45%, and 15%, respectively. Finally, we obtain two models, a classification model to predict the velocity change pattern and a regression model to predict the proportion of distance associated with the speed change in this segment.

The speed patterns are not yet sufficient to meet the goal of energy-efficient train driving because optimizing the train operation (gears) is the most effective way to save energy. Accordingly, the second layer concerns the rules of drivers. Our goal is to derive gear information under different trends of velocity variations. As shown in Fig. 5, we have to find the
3.2 Data processing and feature selection

The quality of machine learning models heavily depends on the available data. In this study, the train data includes route information, locomotive parameters, and gear operation data, which are collected from a train data recorder. In this section, we describe the process of data preprocessing and feature selection in detail.

3.2.1 Data preprocessing

The original data are obtained from a recording instrument as log files. We have to process the raw data into a proper form to train the machine learning models. The most important step in data preprocessing is route segmentation, i.e., cutting the route into sections. This step has two advantages. First, a resultant section has a uniform environmental condition. Second, segmentation enables us to identify similar segments in different routes. As a result, we are able to find common patterns in the driving records collected from multiple routes. For a given section, based on Eq. (3), all environmental factors are translated into resistance and inertial properties. The overall impact can be represented as a uniform parameter by computing the Equivalent Gradient (EG), which has a unit of degree per 1000 meters [21]. We perform the segmentation process according to EG values along a route. In this study, EG takes five different discrete values as listed in Table 1. We further characterize each segment as short, medium, or long according to the length, with the critical values as 1000 and 3000 m.

| Section type               | Label | Resultant gradient |
|----------------------------|-------|--------------------|
| Steep down grade section   | −2    | ≤−3                |
| Gentle down grade section  | −1    | −1 to −3           |
| Gentle grade section       | 0     | −1 to 1            |
| Gentle up grade section    | 1     | 1 to 3             |
| Steep up grade section     | 2     | ≥3                 |

3.2.2 Feature selection

As mentioned, a large number of factors contribute to the energy consumption of a train on a given route. These factors are the features for model training. However, feature selection is essential to avoid overfitting and excessive training time associated with high-dimensional data. The feature selection process should find a balance between complexity (in terms of computation and data storage) and performance (in terms of accuracy and recall rate). Figure 6 lists the details of features selected in this study.

In this paper, we use the Correlation-based Feature Selection (CFS) method [22–24] for feature selection. This method adopts a correlation-based heuristic to evaluate the value of features. CFS searches features according to the degree of redundancy among features. The evaluator aims to find a subset of features that are individually highly correlated with the category but have low intercorrelation. The search process is guided by a numeric measure, such as conditional entropy, to iteratively add features that have the highest correlation with the category. The value of a subset of attributes is evaluated by considering the individual predictive ability along with the degree of redundancy between them. CFS also uses multivariate filters to account for the interactions between features. The equation for CFS is as follows:

$$r_{zc} = \frac{\overline{z}}{\sqrt{k + k(k-1)\overline{II}}}$$

where $r_{zc}$ is the correlation between the feature subsets $zc$ and the class variable, $k$ is the number of feature subsets, $\overline{z}$ is the average of the feature–class correlation, and $\overline{II}$ is the average of the feature–feature intercorrelation. The fraction can be considered as an indicator of how predictive a group of features are, and the denominator represents the degree of redundancy among them. The heuristic handles irrelevant features because they will be poor predictors of a given class. Redundant attributes are identified because they are highly correlated with one or more of the other features.

In this study, CFS is combined with a best-first search strategy [25] to determine the best feature subset. As a greedy strategy, the best-first search sorts the nodes according to the distance from the target and then selects a node to expand on the basis of the estimated distance of nodes. This strategy searches for a better feature subset by traversing the feature set space, while CFS is used as the estimator to measure the quality of
the feature. The entire search process ends when the termination condition is reached. Through the proposed techniques, the 50-dimensional feature space can be reduced to 10 to 15 features.

3.3 Hierarchical combination of ensemble learning methods

After preprocessing the training data and selecting the appropriate features, we need to train the prediction models. As mentioned, we use a hierarchical framework to predict a gear control sequence. In the training process, we have to obtain a velocity prediction model and a gear prediction model. Both models include changes of velocity and the proportion of a velocity in a section. Thus, we need to train a total of four models.

As the training data include discrete and continuous features, we need to predict categories and real values. The training data also have a large number of class labels. Accordingly, we decide to use the tree-based machine learning algorithms. Ensemble learning algorithm is capable of integrating multiple prediction models. Our framework is based on a hierarchical ensemble learning algorithm. Specifically, we use a random forest to predict velocity and gear proportion.

3.3.1 Random forest

We use the random forest algorithm to predict the velocity and gear change. The random forest learning ensemble consists of bagging of unpruned decision tree learners with a randomized selection of features at each split[26]. The basic principle of the algorithm is to train several decision trees and generate multiple models. Then, the trees are combined to form a single, strong learner by averaging or taking the majority vote. The pseudocode is listed in Algorithm 1.

The algorithm works as follows. For each tree in the forest, we select a bootstrap sample from $S$, where $S^{(i)}$ denotes the $i$-th bootstrap. Then, we learn a decision tree using a modified decision tree learning algorithm. The algorithm is modified as follows. At each node of the tree, instead of examining all possible feature splits, we randomly select some subsets of the features $f \subseteq F$, where $F$ is the set of features. The node then splits on the best feature $f$ rather than $F$. In practice, $f$ is significantly smaller than $F$. Deciding on which feature to split is often the most computationally expensive task of decision tree learning. By narrowing down the choice of features, we can drastically speed
up the learning process.

3.3.2 Bagging with REPTree

In the regression prediction stage, we have to predict the velocity and gear proportions. To improve the accuracy, we tried several classifiers and found that the best-performing one is bagging with REPTree\cite{27}.

REPTree generates regression trees by using the information gain as the splitting principle. The learned decision tree is the best fit for the training data, but it likely to be overfitting. To address the problem, a second phase is employed to prune the tree to reduce its dependency on the training data and allow the tree to be generalized. This stage requires a separate pruning dataset, which can be a problem because data are normally scarce. However, REP can be extremely powerful when it is used in combination with boosting. The major means of pruning is replacing a subtree with a leaf that represents the majority of all examples reaching it in the pruning set. This replacement is taken if this modification reduces the error, i.e., the new tree provides an equal or fewer number of misclassifications.

Classification errors in machine learning come from sources such as bias, variance, and noise. Bias refers to the accuracy of the algorithm itself, variance measures the precision or specificity of the algorithm, and noise indicates the intrinsic disturbance in the data. In this study, our main goal is to limit the variance of the model. The variance evaluates how the discrepancies in

### Algorithm 1 Random Forest

**Require:** A training set \( S := (x_1, y_1), ..., (x_n, y_n) \), features \( F \), and number of trees in forest \( B \).

```
function RANDOMFOREST(S, F)
1: for i \in 1, ..., B do
2: \( S^{(i)} \leftarrow \) A bootstrap sample from S (i.i.d sample with replacement); \( h_i \leftarrow \) RANDOMIZEDTREELEARN\( (S^{(i)}, F) \);
3: \( H \leftarrow H \cup h_i \);
4: end for
5: return H
6: end function
```

### Algorithm 2 Bagging Algorithm

**Require:** A training set \( S := (x_1, y_1), ..., (x_n, y_n) \), Classifier \( L \), and number of bootstrap samples (iterations) \( T \).

```
for i \in 1 to T do
1: \( S' \leftarrow \) A bootstrap sample from S (i.i.d. sample with replacement);
2: \( C_i \leftarrow L(S') \);
3: end for
4: \( C^* \leftarrow \arg \max_y \sum_{x \in S} I(C_i(x) = y) \) (the most often predicted label \( y \));
5: return classifier \( C^* \)
```

### Table 2 Trained prediction models.

| Model | Description |
|-------|-------------|
| M1    | Prediction of the velocity change combinations of the slope sections |
| M2    | Prediction of the proportion of the velocity in velocity change combinations |
| M3    | Prediction of the gear change combinations of the velocity section |
| M4    | Prediction of the proportion of the gear in gear change combinations |
we compute the gear control sequence for a trip.

4 Experiments and Analysis of Results

In this paper, the experiment and simulation of the proposed hierarchical ensemble learning prediction energy-efficient driving framework are based on a specific locomotive model on a given route. The locomotive has a gear with 17 levels, specifically, 8 traction levels (1 to 8), a neutral level (0), and 8 braking levels (−1 to −8). A higher absolute value of the gear level provides a higher traction or braking force. Meanwhile, a stronger level of throttle entails higher energy consumption. As the locomotive maintains a constant power output, the energy consumption can be considered as only related to the selection of the gear, which is different from the traction or braking force. For the locomotive used in the experiments, the power characteristics of the brake and traction gears are shown in Figs. 7 and 8, respectively. The route used in this study is a commercial railway line between Sujiatun and Benxi in Shenyang Province, China. Figure 9 illustrates the complex geographical features of the railway line.

Our framework was implemented as an integrated software platform for offline learning and an onboard device for online optimization. The entire proposed framework was tested on a hardware-in-loop test platform shown in Fig. 10. The equipment marked ⑤ is the online trip optimization hardware developed in this study. Other devices serve as simulation and measurement platforms of a freight train. The load inputs in the simulations are given according to the recorded driving data.

We select a total of 633 actual driving records from experienced drivers on different routes with varying environment and loading conditions as the training data. We use the implementation of the machine learning algorithm provided by the Waikato Environment for Knowledge Analysis toolkit to build the prediction models. Then, we use a 10-fold cross-validation procedure to evaluate the models with the results of the four models shown in Table 3.

In Table 3, precision, recall, and F-Measure are used...
Table 3 Evaluation results of trained prediction models.

| Model | Precision (%) | Recall (%) | F-Measure (%) | CC | MAE | RMSE | RAE (%) | RRSE (%) |
|-------|---------------|------------|---------------|----|-----|------|---------|----------|
| M1    | 92.4          | 92.6       | 92.5          | –  | –   | –    | –       | –        |
| M2    | –             | –          | –             | 0.9916 | 0.0186 | 0.0498 | 5.1961 | 12.9661 |
| M3    | 92.0          | 92.2       | 92.0          | –  | –   | –    | –       | –        |
| M4    | –             | –          | –             | 0.9829 | 0.0198 | 0.0619 | 6.6068 | 18.4347 |

This table presents the evaluation results of trained prediction models, including precision, recall, F-Measure, Correlation Coefficient (CC), Mean Absolute Error (MAE), Root-Mean-Squared Error (RMSE), Relative Absolute Error (RAE), and Root Relative Squared Error (RRSE). The results for models M1 and M3 are directly copied from the table, while the results for models M2 and M4 are computed using the following formula:

\[
\text{F1-Measure} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (10)
\]

Correlation Coefficient (CC), Mean Absolute Error (MAE), Root-Mean-Squared Error (RMSE), Relative Absolute Error (RAE), and Root Relative Squared Error (RRSE) are used to evaluate the regression models. The acronyms used are explained as follows:

- **CC**: The correlation is computed between the predicted and actual target values;
- **MAE**: This is a quantity used to measure how close forecasts or predictions are to the eventual outcomes;
- **RMSE**: The error is the amount by which the value implied by the estimator differs from the quantity to be estimated;
- **RAE**: The error is relative to a simple predictor, which is the average of the actual values;
- **RRSE**: This value is the square root of (sum of squares of errors / sum of squares of differences from mean).

A simulated trip operated by the control sequence by the trained model is illustrated in Fig. 11. The black lines are speed and gear curves of experienced drivers, while the grey lines are results derived by the output of our prediction framework. Evidently, our findings are highly consistent with the record of the experienced driver.

We select ten actual driving logs from human drivers with different loading conditions for the trip along the route. Then, we compare the average driving performance in terms of energy efficiency and time deviation with the results from the experiments and simulations between human drivers and the proposed approach. The comparison results are shown in Table 4. The results show that the average energy consumption from the proposed approach is approximately 7.16% lower than the human driving data, while the average time deviation from the train schedule is less than

Table 4 Comparison of driving performance between drivers and proposed approach.

| No. | Load (ton) | EC-Driver (kg) | TC-Driver (s) | EC-Proposed (kg) | TC-Proposed (s) | ES | TD | ES (%) |
|-----|------------|----------------|---------------|------------------|-----------------|----|----|--------|
| 1   | 3272       | 229.29         | 3876          | 209.46           | 3933.5          | 19.83 | 57.5 | 8.65   |
| 2   | 3540       | 259.73         | 3946          | 241.26           | 3980.5          | 18.47 | 34.5 | 7.11   |
| 3   | 3574       | 230.54         | 3928.5        | 216.48           | 3909.5          | 14.06 | –19  | 6.10   |
| 4   | 3603       | 245.27         | 3948          | 232.52           | 3960            | 12.75 | 12   | 5.20   |
| 5   | 3646       | 246.57         | 3941          | 226.55           | 4010.5          | 20.02 | 69.5 | 8.12   |
| 6   | 3678       | 239.18         | 3913          | 221.19           | 3939            | 17.99 | 26   | 7.52   |
| 7   | 3702       | 244.57         | 3943.5        | 226.35           | 3882.5          | 18.22 | –61  | 7.45   |
| 8   | 3997       | 255.62         | 3959          | 235.17           | 3981.5          | 20.45 | 22.5 | 8.00   |
| 9   | 4123       | 259.39         | 3997.5        | 240.45           | 4009            | 18.94 | 11.5 | 7.30   |
| 10  | 4545       | 283.32         | 4257.5        | 266.01           | 4199            | 17.31 | –58.5| 6.11   |
| Avg | 3768       | 249.35         | 3971          | 231.54           | 3980.5          | 17.81 | 9.5  | 7.156  |

Note: EC represents energy consumption, TC indicates time consumption, TD is time deviation, and ES represents energy saving.
10 s. Experiment results validated the superiority of our approach on different routes. In addition, inference with the learned model can be successfully performed on the onboard system with limited computing capability.

5 Conclusion

This paper presented a framework based on a hierarchical ensemble learning approach for the energy-efficient train driving problem. In the framework, driving rules were learned from the driving records of experienced drivers and organized as a decision tree. The offline trained model was deployed in an onboard device, which could generate a control gear sequence for a given trip by considering the locomotive parameters. The proposed approach was validated by hardware-in-the-loop simulation targeting a widely deployed model of commercial diesel locomotive. Experiments proved that the proposed framework enabled an energy saving of more than 7%, and the time deviation from the train timetable was less than 1 min on average. The current study is among the first to explore machine techniques on intelligent train driving problems.

Acknowledgment

This study was sponsored in part by the National Natural Science Foundation of China (Nos. 61872217 and 61527812), Industrial Internet Innovation&Development Project of Ministry of Industry and Information Technology of China, National Science and Technology Major Project (No. 2016ZX01038101), MIIT IT funds (Research and Application of TCN Key Technologies) of China, and the National Key Technology R&D Program (No. 2015BAG14B01-02).

References

[1] J. Huang, Y. D. Deng, Q. W. Yang, and J. G. Sun, An energy-efficient train control framework for smart railway transportation, IEEE Trans. Comput., vol. 65, no. 5, pp. 1407–1417, 2016.
[2] J. Huang, F. Yang, Y. D. Deng, X. B. Zhao, and M. Gu, Human experience knowledge induction based intelligent train driving, in Proc. IEEE/ACIS 16th Int. Conf. Computer and Information Science, Wuhan, China, 2017, pp. 335–340.
[3] S. Ghosh and T. Lee, Intelligent Transportation Systems: New Principles and Architectures. Boca Raton, FL, USA: CRC Press, 2002.
[4] P. Li, L. M. Jia, and A. X. Nie, Study on railway intelligent transportation system architecture, in Proc. 2003 Intelligent Transportation Systems IEEE Int. Conf., Shanghai, China, 2003, pp. 1478–1481.
[5] S. H. Han, Y. S. Byen, J. H. Baek, T. K. An, S. G. Lee, and H. J. Park, An optimal automatic train operation (ATO) control using genetic algorithms (GA), in Proc. IEEE Region 10 Conf. Multimedia Technology for Asia-Pacific Information Infrastructure, Cheju Island, South Korea, 1999, pp. 360–362.
[6] Y. S. Li and Z. S. Hou, Study on energy-saving control for train based on genetic algorithm, (in Chinese). J. Syst. Simul., vol. 19, no. 2, pp. 384–387, 2007.
[7] Y. H. Wang, B. De Schutter, T. J. J. van den Boom, and B. Ning, Optimal trajectory planning for trains—A pseudospectral method and a mixed integer linear programming approach, Transp. Res. Part C: Emerg. Technol., vol. 29, pp. 97–114, 2013.
[8] A. K. Kumar, W. Daum, G. R. Shaffer, P. K. Houpt, J. F. Noffsinger, and D. L. McKay, Trip optimizer method, system and computer software code for operating a railroad train to minimize wheel and track wear, U.S. Patent 9233696, January 12, 2016.
[9] A. Caprara, L. Kroon, and P. Toth, Optimization problems in passenger railway systems, in Wiley Encyclopedia of Operations Research and Management Science, J. J. Cochran, L. A. Jr. Cox, P. Keskincokac, J. P. Khourouf, and J. C. Smith, eds. New York, NY, USA: John Wiley & Sons, 2011.
[10] M. Miyatake and H. Ko, Optimization of train speed profile for minimum energy consumption. IEEE Trans. Electr. Electron. Eng., vol. 5, no. 3, pp. 263–269, 2010.
[11] R. Chevrier, An evolutionary multi-objective approach for speed tuning optimization with energy saving in railway management, in Proc. 13th Int. IEEE Conf. Intelligent Transportation Systems, Funchal, Portugal, 2010, pp. 279–284.
[12] S. Su, X. Li, T. Tang, and Z. Y. Gao, A subway train timetable optimization approach based on energy-efficient operation strategy, IEEE Trans. Intell. Transport. Syst., vol. 14, no. 2, pp. 883–893, 2013.
[13] J. Huang, Y. H. Chen, and A. G. Cheng, Robust control for fuzzy dynamical systems: Uniform ultimate boundedness and optimality, IEEE Trans. Fuzzy Syst., vol. 20, no. 6, pp. 1022–1031, 2012.
[14] J. Huang, Q. M. Huang, Y. D. Deng, and Y. H. Chen, Toward robust vehicle Platooning with bounded spacing error, IEEE Trans. Comput.-Aided Des. Integr. Circ. Syst., vol. 36, no. 4, pp. 562–572, 2017.
[15] J. Huang, Y. H. Chen, and K. H. Guo, Novel approach to multibody system modeling: Cascading and clustering, J. Aerosp. Eng., vol. 27, no. 2, pp. 279–290, 2014.
[16] F. R. Salmasi, Control strategies for hybrid electric vehicles: Evolution, classification, comparison, and future trends, IEEE Trans. Veh. Technol., vol. 56, no. 5, pp. 2393–2404, 2007.
[17] Y. Ding, Y. Bai, F. M. Liu, and B. H. Mao, Simulation algorithm for energy-efficient train control under moving block system, in 2009 WRI World Congress on Computer Science and Information Engineering, 2009, pp. 498–502.
[18] B. R. Ke, C. L. Lin, and C. W. Lai, Optimization of train-speed trajectory and control for mass rapid transit systems, Control Eng. Pract., vol. 19, no. 7, pp. 675–687, 2011.
Guohua Xi received the BS and ME degrees from Shanghai Railway College in 1985 and 1997, respectively, and got the PhD degree from Central South University in 2007. He is now the President of FAW Group Corporation in China. His research interests include system engineering, rail Transportation, etc.

Xibin Zhao received the BS, ME, and PhD degrees from Jiangsu University, China, in 1994, 2000, and 2004, respectively. He is currently an associate professor with the School of Software, Tsinghua University, Beijing, China. His research interests include artificial intelligent, reliability analysis of hybrid network systems, and information system security.

Yan Liu received the BS degree from Xinjiang University, China, in 2014, and ME degree from Tsinghua University, Beijing, China, in 2017. He is now a software engineer at Perfect World Co., Ltd., Beijing, China. His research interests include artificial intelligent, information system security, etc.

Jin Huang received the PhD and BE degrees from Hunan University, Changsha, China, in 2012 and 2006, respectively. He was also a Joint PhD in Georgia Institute of Technology, USA, during 2009–2011. He started his career as the Postdoc and an assistant professor at the School of Software, Tsinghua University, Beijing, China, since 2013 and 2016, respectively. His research interests include artificial intelligence, cyber-physical systems, automation and mechatronics, fuzzy engineering, etc.

Yangdong Deng received the ME and BE degrees from Tsinghua University, Beijing, China, in 1998 and 1995, respectively, and the PhD degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, USA, in 2006. He is now an associate professor in the School of Software, Tsinghua University, Beijing, China. His current research interests include parallel electronic design automation algorithms, parallel program optimization, general purpose computing on graphics processing hardware, etc.