Energy 271 (2023) 126999

Contents lists available at ScienceDirect

Energy

journal homepage: www.elsevier.com/locate/energy

A frequency item mining based embedded feature selection algorithm and its application in energy consumption prediction of electric bus

Li Zhao [a,b], Yuqi Li [a,*], Shuai Li [c], Hanchen Ke [a]

[a] School of Mechanical and Electrical Engineering, Beijing Information Science and Technology University, Beijing, 100192, PR China
[b] Collaborative Innovation Center of Electric Vehicles in Beijing, Beijing, 100192, PR China
[c] Beijing Raise Science Co., Ltd., Beijing, 100012, PR China

ARTICLE INFO

Handling Editor: X Ou

Keywords:
Energy consumption prediction
Embedded feature selection algorithm
Frequency item
Regression learner

ABSTRACT

In the engineering practice of applying embedded feature selection algorithm to construct EV energy consumption prediction model, the constructed regression learners are often affected by random factors appeared in the process of data set sampling, algorithm initialization, computing platform resource scheduling and so on, which makes the prediction results of multiple regression learners constructed with the same feature combination different. This seriously affects the optimization process of energy consumption prediction model, resulting in the failure to find the optimal feature combination, and reduces the accuracy of the prediction results. To solve this problem, an embedded energy consumption prediction model construction method based on frequency item mining and evolutionary computing was proposed. In this algorithm, the combination of input characteristic variables is regarded as individual in the population, the prediction result of regression model is regarded as the fitness function, and the randomness of fitness function is corrected online by the statistical results of frequency items. Simulation results show that the algorithm solves the interference of randomness appeared in the process of resource scheduling, class library function reference, data set segmentation, etc., ensures the stability of feature combination in the optimization process of the prediction model, and gets accurate prediction results.

1. Introduction

In recent years, air pollution and energy emergencies have attracted widespread attention, and energy conservation and emission reduction have become important targets. As one of the main ways of green travel, electric buses are favored by the public for their pollution-free, zero-emission and low-noise characteristics [1–4]. As the basis of path planning, charging strategy optimization and dynamic performance parameter matching [5], energy consumption estimation of electric buses is of great significance not only for charging facility operators, but also for different entities such as power grid companies and users. At present, the data-driven line energy consumption estimation method is gradually favored by many scholars because of its comprehensive consideration of external factors and internal factors [6]. The prediction accuracy depends not only on the selection of regression algorithm, but also on the selection of high-value characteristic variables and their combinations. The purpose of feature selection is to filter out atypical features, reduce the time of model building, avoid overfitting, and enhance the generalization ability of the model.

At present, feature selection methods are mainly divided into three categories: filter, wrapper and embedded method [7]. The filter method calculates the feature weights independently according to the feature data, and the feature selection is independent of the subsequent classifiers or regressors. This method is computationally efficient and versatile, but the classification or regression accuracy is low. Commonly used filter methods include Relief, Chi-squared, etc [8]. Wrapper method uses different learning algorithms (such as support vector machine, decision tree, etc.) to evaluate the selected feature subset [9], which improves the accuracy of feature selection, but its computational complexity and cost are much higher than that of filter feature selection method. In general, the embedded feature selection method and the subsequent classifier/regression learner simultaneously complete the modeling and feature selection, which is conducive to finding the optimal solution, and the computational efficiency is higher than that of the encapsulated feature selection method.

Liu [10] constructed a regression learner using the embedded feature

* Corresponding author.
E-mail address: 2021020114@bistu.edu.cn (Y. Li).

https://doi.org/10.1016/j.energy.2023.126999
Received 8 December 2022; Received in revised form 13 February 2023; Accepted 17 February 2023
Available online 21 February 2023

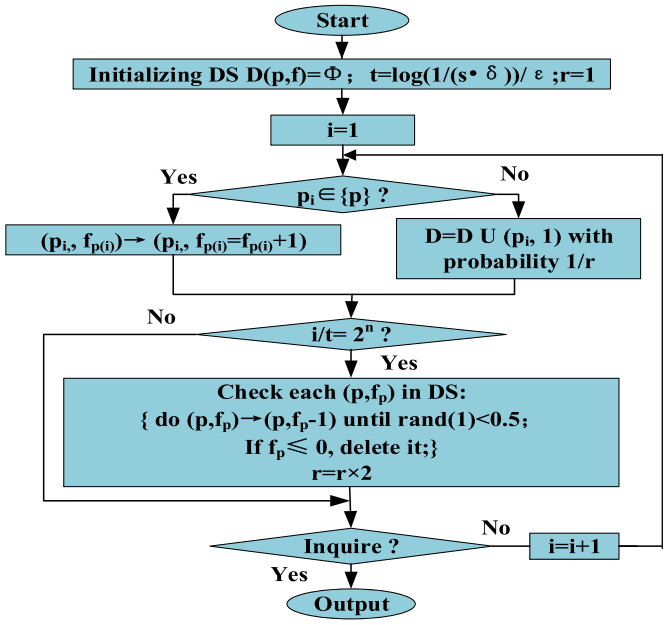


Fig. 1. The flow chart of Sticky Sampling algorithm.

selection algorithm, obtained the feature weight matrix through the local linear embedding (LLE) algorithm, described the reconstruction error with ℓ_1-norm, and found a robust solution by suppressing outliers and noise in the data set. Zheng [11] introduced an additional classifier to jointly learn feature weights with the traditional embedded model. The additional classifier was used to recover the unselected strongly correlated features and replace the weakly correlated features in the selected feature subset, which improved the generalization ability of traditional embedded algorithms. Zhao [12] developed a cost-sensitive embedded feature selection algorithm with $\ell_{2,1}$-norm and added orthogonal constraint terms to ensure that each selected feature is independent. Sebastian Maldonado [13] proposed a new embedded strategy for SVM classification and feature selection, then created a new embedded feature formula to solve the problems they faced, after that used quasi-Newton strategy and Armijo line search to update the scaling factor. Xu [14] proposed a two-stage feature selection method based on filter and embedding to minimize the redundant information in the candidate subset. Fan [15] used ridge regression to calculate the feature selection matrix, and used low-dimensional embedding and cosine similarity to remove most redundant features. Zhang [16] constructed low-dimensional embedding from raw data and designed a regularized optimization function to explore the intrinsic relationship between features as much as possible to ensure the selection of the optimal feature combination. Vikas Kumar [17] processed the label group, embedded the feature vectors into the low-dimensional space through linear mapping, and found the optimal feature combination. Shang [18] combined embedding learning and sparse regression to perform low-dimensional embedding mapping and sparse processing on feature data, which can select representative features faster and more effectively. Fatemeh Amini [19] proposed a two-layer feature selection method, in the first layer, a genetic algorithm was used to select the optimal subset, then in the second layer, an embedded elastic network was used to reduce dimensionality and to eliminate redundant features. Dornaika [20] performed dimensionality reduction work through multi-layer linear embedding, in which each layer of embedding eliminated irrelevant features, and with the used of the multi-layer embedding algorithm, the best feature combination are more easily extracted.

In fact, the prediction accuracy of embedded prediction models depends not only on whether appropriate regression algorithms are used, but also on whether high-valuable input variables and their combinations are selected from a given data set. At present, scholars have proposed many embedded feature selection methods to search for robust solutions of regression learners, improve their generalization ability, reduce trial and error and test costs, solve the problem of data set imbalance, and improve the efficiency of high-dimensional models. However, in engineering practice, due to the introduction of randomness in the process of data set sampling, learning algorithm initialization, computing platform resource scheduling and so on, when using the learner to evaluate the feature combination, different evaluation results will be obtained in each run of the same feature combination. This seriously affects the normal operation of embedded feature selection algorithm, resulting in the failure to find the optimal feature combination, and low accuracy prediction results.

To solve this problem, we propose an embedded feature selection algorithm based on frequency item mining and evolutionary computation. In the proposed algorithm, the combination of input feature variables is regarded as individual in the population, the prediction results of regression models are regarded as the fitness function, and the randomness of the fitness function is corrected online by the statistical results of frequency items. The algorithm is applied in the field of electric vehicle line energy consumption prediction, which solves the problem that the prediction results cannot guide the feature selection due to the random behaviors, such as resource scheduling of simulation platform, class library function reference, and random generation of training set.

2. Related works

2.1. Frequency itemset mining

Frequency itemset mining is the basis of data mining research, which counts the variables that often appear together in a data set and provides some support for possible decision-making. Frequency itemset mining is the basis of many important data mining tasks, such as association rules, correlation analysis, causality, sequence itemsets, local periodicity, and plot segments. Apriori and FP-growth are commonly used algorithms, which are mainly used in traditional relational data schema mining. For data stream, data has the characteristics of fast arrival speed, wide value range, and continuous arrival. Accordingly, the frequency itemset mining algorithm for data streams need to have the following characteristics: one access, continuous processing, limited storage, approximate results, fast response, and so on. Sticky Sampling algorithm and Lossy Counting algorithm are two common algorithms for statistics of frequent items in data streams [21].

Let s be the support threshold, δ be the failure probability, ϵ be the error, i be the current length of the stream, and r be the sampling rate. The incoming stream is divided into windows, the size of which is calculated as $t = \frac{1}{\varepsilon}\log(s^{-1}\delta^{-1})$. The first 2t elements are sampled at rate of r = 1, the next 2t elements are sampled at rate of r = 2, the next 4t elements are sampled at rate of r = 4, and so on. The id of the current window is w. Let D(p,f) be a data structure, in which p is an element that appeared in the data stream and f is its frequency. The traditional Sticky Sampling algorithm can be described as shown in Fig. 1.

2.2. Evolutionary algorithm

Evolutionary algorithm is a self-organizing and adaptive artificial intelligence technology based on Darwinian evolution, which solves problems by simulating the process and mechanism of biological evolution [22]. It generally starts from a set of initial points and uses the value of objective function to guide the evolutionary process. It does not need the derivative information of the objective function or the special knowledge related to the specific problem. Evolutionary algorithms solve optimization problems mainly through three operations: selection, crossover, and mutation. It includes four typical methods: genetic

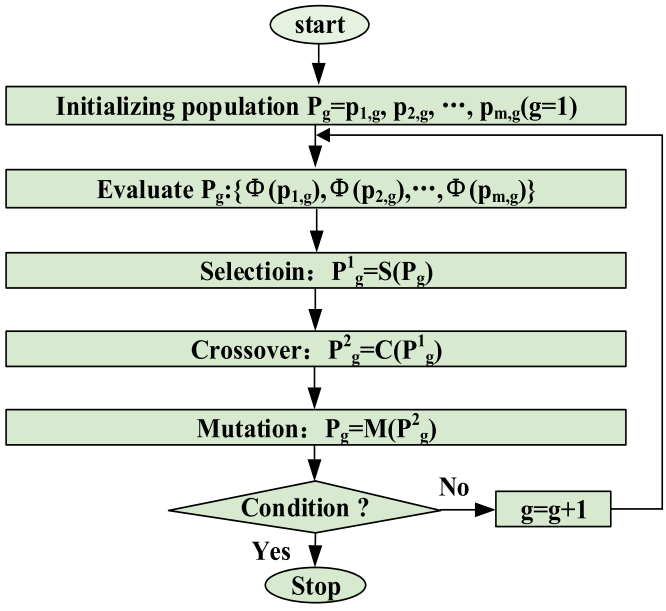


Fig. 2. The flow chart of GA algorithm.

algorithms, genetic programming, evolutionary strategies, and evolutionary programming [23]. Let p ∈ M be an individual, and M be the individual space. The fitness function can be denoted as Φ:M→R. In generation g, population $P_g = p_{1,g}, p_{2,g}, \ldots, p_{m,g}$ can be transformed into the next generation population by operators of S(selection), C(crossover) and M(mutation). Here S, C, and M all refer to macrooperators that transform old groups into new ones. The traditional GA can be described as shown in Fig. 2.

3. Frequency item mining and evolutionary computation based embedded feature selection algorithm

3.1. Randomness in the process of establishing regression model

For most embedded feature selection algorithms, the training process of regression model usually includes several steps: selecting data, training model and evaluating model. In these steps, some random factors tend to interfere with the training process of the regression model, resulting in different prediction accuracy of regression models trained by the same combination of features. For example, to prevent overfitting, cross-validation techniques are used during training. Different data set partitions result in different prediction models. In order to improve the generalization ability, random initialization technique is used in the training process. Different initial parameters will also lead to uncertain prediction results. Generally speaking, the more intelligent the regression model is, the more complex the training process is, the more underlying functions are called, and the more obvious the uncertainty will be. In Matlab R2021a, the same EV energy consumption data set is used for 100 training, and 100 different regression models are obtained. As can be seen from Fig. 3, due to a certain amount of randomness in the processes of data segmentation, model initialization, and resource scheduling, the prediction results of the final regression models are different.

3.2. The influence of stochastic factors on different forecasting models

In fact, among these factors that generate uncertainty, the random generation of training data sets, the random initialization of algorithm parameters, and the random scheduling of system resources are the main factors that generate uncertain results. For different types of regression model, the impact of these factors is different. From Table .1, we can find that for a Gaussian process regression model, the standard deviation (STD) of R^2 generated by algorithm initialization is much larger than that generated by random data sampling, and the standard deviation generated by resource scheduling is much smaller than that generated by random data sampling. For linear SVM regression model, quadratic SVM regression model, and Gaussian SVM regression model, the

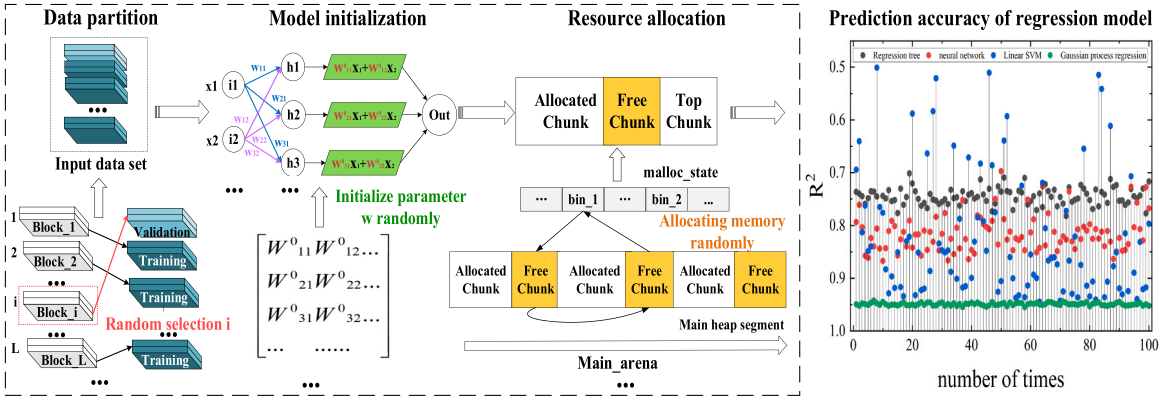


Fig. 3. Randomness in the process of establishing regression models.

Table 1
The influence of stochastic factors on model prediction results.

Regression model	Data extraction		Algorithm initialization		Resource scheduling	
	STD	Average	STD	Average	STD	Average
GPR	0.000000006	0.968404608	0.002731765	0.948985983	6.6949E-16	0.968404593
Linear SVM	0.11795485	0.827226244	0.002484819	0.943304593	7.81071E-16	0.953953338
Quadratic SVM	0.152368133	0.812919722	0.029214291	0.840188692	8.92653E-16	0.980841203
GPR SVM	0.207923035	0.669964386	0.006497166	0.626057425	1.89689E-15	0.676002001

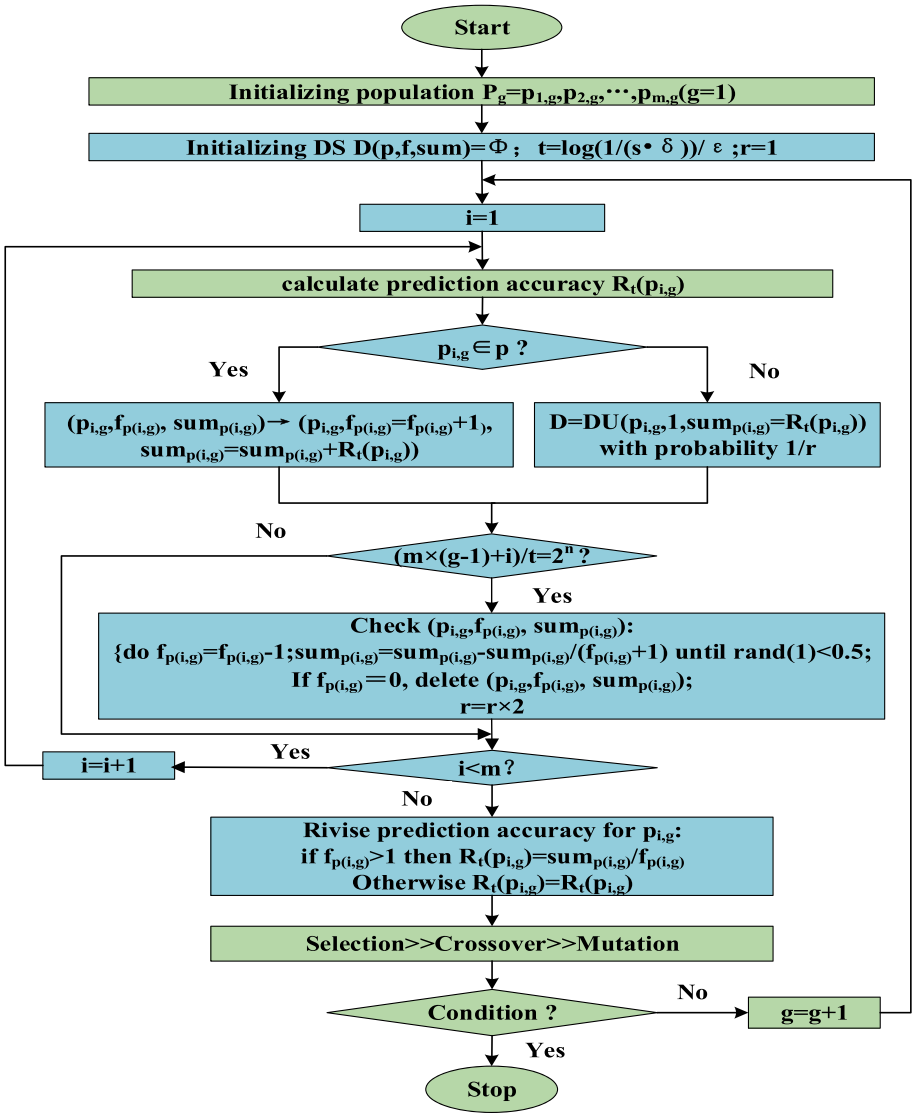


Fig. 4. The flow chart of FI-GAEFS algorithm.

standard deviation of R^2 generated by random data sampling is larger than that generated by algorithm initialization, and the standard deviation of R^2 generated by resource scheduling is much smaller than that generated by algorithm initialization. For different models, the three randomness factors have little influence on the average of the predicted results. Therefore, in the process of feature selection, it is possible to obtain a stable and high-precision prediction model by using the average value of the prediction results of the corresponding model of feature combination as the basis for feature selection. However, for some data sets with high-dimensional feature spaces, it is difficult to bear the cost of one evaluation for each combination, and it is almost impossible to evaluate the average value of the predicted results of multiple evaluations.

3.3. Revising prediction result with frequency items

In embedded feature selection methods, feature selection is often regarded as an optimization problem. The establishment of feature selection matrix and the training of regression model are completed simultaneously. Let D be a data set with n samples (s_1,s_2, …,s_n) and m features (f_1, f_2, …, f_m), R(X) be the accuracy of the regression learning machine built with feature combination X. In general, a feature selection problem requires finding a feature combination X* in the feature space M, such that $\exists X \in M$: $R(X)\leq R(X^*)$. However, in engineering practice, due to the introduction of randomness in the process of data set sampling, learning algorithm initialization, and computing platform resource scheduling, the accuracy R(X) is no longer a determinate value. At time t, the algorithm's prediction accuracy $R_t(X)$ can be expressed as

$$R_t(X)=\mathrm{r}(X)+es_t+ei_t+er_t \quad es,ei,er\in N(0,\delta^2) \tag{1}$$

where es_t, ei_t, and er_t are errors caused by randomness such as data sampling, computation initialization and resource scheduling, respectively. r(X) is the true accuracy. For the embedded feature selection algorithm using random search strategy, the same combination of features will produce different results at different running times, which will cause the algorithm to oscillate around the local optimal solution and fail to converge to the optimal solution.

In general, the average value of multiple calculations is often used to estimate the true prediction accuracy,

$$\overline{\mathrm{R}}(X)=\frac{1}{T}\sum_{t=1}^{T}\left[\mathrm{r}(X)+es_t+ei_t+er_t\right] \tag{2}$$

However, for different feature combinations X, the number of times T appears in the random search process is constantly changing. Thus,

Table 2
Parameters in the data set of No. 51 electric bus.

Parameter	Unit	Parameter	Unit
SOC	%	Longitude	°
Mileage	*km*	Latitude	°
Total voltage	*V*	Acceleration pedal travel value	%
Total current	*A*	Brake pedal state	%
Time	*s*	Driving motor torque	*Nm*
Max battery voltage	*V*	Driving motor speed	*r/min*
Min battery voltage	*V*	Drive motor controller temperature	°C
Motor controller dc bus current	*A*	Motor controller input voltage	*V*

Table 3
Parameters in the appliance energy forecasting data set.

Parameter	Unit	Parameter	Unit
Energy use of appliances in the house(Appliances)	*Wh*	Humidity outside the building (north side)(RH_6)	%
Energy use of light fixtures in the house(lights)	*Wh*	Temperature in ironing room(T7)	°C
Temperature in kitchen area (T1)	°C	Humidity in ironing room(RH_7)	%
Humidity in kitchen area (RH_1)	%	Temperature in teenager room 2 (T8)	°C
Temperature in living room area(T2)	°C	Humidity in teenager room 2 (RH_8)	%
Humidity in living room area (RH_2)	%	Temperature in parents room(T9)	°C
Temperature in laundry room area(T3)	°C	Humidity in parents room(RH_9)	%
Humidity in laundry room area(RH_3)	%	Temperature outside (from Chievres weather station)(T_out)	°C
Temperature in office room (T4)	°C	Pressure (from Chievres weather station)(Press)	mmH
Humidity in office room (RH_4)	%	Humidity outside (from Chievres weather station)(RH_out)	%
Temperature in bathroom (T5)	°C	Wind speed (from Chievres weather station)	*m/s*
Humidity in bathroom(RH_5)	%	Visibility (from Chievres weather station)	*km*
Temperature outside the building (north side)(T6)	°C	Tdewpoint (from Chievres weather station)	A°C
Random variable 1(rv1)		Random variable 2(rv2)	

$$\overline{\mathrm{R}}(X)=\frac{1}{\mathrm{T}(\mathrm{X})}\sum_{\mathrm{t}=1}^{\mathrm{T}(\mathrm{X})}\left[\mathrm{r}(X)+es_t+ei_t+er_t\right] \tag{3}$$

For the embedded feature selection algorithm based on evolutionary algorithm, the number of occurrences of the i th individual in the g th generation population can be expressed by $\mathrm{T}(\mathrm{p_{i,g}})$.

$$\overline{\mathrm{R}}\left(p_{i,g}\right)=\frac{1}{\mathrm{T}\left(p_{i,g}\right)}\sum_{\mathrm{t}=1}^{\mathrm{T}\left(p_{i,g}\right)}\left[\mathrm{r}\left(p_{i,g}\right)+es_t+ei_t+er_t\right] \tag{4}$$

It is impossible to accurately count $T(p_{i,g})$ and retain the prediction result $R_t(p_{i,g})$ relative to it, which will consume a lot of system resources. However, we can use frequency item statistical techniques to calculate the number of occurrences of $p_{i,g}$ and the cumulative value of $R_t(p_{i,g})$.

$$\widetilde{\mathrm{R}}\left(p_{i,g}\right)=\frac{1}{\widetilde{\mathrm{T}}\left(p_{i,g}\right)}\sum_{\mathrm{t}=1}^{\widetilde{\mathrm{T}}\left(p_{i,g}\right)}\left[\mathrm{r}\left(p_{i,g}\right)+es_t+ei_t+er_t\right] \tag{5}$$

Therefore, in the embedded feature selection algorithm based on evolutionary algorithm, we can replace $R_t(X)$ by $\widetilde{R}(p_{i,g})$ when evaluating the fitness of each feature combination.

3.4. FI and EC based embedded feature selection algorithm

Genetic algorithm is an adaptive heuristic search algorithm. In embedded feature selection algorithm, it can be used to find the optimal combination of features in the high-dimensional feature space M. The main difference between traditional genetic algorithm based embedded feature selection algorithm (GAEFS) and the proposed frequency item based GAEFS algorithm (FI-GAEFS) is the calculation process of fitness function. In GAEFS, the fitness values of individuals were calculated with formula (1). In FI-GAEFS, the fitness values of individuals were calculated with formula (5).

The proposed FI-GAEFS algorithm for feature selection is presented below and illustrated in Fig. 4.

(1) Initialization: initialize population P_g, data structure DS(p,f,sum), where g is the generation counter, i is the individual counter, p is the frequency item, f is the frequency of p, sum is the sum of fitness function value of p, t is the window size, and r is sampling rate;
(2) Set the individual counter: i = 1;
(3) Calculate the accuracy of the current regression learner $p_{i,g}$, $R_t(p_{i,g})$;
(4) Judgment: Does the frequency item $p_{i,g}$ appear in the data structure DS? Yes, go Step (5); No, go Step(6);
(5) Update the data structure DS: the occurrence number $f_{p(i,g)} = f_{p(i,g)}+1$, cumulative value of $R_t(p_{i,g})$, $sum_{p(i,g)} = sum_{p(i,g)}+R_t(p_{i,g})$; Go (7);
(6) Add item in data structure DS: Add new individuals $p_{i,g}$, the frequency of it (f = 1), and the accuracy of it ($R_t(p_{i,g})$);
(7) Judgment: Is the frequency item statistics window met? Yes, go Step (8); No, go Step(9);
(8) Update the data structure DS: do $f_{p(i,g)} = f_{p(i,g)}$-1; $sum_{p(i,g)} = sum_{p(i,g)}$-$sum_{p(i,g)}/(f_{p(i,g)}+1)$ until rand(1)<0.5. If $f_{p(i,g)}$=0, delete ($p_{i,g}$, $f_{p(i,g)}$, $sum_{p(i,g)}$). Update sampling rate r = r × 2;
(9) Judgment: Have all the individuals in the population been counted? No, counter i = i+1, go to Step(3); Yes, go to Step (10);
(10) Revise prediction accuracy for individual $p_{i,g}$: if $f_{p(i,g)} > 1$, $R_t(p_{i,g}) = sum_{p(i,g)}/f_{p(i,g)}$; otherwise, $R_t(p_{i,g}) = R_t(p_{i,g})$;
(11) Selection/Crossver/Mutation: Do a traditional selection/crossover/mutation operation;
(12) Check: generation counter g = g+1, repeat steps (2) to (12) until the stopping criterion is met.

4. Simulation experiments

4.1. Experimental platform and data sets

In this section, two energy prediction data sets are used to evaluate the performance of the FI-GAEFS algorithm. Among them, Data set-1 is from the National Alliance of Big Data for New Energy Vehicles, which contains 37 state parameters of Beijing No.51 bus. The recording time is from May 20, 2020 to May 20, 2021, and the acquisition interval is 15 s. Data set-2 is the appliance energy consumption prediction data set, which is from UCI machine learning repository and contains 29 parameters. The recording time is from January 11, 2016 to May 27, 2016, and the acquisition interval is 10 min. The specific parameters are shown in Table 2 and 3 below. The programming language is MATLAB2021a and the hardware configuration is as follows: Intel i7-7700hq processor, 8 GB memory, 64-bit system. In this section, Gaussian process regression algorithm is used as the construction method of regression model. Gaussian process regression is a kind of machine learning regression method based on statistical mathematics theory, which can avoid local optimization in parameter optimization, and can well solve complex problems such as high dimension and nonlinear.

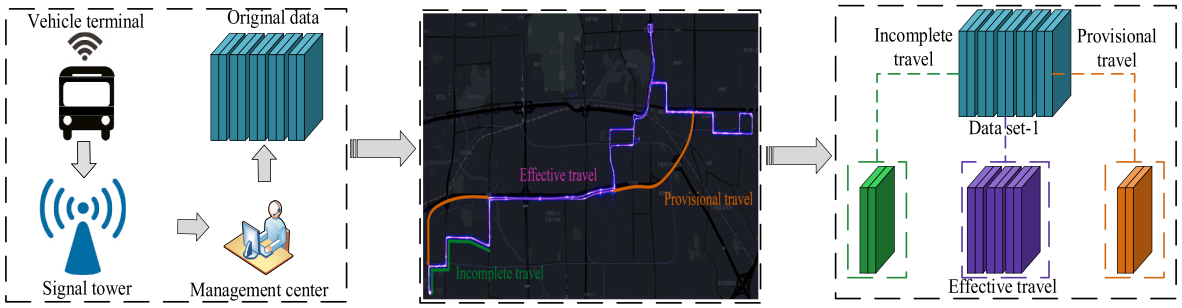
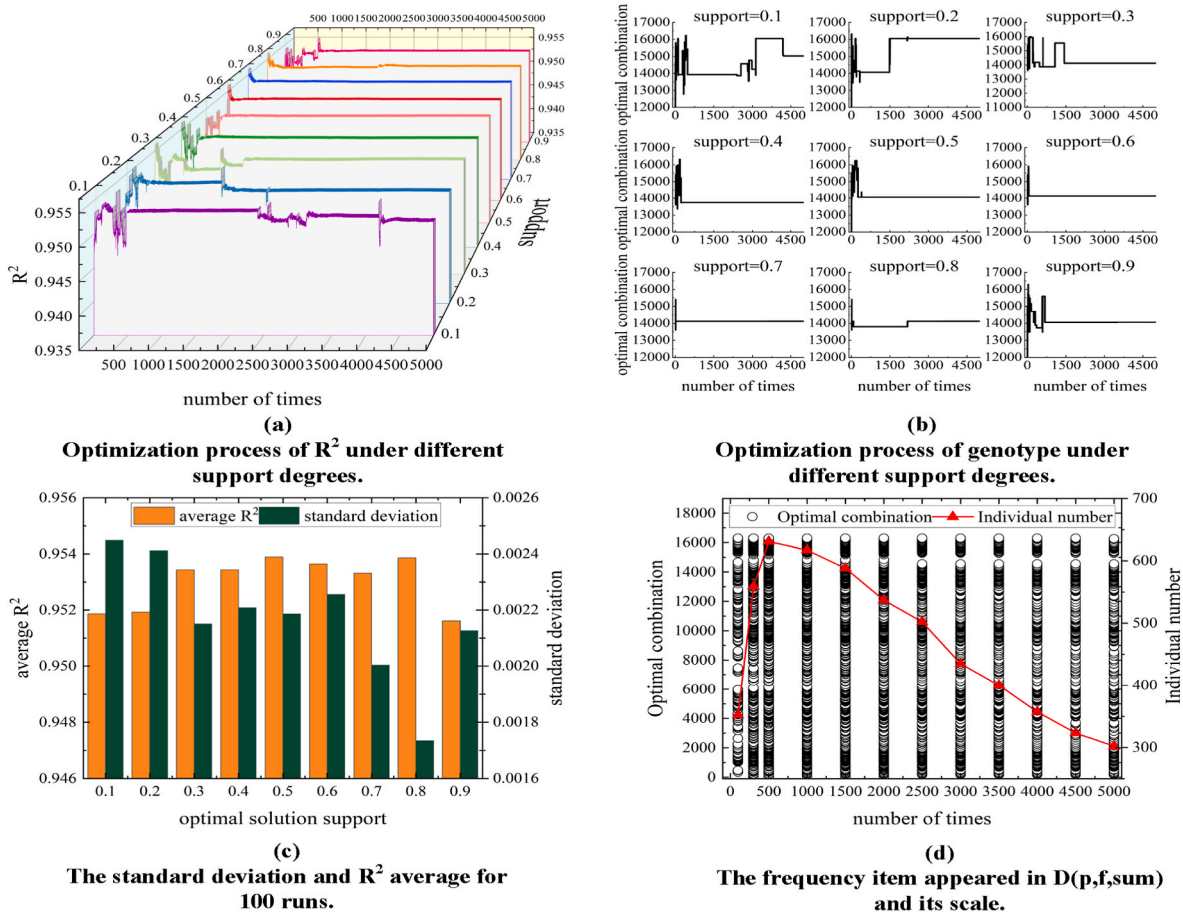


Fig. 5. Diagram of effective route division in Data set-1.



(a)
Optimization process of R^2 under different support degrees.

(b)
Optimization process of genotype under different support degrees.

(c)
The standard deviation and R^2 average for 100 runs.

(d)
The frequency item appeared in D(p,f,sum) and its scale.

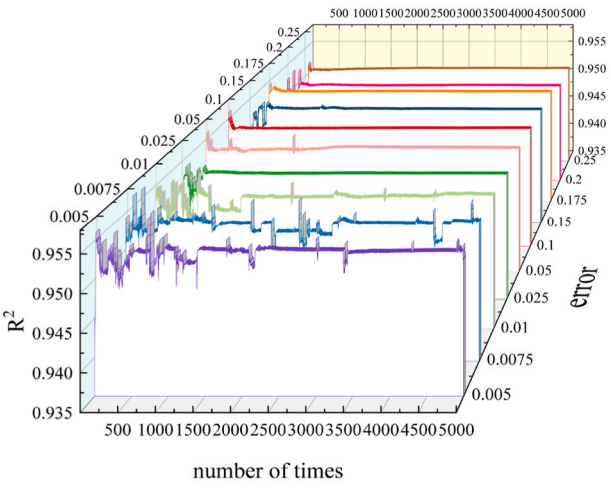
Fig. 6. Influence of different support threshold s.

4.2. *Effective route division*

In the original data set −1, the route of bus 51 is fixed. Its upstream line has 28 stations with a total length of 18 km, and its downstream line has 26 stations with a total length of 15 km. Due to sensor failure, environmental interference, temporary scheduling, and other reasons, some operating lines in the data set are incomplete or out of range. As shown in Fig. 5, the purple line is a complete running route, the orange line is an illegal running route, and the green line is an incomplete running route. In the experiment, the real-time records corresponding to incomplete and invalid route were deleted, and the real-time records sent by the vehicle terminal on the effective route were used to predict the line energy consumption of bus 51.

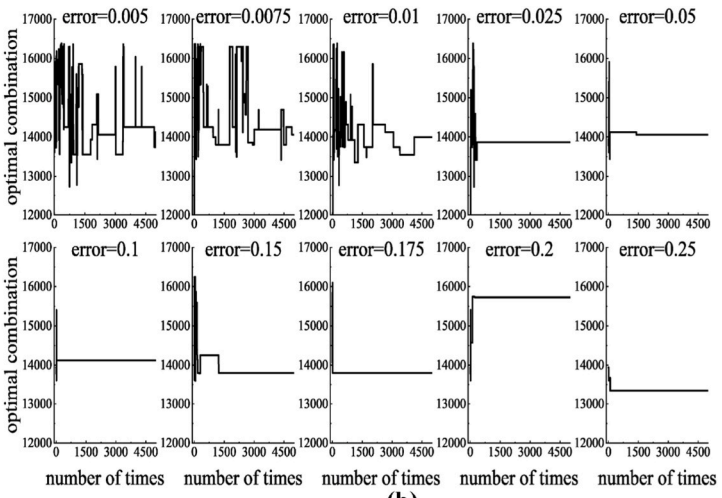
4.3. *Support threshold s*

For FI-GAEFS algorithm, limiting the support s of individuals in the offspring population to be greater than a certain threshold is the basis for ensuring the stability of the algorithm. In the training process of embedded feature selection algorithm, the output result of a regression model constructed by the same input feature combination is uncertain due to the interference of random factors from different sources. A stable and optimal regression model can be obtained by limiting the number of repetitions of these feature combinations using the minimum support parameter s. As can be seen from Fig. 6(a), when the support is set to 0.1, 0.2, …, 0.9, the speed of the algorithm to find the stable solution is different. The high support helps the algorithm find a stable solution quickly. For example, when the support is greater than 0.4, the algorithm can find a stable solution within the first 100 generations. When the support is 0.2, the algorithm needs 1500 generations to find a stable



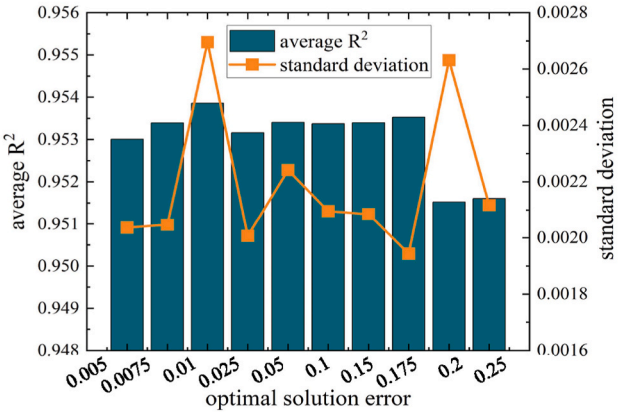
(a)

Optimization process of R^2 under different support degrees.



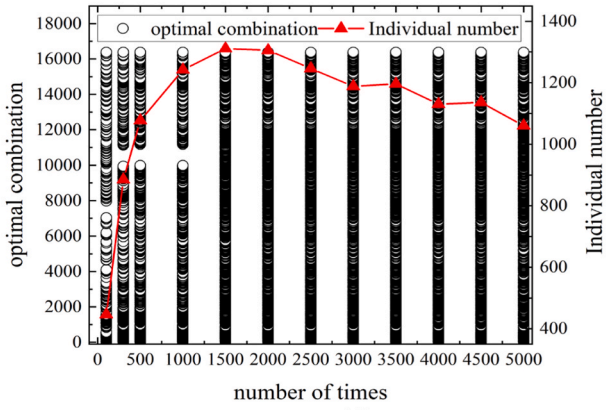
(b)

Optimization process of genotype under different support degrees.



(c)

The standard deviation and R^2 average for 100 runs.



(d)

The frequency item appeared in D(p,f,sum) and its scale.

Fig. 7. Influence of different error threshold ϵ.

Table 4
Pearson correlation coefficient and ReliefF coefficient on Data set-1.

Parameter	PCC	ReliefF	Parameter	PCC	ReliefF
Acceleration pedal travel value	−0.110	0.0001	Brake pedal state	−0.174[b]	0.0004
Drive motor controller temperature	0.084	0.0003	Motor temperature	0.024	0.0004
Max battery voltage	−0.896[a]	0	Min battery voltage	−0.896[a]	0
Motor controller input voltage	−0.897[a]	0.0032	Accumulated mileage	−0.094	0.0002
Total voltage	−0.897[a]	0.0032	Total current	0.797[a]	0
Driving motor torque	−0.093	0.0115	Month	−0.240[a]	0
Motor controller dc bus current	−0.084	0.0054	Hours	−0.205[a]	0

[a] ,At 0.01 level (double tail), the correlation is significant.
[b] ,At 0.05 level (double tail), the correlation is significant.

solution. However, bigger is not always better. When the support s is set to a higher level, better feature combinations with lower support cannot enter the frequency item data structure, and the algorithm cannot rapidly converge to the optimal solution. For example, when the support is set to 0.9, it takes 1000 generations for the algorithm to find a good combination of features. In addition, since different features are mapped to different weight bits on the Y-axis(Fig. 6(b)) during the coding process, the combination of features that are far away on the Y-axis does not mean that their genotypes are far apart. When the characteristics that have little influence on the prediction results appear randomly in the genotypes, the two genotypes apart on the Y-axis seem to be very different, but in fact the main characteristics are basically the same.

As can be seen from Fig. 6(b), the optimal solution varies with the range of support. When the support is set too low (<0.3), the optimal solution found by the algorithm is obviously different from the normal value (>0.3). In addition, we compared the optimal feature combinations found under different support. One hundred regression learners were constructed repeatedly with each combination, and the R^2 and standard deviation of these learners were compared. It can be seen from Fig. 6(c) that learners constructed from feature combinations with too low support have not only too small R^2 but also too large standard deviation. Therefore, the support should be set within a reasonable range (such as >0.3 and < 0.9) to ensure that the algorithm converges to the appropriate optimal solution. In order to show the variation of the storage space occupied by the algorithm during its operation, we gave the feature combinations stored in the frequency item data structure and the number of these combinations. Fig. 6(d) shows the number of feature

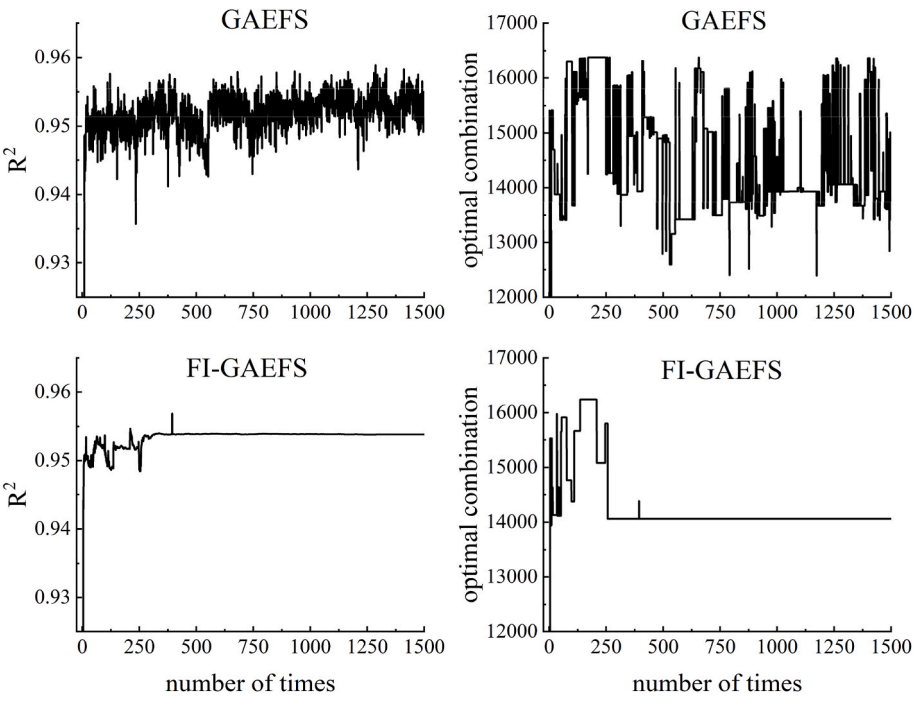


Fig. 8. Comparison between GAEFS algorithm and FI-GAEFS algorithm on Data set-1.

Table 5
Comparison of four algorithms on Data set-1.

Algorithm	Average R^2	Standard Deviation
ReliefF	0.806695088	0.012890603
Pearson correlation coefficient	0.811340292	0.011492252
GAEFS Algorithm	0.952347842	0.002170097
FI-GAEFS Algorithm	0.953890079	0.002186078

combinations and the features in the frequency item space of 5000 generations when the support is 0.6. It can be seen from Fig. 6(d) that the storage space occupied by frequency items does not always increase as the algorithm runs. With the running of the algorithm, it quickly reaches the peak and then gradually falls back.

4.4. Error ϵ

For FI-GAEFS algorithm, limiting the frequency of individuals in a population above a certain threshold is the basis of ensuring algorithm reliability. According to the parameter setting of the sticky sampling algorithm, the frequency f of an individual is determined by the support s, the error ϵ and the number of records N. When the support is fixed, too large error threshold will reduce the stability of the algorithm. When the support was set to 0.2, as shown in Fig. 7, although the algorithm converged to a stable solution with the error parameter increasing, it was not globally or nearly optimal solution ($\epsilon = 0.2$). On the other hand, too small error threshold prolongs the window update time for sampling and clipping the profile data in the data structure DS. Slow window updating and fast population evolution can cause the evolutionary algebra required for convergence to the optimal solution to become longer. As shown in Fig. 7(a and b), when the error is less than 0.01, the algorithm does not converge to a stable optimal solution within 5000 generations.

Under the condition of different error parameter values, 100 regression learners were repeatedly constructed using the optimal combination obtained. By comparing the R^2 and standard deviation of these learners, it can be seen from Fig. 7(c) that too low or too high error parameters are not conducive to improving the accuracy of the learner. When the error is less than 0.01 or greater than 0.2, the standard deviation is larger. Therefore, the error indicator should be set within a reasonable range (e.g. >0.025 and < 0.175). Similar to the analysis of support parameters, the storage space occupied by the algorithm during its operation changes as follows: with the operation of the algorithm, it quickly peaks and then gradually falls back. It peaked in the 1500 generation, this can be seen in Fig. 7(d).

4.5. Comparison of algorithms

For feature selection, the filter method calculates the feature weight independently according to the feature data, and the feature selection process is independent of the subsequent regression learner construction process. This method has high computational efficiency but low regression accuracy. The embedded feature selection method and the subsequent regression learner simultaneously complete the modeling and feature selection. The computational efficiency of this method is generally lower than that of the filter method, but the regression accuracy of it is higher. To compare the performance of different feature selection algorithms, four regression learners were constructed with Data set-1 using Pearson correlation coefficient algorithm (PCC), ReliefF algorithm (RF), genetic algorithm based embedded feature selection algorithm (GAEFS) and the proposed frequency item based GAEFS algorithm (FI-GAEFS), respectively. By comparing the construction speed and prediction accuracy of these four learners, the advantages and disadvantages of these four feature selection algorithms are indirectly shown. Wherein, the sampling frequency m of the ReliefF algorithm is set to 100, and the number of nearest neighbor samples k is set to 10. GA algorithm adopts elite retention strategy, and its crossover rate C and mutation rate M are set as 0.7 and 0.1, respectively.

In Table 4, Pearson correlation coefficient and ReliefF coefficient of the input features appeared in bus line energy consumption data set were shown. In RF algorithm, features with weight coefficients greater than 0.0003 are used to build the regression learner. In PCC algorithm, Pearson correlation coefficient with absolute value greater than 0.2 is used to construct regression learners. For GAEFS algorithm, the selection of feature columns is random. Influenced by random factors in the

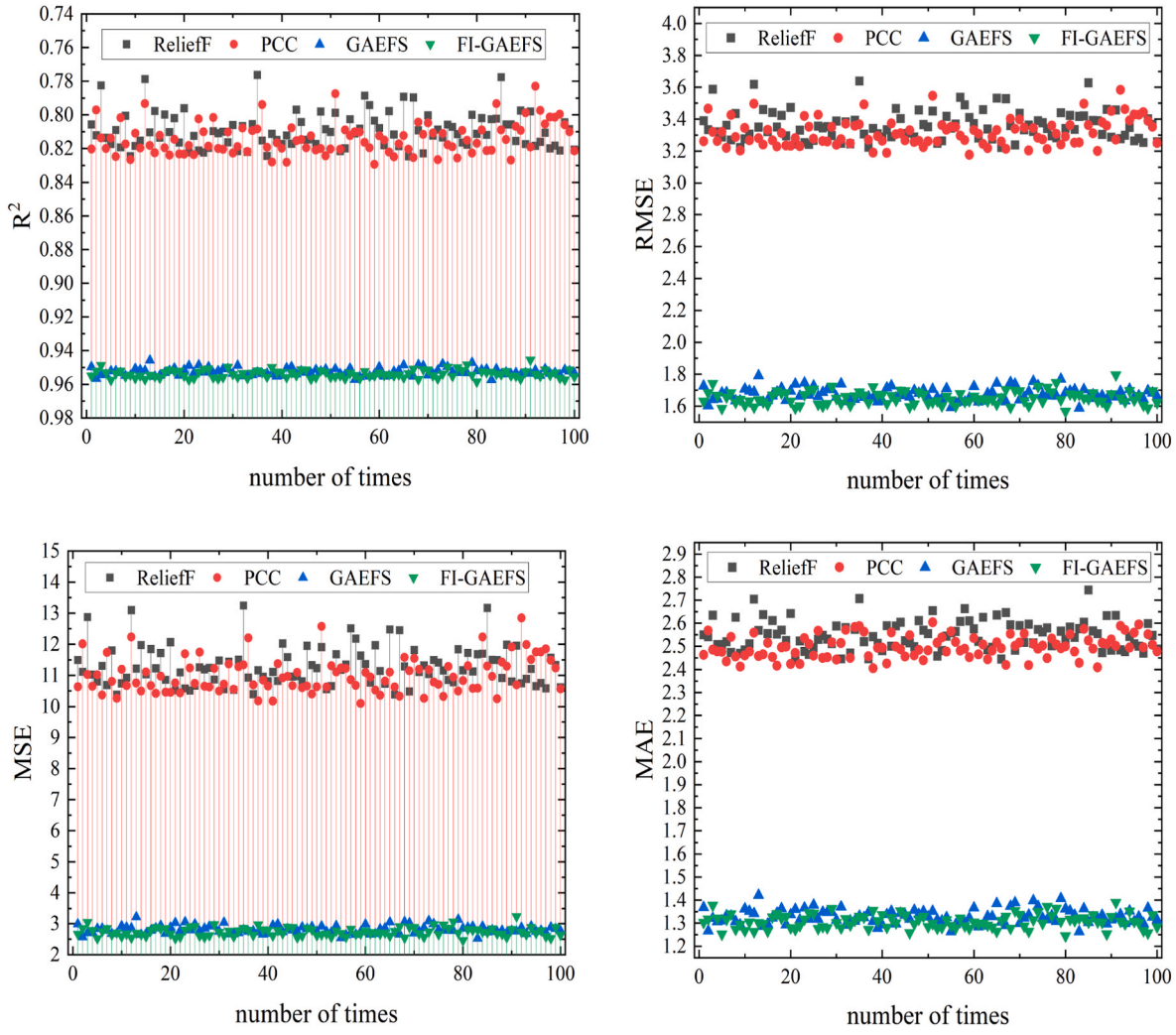


Fig. 9. Prediction accuracy of regression learner constructed by four algorithms on Data set-1.

Table 6
Pearson correlation coefficient and ReliefF parameters on Data set-2.

Parameter	PCC	ReliefF	Parameter	PCC	ReliefF
T1	0.258[a]	0.000390125	T7	0.262[a]	0.000408465
RH_1	0.380[a]	0.000762207	RH_7	−0.235[a]	0.001039359
T2	0.443[a]	0.000399139	T8	0.267[a]	0.000490226
RH_2	−0.268[a]	0.000292416	RH_8	−0.363[a]	0.000977318
T3	0.357[a]	0.000320913	T9	0.517[a]	0.000357963
RH_3	−0.294[a]	0.000262458	RH_9	−0.398[a]	0.000744544
T4	0.446[a]	0.000483187	T_out	0.344[a]	0.000304454
RH_4	0.192[b]	0.000673455	Press	0.197[b]	−8.79E-20
T5	0.079	0.000547498	RH_out	−0.341[a]	0.001795684
RH_5	0.061	0.001210472	Wind speed	0.346[a]	0.00047498
T6	0.338[a]	0.000416882	Visibility	−0.302[a]	0.001556347
RH_6	−0.354[a]	0.003588272	Tdewpoint	0.378[a]	−4.27E-06
rv1	−0.141	0.002182553			

[a] ,At 0.01 level (double tail), the correlation is significant.
[b] ,At 0.05 level (double tail), the correlation is significant.

process of data set sampling, algorithm initialization, computing platform resource scheduling, etc., the prediction result of the regression learner constructed with the same feature combination is also random. As shown in Fig. 8, during the evolution of the population, the R^2 of regression learner constructed by GAEFS algorithm and the genotype of the optimal individual in the population are unstable. This will hinder the normal convergence process of the standard genetic algorithm. However, the regression learner constructed by FI-GAEFS algorithm showed stability after 500 generations, and the genotype of the optimal individual gradually converged to a stable value.

In order to compare the advantages and disadvantages of the feature combinations selected by these four algorithms, we repeatedly constructed 100 regression learners with the optimal combinations found by these four algorithms, respectively. By comparing the prediction accuracy of the 400 learners, we found that the prediction accuracy of RF algorithm and PCC algorithm was low (the average value of R^2 was 0.8067 and 0.8113, and the standard deviation was 0.0129 and 0.0115, respectively). GAEFS algorithm and FI-GAEFS algorithm have high prediction accuracy (the average value of R^2 is 0.9523 and 0.9539, and the standard deviation is 0.00217 and 0.00219, respectively). This can be seen in Table 5. We further compared the prediction accuracy of the regression learner constructed by these four kinds of feature selection algorithms with different indexes and found that similar trends exist. The evaluation indexes were Coefficient Of Determination-R^2, Root Mean Square Error (RMSE), Mean Square Error (MSE), and Mean Absolute Error (MAE), as shown in Fig. 9.

4.6. Comparison of data sets

According to Shannon sampling law, only when the sampling frequency is more than twice the maximum frequency of the signal can the original continuous signal be completely reconstructed from the

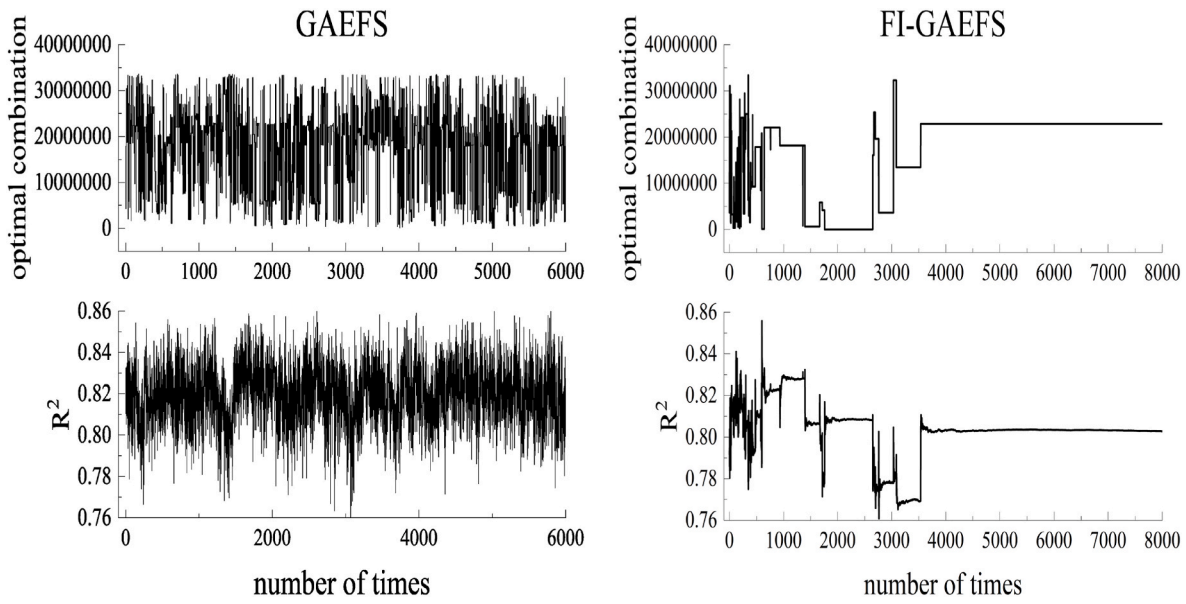


Fig. 10. Comparison of GAEFS and FI-GAEFS algorithms on Data set-2.

Table 7
Comparison of four algorithms on Data set-2.

Algorithm	Average R^2	Standard Deviation
ReliefF	0.668444878	0.030709954
Pearson correlation coefficient	0.798676099	0.021126411
GAEFS	0.781734787	0.021663309
FI-GAEFS	0.805035269	0.020495863

sampled sample. In the energy consumption prediction Data set-1, the collection interval of state parameters for the Beijing No. 51 bus is 15 s. The variation frequency of vehicle speed, acceleration pedal travel value, and other signals in the data set is actually greater than the sampling frequency. In order to minimize the influence of the distortion caused by signal overlap on the algorithm, we use the appliance energy prediction data set in the UCI machine learning knowledge base to test the effectiveness of the proposed algorithm. The sampling interval of this data set is 10 min, and the temperature and humidity of the kitchen, living room, and other rooms change slowly, the change frequency of them is much lower than the sampling frequency.

In Table 6, the Pearson correlation coefficient and ReliefF coefficient of the input features that appeared in the appliance energy prediction data set were shown. In the RF algorithm, features with weight coefficients greater than 0.0003 are used to build the regression learner. In the PCC algorithm, Pearson correlation coefficient with absolute value greater than 0.2 is used to construct regression learners. GAEFS algorithm and FI-GAEFS algorithm were iterated for 6000 and 8000 generations respectively, and the solution of the first 6000 iterations was selected as the optimal solution. As shown in Fig. 10, during the evolution of the population, the R^2 of the regression learner constructed by the GAEFS algorithm and the genotype of the optimal individual in the population are unstable. This will hinder the normal convergence process of the standard genetic algorithm. On the contrary, the regression learner constructed by FI-GAEFS algorithm showed stability in the evolutionary process of 8000 generations, and the genotype of the optimal individual converges to a stable value within 4000 generations.

In order to compare the feature combinations selected by these four algorithms, we repeatedly constructed 100 regression learners with the optimal combinations found by these four algorithms respectively. By comparing the prediction accuracy of the 400 learners, we found that the prediction accuracy of RF algorithm and PCC algorithm was low (the average value of R^2 was 0.6684 and 0.7987, and the standard deviation was 0.0307 and 0.0211, respectively). GAEFS algorithm and FI-GAEFS algorithm have high prediction accuracy (the average value of R^2 is 0.7817 and 0.8050, and the standard deviation is 0.0217 and 0.0205, respectively). This can be seen in Table 7. We further compared the prediction accuracy of the regression learner constructed by these four kinds of feature selection algorithms with different indexes and found that similar trends exist. The evaluation indexes were R^2, RMSE, MSE, and MAE, as shown in Fig. 11. Thus, by comparing the prediction results of different energy consumption prediction data sets, we can verify the effectiveness of the proposed FI-GAEFS algorithm.

5. Conclusions

In the engineering practice of using embedded feature selection algorithm to build electric vehicle energy consumption prediction model, the simulation results are often affected by random factors in the processes of data set sampling, algorithm initialization, computing platform resource scheduling, and so on, resulting in the failure to obtain a stable and optimal energy consumption prediction model. Therefore, a feature optimization method of energy consumption prediction for electric bus based on frequency item mining and genetic algorithm was proposed. The main advantages are: (1) The effect of random factors on the objective function is corrected by frequency item statistics, which ensures the stability of the optimization process of the embedded feature selection algorithm; (2) In the process of evaluating the regression learners, the prediction result of it is regarded as a data stream. Frequency item mining algorithm on the data stream can complete the frequency item statistics in a single, fast, and low-cost way. The main disadvantage is that, in order to effectively execute the stream-frequency item statistics algorithm, a small amount of additional memory is consumed to store the data structure DS used by the algorithm.

Credit author statement

Li Zhao: Conceptualization, Methodology, Project administration, Validation, Writing - review & editing. Yuqi Li: Conceptualization, Data curation, Formal analysis, Software, Validation, Visualization, Writing - original draft. Shuai Li: Conceptualization, Writing - review & editing. Hanchen Ke: Methodology, Writing - review & editing.

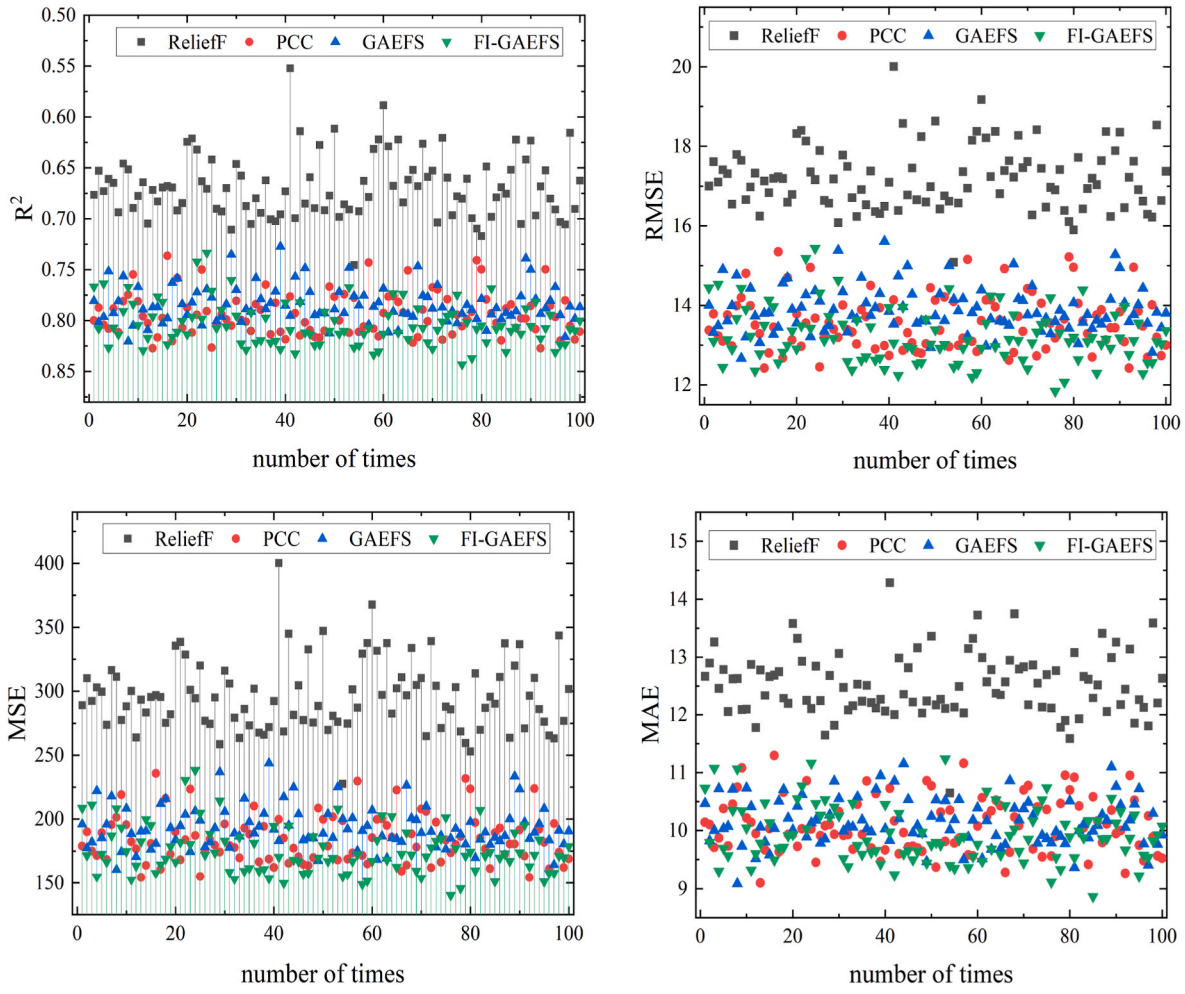


Fig. 11. Prediction accuracy of regression learner constructed by four algorithms on Data set-2.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

Acknowledgments

This work is supported by National Natural Science Foundation of China (52077007).

References

[1] Dzhang X, Zou Y, Fan J, Guo H. Usage pattern analysis of Beijing private electric vehicles based on real-world data. Energy 2019;167:1074–85. https://doi.org/10.1016/j.energ y.2018.11.0 05.

[2] Rodrigues JL, Bolognesi HM, Melo JD, Heymann F, Soares FJ. Spatiotemporal model for estimating electric vehicles adopters. Energy 2019;183:788–802. https://doi.org/10.1016/j.ene rgy.2019.06.117.

[3] Li Z, Khajepour A, Song J. A comprehensive review of the key technologies for pure electric vehicles. Energy 2019;182:824–39. https://doi.org/10.1016/j.energy.2019.06.077.

[4] Huo W, Li W, Zhang Z, Sun C, Zhou F, Gong G. Performance prediction of proton-exchange membrane fuel cell based on convolutional neural network and random forest feature selection. Energy Convers Manag 2021;243. https://doi.org/10.1016/j.enconman.2021.114367.

[5] Zhao L, Ke H, Huo W. A frequency item mining based energy consumption prediction method for electric bus. Energy 2023;263. https://doi.org/10.1016/j.energy.2022.125915.

[6] Li P, Zhang Y, Zhang Y, Zhang Y, Zhang K. Prediction of electric bus energy consumption with stochastic speed profile generation modelling and data driven method based on real-world big data. Appl Energy 2021;298. https://doi.org/10.1016/j.apenergy.2021.117204.

[7] Chen C-W, Tsai Y-H, Chang F-R, Lin W-C. Ensemble feature selection in medical datasets Combining filter, wrapper, and embedded feature selection results. Expet Syst 2020;37. 10.1111/exsy.12553.

[8] Urbanowicz RJ, Meeker M, La Cava W, Olson RS, Moore JH. Relief-based feature selection: introduction and review. J Biomed Inf 2018;85:189–203. https://doi.org/10.1016/j.jbi.2018.0 7.014.

[9] Liu X-Y, Liang Y, Wang S, Yang Z-Y, Ye H-S. A hybrid genetic algorithm with wrapper-embedded approaches for feature selection. IEEE Access 2018;6:22863–74. 10.1109/ACCESS.2018.2818682.

[10] Liu Y, Ye D, Li W, Wang H, Gao Y. Robust neighborhood embedding for unsupervised feature selection. Knowl Base Syst 2020;193. https://doi.org/10.1016/j.knosys.2019.105 462.

[11] Zheng W, Chen S, Fu Z, Zhu F, Yan H, Yang J. Feature selection boosted by unselected features. IEEE Transact Neural Networks Learn Syst 2022;33:4562–74. https://doi.org/10.1109/TNN LS.2021.3058172.

[12] Zhao H, Yu S. Cost-sensitive feature selection via the ℓ2,1-norm. Int J Approx Reason 2019;104:25–37. https://doi.org/10.1016/j.ijar.2018.10.017.

[13] Maldonado S, López J. Dealing with high-dimensional class-imbalanced datasets: embedded feature selection for SVM classification. Appl Soft Comput 2018;67:94–105. 10.1016/j.asoc.2018.02.051.

[14] Xu X, Wu X. feature selection under orthogonal regression with redundancy minimizing. 2020. p. 3457–345346 1. International Conference on Acoustics. Speech and Signal Processing ICASSP.

[15] Fan Y, Chen B, Huang W, Liu J, Weng W, Lan W. Multi-label feature selection based on label correlations and feature redundancy. Knowl Base Syst 2022;241.

[16] Zhang J, Luo Z, Li C, Zhou C, Li S. Manifold regularized discriminative feature selection for multi-label learning. Pattern Recogn 2019;95:136–50. https://doi.org/10.1016/j.patcog.2019. 06.003.

[17] Kumar V, Pujari AK, Padmanabhan V, Kagita VR. Group preserving label embedding for multi-label classification. Pattern Recogn 2019;90:23–34. https://doi.org/10.1016/j.patco g.2019.01.009.
[18] Shang R, Wang W, Stolkin R, Jiao L. Non-negative spectral learning and sparse regression-based dual-graph regularized feature selection. IEEE Trans Cybern 2018;48:793–806. https://doi.org/10.1109/TCYB. 2017.2657007.
[19] Amini F, Hu G. A two-layer feature selection method using Genetic Algorithm and Elastic Net. Expert Syst Appl 2021;166. https://doi.org/10.1016/j.eswa.2020.114072.
[20] Dornaika F. Multi-layer linear embedding with feature subset selection. Knowl Inf Syst 2021;63:1029–43. https://doi.org/10.1007/s10115-020-01535-3.
[21] Manku G, Motwani R. Approximate frequency counts over data streams. VLDB '02. 2002. p. 346–3435 7. https://doi.org/10.14778/2367502.2367508. Proceedings of the 28th International Conference on Very Large Databases.
[22] Katoch S, Chauhan SS, Kumar V. A review on genetic algorithm: past, present, and future. Multimed Tool Appl 2021;80:8091–126. https://doi.org/10.1007/s11042-020-10139-6.
[23] Du W, Zhang M, Ying W, Perc M, Tang K, Cao X, et al. The networked evolutionary algorithm: a network science perspective. Appl Math Comput 2018;338:33–43. https://doi.org/10.1016/j.amc.2018.06.002.