Applied Soft Computing Journal 89 (2020) 106051

Contents lists available at ScienceDirect

Applied Soft Computing Journal

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

Hybridization of cognitive computing for food services

Xiaobo Zhang [a], Senbin Yang [a], Gautam Srivastava [b,c,*], Mu-Yen Chen [d], Xiaochun Cheng [e]

[a] Faculty of Automation, Guangdong University of Technology, Guangzhou, 510006, China
[b] Department of Mathematics and Computer Science, Brandon University, Brandon R7A 6A9, Canada
[c] Research Center for Interneural Computing, China Medical University, Taichung 40402, Taiwan, Republic of China
[d] Department of Information Management, National Taichung University of Science and Technology, Taichung 404, Taiwan
[e] Department of Computer Science, School of Science & Technology, Middlesex University, London NW4 4BT, UK

ARTICLE INFO

Article history:
Received 24 September 2019
Received in revised form 16 December 2019
Accepted 25 December 2019
Available online 2 January 2020

Keywords:
Linear regression
Random forest
Cognitive computing
Food service robots
Data mining
Restaurant industry
Fusion

ABSTRACT

The application of data mining technology to food services and the restaurant industry has certain social value. By predicting customer traffic and needs, a restaurant can prepare a reasonable amount of meals for customers according to predicted needs which is conducive to improving the dining experience of customers and also improving the quality of food preparation and making the restaurant itself operate more efficiently. In recent years, we have seen the use of collaborative robots for use in the fast food industry. In Asia and more specifically in Japan, we have seen many fast-food chains implement the use of robots to better serve their customers. By studying the linear regression algorithm and the random forest algorithm, this paper proposes a new interwoven novel fusion approach of combining both algorithms and applies the new model to restaurant data to assist in the prediction of customer traffic in the restaurant industry. This predictive algorithm using cognitive techniques can assist these newly place robots in the food industry better serve their client base and in doing so make the industry more efficient. Experimental, comparison, and analysis are reported in the paper. The error rate of the fusion solution is reduced by approximately 5.503% compared with the linear regression algorithm and is approximately 3.719% lower than the error rate of the random forest algorithm. Results show that the new fusion algorithm can achieve better prediction results of customer traffic prediction for the restaurant industry. Furthermore, we also provide a new take on the application of data mining technology in the restaurant industry itself.

1. Introduction

Data mining (DM) is the use of related algorithms to extract useful, unknown, comprehensive or user-interested knowledge from a large number of incomplete, noisy, fuzzy, or random data. DM uses the extracted knowledge to establish models for decision support, and provide prediction methods, tools and processes [1–4]. DM can also be thought of as the process of searching for hidden information from an extensive amount of data through algorithms. With the advent of the Internet and the explosion of data in general, DM technology is widely and urgently applied in various fields such as finance, telecommunications, insurance, medical, food service, and many other industries. It mainly uses sorting, analysis, summarization, reasoning and other methods to process a large amount of data, so as to construct and analyze real world problems, and obtain relevant predicted results to make more favorable decisions for future use [5–8]. We have seen similar work undertaken sparingly, however we direct interested readers to the following papers [9–11].

The linear regression (LR) algorithm and the random forest (RF) algorithm are widely machine learning cognitive computing techniques mostly used to facilitate many facets of people's lives. For example, the application of the LR algorithm in household income and expenditures [12], and estimated resistivity [13] are just some of its many uses. After Breiman proposed the RF algorithm, due to its good performance, the algorithm is widely used in:

- classification/regression of gene sequences in biological information [14]
- monitoring and tracking of the human body
- gesture and action recognition
- face recognition
- gender recognition
- behavior and event recognition in the field of computer vision [15,16]

* Corresponding author at: Department of Mathematics and Computer Science, Brandon University, Brandon R7A 6A9, Canada.
E-mail addresses: zxb_leng@gdut.edu.cn (X. Zhang), srivastavag@brandonu.ca (G. Srivastava), mychen@nutc.edu.tw (M.-Y. Chen), X.Cheng@mdx.ac.uk (X. Cheng).

https://doi.org/10.1016/j.asoc.2019.106051
1568-4946/

- speech recognition
- synthesis in the speech domain [17]
- anomaly detection
- metric learning in the field of data mining [18,19]
- application in 3D garment prototyping [20].

Although the LR algorithm and RF algorithm are widely used in people's lives, and in many cases, they are based on the application of a single model. The advantages of each model are fully utilized to improve the effect in general. The stacking algorithm is an integrated learning technology that combines multiple learners and combines the advantages of multiple models. It has become very widely used. Because of its advantages, the stacking algorithm has been extensively studied by researchers and continuously improved in research. Based on the theoretical basis of the stacking algorithm, this paper combines LR and RF, two algorithms to predict data, further optimize it, and improve the performance and effect of our new fusion algorithm. [21,22].

Interactions between consumers and humanoid service robots (HSR) or just service robots will soon be part of routine marketplace experiences and we are already seeing them used in the fast food industry [23,24]. It is unclear, however, whether these robots (compared with human employees) will trigger positive or negative consequences for consumers and companies. To better serve their clientele, an understanding of the customer traffic will better equip employers with the necessary information to train their robots with. For example, more than 10,000 humanoid "Pepper" robots have been sold worldwide since their launch in 2014 [25]. Pepper helps sell coffee machines at 1,000 Nescafe stores in Japan and has worked as a waiter at Pizza Hut in Asia. If these robots were properly trained with customer traffic information, their downtime and numbers good be adjusted accordingly for owners.

We have seen ample recent work relating Artificial Intelligence techniques to niche problems in Industry. Zhang et al. applied AI techniques to the recent emerging Internet of Vehicles field [26]. Lu et al. improved deep sea organism tracking using deep learning [27,28]. We have also seen work in human factors like brain intelligence and wound correction using AI techniques [29,30]. Overall, problems as tackled in this paper have shown to find novel and strong solutions using AI techniques that are either already defined or novel and new like the fusion algorithm presented here.

In this paper, we focus on customer traffic prediction in the restaurant industry. Food services is an important part of our daily lives, and luckily is accompanied by huge behavioral data. If we can make better use of this data, using DM technology to analyze and predict future events, we can assist the restaurant industry to make more reasonable plans and decisions. This will be beneficial to both the customer's dining experience and improve the quality of food. Moreover, the restaurant industry will operate better. Relevant information shows that the application of DM technology in the restaurant industry needs to be improved [22]. For example, a LR model or RF cognitive computing model studied in the past has been shown to still be inaccurate in the restaurant industry [19]. This paper proposes a new solution to this deficiency, integrating the two cognitive computing models into a new model to improve the application of DM technology in the restaurant industry.

The rest of the paper is organized as follows. In Section 2, we first introduce the two main cognitive computing models of LR and RF, and then use DM tools to analyze and predict large amounts of data in the restaurant industry in Section 3. Lastly, in Section 3.3.2, we compare the predicted results between LR, RF, and the combined new fusion model. The experimental results show that the combined model has a better effect and higher accuracy in the customer needs prediction of the restaurant industry. We conclude first with some future directions in Section 4 and finally with some closing remarks in Section 5.

Table 1
Notational table.

Description	Notation
Weight vector	$\vec{w}$
Feature vector	$\vec{x}$
Number of features	n
Loss function	$J(\vec{w})$
Weights	w^j
Threshold	ϵ
Distributed random vector	Θk
Input vector	x
Training data set	T
Data set	D

2. Reviews of linear regression and random forest model

2.1. Notation

We refer our readers to Table 1 for a summary of the Notation used in this paper.

2.2. Linear regression model

For the LR model $f(\vec{x}) = \vec{w} \bullet \vec{x} + b$, the weight vector $\vec{w} = (w^0, w^1, w^2, \dots, w^n)^T$, the feature vector $\vec{x} = (x^0, x^1, x^2, \dots, x^n)^T$, where n is the number of features. The aim of the algorithm is to minimize the error between the predicted value and the actual value, or the value of all the sample points in the data set to the required straight line distance to be the smallest [31,32]. According to this idea, loss function for the mean squared error could be inferred.

$$J(\vec{w}) = \frac{1}{2m}\sum_{i=1}^{m}(f(\vec{x_i}) - y_i)^2 = \frac{1}{2m}\sum_{i=1}^{m}(\vec{w} \bullet \vec{x_i} + b - y_i)^2 \tag{2.1}$$

Solving the mean square error loss function by using the gradient descent method.

Step 1. Randomly initialize parameter $\vec{w}$.

Step 2. Calculate the weights w^j. The equation is as follows, in which a is learning rate, also called step size.

$$w^j : w^i - a\frac{\partial J(\vec{w})}{\partial \vec{w^j}} = w^j - a\frac{1}{m}\sum_{i=1}^{m}\left[(\vec{w} \bullet \vec{x_i} + b - y_i)x_i^0\right] = \vec{w^j} - a\frac{1}{m}\left[(f(\vec{x_i}) - y_i)x_i^j\right] \tag{2.2}$$

Step 3. If the decline distance of all gradients $w^{(j)}$ is less than the set threshold ϵ, then the algorithm ends and output the last parameters, or turns to Step 2.

Commonly, in order to prevent the model from overfitting, it is often necessary to add a regularization term to the loss function when building a linear regression model. Generally, there are regularizations of L1 and L2. The loss function after adding the L1 regularization is expressed as:

$$J(\vec{w}) = \frac{1}{2m}\left[\sum_{i=1}^{m}(\vec{w} \bullet \vec{x_i} + b - y_i)^2 + \lambda \parallel w \parallel_1\right] \tag{2.3}$$

The loss function after adding the L2 regularization is expressed as:

$$J(\vec{w}) = \frac{1}{2m}\left[\sum_{i=1}^{m}(\vec{w} \bullet \vec{x_i} + b - y_i)^2 + \frac{\lambda}{2} \parallel w \parallel_2^2\right] \tag{2.4}$$

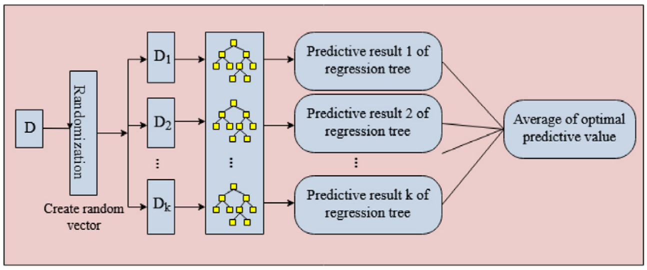


Fig. 1. The process of RF generation.

2.3. Review for random forest algorithm

RF is a classification tree based algorithm [33]. This algorithm requires simulation and iteration and is classified as a method in ML. In the 1980s, Breiman et al. developed an algorithm for classification and regression tree (CART) [34]. By repeating the binary data for classification or regression, it was shown that the computational load will be greatly reduced [35,36]. RF for regression prediction is a regressor composed of a series of tree regressors $h(x, \Theta)k, k = 1, \ldots$, in which Θk is an independent and identically distributed random vector, and each tree has a predicted value for the input vector x [37]. The random forest generation steps are shown in Fig. 1. We analyze the steps in Fig. 1 as follows:

1. $D_1, D_2, \ldots, D_k$, and build k regression trees;
2. In each node of the regression tree randomly selects m out of n indicators, and selects the optimal segmentation index for segmentation;
3. Repeat Step 2 and traverses the k regression trees;
4. Forming a random forest from k regression trees.

Each node (not necessarily a leaf node) of the regression tree will get a predicted value equal to the average of all values belonging to this node. When branching, each threshold of each feature is exhausted to find the best segmentation point, and the measure is to minimize the mean square error. The most reliable branching basis can be found by minimizing the mean square error. Branching until the value of each leaf node is unique or reaches the preset termination condition (such as the upper limit of the number of leaves). If the value of the final leaf node is not unique, then the average of all values on this leaf node is used as the predicted value [38].

The single regression tree in random forests is shown in Fig. 2. We see clearly here that as given threshold values t_i for each decision i of the tree, either a greater than or equal to t will take the left path or less than t will follow the right path respectively.

2.4. Principle of linear regression and random forest algorithm fusion

2.4.1. Review of stacking algorithm

The Stacking algorithm combined with multiple different classification algorithms can be regarded as a special combination strategy [39,40]. Stacking is mainly divided into two layers, the learner in layer 0 is called a primary learner, and the learner in layer 1 is called a secondary learner. First, a plurality of primary learners are trained by using the original feature data as input, and then the output of the primary learner is used as a feature for training the secondary learner, as shown in Fig. 3. The initial training data is sent to n learners, who in turn share their respective results with a secondary learner to create a final classification result.

The specific method is as follows:

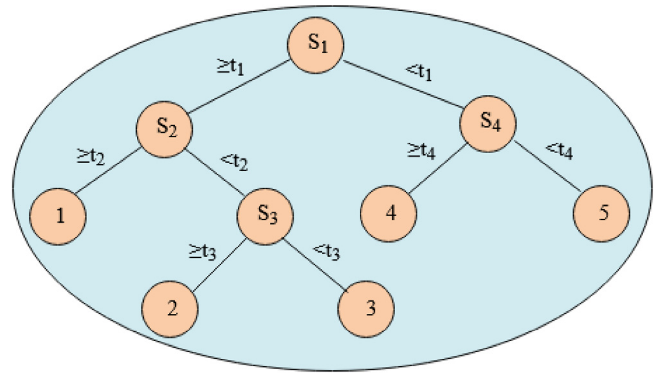


Fig. 2. Single regression tree.

Train the learners in the primary learning phase using the A-fold cross-check method. The initial training data set $T = ((x_1, s_1), (x_2, s_2), \ldots, (x_n, s_n))$ is divided into sub-datasets of similar size $T_1, T_2, \ldots, T_k$, where $T - T_j$ will be used as training data of the m-learning algorithm and obtain a learner L_m^j, and then input T_j as test data into L_m^j. For each sub-dataset, this operation is performed using the m-learning algorithm. Finally, each sample is tested and the result y_{im} is output. If there are n learning algorithms, for each sample x_i after the training, n results will be generated, and they will form a new feature vector $y_i = (y_{i1}, y_{i2}, \ldots, y_{in})$ as the training data of the secondary learner, and the mark will remain the original s_i. We can see details of this algorithm given in Algorithm 1 which runs in $\mathcal{O}(n \log n)$ where n is the size of the input training data.

Algorithm 1 Stacking algorithm for the Fusion method

Require: Input: training data $D = (x_i, y_i)_{i=1}^m$
function Step 1(learn base classifiers)
  for $t \leftarrow 1$ to T **do**
    Learn h_t based on D
  end for
end function
function Step 2(construct new data set of predictions)
  for $i \leftarrow 1$ to m **do**
    $D_h = x_i, y_i$, where $x_i = (h_1(x_1), \ldots, h_t(x_i))$
  end for
end function
function Step 3(learn a meta-classifier)
  learn H based on D_h
  return H
end function
Require: Output: ensemble classifier H

2.4.2. The optimized stacking fusion method

In this paper, the stacking algorithm fusion method is deeply studied, and the LR algorithm and the RF algorithm are fused into a fusion method.

The improved stacking fusion method is as follows and is summarized in Fig. 4:

Step 1. Using linear regression and random forest algorithm as the primary learner, the primary classifier is trained to predict the original data set and obtain the predicted value y_{i1}^{lr} and y_{i1}^{rfr}.

Step 2. Since the prediction effect of the weighted average of the two cognitive computing models is better than the predicted value of a single model, the idea of the stacking fusion algorithm is to train multiple primary learners through the original feature data as input, and then use the output of the primary learner as

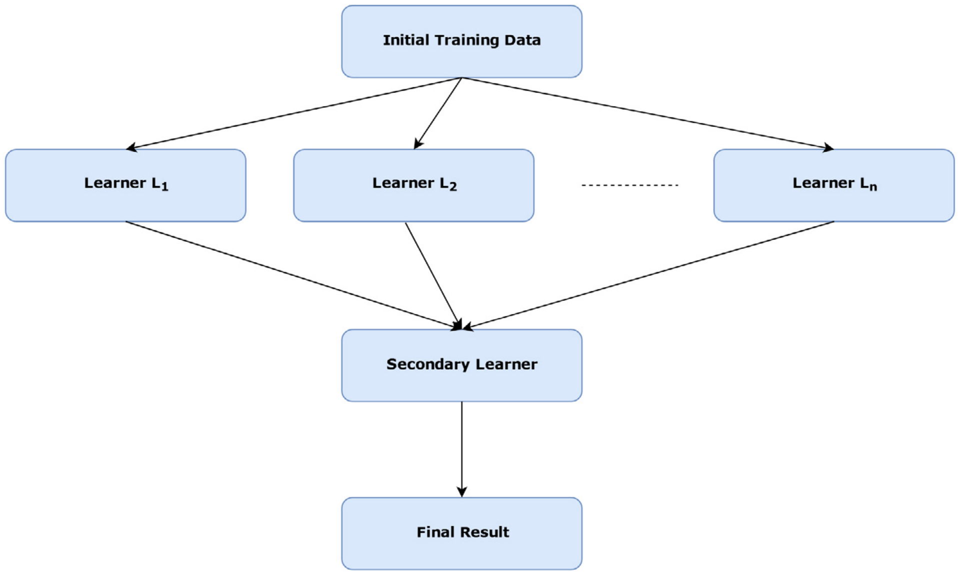


Fig. 3. Stacking method.

a feature for training a secondary learner. Based on this principle, the weighted average can be viewed as an output produced by the primary learner that is closer to the actual value than the single model output, which is used as part of the secondary learner input. A weighted average of the predicted values y_{i1}^{lr} and y_{i1}^{rfr}, obtained from a single model in 1. y_{i1} is the result obtained by weighted averaging, a_1 represents the weight of the obtained linear regression model, and b_1 represents the weight of the obtained random forest model. Then we can get the following equation.

$$y - i1 = y_{i1}^{lr} * a_1 + y_{i1}^{rfr} * b_1 \tag{2.5}$$

Step 3. Using linear regression and random forest as secondary learners respectively, combine the weighted average predicted value with the primary cognitive computing model predicted value as the input of the secondary learner $(y_{i1}^{lr}, y_{i1}^{rfr}, y_{i1})$.

Step 4. Train the secondary learner, output the final strong classifier $H(x)$, and use the strong classifier $H(x)$ to predict the new input data set to get the final predicted value y_{pred}. The optimization method is as shown in Fig. 4

2.4.3. The stacking fusion method versus LR and RF

The main idea of stacking is to call the layer 0 learners as the primary learner, and layer 1 learners as the secondary learner. First, a plurality of primary learners are trained by using the original feature data as input, and then the output of the primary learner is used as the feature for training the secondary learner. This paper uses the LR and RF algorithms as the primary learners and uses the stacking fusion method training the two primary learners through the original feature data as input. The output of the two learners and the weighted average of the two outputs are used as features to train the secondary learner for the cognitive computing model. The secondary learner in this paper uses a linear regression algorithm. This is the main idea of the linear regression and random forest algorithm fusion introduced in this paper.

2.5. Model evaluation criteria

For the prediction of customer traffic, we are more concerned about the traffic error between the predicted value and the actual value [41]. Therefore, in this paper we use the root mean square error (RMLSE) as a standard for evaluating the model's effectiveness. Among them, the root means square error formula is as follows [42]:

$$RMLSE = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(observed - -predicted)^2} \tag{2.6}$$

In which, observed indicates the actual number of passengers, predicted indicates the predicted number of passengers. RMLSE is also called the error index. When RMLSE is smaller, it indicates that the smaller the error, the better the model effect.

Table 2
12 attributes per record after pre-processing.

air_store_id	is_true	visit_date,visitors
day_of_week	is_holiday	prev_day_is_holiday
next_day_is_holiday	air_genre_name	air_area_name
latitude,longitude		

3. Data analysis of model

3.1. Data preprocessing and feature engineering

In this paper, we use the restaurant traffic prediction data from the data in the Kaggle data contest platform known as Recruit Restaurant Visitor Forecasting as originally given in [43] but updated in [44] and then again recently in [45]. After data cleaning and pre-processing, 328,298 records useful for this experiment were extracted. Every record has 12 attributes, as is shown in Table 2.

The pre-processed data does not meet the requirements of the experiment. There are still a large number of potential feature values that have not been mined. At this time, corresponding feature engineering is required. This experiment performs log smoothing on the number of passengers; smooth processing of timing problems, and whether the meal is at the weekend and related features. These operations fully exploit the vast amount of value hidden in the data. After the feature processing, each record has 62 attributes, as is shown in Table 3.

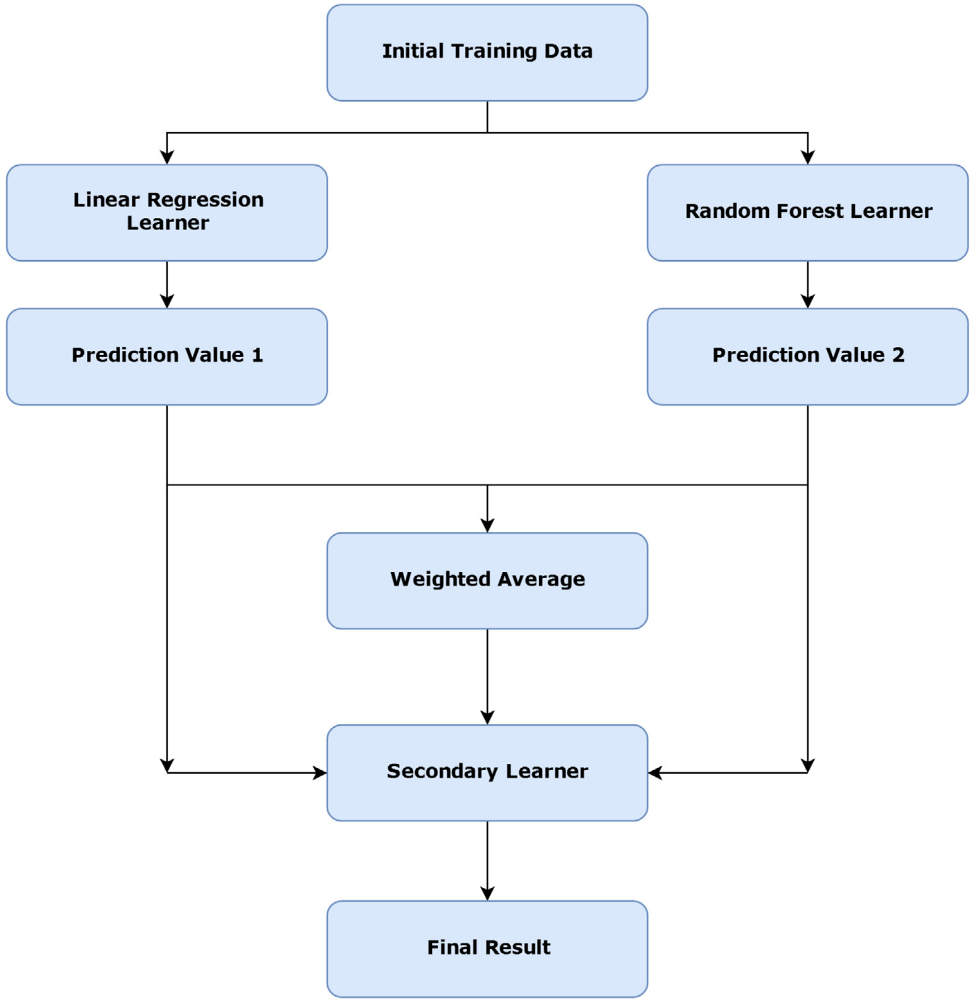


Fig. 4. Optimization algorithm.

Table 3
Attributes per record after feature engineering.

air_store_id	is_true	visit_date
day_of_week	is_holiday	prev_day_is_holiday
next_day_is_holiday	air_genre_name	air_area_name
latitude	longitude	is_outliters
visitors_log	is_weekend	day_of_month
Day_of_week	vc_fewm_mean	vc_fewm_std
vc_log_fewm_mean	vc_log_fewm_std	vc_day_of_week_fewm_mean
vc_day_of_week_fewm_fewm_std	vc_log_day_of_week_fewm_mean	vc_log_day_of_week_fewm_fewm_std
vc_isholiday_fewm_mean		

3.2. Model construction

After pre-processing and feature engineering of the data, the input data is used for model training [45–47]. The data except for the visitors `log` attribute in Table 3 is the training data set, and the visitors `log` attribute is the training label. The input data is shown in Table 4.

Table 5 is an output example for the prediction model of the LR algorithm. Among them, ID indicates the address of the store and the corresponding date, and visitors indicates the predicted number of customers for the store on the corresponding date.

Table 4
Input data of the model.

air_store_id	visit_date	day_of_week	...
air_00a91d42b08b08d9	2019-07-01	Friday	...
air_00a91d42b08b08d9	2019-07-02	Saturday	...
air_00a91d42b08b08d9	2019-07-03	Sunday	...
air_00a91d42b08b08d9	2019-07-04	Monday	...
air_00a91d42b08b08d9	2019-07-05	Tuesday	...
...	...	...	...

3.3. Experimental results and analysis

3.3.1. Predicted values versus actual values for each model

In order to facilitate the observation of experimental phenomena and analysis of the experimental results, this paper compares the predicted traffic with the real traffic for a LR model, RF cognitive computing model and optimized stacking fusion method, as are shown in Figs. 5–7 respectively. The abscissa of Figs. 5–7 indicates the index value of the store, and the ordinate indicates the value obtained by taking the logarithm of the number of

Table 5
Output data of the model.

air_store_id	Visitors
air_00a91d42b08b08d9_2019-04-23	2.706023
air_00a91d42b08b08d9_2019-04-24	10.433438
air_00a91d42b08b08d9_2019-04-25	14.114471
air_00a91d42b08b08d9_2019-04-26	18.580581
air_00a91d42b08b08d9_2019-04-27	14.807312

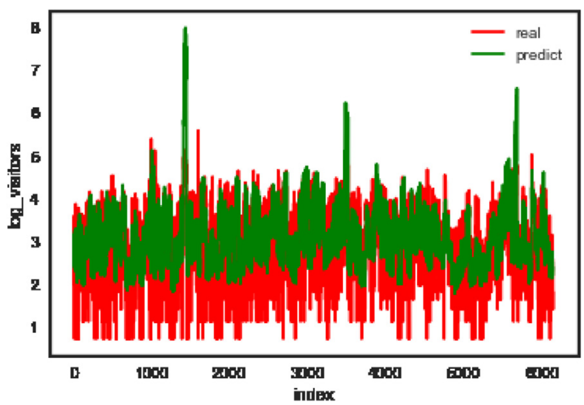


Fig. 5. Contrast diagram of the linear regression model.

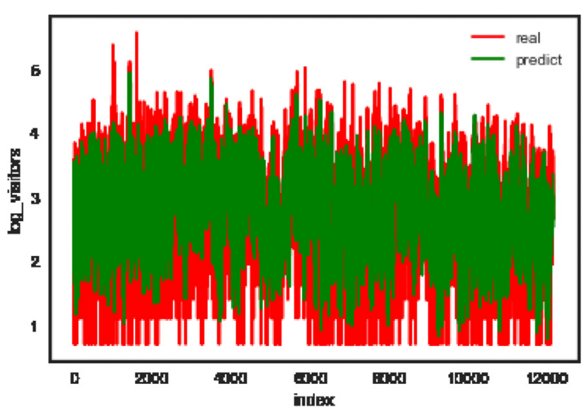


Fig. 6. Contrast diagram of random forest model.

customers. One set of lines in Figs. 5–7 respectively indicate the logarithm of the real traffic for each store under each cognitive computing model, and the broken blue lines respectively indicate the logarithm of the predicted traffic for each store under each hybrid cognitive computing model. The figures reflect the direct quantitative relationship between the actual value and predicted value. It can be clearly seen from Figs. 5–7 that the predicted values are consistent with the actual values for the three models. The three models have shown to be accurate in their predictive ability, clearly showing the strength and novel nature of the results given here.

3.3.2. Comparison of model errors

Examining the data from 1% of total data to 100% of all the data in the experiment, different classifiers were trained for different models. Fig. 8 is the error indices comparison of the LR model with the fusion model, and Fig. 9 is the error indices comparison of the RF model with the fusion model.

In order to further prove the stability and reliability of the experimental results, four different batches of data were randomly selected from the experiment, and 1% of the total data to 100% of each batch of data was taken respectively. The above three

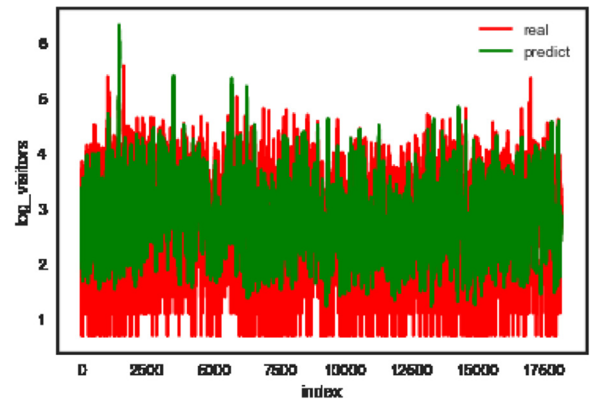


Fig. 7. Contrast diagram of the fusion model.

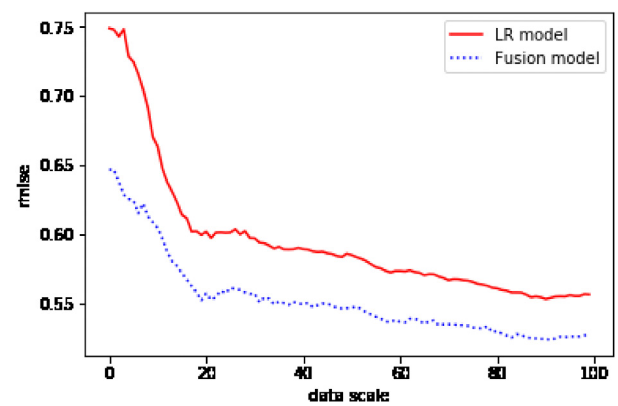


Fig. 8. Error indices comparison of a linear regression model with the fusion model.

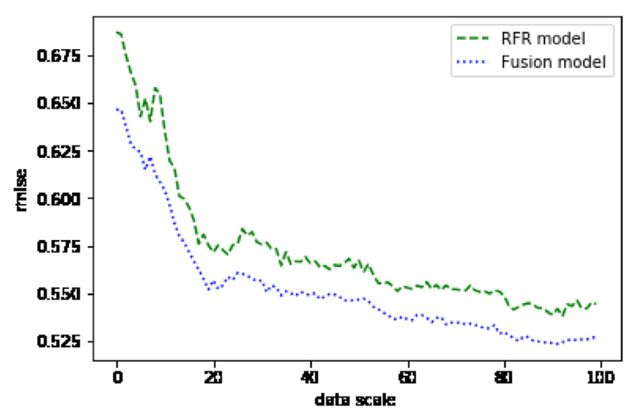
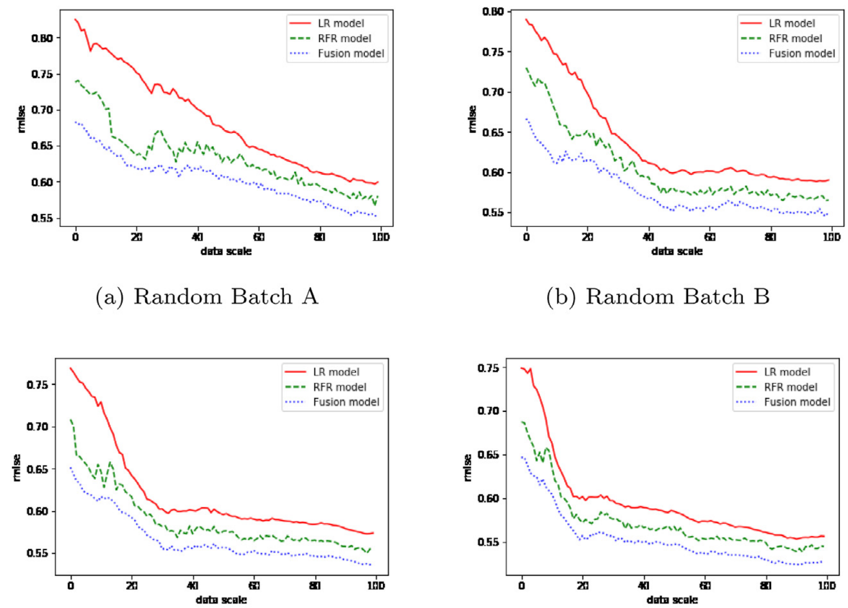


Fig. 9. Error indices comparison of random forest model with the fusion model.

models were trained in turn to obtain three error indices, each batch of data corresponding to a comparison chart, as shown respectively in Figs. 10a, 10b, 10c and 10d. Among the figures, the abscissa in each figure represents the proportion of data under the batch of data, and the ordinate represents the corresponding error indices. The solid line, dashed line and dotted line in the figure respectively represent the prediction error indices of the linear regression model, the random forest model, and the fusion model. The figure visually reflects the change of the prediction error indices of different models under different data sizes. In Fig. 10, the LR model, RFR model, and Fusion model represent the error indices of the linear regression model, the random forest model and the fusion model respectively [46].



(a) Random Batch A (b) Random Batch B

(c) Random Batch C (d) Random Batch D

Fig. 10. Random batches comparison of error indices of different data sizes and models.

3.3.3. Analysis of experimental results

It can be clearly concluded from the analysis of Figs. 5–7 that the predicted values and actual values of the three models are consistent, showing the strength of the fusion model introduced here. The result analysis proves that the three models are suitable for the experiments conducted this paper. By comparing the predicted values and actual values of different models, the degree of fit of the fusion model in Fig. 7 is **better** than that of the LR model in Fig. 5 and the RF model in Fig. 6. It demonstrates that the prediction of the fusion model is better than that of the single model.

Comparing Figs. 8 and 9, we can clearly see that the error indices of the fusion model are significantly lower than that of the LR model and the RF model. In Figs. 10a, 10b, 10c and 10d under identical data size, the effect of the fusion model is **far better** than that of the RF model, and the effect of the RF model is better than that of the LR model. For the fusion model, as the amount of data increases, the error indices of traffic prediction is decreasing. We also see the effect of the model is continuously being optimized, and when the amount of data is large enough, the decline of the error indices are very slow. Overall, we clearly state and show that the effect of the fusion model is better than the two single models independently.

The results obtained can be made use of as a real-time operational use technique to maximize both profit and resource use for any given restaurant. If a given restaurant has the ability to forecast customer traffic in their restaurant effectively, then they can either plan to increase food supply or for slow periods decrease both staff and food supply thus minimizing expenditures. With the onset of humanoid robots in the Food Industry as well, these predicative measures can assist assigning robots to proper tasks within the food establishment as required

4. Future work

During our investigations in this paper, there has been some points that have come to light that will require future addressing. First, one important facet that was overlooked in this work was that customer traffic in restaurants severely varies in different hours of the day. For example, during typical meal times (breakfast, lunch, dinner) the traffic is usually considerably higher than during the in between times of the day. A new model that can attest for this would be helpful in future analysis. Secondly, there is also the need to add some dialog around different types of restaurants. For example, some restaurants only serve dessert items whereas others serve full meals. This fact can be worked into the model in some form in future iterations of the work or taken on by other researchers interested in the finding given here.

5. Conclusion

This paper summarizes the principles of the **linear regression** and **random forest** cognitive computing algorithms, as well as the fusion of the two algorithms. We also optimize and improve the stacking fusion algorithm based on amalgamation of LR and RF. We apply our fusion algorithm to the process of predicting customer traffic in the restaurant industry results show that the application of the fusion algorithm in restaurant traffic prediction shows strong results. These results could effectively be used to better service clientele in restaurants that have begun the use of service robots for customer service. These results will be able to assist the food services industry to make plans and decisions more rationally and in a timely manner. It will also help to enhance the overall customer's dining experience, improve the quality of food services, and at the same time allow the restaurant industry to operate better, obtain greater profits, which enhance the importance of the result obtained here. In this paper, idea of fusing the linear regression and random forest algorithms makes up the shortcomings of the previous models independently, reduces the error indices of restaurant customer prediction, improves the model effect and makes the model more widely applicable. To some extent, as the size of training data increases, the fusion model performance will be continuously improved.

Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.asoc.2019.106051.

References

[1] I.H. Witten, E. Frank, M.A. Hall, C.J. Pal, Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann, 2016, pp. 124–144.

[2] S. Mohan, C. Thirumalai, G. Srivastava, Effective heart disease prediction using hybrid machine learning techniques, IEEE Access 7 (2019) 81542–81554.

[3] X. Cheng, X. Yu, M. Cheng, D. Ouyang, S. Zhong, X. Guo, K. Yang, S. Wang, Mineral resource appraisal based on information synthesis, Cybern. Intell. Chall. Adv. (2003) 23.

[4] L. Sun, S. Yoshida, X. Cheng, Y. Liang, A cooperative particle swarm optimizer with statistical variable interdependence learning, Inform. Sci. 186 (1) (2012) 20–39.

[5] X. Wang, S. Yang, Y. Zhao, Y. Wang, Improved pore structure prediction based on micp with a data mining and machine learning system approach in mesozoic strata of gaoqing field, jiyang depression, J. Pet. Sci. Eng. 171 (2018) 362–393.

[6] M. Kef, L. Chergui, S. Chikhi, A novel fuzzy approach for handwritten arabic character recognition, Pattern Anal. Appl. 19 (4) (2016) 1041–1056.

[7] Time-delayed multiple linear regression for de-noising mems inertial sensors, Comput. Electr. Eng. 76 (2019) 1–12.

[8] Wlrrs: A new recommendation system based on weighted linear regression models, Comput. Electr. Eng. 66 (2018) 40–47.

[9] E.C. Chan, G. Baciu, S. Mak, Cognitive location-aware information retrieval by agent-based semantic matching, Int. J. Softw. Sci. Comput. Intell. (IJSSCI) 2 (3) (2010) 21–31.

[10] H. Watson, The cognitive decision-support generation, Bus. Intell. J. 22 (2) (2017) 5–14.

[11] B.A. TAMA, Data mining for predicting customer satisfaction in fast-food restaurant, J. Theor. Appl. Inf. Technol. 75 (1) (2015).

[12] A. Agrawal, A. Choudhary, Perspective: Materials informatics and big data: Realization of the "fourth paradigm" of science in materials science, Appl. Mater. 4 (5) (2016) 53–208.

[13] S.B. Muhammad, R. Saad, Linear regression models for estimating true subsurface resistivity from apparent resistivity data, J. Earth Syst. Sci. 127 (5) (2018) 64–78.

[14] A. Acharjee, B. Kloosterman, R.G. Visser, C. Maliepaard, Integration of multi-omics data for prediction of phenotypic traits using random forest, BMC Bioinform. 17 (5) (2016) 180–195.

[15] J. Shotton, A.W. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, A. Blake, Real-time human pose recognition in parts from single depth images, in: Cvpr, Vol. 2, 2011, pp. 3–10.

[16] T.F. Cootes, M.C. Ionita, C. Lindner, P. Sauer, Robust and accurate shape model fitting using random forest regression voting, in: European Conference on Computer Vision, Springer, 2012, pp. 278–291.

[17] T. Baumann, Decision tree usage for incremental parametric speech synthesis, in: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2014, pp. 3819–3823.

[18] C. Xiong, D. Johnson, R. Xu, J.J. Corso, Random forests for metric learning with implicit pairwise position dependence, in: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2012, pp. 958–966.

[19] A. Verikas, A. Gelzinis, M. Bacauskiene, Mining data with random forests: A survey and results of new tests, Pattern Recognit. 44 (2) (2011) 330–349.

[20] Y.-J. Liu, D.-L. Zhang, M.M.-F. Yuen, A survey on cad methods in 3d garment design, Comput. Ind. 61 (6) (2010) 576–593.

[21] Y. Wang, M. Liu, J. Yang, G. Gui, Data-driven deep learning for automatic modulation recognition in cognitive radios, IEEE Trans. Veh. Technol. (2019) 1124–1414.

[22] H. Huang, Y. Song, J. Yang, G. Gui, F. Adachi, Deep-learning-based millimeter-wave massive mimo for hybrid precoding, IEEE Trans. Veh. Technol. (2019) 1–14.

[23] M. Mende, M.L. Scott, J. van Doorn, D. Grewal, I. Shanks, Service robots rising: How humanoid robots influence service experiences and elicit compensatory consumer responses, J. Mark. Res. (2019) 0022243718822827.

[24] R. Gallon, N. Lorenzo, M. Josefowicz, Of humans and robots–communication challenges in industry 4.0, Artificial Intelligence (2019).

[25] I. Aaltonen, A. Arvola, P. Heikkilä, H. Lammi, Hello pepper, may i tickle you?: Children's and adults' responses to an entertainment robot at a shopping mall, in: Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction, ACM, 2017, pp. 53–54.

[26] Y. Zhang, R. Wang, M.S. Hossain, M.F. Alhamid, M. Guizani, Heterogeneous information network-based content caching in the internet of vehicles, IEEE Trans. Veh. Technol. 68 (10) (2019) 10216–10226.

[27] H. Lu, T. Uemura, D. Wang, J. Zhu, Z. Huang, H. Kim, Deep-sea organisms tracking using dehazing and deep learning, Mob. Netw. Appl. (2018) 1–8.

[28] H. Lu, Y. Li, T. Uemura, H. Kim, S. Serikawa, Low illumination underwater light field images reconstruction using deep convolutional neural networks, Future Gener. Comput. Syst. 82 (2018) 142–148.

[29] H. Lu, Y. Li, M. Chen, H. Kim, S. Serikawa, Brain intelligence: go beyond artificial intelligence, Mob. Netw. Appl. 23 (2) (2018) 368–375.

[30] H. Lu, B. Li, J. Zhu, Y. Li, Y. Li, X. Xu, L. He, X. Li, J. Li, S. Serikawa, Wound intensity correction and segmentation with convolutional neural networks, Concurr. Comput.: Pract. Exp. 29 (6) (2017) e3927.

[31] M. Zapadka, M. Kaczmarek, B. Kupcewicz, P. Dekowski, A. Walkowiak, A. Kokotkiewicz, M. Łuczkiewicz, A. Buciński, An application of qsrr approach and multiple linear regression method for lipophilicity assessment of flavonoids, J. Pharm. Biomed. Anal. 164 (2019) 681–689.

[32] B. Das, B. Nair, V.K. Reddy, P. Venkatesh, Evaluation of multiple linear, neural network and penalised regression models for prediction of rice yield based on weather parameters for west coast of India, Int. J. Biometeorol. 62 (10) (2018) 1809–1822.

[33] D. Steinberg, P. Colla, Cart: classification and regression trees, in: Top Ten Algorithms in Data Mining, Vol. 9, 2009, pp. 179–191.

[34] D.G. Denison, B.K. Mallick, A.F. Smith, A bayesian cart algorithm, Biometrika 85 (2) (1998) 363–377.

[35] A. Nagpal, V. Singh, Coupling multivariate adaptive regression spline (mars) and random forest (rf): A hybrid feature selection method in action, Int. J. Healthc. Inf. Syst. Inform. (IJHISI) 14 (1) (2019) 1–18.

[36] X. Sun, G. Gui, Y. Li, R.P. Liu, Y. An, Resinnet: A novel deep neural network with feature reuse for internet of things, IEEE Internet Things J. 6 (1) (2018) 679–691.

[37] W. Zhang, X. Cheng, Y. Hu, Y. Yan, Online prediction of biomass moisture content in a fluidized bed dryer using electrostatic sensor arrays and the random forest method, Fuel 239 (2019) 437–445.

[38] M.W. Ahmad, J. Reynolds, Y. Rezgui, Predictive modelling for solar thermal energy systems: A comparison of support vector regression, random forest, extra trees and regression trees, J. Clean. Prod. 203 (2018) 810–821.

[39] Y.-B. Lu, S.-C. Din, C.-F. Zheng, B.-J. Gao, Using multi-feature and classifier ensembles to improve malware detection, J. CCIT 39 (2) (2010) 57–72.

[40] M. Liu, T. Song, G. Gui, Deep cognitive perspective: Resource allocation for NOMA based heterogeneous iot with imperfect SIC, IEEE Internet Things J. (2018).

[41] W. Liu, P.-t. Xu, A study on influencing factors of the helpfulness of online reviews in o2o of restaurant industry–based on tobit model, Chin. J. Manage. Sci. 5 (2016) 168–176.

[42] L.-j. Dong, X.-b. Li, P. Kang, Prediction of rockburst classification using random forest, Trans. Nonferr. Met. Soc. China 23 (2) (2013) 472–477.

[43] W.C. Hu, N. Kaabouch, Big Data Management, Technologies, and Applications, Information Science Reference, 2014.

[44] G. Boomija, A. Anandaraj, S. Nandhini, S. Lavanya, Restaurant visitor time series forecasting using autoregressive integrated moving average, J. Comput. Theor. Nanosci. 15 (5) (2018) 1590–1593.

[45] T. Tanizaki, T. Hoshino, T. Shimmura, T. Takenaka, Demand forecasting in restaurants using machine learning and statistical analysis, Procedia CIRP 79 (2019) 679–683.

[46] X. Zhang, F. Fan, M. Gheisari, G. Srivastava, A novel auto-focus method for image processing using laser triangulation, IEEE Access 7 (2019) 64837–64843.

[47] G.T. Reddy, M.P.K. Reddy, K. Lakshmanna, D.S. Rajput, R. Kaluri, G. Srivastava, Hybrid genetic algorithm and a fuzzy logic classifier for heart disease diagnosis, Evol. Intell. (2019) 1–12.