Expert Systems With Applications 82 (2017) 216–230

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Modified genetic algorithm-based feature selection combined with pre-trained deep neural network for demand forecasting in outpatient department

Shancheng Jiang[a], Kwai-Sang Chin[a,*], Long Wang[a], Gang Qu[b], Kwok L. Tsui[a]

[a] Department of Systems Engineering and Engineering Management, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon Tong, Hong Kong, China
[b] Dalian University Affiliated Xinhua Hospital, 146 Wansui Street, Shahekou District, Dalian City, Liaoning Province, China

ARTICLE INFO

Article history:
Received 14 November 2016
Revised 29 March 2017
Accepted 6 April 2017
Available online 7 April 2017

Keywords:
Deep learning
Feature selection
Modified genetic algorithm
Stacked autoencoder
Demand forecasting in hospital

ABSTRACT

A well-performed demand forecasting can provide outpatient department (OPD) managers with essential information for staff scheduling and rostering, considering the non-reservation policy of OPD in China. Based on the results reported by relevant studies, most approaches have focused on forecasting the overall amount of patient flow and ignored the demand for other key resources in OPD or similar department. Moreover, few studies have conducted feature selection before training a forecast model, which is a significant pre-processing operation of data mining and widely applied for knowledge discovery in expert and intelligent system. This study develops a novel hybrid methodology to forecast the patients' demand for different key resources in OPD, by combining a new feature selection method and a deep learning approach. A modified version of genetic algorithm (MGA) is proposed for feature selection. The key operators of normal genetic algorithm are redesigned to utilize useful information provided by filter-based feature selection and feature combinations. A feedforward deep neural network is introduced as the forecast model, and the initial parameter set is generated from a stacked autoencoder-based pre-training process to overcome the optimization challenges in constructing deep architectures. In order to evaluate the performance of our methodology, it is applied to an OPD located at Northeast China. The results are compared with those obtained from combinations of other feature selection methods and demand forecasting models. Compared with GA and PCA, MGA improves the quality and efficiency of feature selection, with less selected features to get higher forecast accuracy. Pre-trained DNN optimally strengthens the advantage of MGA, compared with MLR, ARIMAX and SANN. The combination of MGA and pre-trained DNN possesses strongest predictive power among all involved combinations. Furthermore, the results of proposed methodology are crucial prerequisites for staff scheduling and resource allocation in OPD. Elite features obtained by MGA can provide practical insights on potential association between manifold feature combinations and demand variance.

1. Introduction

Healthcare managers are continuously confronted by a number of challenges, ranging from limited resources to growing patient demand. Based on the report of World Health Statistics 2015, overcrowding and shortage of medical resources are significant challenges in healthcare systems worldwide, especially in developing countries. In China, the outpatient department (OPD) is an essential part of the overall hospital operations. The OPD normally provides preliminary diagnosis to walk-in patients through a set of nurses, physicians, laboratory examinations, and so on. Under the non-reservation policy, determining fluctuating demand in advance is the premise to relieve OPD overcrowding and resource shortage (Aboagye-Sarfo et al., 2015; Afilal et al., 2016; Hwang et al., 2011; Jones et al., 2009; Jones, Thomas, Evans, Welch, & Haug, 2008; Sahu, Baffour, Harper, Minty, & Sarran, 2014; Xu, Wong, & Chin, 2013). It is suggested that crowding in emergency department, which is similar with OPD, could be reduced considerably by implementing demand management systems (McCarthy et al., 2008). However, methods for demand forecasting in OPD have not been well studied. Most relevant approaches are limited to forecasting the amount of patient flow and ignore the specific demand for key resources in OPD, which contains comparable importance for staff scheduling and resource allocation. Therefore, the objec-

* Corresponding author.
E-mail addresses: scjiang2-c@my.cityu.edu.hk (S. Jiang), mekschin@cityu.edu.hk (K.-S. Chin), long.wang@my.cityu.edu.hk (L. Wang), 1638654878@qq.com (G. Qu).

http://dx.doi.org/10.1016/j.eswa.2017.04.017
0957-4174/

tive of this study is to model and forecast the fluctuating demand for key resources in OPD. Results will aid OPD managers to allocate limited resources and determine staffing level in their decision-making process.

Forecast accuracy has generally been the essential goal of prediction researches. As well demonstrated by Chandrashekar et al.'s study (Chandrashekar & Sahin, 2014), the accuracy of the forecast model not only depends on the model structure and related training algorithm but also on the feature space, which is constructed through the original feature set and feature selection (FS) algorithm. FS is generally used in machine learning applications as part of the pre-processing step, in which a subset of features (i.e. independent variables) is obtained by eliminating features with minimal predictive information (Hinton & Salakhutdinov, 2006; Vieira, Sousa, & Runkler, 2010). Advantages of FS are typically summarized as decreasing training and implementation time, facilitating data understanding, and reducing storage requirement (Ali Jan Ghasab, Khamis, Mohammad, & Jahani Fariman, 2015). However, as for this demand forecasting problem, few approaches conduct FS before fitting the forecast model. The feature space of this demand forecasting problem normally contains seasonal and meteorological factors, and it is risky to include all relevant features in one model because certain features may become useless with changes of the terrain and hospital. For example, in temperate areas patients prefer to visit OPD in rainy days rather than in sunny days, but this rule is inapplicable in tropical areas (Sun, Heng, Seow, & Seow, 2009). Based on model parameters, several researches quantify the relative importance of individual features after fitting the forecast model, in order to explore implications of different features (Kam, Sung, & Park, 2010; Marcilio, Hajat, & Gouveia, 2013). These results might not be reliable with bias introduced by limited model capacity.

Several heuristic optimization algorithms have been successfully applied as search strategy for FS. These algorithms include Particle Swarm Optimization (PSO; Gunasundari, Janakiraman, & Meenambal, 2016; Lin, Ying, Chen, & Lee, 2008; Unler & Murat, 2010), Ant Colony Optimization (ACO; Ali Jan Ghasab et al., 2015; Goodarzi, Freitas, & Jensen, 2009) and Genetic Algorithm (GA; Ghareb, Bakar, & Hamdan, 2016; C.-H. Lin, Chen, & Wu, 2014; Rejer, 2015; Sikora & Piramuthu, 2007). GA has attracted much attention because of its operability and powerful search capability. As an artificial intelligent probabilistic search algorithm, GA has been widely applied to numerous combinatorial optimization problems (Gen & Cheng, 2000) and can be summarized as an evaluation-selection-reproduction cycle. However, poor initialization, hyper-parameter setting, and randomness of crossover and mutation processes are the main problems of GA when it is employed for FS (Ghareb et al., 2016). Traditional GA needs to be modified to improve its efficiency on FS.

Proposed time series prediction models for patient flow forecasting can be divided into four categories: regression-based, time-series, time-series-regression, and artificial intelligence models. The multiple linear regression (Boyle et al., 2012; Jones et al., 2008; Marcilio et al., 2013; McCarthy et al., 2008; Tai, Lee, Shih, & Chen, 2007), autoregressive integrated moving average (Schweigler et al., 2009; Sun et al., 2009; Xu, Wong, Chin, Wong, & Tsui, 2011), and shallow artificial neural network (Menke et al., 2014; Ram, Zhang, Williams, & Pengetnze, 2015) are frequently utilized as forecast models. With limited model capacity and predictive power, these traditional statistical models are not good choices for modeling the diverse demand patterns for different OPD resources. In the era of big data, deep learning algorithms lead to outstanding results in most typical machine learning applications (Guo et al., 2016; LeCun, Bengio, & Hinton, 2015). Motivated by high capacity of deep architectures, this study introduces a pre-trained deep neural network (DNN) as the forecast model, in order to improve the forecasting accuracy and take advantage of the key information provided by FS.

Table 1
Summary of FS applied in demand forecasting domain.

Application	Category	References
Power system	Filter	(Koprinska et al., 2015; Rana, Koprinska, & Khosravi, 2013; Voronin & Partanen, 2014; Jurado, Nebot, Mugica, & Avellana, 2015)
	Wrapper	(Hu, Bao, & Xiong, 2014; Kazemi et al., 2014; Sheikhan & Mohammadi, 2012)
Energy	Filter	(Tonkovic, Zekic-Susac, & Somolanji, 2009)
	Wrapper	(Salcedo-Sanz, Munoz-Bulnes, Portilla-Figueras, & Del Ser, 2015)
Supply chain	Filter	(Tan et al., 2011)
	Wrapper	(Liu et al., 2008)
Healthcare	Wrapper	(Urraca et al., 2015)
	Filter	(Xu et al., 2013)

Overall, our study has added the following improvements: (a) Demand for different key resources in OPD is modeled in one hybrid framework, in which FS and deep learning algorithms are combined. (b) A new modified GA (MGA) is proposed as the FS algorithm. In MGA, a filter-based FS algorithm is merged into initialization, and the feature combination information is utilized in crossover operation to improve the efficiency of optimization process. (c) An unsupervised pre-training algorithm is integrated in the model-training process to overcome the optimization challenges for training deep architectures. The rest of this paper is organized as follows: Section 2 presents the literature review and briefly summarizes related works. Section 3 describes the proposed MGA-based FS algorithm in detail. Section 4 presents the configuration of the DNN model and the mechanism of unsupervised pre-training algorithm. Section 5 discusses the results of the real case study. Section 6 concludes the paper.

2. Literature review

2.1. Feature selection

FS approaches are categorized as filter, wrapper, and embedded methods (Chandrashekar & Sahin, 2014). Filter methods do not rely on any forecast models and they rank all features based on statistical properties. This category includes correlation-based (Koprinska, Rana, & Agelidis, 2015; Tan, Sim, & Yeoh, 2011; Xu et al., 2013), mutual information-based (Doquire & Verleysen, 2013), Chi-square test-based (Mesleh, 2011), and principal component analysis-based approaches (Malhi & Gao, 2004; Rocchi, Chiari, & Cappello, 2004; Uguz, 2011). Filter methods are widely used in high-dimension dataset because of their computational efficiency. Wrapper methods assess feature subsets according to their usefulness to a given predictor or classifier. These methods consider the FS as a search problem, in which different combinations of features are prepared, evaluated, and compared with other combinations. Popular heuristic intelligent optimization algorithms mentioned in Section 1 are applied to guide the search process. Compared with filter methods, wrapper methods exhibit better performance because different feature subsets are evaluated by a new forecast model or learning algorithm in each iteration (Sikora & Piramuthu, 2007). Embedded methods blend FS into the model training process; one typical way of combination is launching regularization strategies while training the model (Chandrashekar & Sahin, 2014). As an example of embedded methods, the LASSO model regularizes the parameters of a linear model with an L1 penalty, thereby reducing the less correlated coefficients to zero. Table 1 shows recently published FS approaches in demand forecasting domain.

Among wrapper methods listed in Table 1, GA-based FS exhibits excellent performance in eliminating redundant features. Kazemi, Hoseini, Abbasian-Naghneh, and Rahmati (2014) proposed a hybrid approach to forecast future electrical energy demand; in this study, a normal GA is applied to determine the most suitable feature subset combined with an adaptive neuro-fuzzy inference system. Sheikhan and Mohammadi (2012) proposed a combination of GA and ACO for FS in the electricity load-forecasting problem; SANN is used as predictor to evaluate the fitness. Urraca, Sanz-Garcia, Fernandez-Ceniceros, Sodupe-Ortega, and Martinez-de-Pison (2015) proposed a GA-SVR framework to forecast hotel room demand, in which GA is not only used for FS but also for parameter tuning and parsimonious model selection. Liu, Yin, Gao, and Tan (2008) applied a GA-based FS to the demand forecasting issues of China's retail industry. Besides demand forecasting, GA-based FS is frequently utilized for knowledge discovery in other hot areas. For example, as for financial market analysis, FS is a general preprocessing mechanism for understanding market behaviors (Cavalcante, Brasileiro, Souza, Nobrega, & Oliveira, 2016). Sexton, Sriram, and Etheridge (2003) found that GA-based FS outperforms the CATLRN on identifying relevant variables to evaluate a bank's financial condition. Tsai and Hsiao (2010) combined GA with PCA and decision trees in order to build a feature selector for stock price prediction. Recently, F. Lin, Liang, Yeh, and Huang (2014) proposed a novel GA-based FS for financial distress prediction problem, in which GA-based wrapper is combined with expert knowledge.

After reviewing these GA-based FS approaches, we find that a typical GA with normal configuration is applied in most studies (Kazemi et al., 2014; Liu et al., 2008; Sheikhan & Mohammadi, 2012; Tsai, Eberle, & Chu, 2013; Urraca et al., 2015). For example, the initial chromosome set is randomly generated, in which population diversity cannot be assured and the frequency of redundant features may affect the efficiency of searching process. Moreover, one-point or two-point-based crossover operators are applied in most approaches. However, during the iterative process, latent information behind feature combinations might be destroyed by randomly generated crossover points. Based on a general idea that an evolutionary algorithm should fit the application to obtain the best search result, a research gap is present and can be bridged by modifying and redesigning normal GA.

2.2. Deep neural network

DNN originates from the traditional shallow artificial neural networks (SANN), which belong to the artificial intelligence model. In recent years, DNN appears frequently in various machine learning applications, such as image recognition (Cireşan, Meier, Masci, & Schmidhuber, 2012; Tang, Deng, Huang, & Zhao, 2015), speech recognition (Dahl, Yu, Deng, & Acero, 2012; Hinton et al., 2012). Although the universal approximation theorem (Cybenko, 1989; Hornik, Stinchcombe, & White, 1989) states that a feedforward network with at least one hidden layer with any "squashing" activation function can approximate any Borel measurable function, this theorem does not restrict the size of the network. As the complexity of an objective function increases, exponential number of hidden units might be required. (Goodfellow, Bengio, & Courville, 2016). Hence, SANN is risky to generalize accurately with a huge number of hidden units; by contrast, deeper architectures are able to reduce the number of hidden units in many circumstances (Goodfellow et al., 2016). Researches in machine learning field demonstrate that the model with greater depth achieves improved generalization in most machine learning applications (Goodfellow, Bulatov, Ibarz, Arnoud, & Shet, 2013; Kahou et al., 2013; Szegedy et al., 2015), because employing deep architectures expresses a useful prior over the space of real functions and improves forecast performance.

Besides typical machine learning applications, DNN has also achieved good performance in time-series modeling and forecasting, with the advantages of deep architectures (Gashler & Ashmore, 2016; Hossain, Rekabdar, Louis, & Dascalu, 2015; Hu, Zhang, & Zhou, 2016). Gashler and Ashmore (2016) utilized a DNN with dynamic parameter tuning method to model artificially generated time series and weekly temperature measurements in Anchorage, Alaska. In this approach, DNN generally predicts the nonlinear temperature trends very well. In M. Hossain et al.'s approach (Hossain et al., 2015), a pre-trained DNN was applied to forecast the weather of Nevada. In the field of energy utilization, Hu et al. (2016) introduced DNN to forecast wind speed, which benefits large-scale wind power penetration. In Wan et al.'s study (Wan et al., 2016), a deep belief network with a restricted Boltzmann machine was implemented to carry out day-ahead prediction of wind speed. Shen, Chao, and Zhao (2015) proposed an improved deep belief network to forecast exchange rates. Aizenberg, Sheremetov, Villa-Vargas, and Martinez-Munoz (2016) utilized a DNN with multi-valued neurons to model the fluctuation of an oilfield asset. Based on literature review, models with deep architectures are rarely applied in the demand forecasting domain.

3. Modified genetic algorithm for feature selection

3.1. Representation and initialization

In this approach, the 0–1 binary representation is applied (Ghareb et al., 2016; Tsai et al., 2013). Each feature subset (i.e., each solution in the search space) is encoded as an n-bit binary chromosome, where the gene with value "1′' means the corresponding feature is selected and "0′' means unselected. The parameter n denotes the number of all available features. Generally, the quality of initial population influences the possibility of finding a good solution. To reduce the risk of poor initialization, we introduce some "excellent chromosomes" which can potentially improve the quality of final solutions and convergence rate, by utilizing the result of a filter method. The algorithm to generate an "excellent chromosome" is summarized in the following pseudocode.

Algorithm 1 indicates that an "excellent chromosome" is obtained based on the result of t-test for multiple linear regression (MLR). In MLR, β_j measures the partial linear effect of feature j on y. For each β_j, the t-test is used to determine the marginal significance of the corresponding feature j, while other features are included in the model. The $|t_j|$ obtained from t-test can be regarded as a criterion to quantify the relative importance of feature j, based on the hypothesis statement of t-test for MLR and also the statistical meaning of t value. Combined with the probability density function of student's t-distribution, higher $|t_j|$implies that the corresponding feature is more significant and contains higher risk to be removed from fitted MLR. Therefore, value "1" on gene j is generated with the probability proportional to $|t_j|$. The "excellent chromosome" is then formulated by combining these individually good features and added into the initial solution pool. Normal chromosomes randomly generated are retained in the initial population to maintain the diversity. The ratio of "excellent chromosome" and normal chromosome is set as 1:2.

3.2. Fitness evaluation

Evaluation of fitness value significantly contributes to the parent chromosome selection and crossover operator. In this minimization problem, the chromosome with lower fitness value will have a higher chance to be selected for crossover. In the present

Algorithm 1 A procedure that produces an n-bit "excellent chromosome," given the training set containing m examples and n features.

Fit a multiple linear regression model to the training set to obtain the estimated regression coefficients $\hat{\beta}_0, \hat{\beta}_1, ..., \hat{\beta}_n$ and corresponding estimated standard deviation $s_{\hat{\beta}_0}, s_{\hat{\beta}_1}, ..., s_{\hat{\beta}_n}$
for $j = 1, ..., n$ **do**
  $t_j \leftarrow (\hat{\beta}_j - 0)/s_{\hat{\beta}_j}$ (t-test for individual regression coefficient β_j)
end for
for $j = 1, ..., n$ **do**
  $p_j \leftarrow |t_j| / \sum_{j=1}^{n} |t_j|$
end for
Generate a 0–1vector$\boldsymbol{c} = [c_1, c_2, ..., c_j]$, $c_j = 1$with the probability ofp_j, or $c_j = 0$with the probability of$(1 - p_j)$.
return $\boldsymbol{c}$

study, the fitness value is evaluated by computing the forecast accuracy of a given feature subset via a restricted DNN model plus the penalty on the size of given feature subset. The fitness function is set as Eq. (1).

$$fitness_{\boldsymbol{c}_i} = MSE_{\boldsymbol{c}_i} + \lambda S_{\boldsymbol{c}_i}. \quad (1)$$

In Eq. (1), $\boldsymbol{c}_i$is the selected chromosome (i.e. feature subset), $S_{\boldsymbol{c}_i}$is the number of features in this subset, and λ is the weight of penalty on the size of this subset. $MSE_{\boldsymbol{c}_i}$(i.e., mean square error) denotes the forecast performance of restricted DNN by using the selected features, and it is calculated through a repeated cross-validation process. As for fitness evaluation, the size of restricted DNN is limited to 3 layers and 20 nodes in each layer to reduce the computation time of training DNNs. Early-stop strategy is also introduced to reduce the number of iterations. Although adding these constraints could likely lead normal DNNs to be under-fitted (Goodfellow et al., 2016), this treatment does benefit the fitness evaluation process. Instead of utilizing a selected feature subset to produce remarkable outcomes, the goal of fitness evaluation process is to quantify the subset's tendency toward remarkable predictive power. Hence, the capacity of the model is unnecessary to be increased in this stage. The capability to explore the underlying relationship between feature subset and output variable is maintained with the high-flexibility of restricted DNN. Moreover, inaccurate results generated by poor initialization of restricted DNN can be regarded as mutation factors for the entire evolution process.

3.3. Parent selection strategy

After fitness evaluation, the parent selection is conducted based on the relative fitness of chromosome through the applied selection strategy. Tournament selection is applied in this study to cooperate with the noisy fitness function and control the selection pressure (Miller & Goldberg, 1995). Tournament selection is regarded as holding a tournament among T competitors (i.e., chromosomes), and T is designated as the tournament size. The winner of the tournament is the chromosome with the lowest fitness value among T competitors. The selection pressure (i.e., the degree to which better individuals are favored) is decreased by decreasing T (Miller & Goldberg, 1995). In the present study, two tournament pools are randomly generated from the population, and each pool consists of T chromosomes. Concerning the noise introduced by fitness function, T is set as $S/4$ and S represents the number of chromosome in the population. Two winners are selected from two pools and are ready for the following crossover operation. Compared with roulette wheel selection, this tournament selection improves the efficiency of selection by evaluating half of individuals, instead of computing the probability of selection for each individual.

Algorithm 2 The fitness-based crossover algorithm for generating a new chromosome C by exchanging information in two parent chromosomes P_1 and P_2.

for $j = 1, ..., n$ **do** (n is the length of chromosome)
  if $P_1^{(j)} = P_2^{(j)}$, **then** ($P^{(j)}$denotes the value of jth gene on the chromosome P)
    $C^{(j)} = P_1^{(j)} = P_2^{(j)}$
  else
    $C^{(j)} = P_1^{(j)}$ with probability of $p_1 = fitness_{P_2}/(fitness_{P_1} + fitness_{P_2})$
    or $C^{(j)} = P_2^{(j)}$ with probability of $p_2 = fitness_{P_1}/(fitness_{P_1} + fitness_{P_2})$
  end if
end for
return C

3.4. Crossover

The crossover operator significantly influences quality and diversity of next generation. One-point or two-point crossover operators are widely applied in GA-based FS; the crossover points are generated based on a fixed probability, and the corresponding segments of parent chromosomes are swapped to produce new chromosomes. This aimless searching strategy may impact the convergence rate. In MGA, a fitness-based crossover algorithm is introduced and redesigned to adapt to this FS problem. The procedure is specified in Algorithm 2.

The fitness-based crossover algorithm is inspired by the idea of utilizing the information behind feature combination. Before interpreting the mechanism of the present crossover algorithm, we should note that a lower fitness value means a higher chance to be selected in this minimization problem. If the fitness values of P_1 and $P_2$are quite different, the feature combination in the individual with lower fitness value is more likely to be inherited. The promising feature combination is maintained and cannot be easily broken off in the entire evolution process until it encounters another "promising combination." Considering another situation, where the fitness values of P_1 and $P_2$are nearly the same, the probability of assigning $P_1^{(j)}$ to $C^{(j)}$ is almost identical to the probability of assigning $P_2^{(j)}$ to $C^{(j)}$. The feature combination is then shuffled and reassembled. A new promising feature combination is created by utilizing the useful information behind both promising feature combinations.

3.5. Replacement

After crossover, the steady-state replacement strategy is employed in MGA. In this very steady-state replacement, the newborn child chromosome produced by two parent chromosomes randomly replaces an individual with higher fitness value than the average. Regarded as a simpler version of generational replacement, the steady-state replacement leads to great improvement in convergence rate, especially for multi-objective optimization problems (Chafekar, Xuan, & Rasheed, 2003; Syswerda, 1991). Two constraints are added before the new chromosome is considered to replace an old one, in order to maintain population diversity dur-

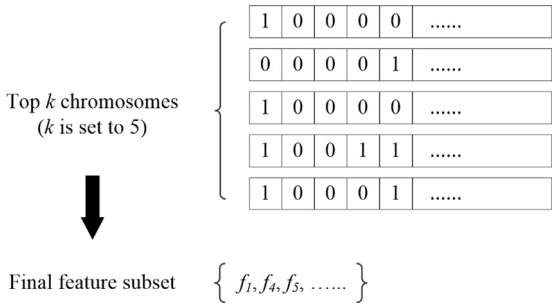


Fig. 1. Decoding operation to build the final feature subset.

ing the iterative process: (a) the fitness value of new chromosome is lower than average; (b) the new chromosome is different from any individuals in current population. If these two constraints cannot be satisfied, the newborn child chromosome is discarded and the crossover is re-executed until an eligible one is obtained.

3.6. Final feature subset

After MGA achieves convergence, that is, the mean fitness value changes minimally over several generations, one or more chromosomes are selected and decoded as the final feature subset. An integration strategy is introduced in the decoding operation of MGA to increase the robustness of the final feature subset (Fig. 1). First, the top k chromosomes with lowest fitness values are retrieved from the final population, in which k is a predefined parameter. A new chromosome is then generated by taking the union of all genes on these k chromosomes. The final feature subset is acquired by decoding this newly created chromosome. From Fig. 1, the final feature subset generated by using the union combination is based on all features that have been chosen by each of the top k chromosomes. In this way, the feature combination information carried by top k chromosomes is inherited and re-combined in the newly created chromosome, and the robustness of the final feature subset is improved.

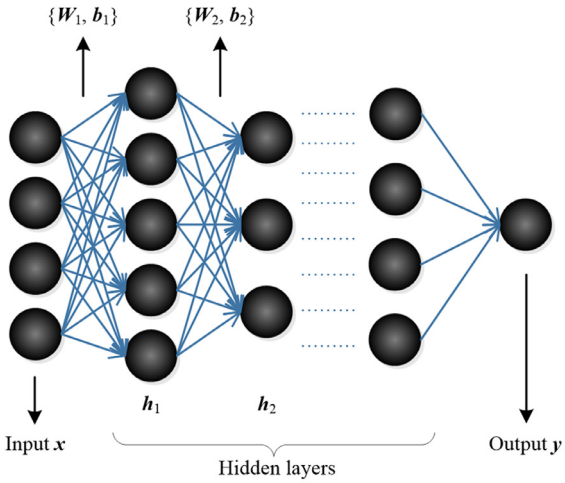
4. Pre-trained deep neural network

4.1. Architecture design

We apply feedforward DNN as the foundational forecast model, which is a quintessential deep learning model (Goodfellow et al., 2016). Similar to classical SANN, feedforward DNN begins with an input layer to match the input features, followed by multiple layers of nonlinear function connected by chain structures, and finally ends with an output layer. This model is categorized as "feedforward" because information flows from the input $\boldsymbol{x}$ through the intermediate computations (multiple hidden layers), and finally to the output $\boldsymbol{y}$. The details of this chain structure are interpreted in Fig. 2.

For feedforward DNN, the key consideration for architecture design is determining the depth of the entire network and the number of units in each layer. By monitoring the validation error of several initial experimentations, the number of hidden layers is set as 3, with 50, 100, and 50 units in each layer, respectively. Another significant design consideration is choosing an appropriate activation function [i.e., the $g(x)$ in Fig. 2]. The hyperbolic tangent activation function is applied (Eq. (2)), which belongs to the sigmoidal activation function and typically performs better than the logistic sigmoid on regression tasks.

$$g(x) = (e^{x} - e^{-x})/(e^{x} + e^{-x}), g(x) \in [-1, 1]. \quad (2)$$



Annotation: feedforward DNN arranges all hidden layers in a chain structure, with each layer being a function of the layer that preceded it. In the above structure, the first layer is given by $h_1 = g_1(W_1^T x + b_1)$, and the second layer is given by $h_2 = g_2(W_2^T h_1 + b^2)$

Fig. 2. Chain structure of feedforward DNN.

4.2. Training algorithm

One significant difference between linear models and DNN is that the nonlinearity of DNNs leads the loss function to be non-convex. Owing to its non-convexity, DNN is normally trained by using iterative, gradient-based algorithms, rather than convex optimization algorithms or linear equation solvers. In this study, the mini-batch stochastic gradient descent is applied as training algorithm, in which the gradients are computed by back-propagation. The details are described in Algorithm 3.

Compared with normal stochastic gradient descent (SGD), the mini-batch presents minor variation because the gradient is calculated by taking the average of a mini-batch of m_0 examples instead of using only one example. While training with a batch size of 1, a relatively small learning rate is required to maintain the stability of iterative process due to the high variance of gradient estimations. Additional iterative steps might be required to satisfy the stopping criterion, and the total runtime can become quite long. By increasing the batch size to m_0, the variance is reduced and the efficiency is then improved. Moreover, mini-batch SGD retains a regularizing effect to DNN with noises brought by sampling process (Wilson & Martinez, 2003).

4.3. Pre-training

Non-convex optimization is generally an extremely difficult task especially for the DNN with a huge number of training examples. Prominent challenges in DNN optimization include ill-conditioning of the Hessian matrix, local minima, long-term dependencies between parameters across layers, inexact gradient estimation, and so on. The breakthrough to overcome these challenges was developed in 2006 with a brand new algorithm for training deep belief networks (Hinton & Salakhutdinov, 2006) and the stacked autoencoders (Poultney, Chopra, & Cun, 2006). These studies are based on a similar approach, in which a greedy unsupervised pre-training is introduced before the normal supervised training process. Although theoretical basis on how unsupervised pre-training works is insufficient, this operation has achieved good performance in various applications (Kim, Calhoun, Shim, & Lee, 2016; Seyyedsalehi & Seyyedsalehi, 2015) and stipulated the development of

Algorithm 3 Mini-batch stochastic gradient descent update at training iteration k, given learning rate ε_k and initial parameter set $\boldsymbol{\theta} = \{\boldsymbol{W}, \boldsymbol{b}\}$.

while stopping criterion not met **do**
  Sample a mini-batch of m_0 examples $\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_{m_0}\}$ from the training set, with corresponding output value $y_i (i = 1, 2, ..., m_0)$.
  Calculate gradient estimate: $\hat{g} \leftarrow (1/m_0)\nabla_{\boldsymbol{\theta}} \sum_{i=1}^{m_0} L(f(\boldsymbol{x}_i; \boldsymbol{\theta}), y_i)$.
  Update parameter set: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \varepsilon_k \hat{g}$
end while

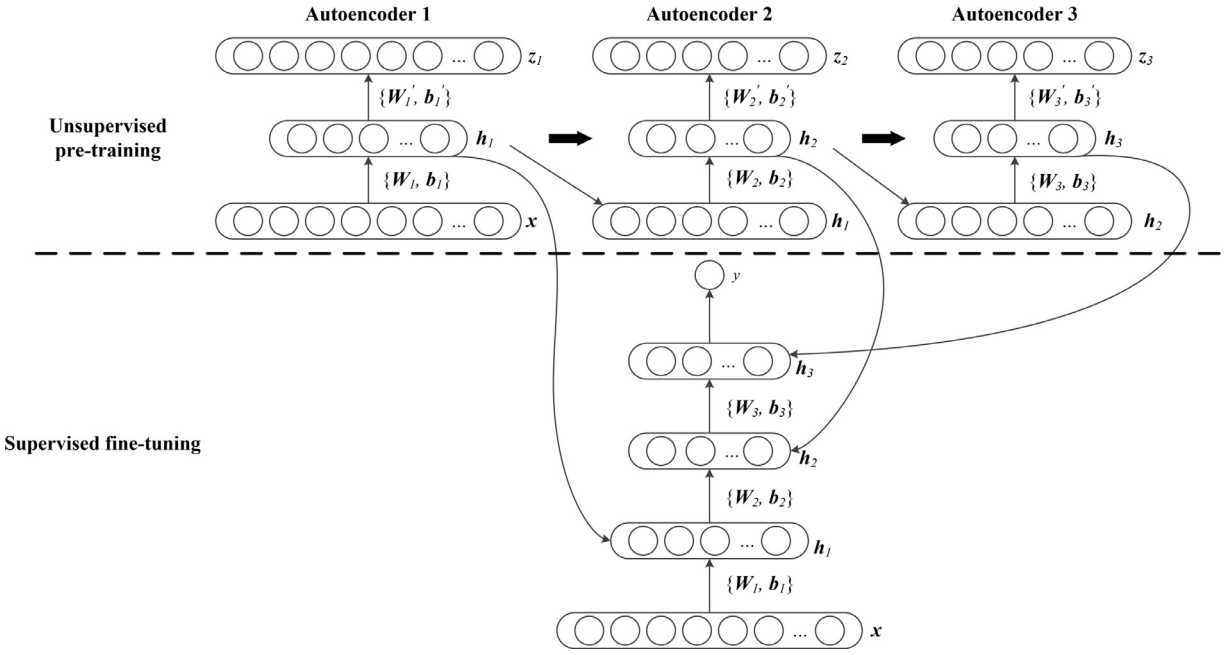


Fig. 3. SAE-based pre-training plus fine-tuning.

deep learning. Stacked autoencoders (SAE)-based pre-training is integrated in the model training process in this study.

Fig. 3 illustrates a 3-hidden-layer DNN pre-trained by SAE. In the pre-training stage, the first layer is pre-trained by Autoencoder 1. The autoencoder is a neural network that reproduces its input; thus, the target output of autoencoder is equal to the input. From Fig. 3, Autoencoder 1 takes an input of $\boldsymbol{x}$ and maps it to a hidden representation $\boldsymbol{h}_1 \in R^{d'}$ as Eq. (3), where $\boldsymbol{x} \in R^d$ belongs to the training set.

$$\boldsymbol{h}_1 = f_1(\boldsymbol{W}_1 \boldsymbol{x} + \boldsymbol{b}_1). \tag{3}$$

In Eq. (3), $f(x)$ is a non-linear activation function, and the hyperbolic tangent is selected for this study. The hidden representation $\boldsymbol{h}_1$ is then decoded to form a reconstruction $\boldsymbol{z}_1 \in R^d$ of the same shape as $\boldsymbol{x}_i$. $\boldsymbol{z}_1 \in R^d$ is calculated through a similar transformation, as shown in Eq. (4).

$$\boldsymbol{z}_1 = f_1'(\boldsymbol{W}_1' \boldsymbol{h}_1 + \boldsymbol{b}_1'). \tag{4}$$

$f'(x)$ is set as a linear transfer function in this approach, and tied weights are not applied. Under this mechanism, the parameter sets of autoencoders are trained iteratively to minimize the mean reconstruction error (i.e., $L(\boldsymbol{x}\boldsymbol{z}_1)$, Eq. (5)) over the entire training set.

$$L(\boldsymbol{x}\boldsymbol{z}_1) = \|\boldsymbol{x} - \boldsymbol{z}_1\|^2. \tag{5}$$

Similarly, Autoencoder 2 is pre-trained by taking the hidden layer of Autoencoder 1 as an input. Once the first l layers are trained, $(l+1)$th layer is able to be trained because the hidden representation from the layer below is available. After all layers are pre-trained, the network undergoes the fine-tuning stage by stacking autoencoders and conducting the normal model-training process. In this approach, we apply supervised fine-tuning, in which parameters and hidden representations obtained from autoencoders are used as initialization to train the final feedforward DNN with mini-batch SGD algorithm.

Several findings on the success of SAE-based pre-training can be derived from the operation process of SAE. Compared with convex optimization that theoretically converges starting from any initializations, non-convex loss functions trained by mini-batch SGD cannot guarantee convergence. The non-convex optimization is sensitive to the values of initial parameters. As an unusual form of regularization (Erhan et al., 2010), SAE-based pre-training boosts the mini-batch SGD, by minimizing variance while introducing bias toward the configuration of initial parameter space. Through this regularization, the parameter values are driven to the appropriate range. Another reasonable conjuncture is that unsupervised SAE prompts the initial parameters to some points which render the following optimization process more effective. The convergence rate is enhanced by avoiding local minima. To validate the effect of SAE-based pre-training, a numerical experimentation is conducted using one of our available datasets. The results (Fig. 4) are obtained from training DNN through two different methods: assigning initial parameters randomly and using SAE-based pre-training to initialize the parameters.

Fig. 4 shows the root-mean-square error (RMSE) on a validation set obtained from pre-trained and non-pre-trained DNNs. Fig. 4-(a) indicates that the pre-trained DNN yields significantly less validation errors than the non-pre-trained DNN with the same batch size (except for batch size equal to 100), controlling the number of training iteration to 100. The SAE-based pre-training helps DNN achieve improved performance by initializing parame-

Table 2
Datasets collected for analysis.

Dataset	Dependent variable	Number of original features	Mean	Standard deviation
D1	Number of daily patient arrivals to OPD		186.007	91.749
D2	Number of daily color Doppler ultrasonography (CDU) orders made in OPD	60 (including 17 calendar-based indicators, 4 holiday-related indicators and 39 meteorological factors)	6.602	4.953
D3	Number of daily computed tomography (CT) orders made in OPD		5.471	4.144
D4	Number of daily laboratory orders (e.g. blood tests and urine tests) made in OPD		26.119	15.268

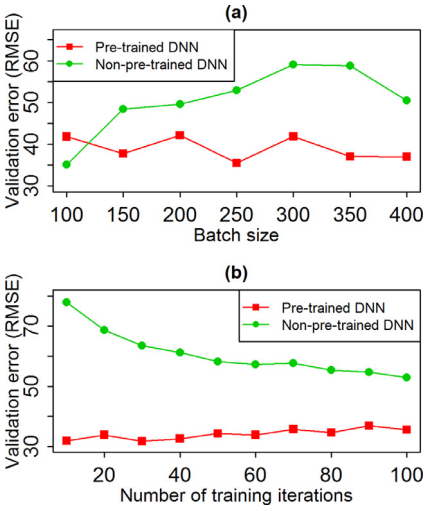


Fig. 4. Root-mean-square error of DNNs trained by two different methods.

ters to a healthy range. Fig. 4-(b) presents the significant difference between two sets of validation errors as changes of the number of iterations, controlling the batch size to 250. The random initialization tends to place parameters in a poorly generalized region, and takes time to diminish the training error. Therefore, the SAE-based pre-training algorithm significantly speeds up the convergence.

5. Real case study

5.1. Data description and experimental setup

Fig. 5 outlines the main steps of implementing the present methodology in an actual OPD. This real case study is conducted in a tertiary referral hospital located at a major city of China. The city has a warm continental climate, characterized by extreme differences in temperatures between summer and winter. OPD administrative data and order data are obtained from the hospital information system (HIS) during the period of 1 January 2014 to 31 December 2015. The original records are collected from different databases and assembled to formulate four datasets which reflect daily demand for different key resources in the OPD. All these four datasets share the same original features and relevant details are shown in Table 2.

For D1 to D4, the first 527 instances are used for feature selection, model training and parameter validation, and the rest instances are used for testing. All dependent variables of four training sets are chronologically plotted in Fig. 6. These time series

Table 3
Independent variables included in original feature set.

Feature type	Feature No.	Description
Calendar-based factors	X1 to X11	Dummy variables for month of the year
	X12 to X17	Dummy variables for day of the week
	X18 to X21	Dummy variables for holiday-related indicators
Meteorological factors	X22 to X24	Maximum, mean, and minimum temperature
	X25 to X27	Maximum, mean, and minimum dew point
	X28 to X30	Maximum, mean, and minimum humidity
	X31 to X33	Maximum, mean, and minimum air pressure
	X34 to X36	Maximum, mean, and minimum visibility
	X37 to X38	Maximum and mean wind speed
	X39	Precipitation
	X40	Cloud cover
	X41	Snow signal
	X42	Wind direction
	X43 to X60	Changes of X22 to X39

plots indicate that four different kinds of demands show a similar seasonal pattern. To achieve any meaningful statistical analysis of time series, the stationarity test is carried out after removing the seasonality, including augmented Dickey-Fuller (ADF) test, autocorrelation function (ACF) plot, and partial autocorrelation function (PACF) plot (Fig. 7). As shown in Fig. 7, small p-values obtained by ADF tests allow us to reject the non-stationary hypothesis for D1 to D4. Furthermore, both the cutting-off pattern of ACF plot and the tailing-off pattern of PACF plot confirm the temporal stationarity.

All the original features are listed and numbered in Table 3. The calendar-based factors are converted into 0–1 dummy variables. Holiday-related indicators include whether a public holiday or not and whether before or after a public holiday. Meteorological factors are retrieved from an online database (The Weather Channel Interactive, 2017). Changes of these factors are obtained by calculating the increment compared with one day before.

As is known, a longer prediction horizon can facilitate OPD managers to make stable and robust downstream decision makings. In addition, both managers and staff will have more buffer time to get used to the periodic schedule if the prediction horizon is long enough. However, with the increase of prediction horizon, the meteorological data obtained from weather forecast would become less reliable, which may affect the accuracy of predicted val-

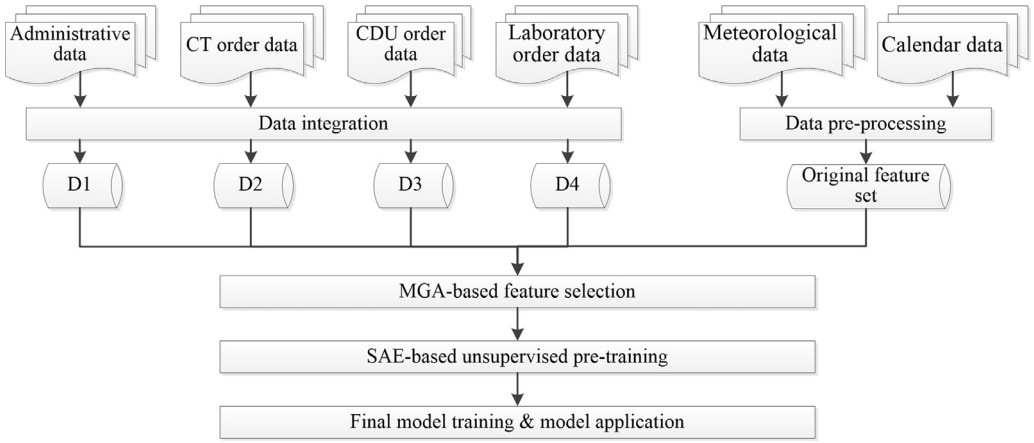


Fig. 5. The main steps of our case study.

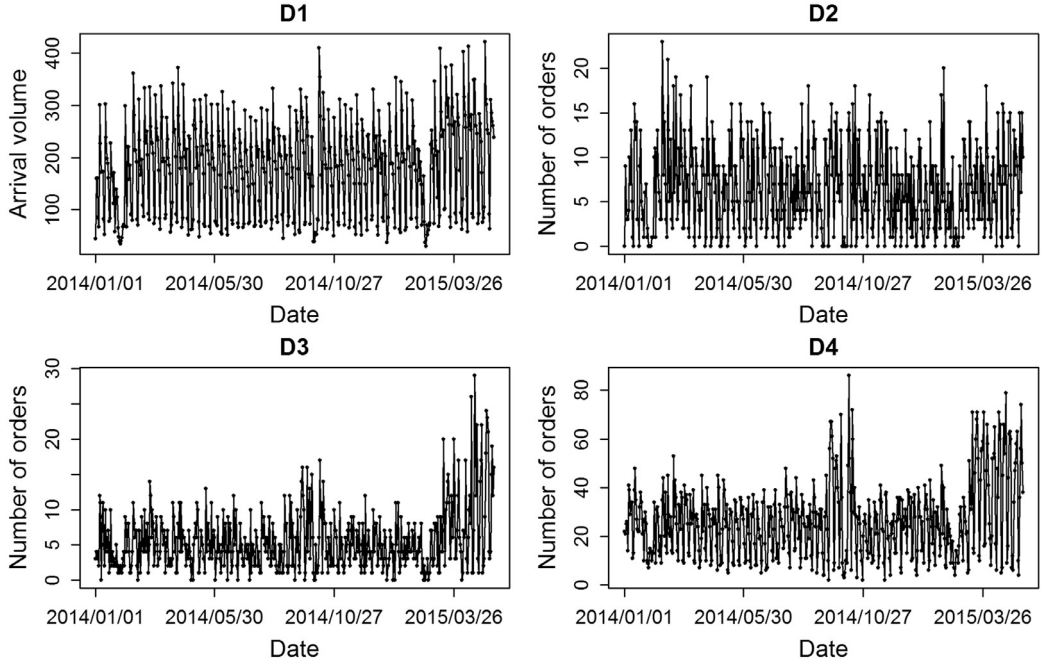


Fig. 6. Time series plots for dependent variables of four training sets.

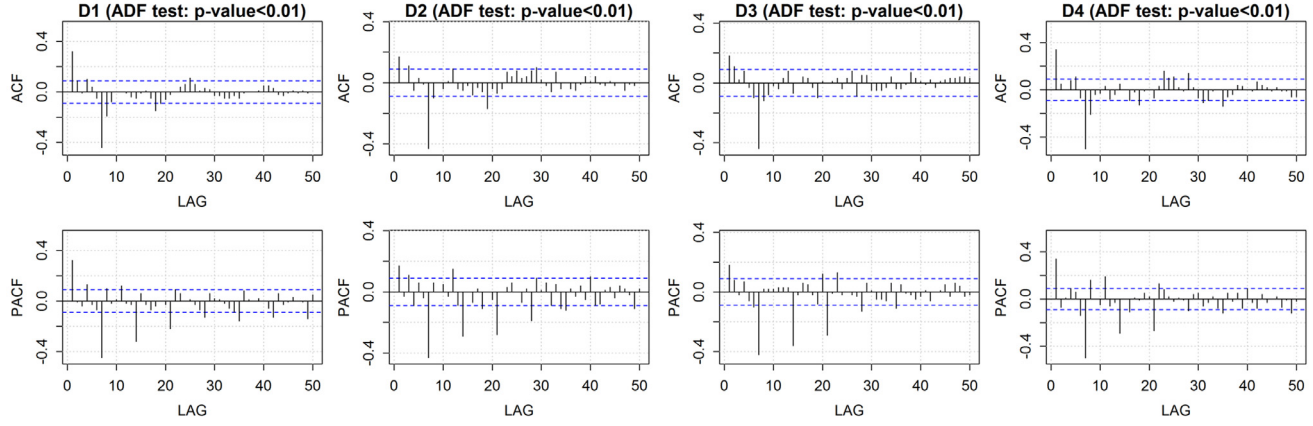


Fig. 7. The ADF test, ACF and PACF plot for dependent variables of D1 to D4.

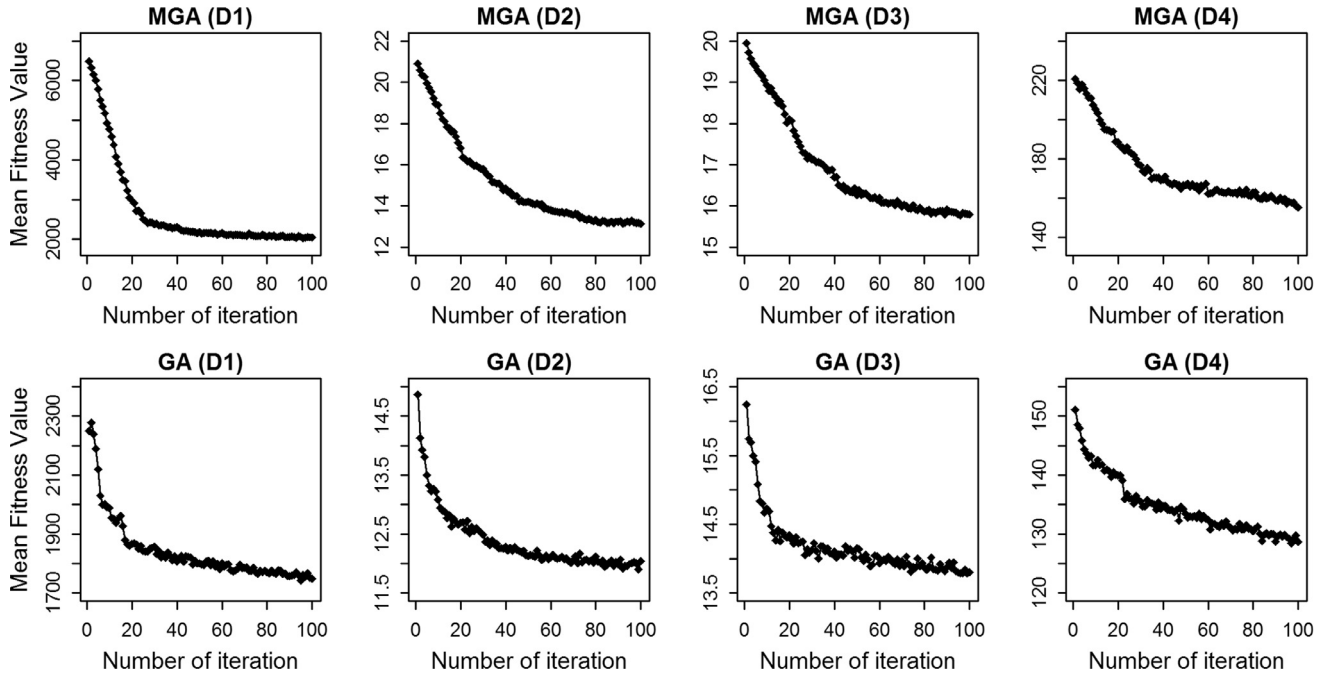


Fig. 8. Convergence process of MGA and GA in four datasets.

ues. To trade-off between the requirement of OPD and the reliability of weather forecast, the prediction horizon is fixed to 28 days ahead.

The performance of neural network-based models is sensitive to input representation; as such, all independent variables are normalized through Eq. (6).

$$x_i' = [x_i - \min(x)]/[\max(x) - \min(x)]. \quad (6)$$

It is worth mentioning that the main objective of FS is to obtain fewer but more effective features and meanwhile maintaining or improving the model's forecast accuracy. In Section 5.2, comparative experimentations are carried out to test if MGA is able to fulfill this objective. A normal GA-based FS algorithm and principal components analysis (PCA) are implemented as benchmark algorithms. In this stage, the convergence efficiency of MGA is tested in Section 5.2.1, by comparing iterative processes between MGA and GA. The reduction of feature dimension is tested in Section 5.2.2. The quality of acquired feature subsets is tested by using selected features to train the pre-trained DNN (Section 5.2.3) and other frequently-used models on this demand forecasting problem (Section 5.2.4). These models include multiple linear regression (MLR), autoregressive integrated moving average with exogenous variables (ARIMAX), and SANN. Furthermore, the original feature set is employed to carry out the same forecast tasks during this stage. The results are utilized to test if these three FS algorithms improve the forecast performance. In Section 5.3, the effectiveness of combining MGA and pre-trained DNN is examined by comparing it with all other possible combinations. In Section 5.4, some management implications are explored based on experimental outcomes. Overall, RMSE is set as the metric to evaluate forecast accuracy. All involved experiments are run on R version 3.1.2 and Python 2.7.10. The computer used is Inter(R) Core(TM) i7-4600 CPU, 8GB RAM, operated by Windows 64-bit Operating System.

5.2. Effectiveness of MGA on FS tasks

5.2.1. Convergence efficiency of MGA and GA

Several hyper-parameters are pre-defined before implementing MGA and GA. The population size of chromosomes is controlled as 30 for both MGA and GA. For MGA, the hyper-parameter λ is approximately set to 0.3% of the validation error obtained by training DNN with the original feature set. For GA, the crossover rate is set as 0.8 and the mutation rate is 0.1. The total number of iterations is fixed to 100 times, and the mean fitness values are monitored to determine the convergence of two algorithms. The result is shown in Fig. 8.

By comparing the scale of vertical-axes, MGA achieves a more significant decrement in mean fitness value than GA. For D2 and D3, the mean fitness value of MGA becomes stable around 80 iterations, and for D1 and D4, it is around 60 iterations, although slight fluctuations can be noticed because of imperfect evaluation of DNN. The stationary interval cannot be captured in GA (D1) and GA (D4), and the mean fitness value continuously decreases during the entire evolution process. Hence, MGA improves the convergence efficiency because of the initialization strategy and fitness-based crossover operator.

Table 4
Dimensionality reduction of MGA, GA, and PCA for the four datasets.

Dataset	Number of selected features			Reduction rate		
	MGA	GA	PCA	MGA	GA	PCA
D1	21	30	33	65%	50%	45%
D2	14	29		76.67%	51.67%	
D3	16	20		73.33%	66.67%	
D4	20	36		66.67%	40%	

5.2.2. Feature dimensionality reduction

Dimensionality reduction is a significant performance evaluation criterion of any FS approach. Under similar generalization error, higher dimensionality reduction indicates that the corresponding FS algorithm is able to extract more crucial information from the original feature space. Additionally, a higher dimensionality reduction improves the efficiency of model-training process. The number of selected features through MGA, GA, and PCA are analyzed, and the results are shown in Table 4. For PCA, principal components are retained with 95.48% cumulative importance.

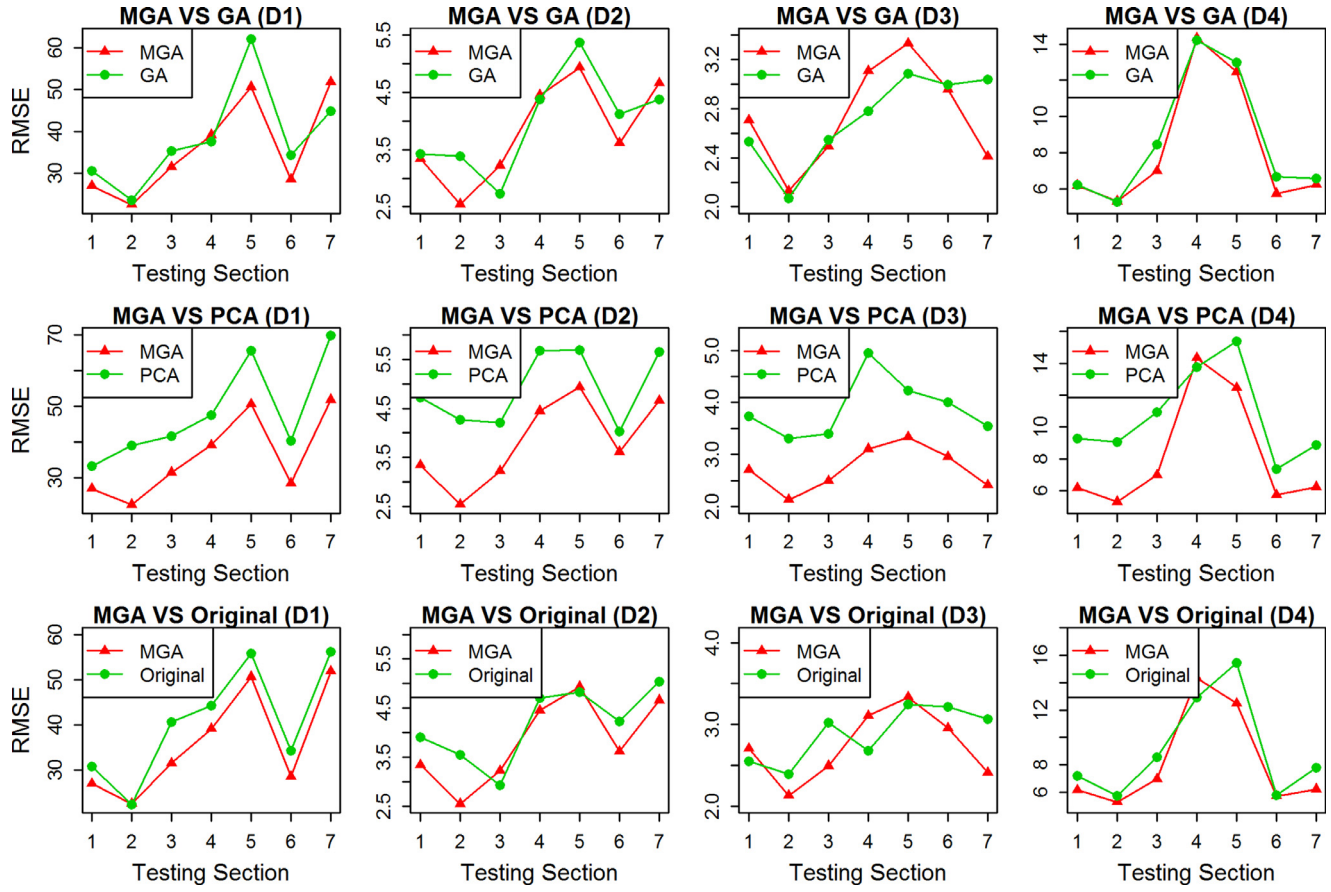


Fig. 9. Impact of different FS approaches on the forecast accuracy of Pre-trained DNN.

Table 4 shows that MGA achieves the highest reduction rate among four datasets, indicating that MGA outperforms normal GA and PCA in reducing the number of original features. This success is attributed to the fitness evaluation process of MGA, in which the penalty on the size of feature subset is introduced. Numerous features tend to diminish in the next generation, whereas elites strive for survival to maintain the quality of population.

5.2.3. Impact of MGA on pre-trained DNN

In this stage, multiple pre-trained DNNs are trained and tested with original features and feature subsets obtained by MGA, GA, and PCA. Generalization error on testing sets are compared and analyzed. A cumulative training set is applied to increase the reliability of forecast results, concerning the actual application environment and the time-order of training examples. The whole testing set is divided into seven sequential sections each containing 28 examples (according to the fixed prediction horizon). The first forecast model is trained with only the original training set, and its forecast accuracy is calculated on the first section of the testing set. Subsequently, the second forecast model is trained with the original training set plus the first section of testing set, and the forecast accuracy is determined with the second section of the testing set. Each testing section is added to the previous training set after the last prediction is accomplished. This training process can be regarded as a sequential cross-validation process, and the results are shown in Figs. 9. The mean RMSEs of seven testing sections are listed in Table 5.

Fig. 9 and Table 5 show the forecast accuracy of FS approaches, with the cooperation of pre-trained DNN. In Fig. 9, the forecast results with MGA shows a relatively stationary pattern without abnormality. Generally, with respect to pre-trained DNN, normal GA still achieves acceptable results but requires more features than MGA. PCA yields the lowest accuracy in all the four datasets. Based on the mean RMSE listed in Table 5, MGA achieves a significantly higher accuracy for D1 and D4. In Dataset D1, the mean RMSE of MGA is 35.885, which is 6.4%, 25.5%, and 11.7% lower than GA, PCA, and the original, respectively. In Dataset D4, the mean RMSE of MGA is 8.176, which is 5.2%, 23.3%, and 9.9% lower than GA, PCA, and the original, respectively. For Datasets D2 and D3, the performance of MGA is similar to that of GA, and MGA achieves 8.3% and 5.1% lower mean RMSE than the original, respectively. Overall, these comparative results demonstrate that MGA has a significant positive effect on training pre-trained DNN.

Table 5
Mean RMSE of pre-trained DNNs, trained and tested with original features and feature subsets obtained by MGA, GA, and PCA.

Dataset	Feature set			
	MGA	GA	PCA	Original
D1	35.885	38.349	48.204	40.624
D2	3.824	3.970	4.890	4.169
D3	2.734	2.721	3.880	2.881
D4	8.176	8.623	10.664	9.075

5.2.4. Impact of MGA on other frequently-used demand forecasting models

To further validate the universality of feature subsets obtained by MGA, other frequently-used demand forecasting models are trained and tested with original features and feature subsets obtained by MGA, GA, and PCA. Results are shown in Figs. 10–12 and

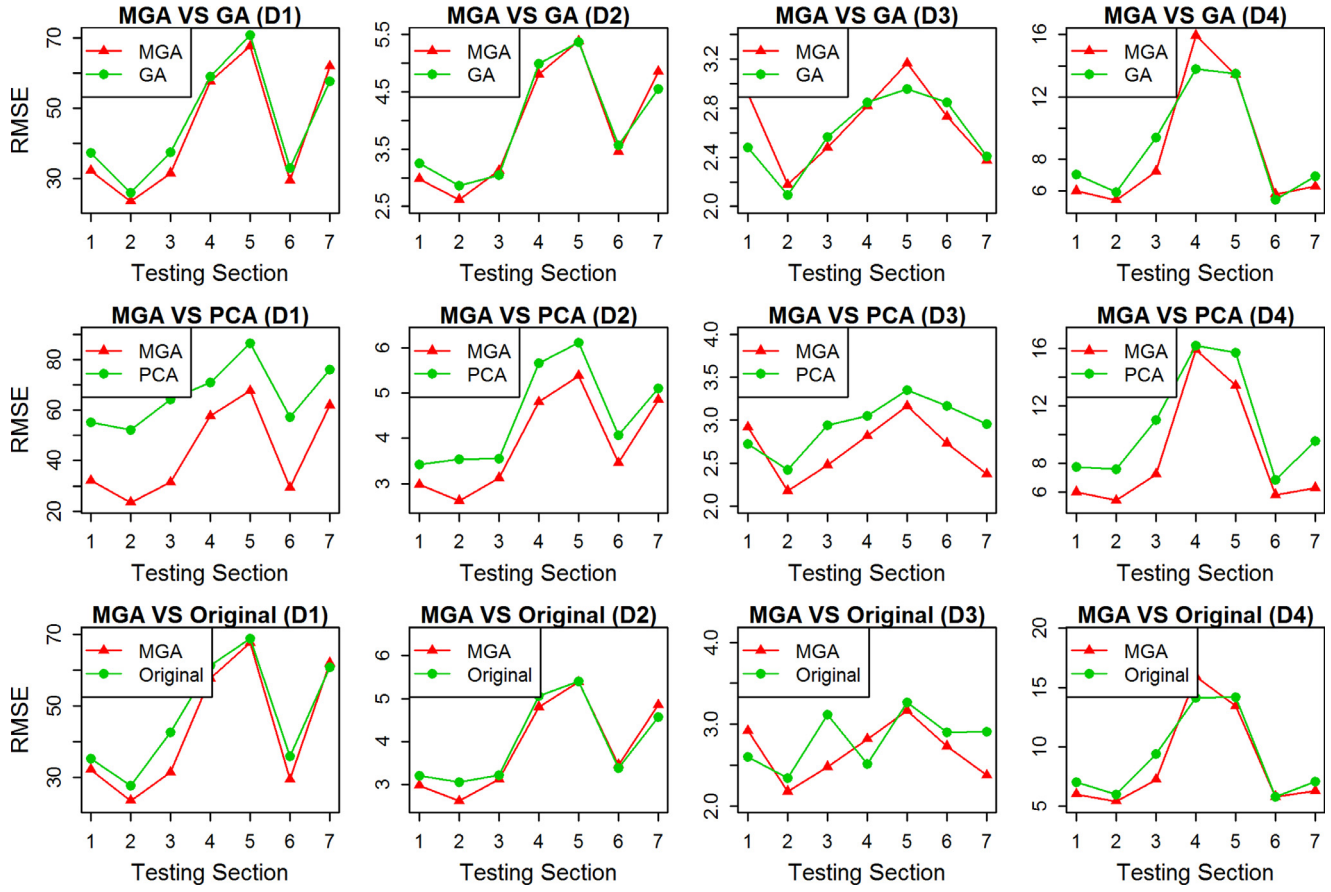


Fig. 10. Impact of different FS approaches on the forecast accuracy of MLR.

Table 6
Mean RMSE of MLR, ARIMAX, and SANN, with respect to different FS methods. The best result is shown in bold face.

Forecast model	D1				D2			
	MGA	GA	PCA	Original	MGA	GA	PCA	Original
MLR	**43.382**	45.864	66.065	47.437	**3.889**	3.949	4.497	3.984
ARIMAX	40.104	**38.250**	84.183	39.417	**3.951**	3.973	4.506	3.998
SANN	**48.962**	53.936	52.425	65.490	**3.921**	3.987	4.670	4.028
	D3				D4			
	MGA	GA	PCA	Original	MGA	GA	PCA	Original
MLR	2.666	**2.599**	2.946	2.806	**8.580**	8.860	10.665	9.081
ARIMAX	2.767	**2.646**	3.016	2.813	**8.910**	9.061	11.392	9.024
SANN	**2.660**	2.695	3.541	2.968	**8.703**	9.256	12.320	10.894

mean RMSEs are listed in Table 6. Similar results are acquired from MLR and SANN, in which MGA achieves the highest forecast accuracy, whereas PCA performs worst in almost all cases. For MLR with datasets D1 and D4, MGA achieves 5.4% and 3.2% lower mean RMSE than GA, 34.3% and 19.5% lower than PCA, and 8.5% and 5.5% lower than the original. For SANN with D1 and D4, MGA achieves 9.2% and 6.0% lower than GA, 6.6% and 29.4% lower than PCA, and 25.2% and 20.1% lower than the original. In datasets D2 and D3, MGA and GA still improve the forecast accuracy than using original features, although the decrease on mean RMSE is insignificant. For ARIMAX, no one FS algorithm dominates others in improving forecast accuracy. For example, in dataset D1, GA improves forecast accuracy and proposed MGA slightly worsens it compared with the original. By contrast, in dataset D4, GA exhibits worse forecast accuracy compared with using the original, whereas MGA improves it. The reason behind it is that the parameter-fitting and forecast performance of ARIMAX depend more on historical trend of dependent variables, as a time series regression model. Based on these comparative analyses, it can be demonstrated that feature subsets selected by MGA have a universal adaptability to other demand forecasting models. The proposed MGA-based FS has a positive effect on both model-training efficiency and forecast accuracy, if the forecast model is sensitive to input features.

5.3. Combination of MGA and pre-trained DNN

Experimental results were further analyzed to validate the predictive power of pre-trained DNN, and the combination of MGA and pre-trained DNN as well. The comparison is depicted through bar charts shown in Fig. 13.

Fig. 13 indicates that pre-trained DNN obtains remarkably robust results under different FS approaches and different datasets. In dataset D1 and D4, pre-trained DNN achieves the lowest mean RMSE with respect to MGA, GA, and PCA, though a slightly higher

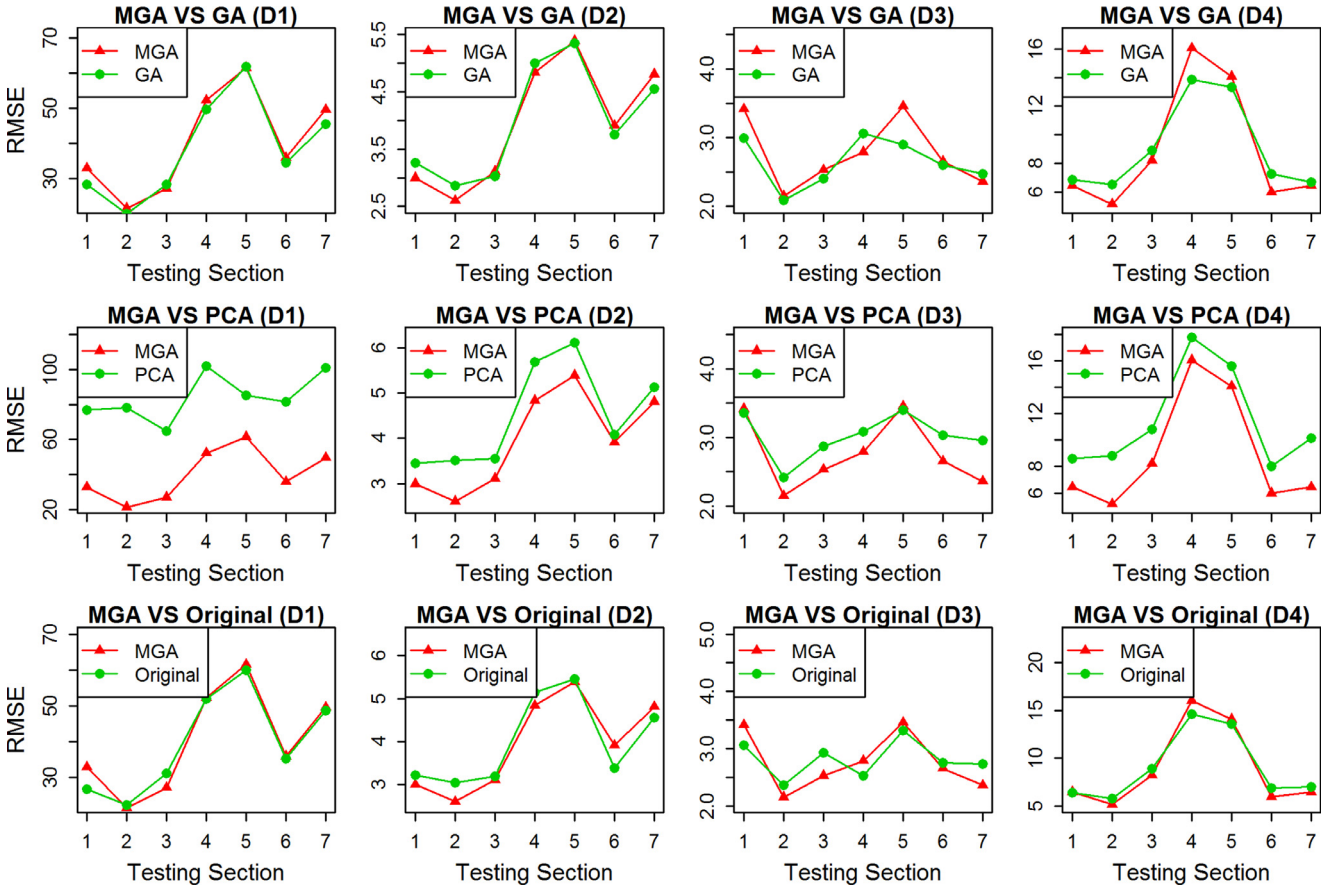


Fig. 11. Impact of different FS approaches on the forecast accuracy of ARIMAX.

mean RMSE than ARIMAX with the original (D1). The significant difference between pre-trained DNN and SANN indicates that neural networks with shallow architectures and random initialization are difficult to converge, especially with a large number of noisy inputs. Furthermore, poor performances of SANN validate the necessity of introducing deep architectures and unsupervised pre-training processes.

Among all involved combinations between FS methods and forecast models, MGA plus pre-trained DNN outperforms others in all pairwise comparisons for D1, D2, and D4. As for D3, MGA plus pre-trained DNN can maintain the forecast accuracy to a desired level though the dependent variable is well explained by linear methods. The results indicate that pre-trained DNN obviously stands out among the parallel experiments and optimally strengthens the advantage of MGA. Therefore, the combination of MGA and pre-trained DNN possesses strong predictive power and universal property, which can be deployed as a universal tool to fulfill different demand forecasting tasks in OPD.

5.4. Management implications for practitioners

Besides improving theoretical methods on feature selection and time series prediction, the results of our study have crucial implications on resource allocation and surge capacity for OPD. The 28-day-ahead prediction satisfies the timeliness of downstream decision-makings, such as fine-tuning staff schedules and procurement plans. Improved forecast accuracy further ensures the quality and robustness of them, as well as the reliability of obtained feature subsets. Table 7 lists the detail of feature subsets obtained by MGA. To provide some practical insights for OPD managers, some similarities and differences of feature combinations are extracted from four datasets. From Table 7, we find that nearly all the day-of-week indicators are selected in D1 to D4, which indicates a common weekly seasonality. For month-of-year indicators, X7 and X8 are selected in D1 and D2, so it implies that the patient flow and the demand for CDU test varies in summer. For holiday-related indicators, both X18 and X20 are selected in D1, D2, and D4, which indicates that the day after holiday is just as meaningful to OPD demand forecast as the day on holiday. Compared with calendar-based factors, the effect of meteorological factors shows huge differences among four datasets. As for the temperature-related factors, the demand for CT test (D3) is affected by current temperature (X22, X23, X25, X27), whereas the patient flow (D1) is more sensitive to temperature changes (X44, X45, X47). The effect of temperature on demand for laboratory test (D4) presents a mixed mode (X22, X25, X48). In particular, the demand for CDU test (D2) is less sensitive to temperature or temperature change, in which only X27 is selected. For other meteorological factors, the demand

Table 7
The detail of feature subsets obtained by MGA-based FS.

Dataset	Feature subset obtained by MGA	
	Calendar-based factors	Meteorological factors
D1	X7, X8, X10, X12, X13, X14, X15, X16, X17, X18, X20	X22, X32, X38, X39, X44, X45, X47, X49, X51, X57
D2	X6, X7, X8, X12, X13, X14, X15, X16, X18, X20	X27, X39, X50, X51
D3	X1, X6, X9, X12, X13, X14, X15, X16, X18	X22, X23, X25, X27, X53, X54, X55
D4	X2, X3, X6, X11, X12, X13, X14, X15, X16, X18, X19, X20, X21	X22, X25, X32, X34, X39, X41, X48

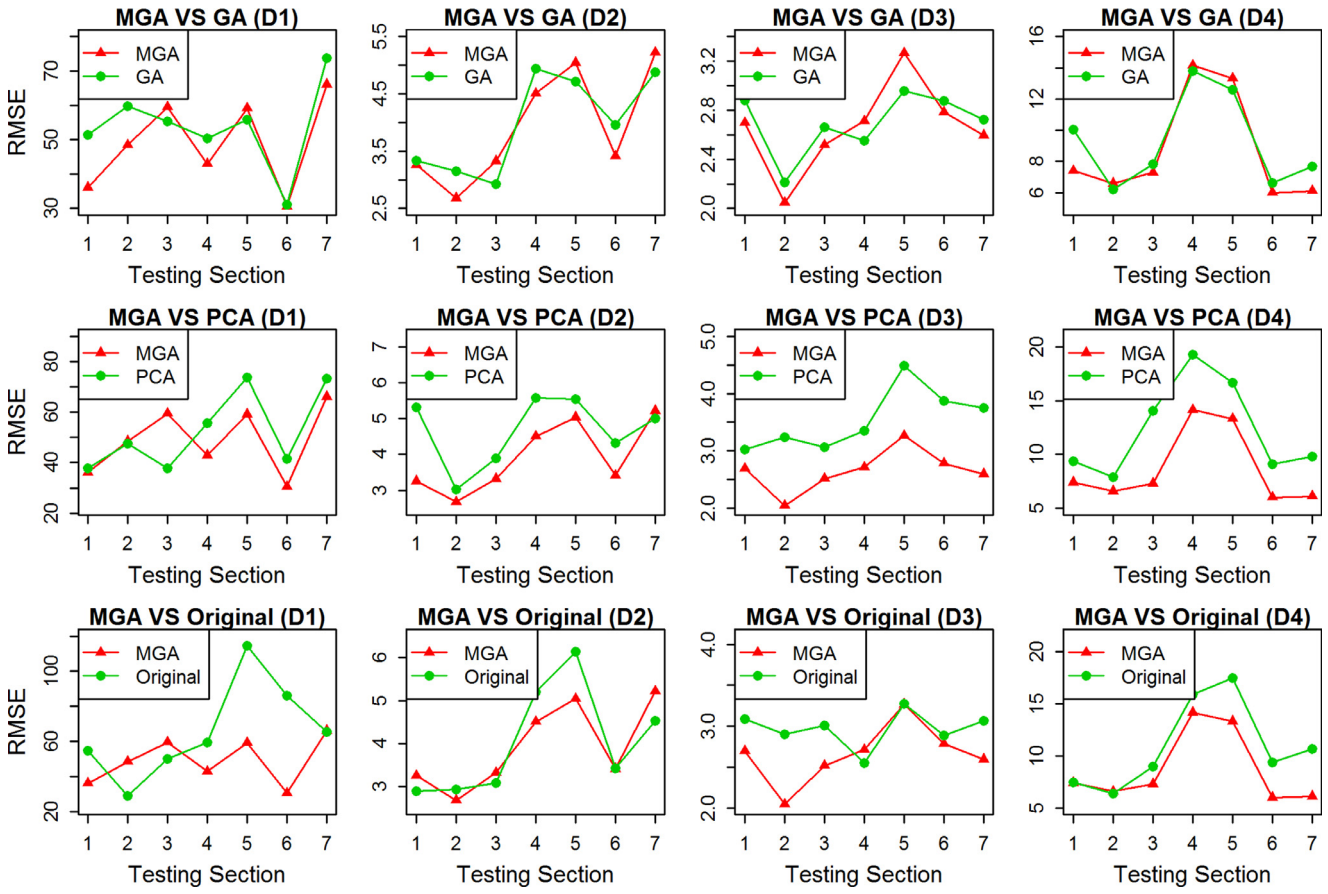


Fig. 12. Impact of different FS approaches on the forecast accuracy of SANN.

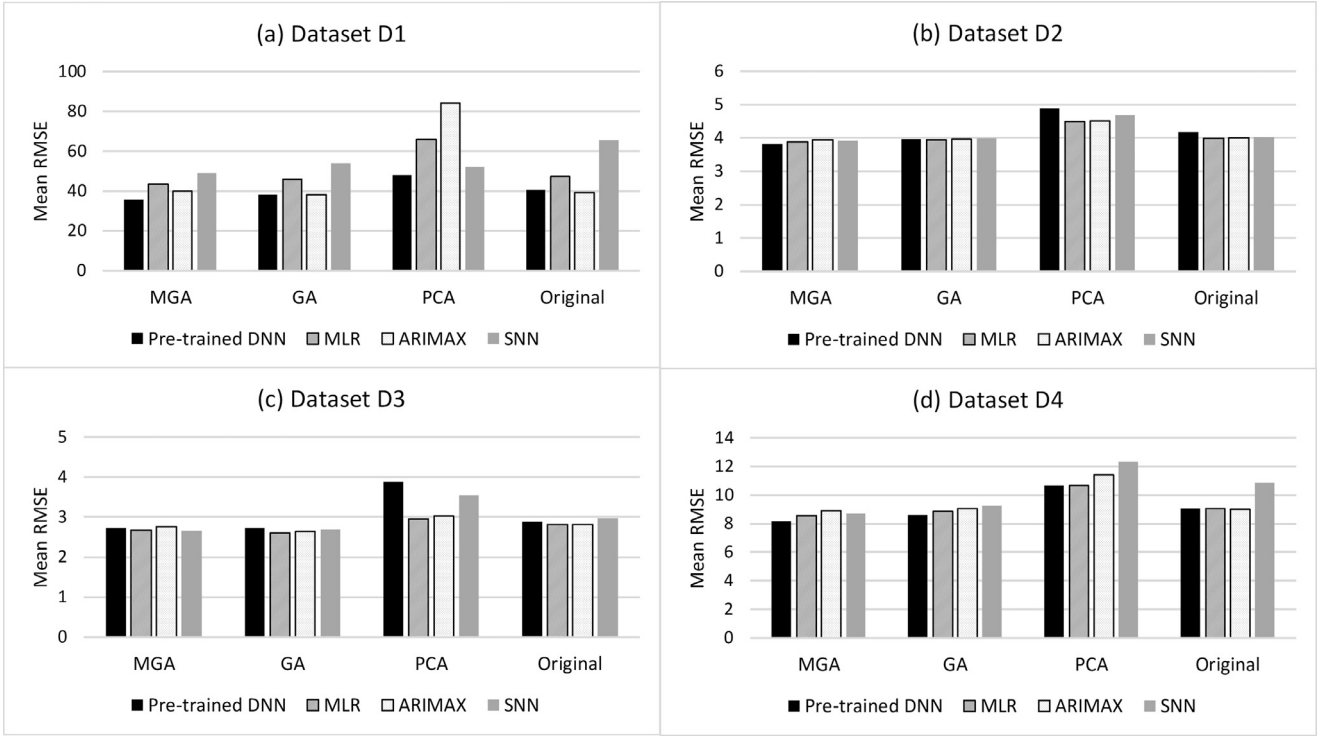


Fig. 13. The mean RMSE on seven testing sections generated by all involved combinations.

for CDU test (D2) relies solely on humidity related factors (X39, X50, X51). Changes of air pressure and visibility are both included in D3, which implies that the air quality has an important influence on the demand for CT test. The snow signal is merely selected in D4, though the city is snowy in winter.

6. Conclusion

This study proposes a novel hybrid methodology to generate a more accurate and robust model for demand forecasting in OPD, by combining FS and demand forecasting models. As for FS, a modified version of GA named as MGA is proposed to extract elite features with significant effect on demand. In MGA, key operators of GA are re-designed to enhance the search ability for FS. A *t*-statistic-based filter method is utilized in the initialization process, and the imperfect evaluation of fitness values is implemented via restricted DNN. The crossover operator takes advantage of individuals' fitness values and is utilized to explore the potential information of feature combination. As for model selection, a well-designed DNN is selected as the forecast model to increase the forecast accuracy. An SAE-based pre-training algorithm is integrated in the model-training process to overcome the challenges of optimization.

Based on the results of comparative experiments, the main contributions of our study are categorized as two fold. Theoretically, compared with GA and PCA, MGA improves the quality and efficiency of FS, with less selected features and universally higher forecast accuracy under different forecast models. Pre-trained DNN obviously stands out among the parallel experiments and optimally strengthens the advantage of MGA, compared with MLR, ARIMAX and SANN. The combination of MGA and pre-trained DNN possesses strongest predictive power among all involved combinations, and it can be deployed as a universal tool to fulfill different demand forecasting tasks in OPD. Practically, our hybrid methodology has been applied for demand forecasting in an actual OPD. Predicted values with high accuracy have crucial implications on resource allocation and surge capacity for OPD. Elite features obtained by MGA can provide practical insights on potential association between manifold feature combinations and demand variance, which is a complement of existing expertise knowledge. Furthermore, our hybrid method would be generalized as a tool for knowledge discovery and data mining in other intelligent systems with similar characteristics, such as load forecasting for power systems planning and operation, demand forecasting in supply chain management, etc.

In future, more effort will be made to improve the planning and implementation of proposed methodology. For MGA and other similar wrapper FS methods, one promising direction for improving their efficiency is exploring how to strengthen the advantage of filters. In the current study, the results of a filter-based FS approach have benefited MGA in the initialization process. Moreover, core operators or ideas in a certain filter method would be summarized as a rule integrated in wrapper methods. In this regard, the efficiency of wrapper methods might be further improved. Another research topic for future work is the generalization of proposed methodology. Additional combinations will be established between MGA and other deep architectures to accomplish a wider range of classification or regression tasks.

Acknowledgments

This work is partly supported by the Hong Kong RGC Grant No. T32-102/14N and the NSFC Key Project Grant no. 71231007.

References

Aboagye-Sarfo, P., Mai, Q., Sanfilippo, F. M., Preen, D. B., Stewart, L. M., & Fatovich, D. M. (2015). A comparison of multivariate and univariate time series approaches to modelling and forecasting emergency department demand in Western Australia. *Journal of Biomedica, 57*, 62–73.

Afilal, M., Yalaoui, F., Dugardin, F., Amodeo, L., Laplanche, D., & Blua, P. (2016). Forecasting the emergency department patients flow. *Journal of Medical Systems, 40*, 1–18.

Aizenberg, I., Sheremetov, L., Villa-Vargas, L., & Martinez-Munoz, J. (2016). Multilayer neural network with multi-valued neurons in time series forecasting of oil production. *Neurocomputing, 175*, 980–989.

Ali Jan Ghasab, M., Khamis, S., Mohammad, F., & Jahani Fariman, H. (2015). Feature decision-making ant colony optimization system for an automated recognition of plant species. *Expert Systems with Applications, 42*, 2361–2370.

Boyle, J., Jessup, M., Crilly, J., Green, D., Lind, J., Wallis, M., et al. (2012). Predicting emergency department admissions. *Emergency Medicine Journal, 29*, 358–365.

Cavalcante, R. C., Brasileiro, R. C., Souza, V. L. F., Nobrega, J. P., & Oliveira, A. L. I. (2016). Computational intelligence and financial markets: A survey and future directions. *Expert Systems with Applications, 55*, 194–211.

Chafekar, D., Xuan, J., & Rasheed, K. (2003). Constrained multi-objective optimization using steady state genetic algorithms. In E. CantuPaz, J. A. Foster, K. Deb, L. D. Davis, R. Roy, U. M. Oreilly et al. (Eds.), *Genetic and evolutionary computation - Gecco 2003, pt I, proceedings* (Vol. 2723, pp. 813–824). Berlin: Springer-Verlag Berlin.

Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering, 40*, 16–28.

Cireşan, D., Meier, U., Masci, J., & Schmidhuber, J. (2012). Multi-column deep neural network for traffic sign classification. *Neural Network, 32*, 333–338.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematical Control and Signal System, 2*, 303–314.

Dahl, G. E., Yu, D., Deng, L., & Acero, A. (2012). Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech and Language Processing, 20*, 30–42.

Doquire, G., & Verleysen, M. (2013). Mutual information-based feature selection for multilabel classification. *Neurocomputing, 122*, 148–155.

Erhan, D., Bengio, Y., Courville, A., Manzagol, P.-A., Vincent, P., & Bengio, S. (2010). Why does unsupervised pre-training help deep learning. *Journal of Machine Learning Research, 11*, 625–660.

Gashler, M. S., & Ashmore, S. C. (2016). Modeling time series data with deep Fourier neural networks. *Neurocomputing, 188*, 3–11.

Gen, M., & Cheng, R. (2000). *Genetic algorithms and engineering optimization*: Vol. 7. John Wiley & Sons.

Ghareb, A. S., Bakar, A. A., & Hamdan, A. R. (2016). Hybrid feature selection based on enhanced genetic algorithm for text categorization. *Expert Systems with Applications, 49*, 31–47.

Goodarzi, M., Freitas, M. P., & Jensen, R. (2009). Ant colony optimization as a feature selection method in the QSAR modeling of anti-HIV-1 activities of 3-(3,5-dimethylbenzyl)uracil derivatives using MLR, PLS and SVM regressions. *Chemometrics and Intelligent Laboratory Systems, 98*, 123–129.

Goodfellow, I., Bengio, Y., & Courville, A.. *Deep learning* Accessed 22.10.16 http://www.deeplearningbook.org.

Goodfellow, I. J., Bulatov, Y., Ibarz, J., Arnoud, S., & Shet, V. (2013). Multi-digit number recognition from street view imagery using deep convolutional neural networks. arXiv preprint arXiv:1312.6082.

Gunasundari, S., Janakiraman, S., & Meenambal, S. (2016). Velocity bounded boolean particle swarm optimization for improved feature selection in liver and kidney disease diagnosis. *Expert Systems with Applications, 56*, 28–47.

Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., & Lew, M. S. (2016). Deep learning for visual understanding: A review. *Neurocomputing, 187*, 27–48.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A. r., Jaitly, N., et al. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Process Magazine, 29*, 82–97.

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science, 313*, 504–507.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks, 2*, 359–366.

Hossain, M., Rekabdar, B., Louis, S. J., & Dascalu, S. (2015). Forecasting the weather of Nevada: A deep learning approach. In *2015 international joint conference on neural networks (IJCNN)* (pp. 1–6).

Hu, Q., Zhang, R., & Zhou, Y. (2016). Transfer learning for short-term wind speed prediction with deep neural networks. *Renewable Energy, 85*, 83–95.

Hu, Z. Y., Bao, Y. K., & Xiong, T. (2014). Comprehensive learning particle swarm optimization based memetic algorithm for model selection in short-term load forecasting using support vector regression. *Application Soft Computing, 25*, 15–25.

Hwang, U., McCarthy, M. L., Aronsky, D., Asplin, B., Crane, P. W., Craven, C. K., et al. (2011). Measures of crowding in the emergency department: A systematic review. *Academic Emergency Medicine, 18*, 527–538.

Jones, S. S., Evans, R. S., Allen, T. L., Thomas, A., Haug, P. J., Welch, S. J., et al. (2009). A multivariate time series approach to modeling and forecasting demand in the emergency department. *Journal of Biomedical Informatics, 42*, 123–139.

Jones, S. S., Thomas, A., Evans, R. S., Welch, S. J., Haug, P. J., & Snow, G. L. (2008). Forecasting daily patient volumes in the emergency department. *Academic Emergency Medicine, 15*, 159–170.

Jurado, S., Nebot, A., Mugica, F., & Avellana, N. (2015). Hybrid methodologies for electricity load forecasting: Entropy-based feature selection with machine learning and soft computing techniques. *Energy, 86*, 276–291.

Kahou, S. E., Pal, C., Bouthillier, X., Froumenty, P., Gülçehre, C., & Memisevic, R. (2013). Combining modality specific deep neural networks for emotion recognition in video. In *Proceedings of the 15th ACM on international conference on multimodal interaction* (pp. 543–550). ACM.

Kam, H. J., Sung, J. O., & Park, R. W. (2010). Prediction of daily patient numbers for a regional emergency medical center using time series analysis. *Healthcare Informatics Research, 16*, 158–165.

Kazemi, S. M. R., Hoseini, M. M. S., Abbasian-Naghneh, S., & Rahmati, S. H. A. (2014). An evolutionary-based adaptive neuro-fuzzy inference system for intelligent short-term load forecasting. *International Transactions in Operational Research, 21*, 311–326.

Kim, J., Calhoun, V. D., Shim, E., & Lee, J.-H. (2016). Deep neural network with weight sparsity control and pre-training extracts hierarchical features and enhances classification performance: Evidence from whole-brain resting-state functional connectivity patterns of schizophrenia. *NeuroImage, 124*, 127–146 Part A.

Koprinska, I., Rana, M., & Agelidis, V. G. (2015). Correlation and instance based feature selection for electricity load forecasting. *Knowledge-Based Systems, 82*, 29–40.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*, 436–444.

Lin, C.-H., Chen, H.-Y., & Wu, Y.-S. (2014). Study of image retrieval and classification based on adaptive features using genetic algorithm feature selection. *Expert Systems with Applications, 41*, 6611–6621.

Lin, F., Liang, D., Yeh, C.-C., & Huang, J.-C. (2014). Novel feature selection methods to financial distress prediction. *Expert Systems with Applications, 41*, 2472–2483.

Lin, S. W., Ying, K. C., Chen, S. C., & Lee, Z. J. (2008). Particle swarm optimization for parameter determination and feature selection of support vector machines. *Expert Systems with Applications, 35*, 1817–1824.

Liu, Y., Yin, Y. F., Gao, J. J., & Tan, C. L. (2008). Wrapper feature selection optimized SVM model for demand forecasting. In G. J. Wang, J. Chen, M. R. Fellows, & H. D. Ma (Eds.), *Proceedings of the 9th international conference for young computer scientists: Vol. 1–5* (pp. 953–958).

Malhi, A., & Gao, R. X. (2004). PCA-based feature selection scheme for machine defect classification. *IEEE Transactions on Instrumentation and Measurement, 53*, 1517–1525.

Marcilio, I., Hajat, S., & Gouveia, N. (2013). Forecasting daily emergency department visits using calendar variables and ambient temperature readings. *Academic Emergency Medicine, 20*, 769–777.

McCarthy, M. L., Zeger, S. L., Ding, R., Aronsky, D., Hoot, N. R., & Kelen, G. D. (2008). The challenge of predicting demand for emergency department services. *Academic Emergency Medicine, 15*, 337–346.

Menke, N. B., Caputo, N., Fraser, R., Haber, J., Shields, C., & Menke, M. N. (2014). A retrospective analysis of the utility of an artificial neural network to predict ED volume. *American Journal of Emergency Medicine, 32*, 614–617.

Mesleh, A. M. d. (2011). Feature sub-set selection metrics for Arabic text classification. *Pattern Recognition Letters, 32*, 1922–1929.

Miller, B. L., & Goldberg, D. E. (1995). Genetic algorithms, tournament selection, and the effects of noise. *Complex Systems, 9*, 193–212.

Poultney, C., Chopra, S., & Cun, Y. L. (2006). Efficient learning of sparse representations with an energy-based model. In *Advances in neural information processing systems* (pp. 1137–1144).

Ram, S., Zhang, W., Williams, M., & Pengetnze, Y. (2015). Predicting asthma-related emergency department visits using big data. *IEEE Journal of Biomedical and Health Informatics, 19*, 1216–1223.

Rana, M., Koprinska, I., & Khosravi, A. (2013). Feature selection for neural network-based interval forecasting of electricity demand data. In V. Mladenov, P. KoprinkovaHristova, G. Palm, A. E. P. Villa, B. Appollini, & N. Kasabov (Eds.). In *Artificial neural networks and machine learning - ICANN 2013: Vol. 8131* (pp. 389–396).

Rejer, I. (2015). Genetic algorithm with aggressive mutation for feature selection in BCI feature space. *Pattern Analysis & Applications, 18*, 485–492.

Rocchi, L., Chiari, L., & Cappello, A. (2004). Feature selection of stabilometric parameters based on principal component analysis. *Medical & Biological Engineering & Computing, 42*, 71–79.

Sahu, S. K., Baffour, B., Harper, P. R., Minty, J. H., & Sarran, C. (2014). A hierarchical Bayesian model for improving short-term forecasting of hospital demand by including meteorological information. *Journal of the Royal Statistical Society. Series A. Statistics in Society, 177*, 39–61.

Salcedo-Sanz, S., Munoz-Bulnes, J., Portilla-Figueras, J. A., & Del Ser, J. (2015). One-year-ahead energy demand estimation from macroeconomic variables using computational intelligence algorithms. *Energy Conversion and Management, 99*, 62–71.

Schweigler, L. M., Desmond, J. S., McCarthy, M. L., Bukowski, K. J., Ionides, E. L., & Younger, J. G. (2009). Forecasting models of emergency department crowding. *Academic Emergency Medicine, 16*, 301–308.

Sexton, R. S., Sriram, R. S., & Etheridge, H. (2003). Improving decision effectiveness of artificial neural networks: A modified genetic algorithm approach. *Decision Science, 34*, 421–442.

Seyyedsalehi, S. Z., & Seyyedsalehi, S. A. (2015). A fast and efficient pre-training method based on layer-by-layer maximum discrimination for deep neural networks. *Neurocomputing,, 168*, 669–680.

Sheikhan, M., & Mohammadi, N. (2012). Neural-based electricity load forecasting using hybrid of GA and ACO for feature selection. *Neural Computing & Applications, 21*, 1961–1970.

Shen, F. R., Chao, J., & Zhao, J. X. (2015). Forecasting exchange rate using deep belief networks and conjugate gradient method. *Neurocomputing, 167*, 243–253.

Sikora, R., & Piramuthu, S. (2007). Framework for efficient feature selection in genetic algorithm based data mining. *European Journal of Operational Research, 180*, 723–737.

Sun, Y., Heng, B. H., Seow, Y. T., & Seow, E. (2009). Forecasting daily attendances at an emergency department to aid resource planning. *BMC Emergency Medicine, 9*, 1–9.

Syswerda, G. (1991). A study of reproduction in generational and steady state genetic algorithms. *Foundations of Genetic Algorithms, 2*, 94–101.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., et al. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1–9).

Tai, C.-C., Lee, C.-C., Shih, C.-L., & Chen, S.-C. (2007). Effects of ambient temperature on volume, specialty composition and triage levels of emergency department visits. *Emergency Medicine Journal, 24*, 641–644.

Tan, D. W., Sim, Y. W., & Yeoh, W. (2011). Applying feature selection methods to improve the predictive model of a direct marketing problem. In J. M. Zain, W. M. B. Mohd, & E. ElQawasmeh (Eds.). In *Software engineering and computer Systems, pt 1: 179* (pp. 155–167).

Tang, J. X., Deng, C. W., Huang, G. B., & Zhao, B. J. (2015). Compressed-domain ship detection on spaceborne optical image using deep neural network and extreme learning machine. *IEEE Transactions on Geoscience and Remote Sensing, 53*, 1174–1185.

The Weather Channel Interactive, I. Weather Underground. (2017). https://www.wunderground.com/q/zmw:00000.1.54662. Accessed 20.03.17.

Tonkovic, Z., Zekic-Susac, M., & Somolanji, M. (2009). Predicting natural gas consumption by neural networks. *Tehnički Vjesnik, 16*, 51–61.

Tsai, C.-F., Eberle, W., & Chu, C.-Y. (2013). Genetic algorithms in feature and instance selection. *Knowledge-Based System, 39*, 240–247.

Tsai, C.-F., & Hsiao, Y.-C. (2010). Combining multiple feature selection methods for stock prediction: Union, intersection, and multi-intersection approaches. *Decision Support Systems, 50*, 258–269.

Uguz, H. (2011). A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm. *Knowledge-Based System, 24*, 1024–1032.

Unler, A., & Murat, A. (2010). A discrete particle swarm optimization method for feature selection in binary classification problems. *European Journal of Operational Research, 206*, 528–539.

Urraca, R., Sanz-Garcia, A., Fernandez-Ceniceros, J., Sodupe-Ortega, E., & Martinez-de-Pison, F. J. (2015). Improving hotel room demand forecasting with a hybrid GA-SVR methodology based on skewed data transformation, feature selection and parsimony tuning. In E. Onieva, I. Santos, E. Osaba, H. Quintian, & E. Corchado (Eds.). In *Hybrid artificial intelligent systems: Vol. 9121* (pp. 632–643).

Vieira, S. M., Sousa, J. M. C., & Runkler, T. A. (2010). Two cooperative ant colonies for feature selection using fuzzy models. *Expert Systems with Applications, 37*, 2714–2723.

Voronin, S., & Partanen, J. (2014). Forecasting electricity price and demand using a hybrid approach based on wavelet transform, ARIMA and neural networks. *International Journal of Energy Research, 38*, 626–637.

Wan, J., Liu, J. F., Ren, G. R., Guo, Y. F., Yu, D. R., & Hu, Q. H. (2016). International Journal of Pattern Recognition and Artificial Intelligence, *30*(5), 1650011. doi:10.1142/S0218001416500117.

Wilson, D. R., & Martinez, T. R. (2003). The general inefficiency of batch training for gradient descent learning. *Neural Network, 16*, 1429–1451.

Xu, M., Wong, T. C., & Chin, K. S. (2013). Modeling daily patient arrivals at Emergency Department and quantifying the relative importance of contributing variables using artificial neural network. *Decision Support Systems, 54*, 1488–1498.

Xu, M., Wong, T. C., Chin, K. S., Wong, S. Y., & Tsui, K. L. (2011). Modeling patient visits to accident and emergency department in Hong Kong. In *Industrial engineering and engineering management (IEEM), 2011 IEEE international conference on* (pp. 1730–1734).