Mitigating Effect of Communication Link Failure in Smart Meter-Based Load Forecasting

Chapter · April 2020

DOI: 10.1007/978-981-15-2854-5_26

Sneha Rai

National Institute of Technology Patna

8 PUBLICATIONS **6** CITATIONS

SEE PROFILE

Mala De

National Institute of Technology Patna

48 PUBLICATIONS **288** CITATIONS

SEE PROFILE

Mitigating Effect of Communication Link Failure in Smart Meter-Based Load Forecasting

Vibhas Kumar Vats, Sneha Rai, Suddhasil De and Mala De

Abstract With the ever-increasing number of smart meter installations, an enormous amount of power consumption data is collected by these meters in real time. Availability of this large amount of power consumption data has changed the way power system analyses were done traditionally; one such area being load forecasting. The load forecasting is now largely data-driven and hence failure to receive data from the smart meters lead to forecasting errors. The present paper targets to solve this problem by introducing a novel classification-based load forecasting methodology that enables day-ahead load predication in case of missing data due to communication link failure. In this method, the loads are classified or clustered in sub-classes based on amount of consumption and then day-ahead forecasting is done using this clustered load data. The proposed methodology is demonstrated using the data collected in a practical smart system installed at NIT Patna campus.

Keywords Load · Forecasting · Communication failure · Load classification · Classification/clustering based load forecasting

1 Introduction

In recent years, the deployment of large number of smart meters throughout the distribution system has enabled collection of large set of power consumption data in real-time. Availability of this huge data has opened up new avenues in the area of

V. K. Vats (✉) · S. Rai · M. De
Department of Electrical Engineering, NIT Patna, Patna, India
e-mail: vibhasvats.india@gmail.com

S. Rai
e-mail: sneharai0212@gmail.com

M. De
e-mail: mala.de.in@ieee.org

S. De
Department of Computer Science and Engineering, NIT Patna, Patna, India
e-mail: suddhasil.de@acm.org

V. Nath and J. K. Mandal (eds.), *Nanoelectronics, Circuits and Communication Systems*, Lecture Notes in Electrical Engineering 642,
https://doi.org/10.1007/978-981-15-2854-5_26

power system analysis, operation and planning. More powerful data analysis methods are needed to handle this huge volume of data [1, 2]. With the increased accessibility of this data, customer classification based on multiple aspects has gained its popularity in recent time [3–6]. Customer classification [7] is the process of dividing customers into certain groups decided by energy usage patterns, i.e., daily usage pattern, maximum and minimum load, etc. Customer classification enables utility for better understanding of customer behavior, which will lead to improved demand response (DR) [8], capacity planning and load forecasting, etc. [9–11].

Load classification is implemented for household customers for variety of applications using the daily consumption data available from the smart meters. A load profiling methodology based on self-organization techniques is described in [3] that identifies existing load patterns, classifies customers to classes, and generate typical load profiles. A household classification method based on supervised learning is presented in [4] using smart meter data, which indicates the number of persons living in a household, and their habits. A methodology is presented in [5] to classify any new residential customer to existing classes with limited available data by using model-based feature selection for the classification. It shows that the customer classification can be completed satisfactorily even with one week metering data; with increased data size the accuracy is improves.

The work proposed in [6] targets to classify residential loads to predict energy efficiency relevant characteristics (heating, size/age of house, number of people reside in, etc.) from the consumption traces. This method can provide information required by the utility to develop energy policy targeted to residential customers.

One very important application area of this big data available due to smart metering is load forecasting that leads to load management and generation planning [12, 13]. Load clustering or load classification is also used for load forecasting in [13]. In this work a number of loads are clustered together based on their consumption profile to reduce the prediction criterion with increased scalability.

From the above discussion, we can find out that load classification is largely used in existing works to group residential loads for generation planning or to find out other statistical data from energy consumption pattern, like number of residents in building, type of appliances they use, etc. There are also limited works where load classification is used for load forecasting with a target to improve scalability. The present paper also intends to use load classification for load forecasting but with a very different focus.

The main contribution of this paper is use of load classification in load forecasting in case of missing data due to communication failure. With very large number of smart meter placed throughout different nodes of a network which are geographically not close by, communication link [14] plays a vital role in availability of the real-time data at the utility operator’s premises/computer. Some of the data may be unavailable due to sensor mis-operation or may get lost while being transferred through the communication link. In this scenario load classification is shown to be very useful in proper load forecasting where exact load consumption of a node is not required, rather the information that it belongs to a particular load group is enough. In this method all the individual loads are grouped in some sub-classes based on their consumption pattern and the day-ahead load forecasting for all 24 h will provide the information that which hourly load belong which group of load.

The remainder of this paper is organized as follows: load classification methods are presented in Sect. 2. Section 3 describes the proposed load forecasting method and Sect. 4 presents NIT Patna campus smart metering infrastructure. Validation of effectiveness of the proposed method shown in Sect. 5. Section 6 presents the comparison with other established works and finally Sect. 6 concludes the work.

2 The Load Classification Method

Efficient forecasting of load makes an electrical power management system more dependable and robust. It also makes the generation and distribution of power more economic, especially for high power demand educational institutions, industries or a society as well. There are numerous methods that are utilized for proper forecasting of load these days. One of such big platforms is using various methods available for statistical analysis. One of the imminent requirements for statistical analysis is having a large set of data, which in our case is available from the smart meters installed at different nodes of NIT Patna campus.

A classification method classifies the load in different segments or classes (range of classes can be set as per requirement). Different load classification algorithms have been used in the existing literature. The selection of algorithm for load classification plays an important role in quality of performance of the classification method [15]. A classification method based on decision tree, K-Nearest Neighbor (KNN) and support vector machine (SVM) provides better result with 20% to 29% error. Hence, in the present paper we have used the following techniques for load classification.

- Random forest classification method
- Support vector classification method (SVC) method
- K-Nearest Neighbor (KNN) method

Here, we are presenting these three different classification methods that will be used to predict the segmentation in which the load will fall in.

2.1 K-Nearest Neighbor Classification Method (KNN)

KNN is a non-parametric method, i.e., it does not assume explicitly the function $f(x)$ used for classification of the available data, rather it finds out the K (specified by user) closest training values of prediction point X_0 and marks them as N_0. Then it estimates the average of all training responses in N_0 and put the prediction point X_0 in the class closest to its average value. The flexibility of $f(x)$ depends on the value of K that is chosen by the user. The function $f(x)$ is more flexible with low bias and high variance at smaller values of K as it has only very small value to compute average (only one if $K = 1$) and it leads to overfitting of data points by $f(x)$. But with increase in value of K, it will have more observation to take average on and hence its prediction improves. In any case, making K very large will lead to deviation from the actual model of function.

Value of K can be chosen by analyzing the performance of KNN model by checking the prediction achieved. The optimal value of K is considered to be the one in which changing the value of K does not leads to any sharp change in the prediction efficiency. The efficient working of the model also depends upon the standard scaling of data when all the variables used in the model are not measured in same units. The standardisation of data for this work is done using StandardScaler tool of data preprocessing methods of Sci-kit learn module in Python.

Table 1 shows that the variation in percentage corrects classification with change in value of K. It is evident that at start with smaller K the effectiveness of model is less and by increasing the value of K we can actually increase the effectiveness of the (remove) this model. But as the value of K grows large the percentage correct classification stagnates, not providing much improvement in the model. Any value between 20 and 40 can be chosen for our model. We have chosen 25 as our value of K.

2.2 Random Forest Method

This method uses a tree classification for forming leaves, i.e., the end point observation which are close to each other forming a node and branches i.e. two or more such nodes joined to the closest node with similar character. It predicts that each test observation belongs to the region of most commonly occurring class of training observations. In doing this, random forest classification uses bootstrap method of reducing the variance, gini criterion and gini index to track the classification error rate while fitting the model on training data set [16]. A value of number of estimators is chosen by user which provides the total number of trees in the forest. Random forest method does not need standardization of predictor variables.

Bootstrapping is classic method to reduce variance (hence improving prediction accuracy) by taking many training sets from the population (a large collection of data) to build a separate prediction model for each training set. The final model uses average of all predictions.

While forming the trees, random forest method forms nodes to join two related branches of tree. The effectiveness or purity of this node is given by the gini index; it is a method to calculate the classification error rate while fitting the model. It calculates the total variance across all classes in the forest. A small gini index can indicate that one node contains observation from only one class, predominantly.

Random forest method de-correlates the trees in the forest to further reduce the variance over other tree classification methods, which in return significantly improve the prediction result. For de-correlation of the trees random forest method forces each split to consider only a subset of the predictors having size M out of total P

Table 1 Number of correct classification percent with change in value of K

Correctness (%)	$K = 1$	$K = 5$	$K = 8$	$K = 13$	$K = 17$	$K = 20$	$K = 25$	$K = 30$	$K = 35$	$K = 40$
Correct classification	84	86	84	87	87.86	88.48	88.48	89.2	89.2	89.2
Incorrect classification	16	14	16	13	12.24	11.52	11.52	10.8	10.8	10.8

predictors. Therefore, on average $(P - M)/P$ of the splits will not even consider the strong predictor, and so other predictors will have more of a chance.

2.3 Support Vector Classification Method

SVC is an extension of the maximal margin classifier that can be applied in a broader range of cases. The idea is to divide the P-dimensional space of variables into separate hyperplanes. The division hyperplane, that is the place in p-dimensional space which divides the training observation into different classes, equation can be given as

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p = 0$$

The observation variable X do not explicitly need to satisfy the equation of hyperplane, rather it has to lie in either side of hyperplane. For the purpose of data separation, the value of $f(x)$ is computed for the test observation X and based on the sign of $\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p = 0$. It is separated, while on the basis of magnitude of $f(x)$ it is classified into separate classes. The magnitude of $f(x)$ gives an idea of closeness to class, with small $f(x)$ value for any X it can is said to be closer to the hyperplane and thus around the periphery of the two-separating class, while for larger value of $f(x)$ the corresponding observation X lies away from separating hyperplane and thus close to its class of assignment.

Since it is not always possible to perfectly classify all observation to some definite separate class, support vector classifier provides "soft margin". Soft margin allows some user specified margin to be used on the incorrect side of margin or even on the incorrect side of hyperplane.

The maximization equations for hyperplane have three different parameters that control the soft margin and observations that are allowed in that margin.

C: It is a non-negative tuning parameter controlling the bias-variance trade off by binding the sum of ϵ_i's (slack variables) so as to determine number and severity of violation of margin that will be tolerated by the hyperplane. Smaller value of C means low bias but high variance and vice versa.

M: It defines the width of the margin for hyperplane equation.

Slack variables: It tells us the relative position of an observation with respect to the margin. For an observation which is on the correct side of margin $\epsilon_i = 0$, $\epsilon_i > 0$ for an observation to be on the wrong side of margin and $\epsilon_{\mathrm{i}} > 1$ if an observation is on the wrong side of hyperplane. The observation which is falling either inside the margin or on the border is termed as support vectors.

The classification of observation is done using a function, called kernel function. For this model we have used 'radial basis function' or 'rbf' kernel. The accuracy of the kernel is affected by a positive constant 'gamma'. We have used Gridsearch approach to reach a suitable value of C and 'gamma' with 'rbf' kernel. In Gridsearch method we have tried a number of values of C and 'gamma' to find out the best value of score. Corresponding to best score, C and 'gamma' value is chosen for final fitting

Table 2 Variation of prediction percent with gamma and *C*

Kernel = ‘adial basis function’, degree = 4							
Value of *C*	1000	100	50	25	10	1	0.1
Value of gamma	0.001	0.001	0.01	0.01	0.01	0.01	0.01
Correct prediction%	85.61	86.33	84.17	85.61	86.33	88.48	89.21

of model. We can also control the polynomial nature of function while instantiating the SVC with the term ‘degree’ as an argument in code. For our model it is set to 4.

Table 2 shows the prediction accuracy of model with respect to different values of *C* and ‘gamma’ for ‘rbf’ kernel and degree of polynomial set at 4.

The efficient working of the model also depends upon the standard scaling of data when all the variables used in the model are not measured in same units. The standardisation of data for this work is done using StandardScaler tool of data preprocessing methods of Sci-kit learn module in Python.

3 Load Forecasting Using Load Classification

The loads connected at different nodes of the system are classified in different groups using the smart meter data. These classes decided in this work based on the amount of power consumed at different nodes. Five different load classes are considered here which can easily be increased/decreased into larger/smaller number of groups/classes depending on the requirement. This class of data for 30 days is used to predict the load for the next day using linear regression analysis. The resulting predicted load will lead to hourly loads categorized into those defined classes.

During the load forecasting or prediction method only the number of loads and their time of occurrence for any particular load class is required to be known. Therefore, if few or some of the data is lost due to communication failure then also without knowing the exact consumption value for a load for a particular time period, load class can be predicted for the next day using this data. Hence, the proposed method presents a methodology for load forecasting even in case of loss of partial data due to communication failure. The main steps of the load forecasting method are presented below.

1. Collect 30-day load data for each 15-min interval for the different nodes of the system.
2. Decide the number of classes (here five) to be categorized and their range.
3. Classify the loads into these classes using methods described in Sect. 2.
4. Using these five class data for the 30-day period predict the load for the next day using regression analysis.
5. Compare the predicted load classes to actual load classes to determine the precision of load forecasting method.

The above-mentioned method is tested on a practical academic institute smart grid present at NIT Patna campus.

4 The NIT Patna Campus Smart Grid

Smart meters are connected at the central substation and different strategic locations of NIT Patna campus which collects the data for every 15 s interval. This data consists of active and reactive load connected at any node in the network along with voltages, power factor, load current, etc., for three phases. The NIT Patna campus system is a mixture of commercial, residential and big laboratory loads which provide a wide-variations in load profile during weekdays and weekends and also at different times of the day. Hence, Applicability of the proposed method on NIT Patna campus data shows the method's applicability to any type of distribution system.

5 The Results

For the NIT Patna system described in the previous section, the proposed load forecasting method is tested and validated and the results are presented below. A set of six data points have been taken by averaging all data for every 10 min sample. So for a day we have $6 \times 24 = 144$ data points. All these data have been classified in five segments based on the load pattern.

The whole process of load prediction can be divided into two parts. First, we train the model on large number of data and then we use separate data (previously unseen by the model) to do the prediction. There are many factors that can be considered as variables for training and testing purposes. In our model, we have chosen Power Factor, Volt-LL, Volt-LN and current as the prediction criterion. We have trained our model using the data of over a month period, in total 144×30 data points, so we have also considered the effect of days in our model (since the test system is an educational institute, which has large variation in day to day load pattern). We have also considered the time on a day in this model to train and test. The heatmap graph below shows the correlation among all these variables are shown in Fig. 1.

It can be seen from the figure that we have a positive correlation for load, PF, Volt, days, and load current. We have very strong correlation between load and load current and significantly positive correlation for voltage and hour of day, Load and hours of day, etc.

For KNN model we have selected $K = 25$. The result is represented in Table 3 in the form of 'confusion matrix' which shows the correct as well as misplaced classification. Initially we divided the whole load pattern into five classes. The classification prediction done by model is compared with the actual value and the comparison is represented in confusion matrix shown in Table 3.

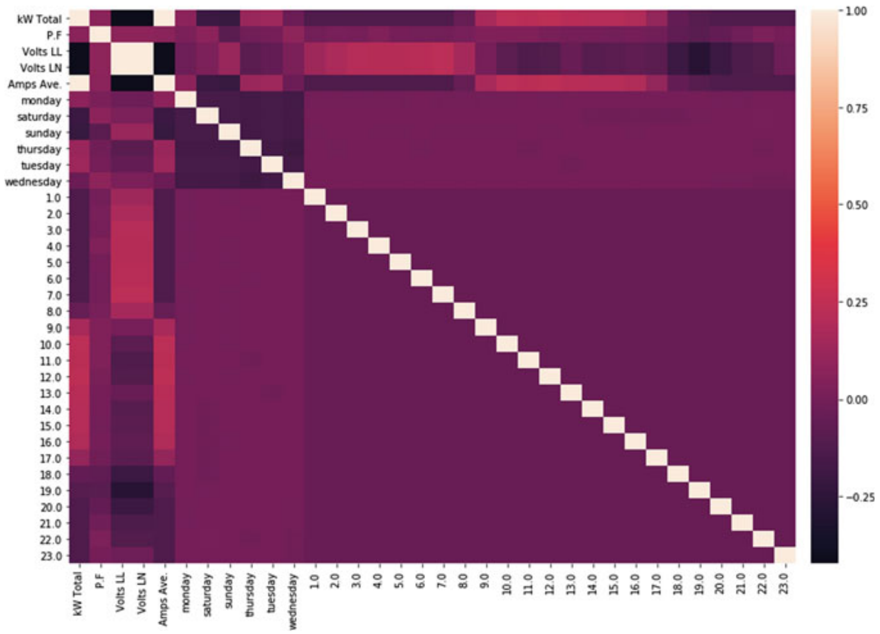

Fig. 1 Correlation between variables under consideration for the model

Table 3 Confusion matrix for KNN method for $K = 25$

Confusion matrix for $K = 25$					
Classes	C-0 (0–5 kW)	C-1 (5–15 kW)	C-2 (15–25 kW)	C-3 (25–35 kW)	C-4 (>35 kW)
C-0	81	3	0	1	0
C-1	2	4	1	3	0
C-2	0	0	0	0	0
C-3	0	2	1	38	3
C-4	0	0	0	0	0

C-0, C-1, C-2, C-3, C-4 shows the five classes in which the load is classified. These five groups represent very low load, low load, normal load, high load, and very high load groups of loads respectively. This means all the loads in the different nodes of NIT Patna campus for 24 h. between 0 and 5 kW will be grouped under C-0. Similarly, the other load groups are formed and their corresponding range of load values is shown in the table.

For random forest method similar result is presented in Table 4. With a single data sample to train from, this method forms 2000 number of estimators while bootstrapping, gini index is used for tracking and minimizing the error rate at each step for tree formation. De-correlation further brings down the variance in the system. Table 2

Table 4 Confusion matrix for Random forest method of classification

Classes	C-0	C-1	C-2	C-3	C-4
C-0	83	0	0	0	0
C-1	4	4	1	0	0
C-2	0	1	0	1	0
C-3	0	3	11	28	0
C-4	0	0	0	3	0

presents a confusion matrix which gives the classification details for random forest model. This method gives 83% precise prediction of classes and the misplaced classes are also grouped in the immediate adjoining class to the actual class of prediction.

The outcome of SVC classification can be constructed as per our requirement. The correction prediction of model depends on the value of C and the value of 'gamma' as shown in Table 5. Two corresponding confusion matrices in presented in Tables 5 and 6 for two distinct value of C and 'gamma'.

It can be seen from the table that for large value of C (refer Table 5), which allows for more number of observation to be in soft margin, the prediction is spread over all columns but for smaller value of C (refer Table 6), which do not allow more number of observations to be in soft margin, the prediction is concentrated. We even have two such classification classes where nothing could be classified due the strict margin as compared to classification represented in Table 5. Therefore, it depends

Table 5 SVC confusion matrix for $C = 1000$ and gamma $= 0.001$

Confusion matrix for $C = 1000$ and gamma $= 0.001$					
Classes	C-0	C-1	C-2	C-3	C-4
C-0	83	0	0	0	0
C-1	4	4	1	0	0
C-2	0	1	0	1	0
C-3	0	2	8	32	0
C-4	0	0	0	3	0

Table 6 SVC confusion matrix for $C = 0.1$ and gamma $= 0.01$

Confusion matrix for $C = 0.1$ and gamma $= 0.01$					
Classes	C-0	C-1	C-2	C-3	C-4
C-0	83	0	0	0	0
C-1	7	0	0	2	0
C-2	1	0	0	1	0
C-3	1	0	0	41	0
C-4	0	0	0	3	0

Table 7 Comparison of the proposed method with other established methods

Classifier method		Accuracy (%)		
		Proposed method	Ref [18]	Ref [17]
KNN	$K = 1$	84	70.18	–
	$K = 5$	86	73.17	–
SVM		89.21	73.34	87.65
Random forest		83	71.74	87.99

on the requirement that which kind of classification we need, to set the value of C and 'gamma' for the SVC model.

Hence, we see from the above discussion that the load forecasting or load prediction using load classification has high value of accuracy, more than 83% for all the cases and even more the 89% for some cases using KNN and SVC method. For the remaining 11–17% loads the mismatch in predication was different by a single class or group, this means, a load is grouped to its immediate higher or lower group than the actual group. Therefore, it can be concluded that the proposed load classification-based load forecasting technique results into good accuracy.

6 Comparison with Other Methods

There are similar works in the area of load forecasting 18] as well as other area like medical applications [18]. The following table shows the comparison of accuracy of the proposed method with these above-mentioned methods (Table 7).

It can be seen from the above table that the proposed method shows better accuracy compared to the other two established methods.

7 Conclusion

This paper presented a load forecasting methodology using load classification. The proposed method uses the large amount of smart meter data to classify the loads first then these load classes are used to predict next day load pattern. The proposed methods work with good accuracy even sin case of missing data due to communication failure which is a very common phenomenon in a smart meter-based power system environment. Three most efficient and commonly used classification algorithms were investigated and results for the same are evaluated for NIT Patna smart grid system. The evaluation results show that the proposed method has high degree of accuracy in predicting day-ahead load profile.

Acknowledgements This work is supported by the Science and Engineering Research Board, Dept of Science and Technology (DST), Government of India, grants number ECR/2017/001027.

References

1. C. Chen, S. Phan, Data analytics: from smart meters to smart decisions. Electr. Eng. Comput. Sci. Newsl., Expon. **7** (2018)
2. P.D. Diamantoulakis, V.M. Kapinas, G.K. Karagiannidis, Big data analytics for dynamic energy management in smart grids. Big Data Res. **2**(3), 94–101 (2015)
3. G. Grigoras, O. Ivanov, M. Gavrilas, Customer classification and load profiling using data from Smart Meters, in *12th Symposium on Neural Network Applications in Electrical Engineering (NEUREL)* (Belgrade, 2014), pp. 73–78
4. P. Carrollpaula, T. Murphy, M. Hanley, D. Dempsey, J. Dunne, Household classification using smart meter data. J. Off. Stat. **34**(1), 1–25 (2018)
5. J.L. Viegas, S.M. Vieira, R. Melício, V.M.F. Mendes, J.M.C. Sousa, Classification of new electricity customers based on surveys and smart metering data. Energy, pp. 804–817 **107** (2016)
6. K. Hopf, M. Sodenkamp, I. Kozlovkiy et al., Feature extraction and filtering for household classification based on smart electricity meter data. Comput. Sci. Res. Dev. **31**(3), 141–148 (2016)
7. J. Jacques, C. Preda, Functional data clustering: a survey. Adv. Data Anal. Classif. **8**, 231–255 (2014)
8. Y. Wang, Q. Chen, C. Kang, M. Zhang, K. Wang, Y. Zhao, Load profiling and its application to demand response: a review. Tsinghua Sci. Technol. **20**, 117–129 (2015)
9. G. Chicco, Customer behaviour and data analytics, in *2016 International Conference and Exposition on Electrical and Power Engineering (EPE)* (Iasi, 2016), pp. 771–779
10. M. Chaouch, Clustering-based improvement of nonparametric functional time series forecasting: application to intra-day household-level load curves. IEEE Trans. Smart Grid **5**, 411–419 (2014)
11. F.L. Quilumba, W.J. Lee, H. Huang, D.Y. Wang, R.L. Szabados, Using smart meter data to improve the accuracy of intraday load forecasting considering customer behavior similarities. IEEE Trans. Smart Grid **6**, 911–918 (2015)
12. Y. Wang, Q. Chen, T. Hong, C. Kang, Review of smart meter data analytics: applications, methodologies, and challenges. IEEE Trans. Smart Grid, https://doi.org/10.1109/tsg.2018.2818167
13. B. Auder, J. Cugliari, Y. Goude, J.M. Poggi, Scalable Clustering of Individual Electrical Curves for Profiling and Bottom-Up Forecasting. Energies **11**(7), 1–22 (2018)
14. Y. Yan, Y. Qian, H. Sharif, D. Tipper, A survey on smart grid communication infrastructures: motivations, requirements and challenges. IEEE Commun. Surv. Tutor. **15**, 5–20 (2013)
15. M. Azaza, F. Wallin, Evaluation of classification methodologies and features selection from smart meter data. Energy Proceedia **142**, 2250–2256 (2017)
16. G. James, D. Witten, T. Hastie, R. Tibshirani, An Introduction to Statistical Learning, Springer Texts in Statistics (2013)
17. M.A. Hambali, M.A. Oladunjoye, Electric Power Load Forecast Using Decision Tree Algorithms. Comput., Inf. Syst., Dev. Inform. Allied Res. J. **7**(4), 29–42 (2016)
18. J.P. Kandhasamy, S. Balamurali, Performance analysis of classifier models to predict diabetes mellitus. Procedia Comput. Sci. **47**, 45–51 (2015)