Short-Term Electricity Load Forecasting Using Modified Hidden Markov Model

Poras Khetarpal, Neelu Nagpal, Mahesh Kumar, D. Lakshmi, and Neelam Kassarwani

Abstract This study proposes a modified Hidden Markov Model (HMM) as a method for predicting electricity on an hourly basis in the Delhi region. Typically, load prediction involves utilizing statistical techniques that need significant modifications in the data to adapt to the random nature of the energy demand. Alternatively, data-based methods like artificial neural networks (ANNs) rely heavily on data to deliver reliable findings. An attempt is made to implement HMM taking into account short-term electricity demand as a non-stationary time series. This provides satisfactory prediction results even with limited data. Furthermore, the proposed modified HMM technique outperforms alternative techniques in terms of computational time and complexity.

Keywords Electricity demand · Hidden Markov Model · Short-term electricity forecasting

1 Introduction

Short-term demand for electricity forecasting is vital for power generation and distribution. It is useful for the electricity sector in demand estimation, distribution planning, scheduling, demand side management, etc. Electricity forecasts are categorized as long-term, medium, and short-term forecasts. Long-term electricity prediction is obtained using several months of data. Long-term electricity demand forecasts are

P. Khetarpal · M. Kumar
IT Department, Bharati Vidyapeeth's College of Engineering, New Delhi 110063, India

N. Nagpal (✉) · N. Kassarwani
EEE Department, Maharaja Agrasen Institute of Technology, Delhi 110086, India
e-mail: nagpalneelu1971@ieee.org

D. Lakshmi
School of Computing Science and Engineering, VIT Bhopal University, Bhopal, MP 466114, India
e-mail: lakshmi.d@vitbhopal.ac.in

H. Malik et al. (eds.), *Renewable Power for Sustainable Growth*,
Lecture Notes in Electrical Engineering 1086,
https://doi.org/10.1007/978-981-99-6749-0_4

generally easier to obtain than short-term forecasts since seasonal patterns are clearly visible in the long-term data. Hence, this type of data is easier than a forecast model. On the other hand, short-term electricity demand, especially hourly and half-hourly demand is highly stochastic in nature showing no such clear trends as compared to long-term electricity demand forecasting [1]. Hourly short-term forecast is required for manufacturing and planning-related purposes so that the demand is not under- or over-valued. A variety of methodologies have been developed to estimate short-term demand including intelligent methods [2–4], statistical methods [5], machine learning methods [6, 7], and many more. In the case of statistical technique, detrending and other modifications to the non-stationary data are necessary. In recent years, other methods are used such as artificial neural networks (ANNs). ANN requires years of data and high computational power to produce forecasts [8].

HMM is suitable for application in the non-stationary nature of the time series and stochastic processes. It has found applications in various fields such as stock market prediction, speech recognition, and other varied fields [9–14]. In this study, the HMM algorithm is modified using Bayesian inference for better performance. In addition, the Markov Chain Monte Carlo (MCMC) approach is utilized to determine the parameters of forecasting model rather than the conventional Baum–Welch procedure. The HMM and its modified version are used for short-term electricity prediction using hourly electricity demand of 2 years (2018–2019) for Delhi city. Following parameter estimation using these HMM methods, forecast results are compared to test data in terms of mean absolute error (MAE) and root mean square error (RMSE). In addition, other forecasting models based on auto-regression (AR), auto-regressive moving average (ARIMA), exponential moving average (EMA), and long short-term memory networks (LSTM) techniques are used. The workflow structure of the proposed model is shown in Fig. 1. The main contributions of work are summarized below:

- This study explores the application of a modified HMM technique to perform the STLF forecasting model which is simple and requires limited training data and less computational time.
- This study looks at how well the suggested forecast model performs for STLF in a metropolitan city with variable weather, including excessive heat in the summer and extreme cold in the winter. Further, the load pattern in this city is unstable due to its fast-paced urban development.
- Viterbi algorithm (VA) is used to train the model by determining the parameters using the Baum–Welch algorithm.
- Comparative analysis with various alternative forecasting approaches such as AR, ARIMA, EMA, LSTM, and HMM to validate the suggested forecasting technique's superior performance over others.

The paper is structured as: The introduction and background of the present work are presented in Sect. 1. A description of HMM method is presented in Sect. 2. The proposed electricity forecasting approach and its implementation are discussed in Sect. 3. A simulation study and result analysis are presented in Sect. 4 to validate the performance of the modified HMM forecasting approach. Section 5 presents the conclusion along with the future direction of this work.

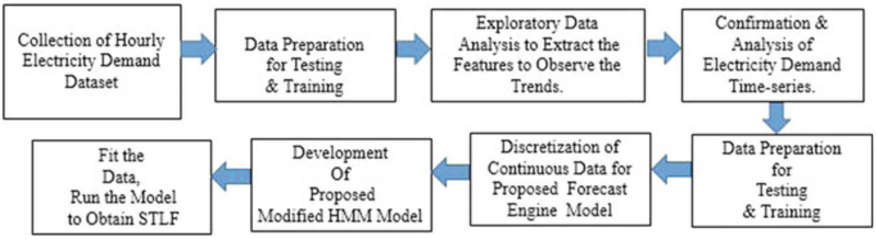


Fig. 1 Workflow structure of proposed forecasting model

2 Hidden Markov Model

HMM is useful for non-stationary data due to its stochastic nature. Being a probabilistic model, it is applicable to a time series which can be assumed as a Markov process. This is a kind of process in which the probability of each event happening in a particular sequence is determined only by its preceding state. The distribution of conditional probabilities for the subsequent state depends entirely on the previous state. It is mathematically represented as:

$$\begin{aligned} &\text{Prob}\,(Y_n = y_n | Y_{n-1} = y_{n-1}, \ldots Y_0 = y_0) \\ &= \text{Prob}\,(Y_n = y_n | Y_{n-1} = y_{n-1}) \end{aligned} \tag{1}$$

In HMM, the input sequence is assumed to be the "hidden" state as the input variable is not actually being directly observed. The observed variables are calculated based on the transition probabilities of the input (or hidden) sequence. The HMM can be represented with its parameters in Fig. 2. The observable Y (forecast output) is obtained by a sequence of hidden states, y (sequenced input). Assume that the transitions between hidden states have the Markov property. They are obtained by the transition probability matrix, A, by the emission probability matrix, B (sometimes θ). Also, π denotes the initial probability. With the following model parameters and observed data, the sequence of hidden states is estimated. The model parameters are determined by the iterative Baum–Welch algorithm [15] for expectation–maximization where the forward–backward technique is used. The Baum–Welch method identifies local maxima as follows:

$$\theta^* = \arg\max_\theta \text{Prob}(Y|\theta) \tag{2}$$

where θ maximizes the probability of observing X. Viterbi algorithm [16] discovers the most probable sequence of hidden states. For an observation i, this algorithm computes the likelihood in terms of probability of observing ith element in state l which is $e_l(i)$ as below

$$p_l(i\ x) = e_l(i) \max_k \left(p_k(j, x-1) \cdot p_{kl}\right) \tag{3}$$

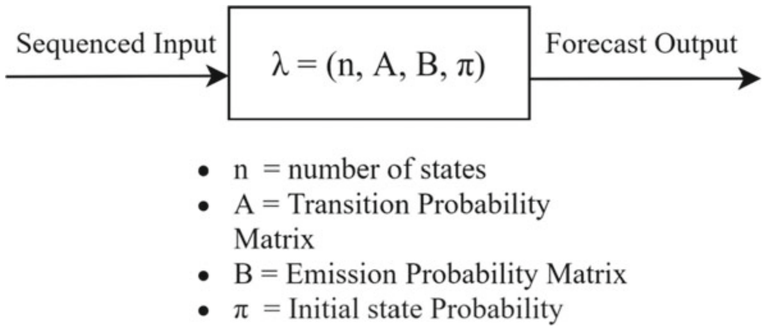


Fig. 2 HMM

This algorithm calculates the likelihood that, given observation i, the path would end in state k. $p_k(j, x-1)$ = Likelihood of the most plausible path endpoint in position $x-1$ and in the state k with the ith element. $p_k l$ = probability of transition from state l to state k.

The Viterbi method is used in the model to extract the sequence of hidden states with a series of observations. HMM parameters are then obtained using an observation likelihood matrix with the Baum–Welch procedure (known as the forward–backward algorithm). It begins with some given initialized values of the HMM parameters, and it follows the two steps for multiple iterations. This method involves two steps as follows:

E-step: The expectation step

$$\begin{aligned} \chi &= \frac{\gamma_l(i)\sigma_l(j)}{\gamma_T q_F}, \quad \forall t, j \\ \xi_t(i, j) &= \frac{\gamma_l(i)\alpha_{ij}\beta_j(o_{t+1}\sigma_{t+1}(j))}{\gamma_T(q_F)}, \quad \forall t, i, \text{ and } j \end{aligned} \tag{4}$$

M-step: The maximization step

$$\begin{aligned} \widehat{\alpha}_{ij} &= \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \sum_{k=1}^{N} \xi_t(i, k)} \\ \widehat{\beta}_{v_k} &= \frac{\sum_{t=1 \text{ s.t. } O_1 = v_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)} \end{aligned} \tag{5}$$

During E-step, "*A*" parameter is used for expected state occupancy counts, and "*B*" parameter is used for expected state transition counts. And further during the M-step, they are utilized to recalculate the probability for *A* and *B*. In the modified HMM, Bayesian HMM with inference performed via MCMC is used [17]. The Baum–Welch algorithm is not used to determine parameters in this case; instead, MCMC is used. HMM is a dynamic model as the probability of a state change is a function of time, and it is subject to change. The number of latent or hidden states in modified HMM varies as part of the fitting process. This is done using hierarchical Dirichlet prior (HDP), and then MCMC sampling is done on the hidden states to estimate the model parameters. MCMC updates the multinomial regression coefficients σ, the state model parameter θ, and the hidden variable Z.

Let $Z_t = (Z_1, \ldots, Z_t)$ be the past observed process and $\zeta_t = (\zeta_1, \ldots, \zeta_t)$ represent the order of states from time = 1 to time = t. $f_z(.)$ denotes the normal probability density function of $Z_t \vee Z_t = s, s \in S$, then the formulation of the joint likelihood function considering the observed data is represented as

$$\begin{aligned} \pi\left(y^T, z^T|\theta, \sigma\right) &= \pi\left(y^T|z^T\theta, \sigma\right)\pi\left(z^T|\theta, \sigma\right) \\ &= f_{z1}(y1)p_{z1}^{(1)}, z2(y2)\ldots p_{zT-1,zT}^{(T-1)}, f_{zT}(yT) \\ P_{ij}^t &= \frac{\exp(x_t\sigma_j)}{\sum_{l=1}^{m}\exp(x_t\sigma_{il})}; \quad \text{for } i, j = 1, \ldots, m. \end{aligned} \tag{6}$$

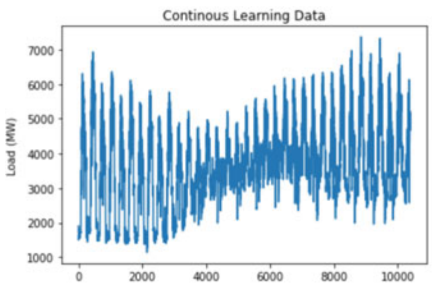
The time-dependent values of the transition matrix are obtained based on the parameter σ from the presented equations.

3 Proposed HMM as Forecasting Model

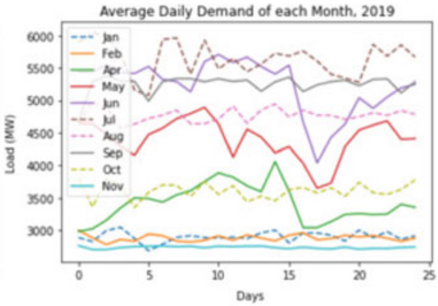
The dataset of hourly electricity demand used was obtained from the state load dispatch center, Delhi. This center is responsible for scheduling and distributing power within Delhi, exercising supervision and control over the intrastate transmission system, and monitoring grid operations, among other responsibilities. The data size was hourly electricity demand for 24 months from January 2018 to December 2019. It was divided into training and learning data (10,000 and 4000 dataset points). After exploratory data analysis, long-term and short-term trends were analyzed within the electricity demand. Since Delhi has fairly consistent and distinct seasons, long-term seasonality is observed right away. Then feature engineering and selection are done based on the observed data's tendencies.

3.1 Exploratory Analysis

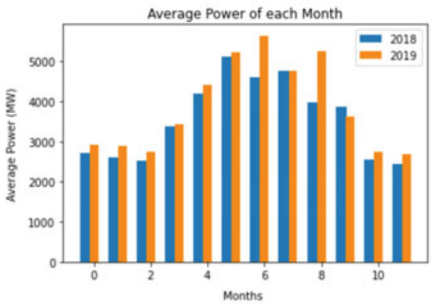
The long-term trends in Delhi's electricity demand are notably visible because of the seasonal weather changes [18]. The monthly change in demand is drastic from the lowest demand being in the winter months (November, December, January) to the highest demand being in the hottest summer months (June, July). The monthly trends are fairly consistent throughout the years. The daily and weekly electricity demands were also found to be consistent within the same month of different years as shown in Fig. 3a–f. The weekend demand was found to be only slightly less than



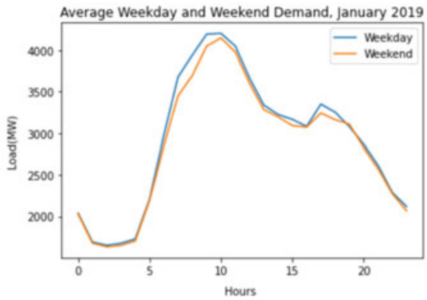
(a) Hourly electricity demand of Delhi.



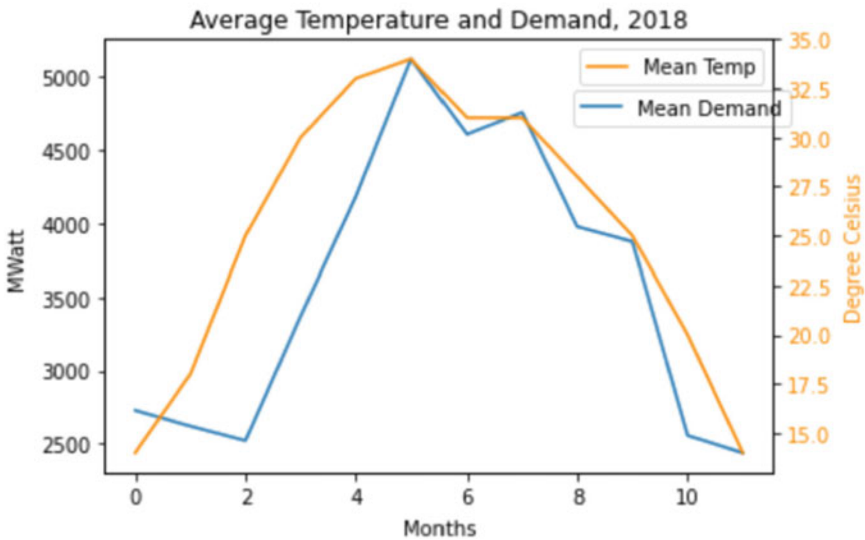
(b) Average daily demand of each month.



(c) Total average power of each month.

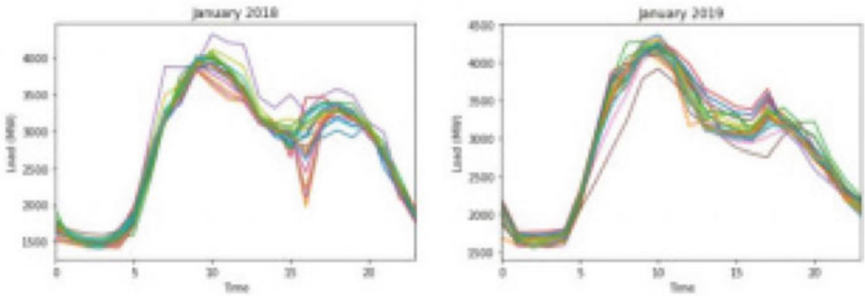


(d) Weekend vs weekday demand.



(e) Average monthly temperature and demand

Fig. 3 Data preparation



(f) Daily demand of January.

Fig. 3 (continued)

the weekday demand and with the same trends. The daily and monthly temperatures were found to have a high correlation with the electricity demand. It is pertinent to mention that seasonal temperature effects as well as hourly temperature effects were prominent.

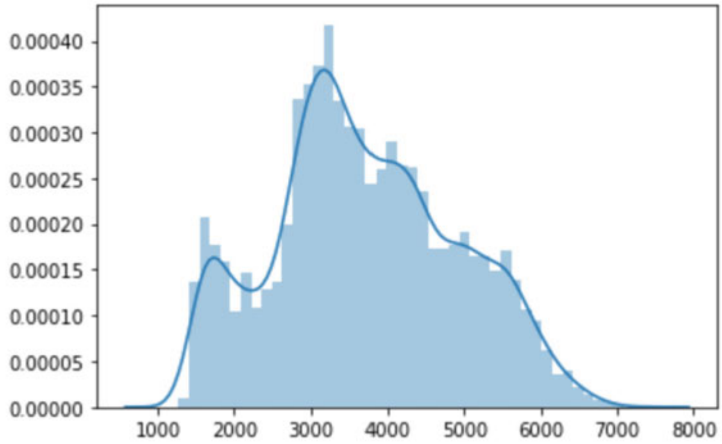
3.2 Data Preparation

First of all, the non-stationary nature of hourly electricity demand time series was confirmed using two methods: (i) augmented Dickey–Fuller (ADF) [19]; (ii) testing and plotting rolling mean and standard deviation.

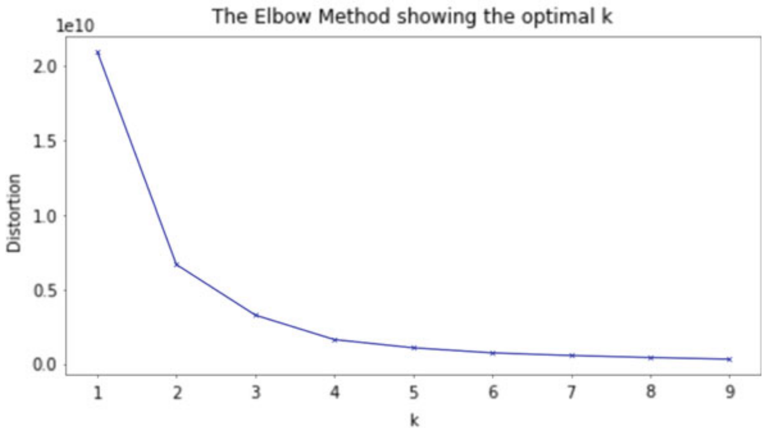
As shown in Fig. 4, the results of the ADF test have proved the non-stationarity of the time series with a *P*-value significantly greater than 0.05. The rolling mean and

```
Augmented Dickey-Fuller Test Results:
ADF Test Statistic        -0.942611
P-Value                    0.773616
# Lags Used                0.000000
# Observations Used       23.000000
Critical Value (1%)       -3.752928
Critical Value (5%)       -2.998500
Critical Value (10%)      -2.638967
dtype: float64
Is the time series stationary? False
```

Fig. 4 ADF test on demand of a single day



(a) Distribution of Energy Demand in MW.



(b) The Elbow Method with optimum number of states.

Fig. 5 Data analysis

standard deviation were also found to be nonlinear. Since the statistical properties changed over time, it is a non-stationary time series. As illustrated in Fig. 5a, the distribution of demand was then analyzed in order to discretize the continuous data, as the HMM requires input in discrete sequences of states. An optimum number of states are computed with the use of K-means clustering [20] and the period for discretization. The elbow curve method was employed to confirm the ideal number of states for the present data as shown in Fig. 5b.

3.3 Modified HMM

The model was trained with the Viterbi algorithm, and the parameters (A, B) were determined by the Baum–Welch algorithm. The test data of 10,000 dataset points was used to train the model. The transition matrix obtained along with the emission probability matrix is shown in Fig. 6.

The number of EM algorithm iterations in the training proceeds until it reaches convergence. The number of iterations converged the EM algorithm, and this is confirmed using the HMM learn library as shown in Fig. 7.

After obtaining all the parameters, the future values are predicted and compared against test data. The modified HMM is then implemented; initially, Markov Chain Monte Carlo (MCMC) sampling is utilized to generate the starting-state probabilities, emission probabilities, and transition probability matrix as shown in Fig. 8.

```
Transition matrix

[[9.93734167e-01 1.29894375e-72 6.26583284e-03 6.04369063e-62]
 [1.96383902e-97 9.91559259e-01 2.39588751e-04 8.20115240e-03]
 [8.87642934e-03 8.25881303e-04 9.87849052e-01 2.44863708e-03]
 [2.30126829e-04 7.59934718e-03 1.38695936e-03 9.90783567e-01]]
```

Fig. 6 Emission probability matrix

```
from hmmlearn.base import ConvergenceMonitor
model.monitor_
ConvergenceMonitor(0.01,100,verbose=False)
model.monitor_.converged
```

```
True
```

Fig. 7 HMM learn library

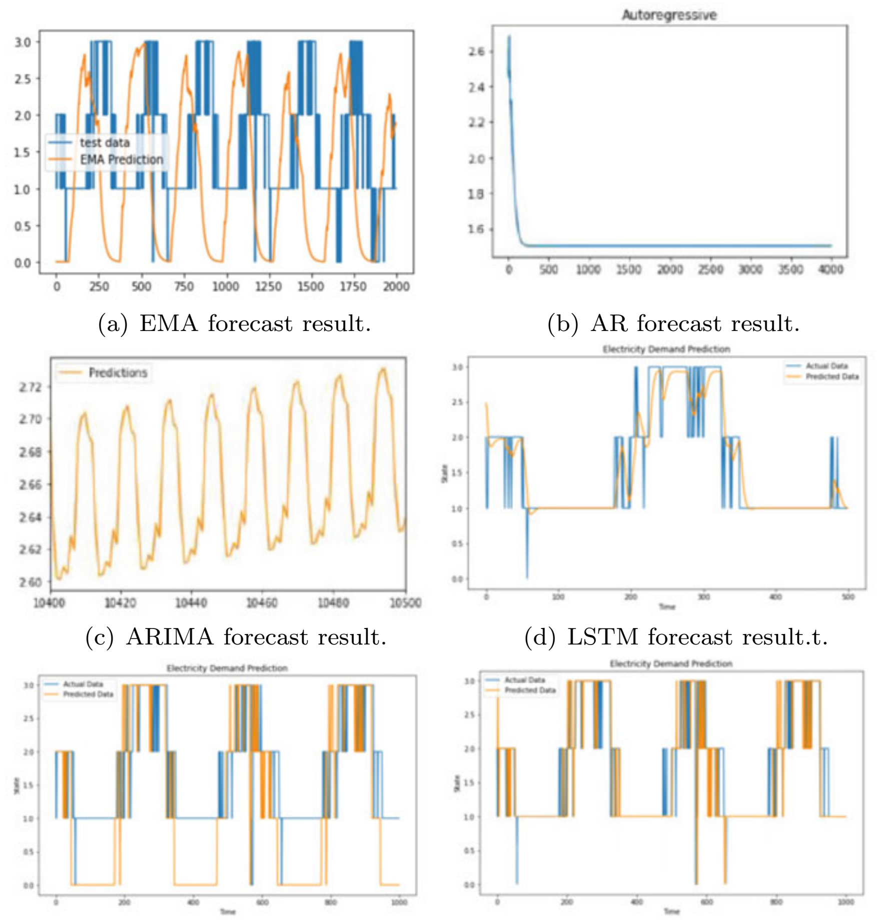
```
Transition Probability

[0.925, 0.001, 0.002, 0.072]
[0.039, 0.015, 0.032, 0.913]
[0.174, 0.042, 0.319, 0.466]
[0.237, 0.383, 0.063, 0.316]
```

Fig. 8 Transition probability matrix

It is seen in this section that the acquired transition and emission probabilities are used to predict future electricity demand values based on training data using the HMM. The obtained model parameters from MCMC are then used to forecast the next values using this HMM, and this is termed a modified HMM approach.

4 Results

The program is developed in Python programming language on Google Colaboratory web IDE. The IDE offers a 2.30 GHz CPU clock, Haswell as the CPU family, and 2 Core CPUs. Pmdarima, Statsmodel, Pandas, and Bokeh packages are used for statistical models, whereas NumPy, Pandas, Bokeh, Sklearn, and XGBoost packages are used for machine learning models. The program is developed using the available data (refer to Sect. 3.1) on a personal computer. Different forecast approaches (such as AR, ARIMA, and EMA, as well as the LSTM and HMM) are developed and simulated, and the corresponding results are shown in Fig. 9a–f. A study is conducted to compare the forecasting performance of the recommended modified HMM with other traditional models, including LSTM and HMM. As mentioned above, the modified HMM with MCMC obtained transition and emission probabilities that are used to obtain the forecast as shown in Fig. 9f. From the results, it is indicated that statistical methods AR and ARIMA fail to perform well (as expected) on non-stationary time series. Each forecast outcome is compared to the test data, and the error metrics obtained for these models, namely MAE and RMSE, are displayed (refer Table 1). LSTM and the modified HMM have the lowest error metrics out of the five which can be inferred from Fig. 10 and Table 1. This comparative analysis validates the superior modified HMM forecast, which has the lowest MAE = 0.616 and RMSE = 0.165, respectively. In contrast, the LSTM [21–24] is known to produce the most accurate forecasts and performs well, but it required significantly more computation time and processing power than the HMM and modified HMM. LSTM model produces the best accuracy of 97.765%, and then modified HMM performed with the accuracy of 97.446. HMM has comparable error metrics to LSTM with a higher MAE and the lowest RMSE. On the other hand, LSTM, being a deep learning-based method, requires a huge dataset, and its computational time is large as compared to HMM and modified HMM which requires less data and puts less computational burden on the processor. Also, the modified HMM method is very near to accuracy as compared to LSTM. Further, the temporal complexity of the two algorithms, LSTM and MCMC for parameter estimation in modified HMM, are compared using empirical values of the runtime of the algorithms. Considering the size of the present training data, the forecast model upon which the program is run, 100 iterations of LSTM take 30–40 min to train, while 100 MCMC steps take 8 s per iteration making 13 min in total. It is known that for even larger-sized data, LSTM and other RNN models



(a) EMA forecast result.

(b) AR forecast result.

(c) ARIMA forecast result.

(d) LSTM forecast result.t.

(e) HMM forecast result

(f) Modified HMM forecast.

Fig. 9 Forecast results using different approaches

take a lot of training time. This result analysis concludes the work in the following section.

5 Conclusion

This research proposes a modified HMM for short-term electricity demand forecasting. HMM is modified using Bayesian inference, using MCMC sampling to obtain the model parameters, as short-term electricity demand is a non-stationary time series with stochastic nature. From the result analysis, the overall performance of HMM

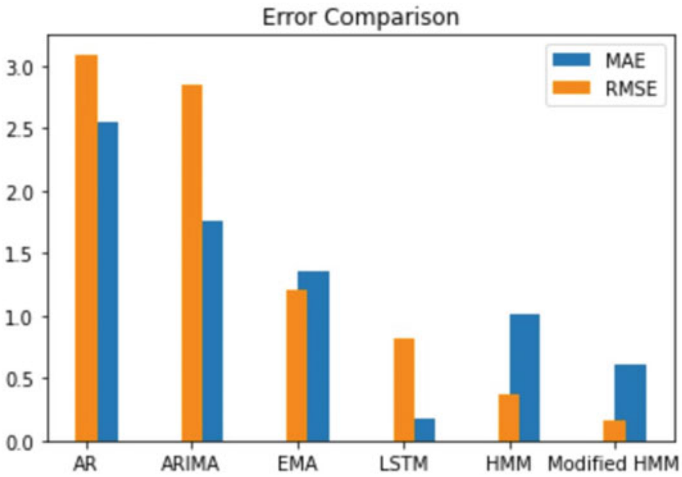


Fig. 10 Comparative analysis using MAE and RMSE

Table 1 Comparative performance analysis of forecasting models

Performance indices ↓	AR	ARIMA	EMA	LSTM	HMM	Proposed HMM
MAE	2.557	1.76	1.357	0.177	1.016	0.616
RMSE	3.094	2.849	1.2	0.815	0.371	0.165
% Accuracy	86.675	91.138	93.108	97.765	96.411	97.446

and its modification proves better in predicting the future electricity forecast values than all other statistical methods if accuracy, computation time, and dataset size are considered together. Also, modified HMM and LSTM have better error metrics, but LSTM is way more time-consuming, computationally complex, and takes a lot of processor power and time as compared to the HMM, whereas the MCMC sampling takes considerable computational time as compared to Baum–Welch HMM. Subsequently, it is concluded that the proposed modified HMM is capable of performing efficient short-term electricity load forecasting with limited training data, lower complexity, and overall ease of use. With the addition of weather conditions, the performance of the modified HMM approach can be improved even further. As a future direction of work, other inference methods can also be investigated for improved results. Furthermore, HMM can be incorporated into AI-based models to improve forecast accuracy into AI-based models to improve forecast accuracy when used for real-time applications.

References

1. Alhelou HH, Parthasarathy H, Nagpal N, Agarwal V, Nagpal H, Siano P (2022) Decentralized stochastic disturbance observer-based optimal frequency control method for interconnected power systems with high renewable shares. IEEE Trans Ind Inform 18(5):3180–3192. https://doi.org/10.1109/TII.2021.3107396
2. Nilakanta Singh K, Robindro Singh K (2021) A review on deep learning models for short-term load forecasting. In: Choudhary A, Agrawal AP, Logeswaran R, Unhelkar B (eds) Applications of artificial intelligence and machine learning. Lecture notes in electrical engineering, vol 778. Springer, Singapore. https://doi.org/10.1007/978-981-16-3067-5-53
3. Sadaei HJ, de-Lima e-Silva PC, Guimarães FG, Hisyam LM (2019) Short-term load forecasting by using a combined method of convolution neural networks and fuzzy time series. Energy 175:365–377. ISSN 0360-5442
4. Saravanan S, Kannan S, Thangaraj C (2012) Forecasting India's electricity demand using artificial neural network. In: IEEE-international conference on advances in engineering, science and management (ICAESM-2012), Tamil Nadu, pp 79–83
5. Taylor JW, McSharry PE (2007) Short-term load forecasting methods: an evaluation based on European data. IEEE Trans Power Syst 22(4):2213–2219
6. Son M, Moon J, Jung S, Hwang E (2019) A short-term load forecasting scheme based on auto-encoder and random forest. In: Ntalianis K, Vachtsevanos G, Borne P, Croitoru A (eds) Applied physics, system science and computers III. APSAC 2018. Lecture notes in electrical engineering, vol 574. Springer, Cham. https://doi.org/10.1007/978-3-030-21507-1-21
7. Subbiah SS, Chinnappan J (2022) Short-term load forecasting using random forest with entropy-based feature selection. In: Raje RR, Hussain F, Kannan RJ (eds) Artificial intelligence and technologies. Lecture notes in electrical engineering, vol 806. Springer, Singapore. https://doi.org/10.1007/978-981-16-6448-9-8
8. Singhal R, Choudhary NK, Singh N (2020) Short-term load forecasting using hybrid ARIMA and artificial neural network model. In: Dutta D, Kar H, Kumar C, Bhadauria V (eds) Advances in VLSI, communication, and signal processing. Lecture notes in electrical engineering, vol 587. Springer, Singapore. https://doi.org/10.1007/978-981-32-9775-3-83
9. Gupta A, Dhingra B (2012) Stock market prediction using hidden Markov models. In: Students conference on engineering and systems, Allahabad, Uttar Pradesh, pp 1–4
10. Varga AP, Moore RK (1990) Hidden Markov model decomposition of speech and noise. In: International conference on acoustics, speech, and signal processing, Albuquerque, NM, vol 2, pp 845–848
11. Starner T, Pentland A (1995) Real-time American sign language visual recognition from video using hidden Markov models. Master's thesis, MIT, Feb 1995. Program in Media Arts
12. Jia L, Najmi A, Gray RM (2000) Image classification by a two-dimensional hidden Markov model. IEEE Trans Signal Process 48(2):517–533. https://doi.org/10.1109/78.823977
13. Chen MY, Kundu A, Zhou J (1994) Off-line handwritten word recognition using a hidden Markov model type stochastic network. IEEE Trans Pattern Anal Mach Intell 16(5):481–496. https://doi.org/10.1109/34.291449
14. Srivastava A, Kundu A, Sural S, Majumdar A (2008) Credit card fraud detection using hidden Markov model. IEEE Trans Depend Secure Comput 5(1):37–48. https://doi.org/10.1109/TDSC.2007.70228
15. Turin W (1998) Unidirectional and parallel Baum–Welch algorithms. IEEE Trans Speech Audio Process 6(6):516–523. https://doi.org/10.1109/89.725318
16. David G, Forney J (2005) The Viterbi algorithm: a personal history. https://arxiv.org/pdf/cs/0504020
17. Djuric PM, Chun J (2002) An MCMC sampling approach to estimation of non-stationary hidden Markov models. IEEE Trans Signal Process 50(5):1113–1123. https://doi.org/10.1109/78.995067

18. Hor CL, Watson SJ, Majithia S (2005) Analyzing the impact of weather variables on monthly electricity demand. IEEE Trans Power Syst 20(4):2078–2085. https://doi.org/10.1109/TPWRS.2005.857397
19. Fuller WA (1976) Introduction to statistical time series. Wiley, New York. 0-471-28715-6
20. Kanungo T, Mount DMN, Netanyahu S, Piatko CD, Silverman R, Wu AY (2002) An efficient k-means clustering algorithm: analysis and implementation. IEEE Trans Pattern Anal Mach Intell 24(7):881–892. https://doi.org/10.1109/TPAMI.2002.1017616
21. Sepp H, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735. PMID 9377276
22. Kong W, Dong ZY, Hill DJ, Luo F, Xu Y (2018) Short-term residential load forecasting based on resident behaviour learning. IEEE Trans Power Syst 33(1):1087–1088. https://doi.org/10.1109/TPWRS.2017.2688178
23. Tomar A et al (eds) (2022) Proceedings of 3rd international conference on machine learning, advances in computing, renewable energy and communication: MARC 2021, vol 915. Springer Nature, pp XV, 781. ISBN: 978-981-19-2830-7. https://doi.org/10.1007/978-981-19-2828-4
24. Ahmad MW et al (eds) (2022) Intelligent data analytics for power and energy systems. Springer Singapore, pp XXII, 641. ISBN: 978-981-16-6081-8. https://doi.org/10.1007/978-981-16-6081-8