Deep Learning Based Forecasting of COVID-19 Hospitalisation in England: A Comparative Analysis

Ajao-olarinoye, M., Palade, V., Mousavi, S., He, F. & Wark, P. A.

Original citation & hyperlink:

Ajao-olarinoye, M, Palade, V, Mousavi, S, He, F & Wark, PA 2024, Deep Learning Based Forecasting of COVID-19 Hospitalisation in England: A Comparative Analysis. in M Arif Wani, M Boicu, M Sayed-Mouchaweh, PH Abreu & J Gama (eds), 2023 International Conference on Machine Learning and Applications (ICMLA). Proceedings - 22nd IEEE International Conference on Machine Learning and Applications, ICMLA 2023, Institute of Electrical and Electronics Engineers Inc., pp. 1344-1349, 22nd IEEE International Conference on Machine Learning and Applications, ICMLA 2023, Jacksonville, Florida, United States, 15/12/23.
https://doi.org/10.1109/ICMLA58977.2023.00203

DOI 10.1109/ICMLA58977.2023.00203
ISSN 1946-0740
ESSN 1946-0759

Publisher: IEEE

Deep Learning based Forecasting of COVID-19 Hospitalisation in England: A Comparative Analysis

Michael Ajao-olarinoye*, Vasile Palade*, Seyed Mosavi*, Fei He*, and Petra Wark†,
*Centre for Computational Science and Mathematical Modelling, Coventry University, Coventry, United Kingdom
†Research Institute for Health and Wellbeing, Coventry University, Coventry, United Kingdom

Abstract—**In the midst of the COVID-19 pandemic, it was essential to accurately forecast the demand for hospitalisation resources to achieve an effective allocation of healthcare resources. This paper explores the potential of various deep learning (DL) models, namely basic Recurrent neural networks (RNNs), Long short-term memory networks (LSTMs), Gated recurrent units (GRU), Bidirectional RNNs, and Sequence-to-Sequence architectures with the inclusion of attention mechanisms, to forecast the demand for hospitalisation resources (mechanical ventilators) in England during the COVID-19 pandemic. The implementation of simulated annealing (SA) as a hyperparameter tuning method produced certain model structures and good results in terms of prediction accuracy. Our findings show that the LSTM-based models (LSTM_SA), achieved the lowest mean average error (MAE), outperforming other architectures used in this study. The results of this study show the potential of DL models to forecast the demand for resources and could help inform the distribution of hospitalisation resources in England during the COVID-19 pandemic.**

Index Terms—**Deep learning, COVID-19, Hospitalisation forecasting, RNN, LSTM, GRU, Attention mechanism**

I. Introduction

The COVID-19 pandemic has underlined the need for an optimal allocation of hospitalisation resources to control disease spread and prevent mortality. The challenges faced by healthcare systems prompt the need to develop effective strategies to forecast hospitalisation demand. Hospitalisation services, such as beds in intensive care units (ICU) and mechanical ventilators, can be forecasted for a given time horizon within a certain location or health system.

In 2020, the United Kingdom saw a sharp increase in the need for hospitalisation resources for those suffering from severe acute respiratory diseases due to the COVID-19 pandemic[1]. Initially, national health services (NHS) had only 7,400 mechanical ventilators, but the target was set to acquire 30,000 by the end of June 2020 to cope with the pandemic surge[2].

Mathematical models such as SEIR and its variants, machine learning (ML) including regression models, and deep learning, have been used to forecast the number of cases, recoveries, and deaths [1]–[5]. Deep learning techniques, including RNN, LSTM, GRU [6], graph neural networks (GNN), and others, have been used to accurately forecast the demand for beds and mechanical ventilators in the ICU during a pandemic [7]. Goic et al. [8] proposed a composite approach by combining autoregressive, ML and epidemiological models to forecast the short-term use of ICU beds at regional levels, demonstrating the superiority of combined models over individual ones. Tello et al. [9] used a ML methodology with a support vector regression model to predict the weekly demand for hospital beds, helping to strategically allocate resources to emergency departments.

A recent study by Borges and Nascimento [10] highlighted the difficulties posed by the COVID-19 pandemic, their research explored the combination of Prophet and LSTM to forecast the demand for ICU beds in Brazil. Recently, in [11], the authors presented Variational AutoEncoder (VAE) as the best performing model to forecast COVID-19 cases in different counties. Since we are not looking at COVID-19 cases, but data on hospitalisation resources usage in England, we cannot compare our results with the work presented in [11]. Although the models used will be useful in forecasting the demand for hospitalisation resources in England. In [12], the EpiBeds model is introduced to forecast the impact of COVID-19 on hospital capacity in England, EpiBeds highlighting the value of integrated data in accurately predicting healthcare demands.

In this paper, we explore the potential of various deep learning models, namely LSTM, Vanilla RNNs, GRUs, Bidirectional RNNs, and Sequence-to-Sequence (Seq2Seq) RNNs with an attention mechanism, to predict the demand for hospitalisation resources in England.

In summary, the contributions of this paper are listed below:

- This study leverages similar data feature utilised in the EpiBeds model [12] for daily forecasting in England using deep learning based approaches.
- The study introduces various deep learning models for single-step and multi-step forecasting of hospitalisation demands.
- Exploring simulated annealing as a technique for hyperparameter tuning across different RNN architectures employed in this research.

The rest of the paper is organised as follows. Section II explains the deep learning techniques used in the paper. Section III describes the data preprocessing, forecasting metrics, and experiments conducted. Section IV reports the experimen-

Emails: olarinoyem@coventry.ac.uk, ab5839@coventry.ac.uk, ad0204@coventry.ac.uk, ad0067@coventry.ac.uk, ac4710@coventry.ac.uk,

[1]https://www.kingsfund.org.uk/publications/critical-care-services-nhs
[2]https://www.nao.org.uk/reports/increasing-ventilator-capacity-in-response-to-covid-19/

tal results. Finally, Section V summarises the findings and concludes the paper.

II. Deep Learning Forecasting Architectures

In this study, we developed a variety of deep learning models to forecast the demand for hospitalisation resources (ICU beds, mechanical ventilators) in England, represented as time series data. In particular, models such as RNN, LSTM, and GRU have been shown to be effective on time series data [7]. Given the nature of our data, we treated the problem as a univariate time series forecasting task by selecting the daily use of ventilators in England from the COVID-19 hospital activity data. Formally, the relationship for a given time step t based on a history of N steps that are daily data can be expressed as

$$y_t = h(y_{t-1}, y_{t-2}, \ldots, y_{t-N}) \tag{1}$$

where y_t denotes the data point at time t and h represents the function mapping the historical data points to the future. The primary objective of the deep learning models to be developed is to accurately learn and approximate the function in Equation 1.

A. Recurrent neural networks

In 1986, Rumelhart et al. [13] introduced the backpropagation algorithm, a technique used for training artificial neural networks (ANN) such as feedforward neural networks (FFNs) and RNNs. RNNs unlike FFNs, are particularly adept at handling sequential data, as they are capable of recognising patterns in time series data. Traditional RNNs are known to have the problem of vanishing and exploding gradients [14] which hampers their ability to learn long-term dependencies. To address this, Hochreiter and Schmidhuber [15] developed LSTM networks in 1997, which have memory cells and gates to control the flow of information. This design ensures that gradients are propagated stably, making LSTMs especially effective for learning long-term dependencies in data.

For an input x_t at time t and the previous hidden state H_{t-1}, LSTMs operate through three key gates:

- **Input Gate (I_t):**

$$I_t = \sigma(W_{xi}x_t + W_{hi}H_{t-1} + b_i) \tag{2}$$

- **Forget Gate (F_t):**

$$F_t = \sigma(W_{xf}x_t + W_{hf}H_{t-1} + b_f) \tag{3}$$

- **Output Gate (O_t):**

$$O_t = \sigma(W_{xo}x_t + W_{ho}H_{t-1} + b_o) \tag{4}$$

These gates function together to update the memory cell as:

$$C_t = F_t \odot C_{t-1} + I_t \odot \tanh(W_{xc}x_t + W_{hc}H_{t-1} + b_c) \tag{5}$$

and to compute the hidden state as:

$$H_t = O_t \odot \tanh(C_t) \tag{6}$$

The LSTM architecture contains weight matrices W, biases b, sigmoid function σ, and element-wise multiplication $\odot$.

The GRU proposed by Cho et al. in 2014 [16], which is based on the LSTM, was created as a simplified version that is still capable of managing long-term temporal dependencies. As time passed, these structures have became more complex, including stacked and bi-directional versions. These improvements allow the model to use data from both past and future states. Therefore, RNN, LSTM, and GRU and their variants are used here to forecast the demand for hospitalisation resources.

B. Sequence-to-Sequence Models

Seq2Seq models, which are characterised by their encoder-decoder architecture, have been highly successful in a variety of tasks, such as machine translation and time series forecasting. This approach was first popularised in the works of Sutskever et al. [17] and Cho et al. [16] in (2014). The core concept of the Seq2Seq model is to encode sequential data in a latent representation space that is then decoded to generate the desired output sequence. This latent representation captures the patterns and structures in the data, similar to how humans can recognise patterns and features in complex entities.

A variety of techniques can be used to construct a Seq2Seq model. Popular approaches include RNNs, which can be used alone or in combination with another RNN. As an example, an RNN-to-RNN model can be used, where both the encoder and decoder are RNN-based architectures (Figure 1). During training, the decoder is provided with the true output from the previous time step, this a technique known as "teacher forcing". This involves feeding the actual output from the training data into the current step of the decoder instead of the decoder's own predictions from the prior step. This method can speed up convergence, stabilise training, and often lead to more accurate models, particularly for longer sequences. For inference, the model's own predictions are fed back into the decoder to generate future steps in the sequence.

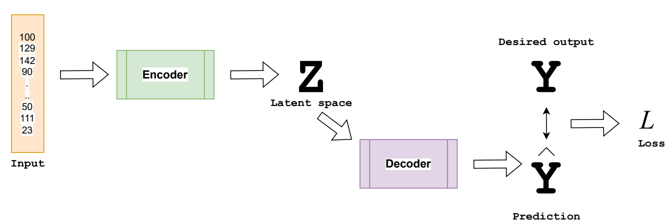

Fig. 1. The encoder-decoder architecture

C. Enhanced Forecasting Through Attention Mechanisms

Accurate forecasting of hospitalisation demands poses a complex challenge, primarily due to the need to capture intricate long-term dependencies and temporal correlations across multiple time steps. While traditional Seq2Seq models provide a foundational framework for modelling such dependencies, their efficacy diminishes when faced with extended sequences or intricate temporal patterns.

To overcome these limitations, we employ the attention mechanism, initially conceived to address challenges in machine translation [18] and later refined by Vaswani et al. [19]. Unlike traditional Seq2Seq models, which are limited by the encoder's final hidden state, the attention mechanism allows our model to dynamically focus on different portions of the input sequence at each step of output generation.

$$\mathcal{A}(q, K, V) = \sum p(a(k_i, q)) \times v_i \tag{7}$$

In this generalised attention framework, q represents the query generated by the decoder, K are the keys derived from the encoder, and V are the values that are weighted by the attention distribution. Here, $p(a(k_i, q))$ denotes the attention weight computed for each key k_i using the query q, and v_i is the corresponding value. This formulation allows our model to adapt the attention distribution, providing a more robust and flexible framework that is particularly suited for complex forecasting tasks such as hospitalisation demand prediction.

III. Experimental Setup and Data analysis

This study examined the use of both single-step and multi-step forecasting to anticipate the demand for hospitalisation resources in England. Single-step forecasting was used, which trains the model to predict a future time point based on past and current data. This approach is especially useful for short-term predictions. However, its capacity to capture long-term patterns is uncertain. To address this issue, multi-step forecasting was later used, with a window size of fourteen days, twice the forecast horizon. Sequence-to-sequence (seq2seq) deep learning models were used to predict a horizon of seven days. The aim was to forecast the subsequent H timesteps, $y_{t+1}, \ldots, y_{t+H}$, where $H > 1$. Despite its practical relevance, multistep forecasting has been relatively unexplored due to its complexity; as we look further into the future, predictions become increasingly uncertain due to the intricate interactions between the forecasted steps. The framework strategies for the experiments used in this paper are illustrated in Figure 3.

A. Data Preprocessing

The daily fluctuations in the time series data and the ever-changing context of the pandemic make forecasting difficult because of the stationarity property. To identify the underlying trends and seasonality, we performed an exploratory data analysis on the variable MVbeds (daily usage of mechanical ventilator beds) from the COVID-19 hospital activity data set [20]. In Figure 2 the rolling mean of 7 and 30 days of the data is shown, indicating the present trends. The Augmented Dickey-Fuller (ADF) test was used to determine whether the data were stationary; however, the null hypothesis with $p > 0.05$ was not met. To address this problem, we applied an AutoStationary Transformer approach to the data. This approach was proposed in a time series book by Joseph Manu [21].

The AutoStationary Transformer executes a sequence of operations to guarantee data stationarity. To begin with, it performs a statistical test to detect trends and if a trend is identified, it applies a detrending transformation. After that, it inspects the data for seasonality and, if necessary, applies a deseasonalisation transformation. Finally, it evaluates the data for heteroskedasticity (variance that changes over time), and if this condition is observed, it uses a Box-Cox transformation to stabilise the variance. The data set used to train the deep learning models was collected from April 8, 2020 to May 31, 2023. It was divided into three parts: training, validation, and testing. 861 samples, which is 74.93% of the data set, were allocated for training up to August 16, 2022. The following segment, 115 samples (10.01%) from 17 August 2022, was allocated for validation. Lastly, 173 samples (15.06%) from 10 December 2022 were kept for testing to guarantee an unbiased assessment of the model's performance.

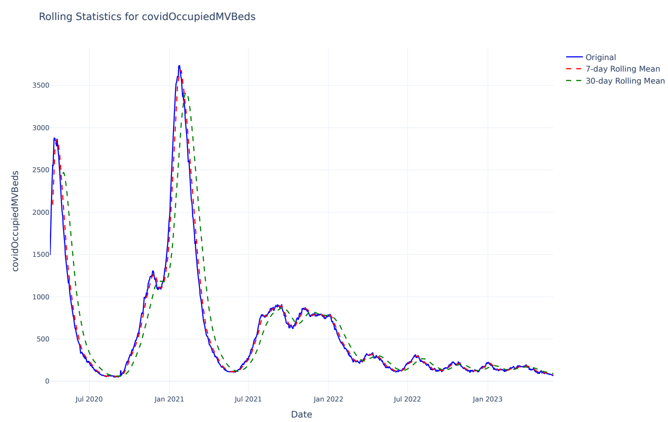

Fig. 2. COVID-19 daily cases of patients occupying mechanical ventilator beds in England

B. Hyperparameter Optimisation using Simulated Annealing

Hyperparameter tuning is crucial for optimising the performance of ML models. Given the vast hyperparameter space, efficient optimisation techniques are imperative. In this work, we employ Simulated Annealing (SA) [22], a probabilistic optimisation method inspired by the annealing process in material science, to fine-tune the hyperparameters of the models [23]. In this instance, multiple objectives are passed to the algorithm (Algorithm 1). Starting with an initial set of hyper-parameters, the SA algorithm was run for a specified number of iterations to find a (near) optimal set of hyperparameters in the search space presented in Table I.

TABLE I
Defined search space for hyperparameters.

Hyperparameter	Search Space
rnn_type	GRU, LSTM, RNN
attention_mechanism	general, concat, additive, dot
hidden_size	32, 256
num_layers	1, 10
bidirectional	True, False
learning_rate	1×10^{-2}, 1×10^{-6}

Algorithm 1 Hyperparameter Optimisation with SA

Input: $f(\mathbf{x})$ - validation loss, $N(\mathbf{x})$ - neighbour function, $T(t)$ - cooling schedule, $\mathbf{x_0}$ - initial hyperparameters, T_0 - initial temperature, k - iterations.
Output: $\mathbf{x}_{\text{best}}$ - optimal hyperparameters.
procedure SIMULATEDANNEALING($f, N, T_0, \mathbf{x_0}, k$)
  Initialize $\mathbf{x}, f_{\text{best}}, \mathbf{x}_{\text{best}}, T \leftarrow \mathbf{x_0}, f(\mathbf{x_0}), \mathbf{x_0}, T_0$
  for $t = 1$ to k **do**
    $\mathbf{x}' \sim N(\mathbf{x})$, $\Delta f = f(\mathbf{x}') - f(\mathbf{x})$
    if $\Delta f < 0$ or rand$(0,1) < \exp(-\Delta f / T)$ **then**
      $\mathbf{x} \leftarrow \mathbf{x}'$
      if $f(\mathbf{x}') < f_{\text{best}}$ **then**
        $f_{\text{best}}, \mathbf{x}_{\text{best}} \leftarrow f(\mathbf{x}'), \mathbf{x}'$
      end if
    end if
    $T \leftarrow$ cooling(T)
  end for
  return $\mathbf{x}_{\text{best}}$
end procedure

C. Experimental Configuration

The experiments were carried out on a laptop using Windows Subsystem for Linux, equipped with an AMD Ryzen 7 5000 series processor, 16GB of DDR4 RAM, and an NVIDIA RTX 3050 GPU. Pytorch and Lightning were used for training and evaluation, referencing the metrics defined in IV-A. Additional Python libraries facilitated data preprocessing, exploratory analysis, and plotting.

The ease of hyperparameter modification was provided by Pytorch Lightning. For reproducibility, the initial training conditions were standardised as follows: hidden size fixed at 64, two layers per model, batch size of 64, and bidirectionality determined within the model structure. All Seq2Seq models were bidirectional, with a learning rate of 1×10^{-2} and a teacher forcing ratio of 1 for models that employ attention mechanisms. Training spanned a minimum of five and a maximum of 100 epochs, with early stopping based on a patience of three in monitoring validation loss. The initial attention mechanism for Seq2Seq models was set to general. After preprocessing and during training, the transformed data were standardised using the formula:

$$\text{standardized series} = \frac{\text{series} - \text{mean}}{\text{std}} \tag{8}$$

where the mean and standard deviation were computed from the training data. Data loading and normalisation processes were facilitated by the PyTorch data loader function.

IV. RESULTS

A. Evaluation Metrics

The performance of the forecasting models was assessed using the following four key metrics: Mean Absolute Error (MAE), averaging absolute differences between predicted and observed values, offers a direct measure of the accuracy of the forecast (Eq 9): the mean squared error (MSE) computes the mean squared differences between the forecast and actual values, highlighting larger errors (Eq 10); the mean absolute scaled error (MASE) normalises the forecast errors, providing a scale-independent evaluation (Eq 11), and Forecast Bias (FB) identifies consistent over or under forecasting tendencies, capturing systemic bias in predictions (Eq 12). These metrics jointly facilitate a comprehensive model evaluation. Although MAE and MSE highlight the magnitude and severity of errors, MASE ensures scale independence and FB reveals systemic biases.

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N} |f_i - y_i| \tag{9}$$

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N} (f_i - y_i)^2 \tag{10}$$

$$\text{MASE} = \frac{\frac{1}{N}\sum_{i=1}^{N} |f_i - y_i|}{\frac{1}{N-1}\sum_{j=1}^{N} |y_j - y_{j-1}|} \tag{11}$$

$$\text{FB} = \frac{\sum_{i=1}^{N} f_i - \sum_{i=1}^{N} y_i}{\sum_{i=1}^{N} y_i} \tag{12}$$

In these equations, N signifies the total number of observations, while f and y represent the forecasted and observed values, respectively.

B. Single-step forecasting model

The comparison of the vanilla RNN, LSTM and GRU models using the single-step forecast of 1 day ahead for each point revealed that the LSTM model was the most successful. It had a MAE of 0.1966, which was approximately 20% lower than the MAE of 0.2441 for the RNN model. The GRU model had a MAE of 0.1937, which was also lower than the RNN model by around 21%. Similarly, the LSTM also demonstrated superiority in terms of MASE, with roughly a 10% advantage over the other two models. Figure 4 shows the forecast result for 1 month of the three models, including the SA optimised model. The results of the LSTM model indicate its strength and potential to be further developed and applied to the task of forecasting hospitalisation demand from time series data. All three models had a small forecast bias, but the LSTM and GRU models had a negative bias, indicating that they slightly underestimated the forecast, as evidenced by the forecast bias metric. The Seq2Seq model was trained using the single-step method, the results were acceptable except for the RNN-RNN Seq2Seq model, which produced results that were underforecasting more than the others, and can be seen in Figure 4.

C. Multistep forcasting models

The effectiveness of Seq2Seq architectures with attention mechanisms was demonstrated in multistep models. Two LSTM models were used as Seq2Seq encoder-decoder models, with the same parameters except for the last result in Table III.

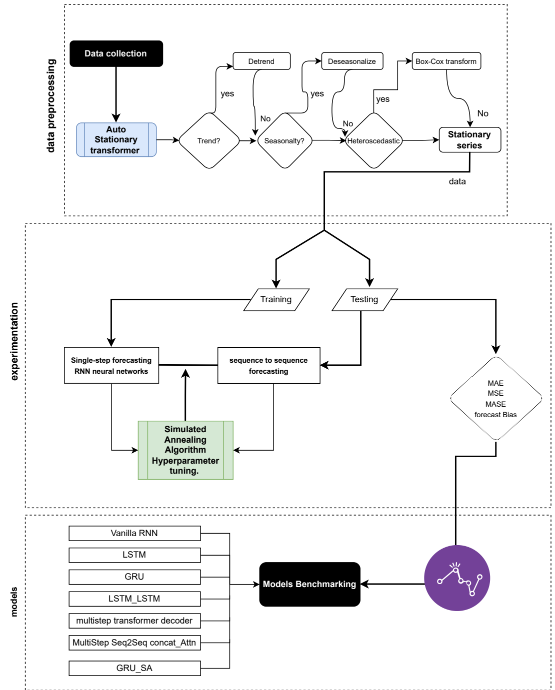

Fig. 3. Flowchart of the preprocessing of data, experimentation and evaluation of models

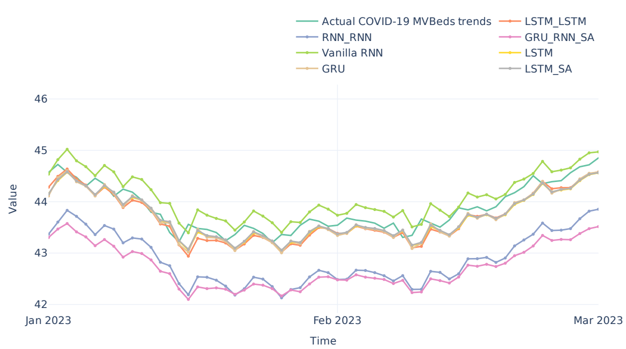

Fig. 4. Comparative analysis of the actual vs forecasted trends from January 2023 till March 2023 for MVbeds demands single step models

Figure 5 shows the 1 month forecast of the trained models. The model with the concat attention mechanism had the best MAE of 0.2326 and the lowest MSE of 0.0907. This implies that attention mechanisms can be beneficial in improving the predictability of multi-step models, although there is still room for improvement, as indicated by the MASE and Forecast Bias metrics. SA optimisation was also applied to find the optimal attention mechanism and other parameters to produce a refined model for the forecasting of hospitalisation demand.

The result presented in Table II was the result of using SA to search the parameters, performed in the three instances of experimentation that were trained on the data. i.e., single-step forecasting with RNNs architecture, Seq2Seq models, and multistep Seq2Seq models with attention network. Training and evaluation to determine the optimal parameters took a while to compute with the available computational power. The results present the models that were selected, the number of hidden_size and the number of layers for each model structure, as well as the learning rate. The LSTM model with a hidden layer size of 143 took the shortest time to compute, while the dot-attention (LSTM-LSTM) seq2seq model exhibited a significantly higher computational demand in seconds.

TABLE II
BEST PARAMETERS AND TOTAL RUNNING TIME FOR VARIOUS MODELS

SA model	hidden size	num layers	learning rate	Time (sec)
LSTM	143	3	0.0065	1203.04
GRU_RNN	172	3	0.009	1507.03
dot_Attn	48	5	0.006	4072.23

TABLE III
PERFORMANCE COMPARISON OF ALL MODELS

Algorithm	MAE	MSE	MASE	Forecast Bias
RNN	0.2441	0.0819	0.6622	0.23%
LSTM	0.1966	0.0549	0.5333	-0.17%
GRU	0.1937	0.0547	0.5255	-0.16%
LSTM_SA	0.1915	0.0525	0.5194	-0.16%
LSTM_LSTM	0.2058	0.0584	0.5584	-0.17%
RNN_RNN	0.9134	0.8697	2.4778	-0.91%
GRU_RNN_SA	1.1949	1.4545	3.2416	-1.19%
MultiStep_general_Attn	0.2314	0.0901	0.6277	0.01%
MultiStep_concat_Attn	0.2384	0.0938	0.6468	0.06%
MultiStep_additive_Attn	0.2707	0.1180	0.7343	0.17%
MultiStep_dot_Attn_SA	0.2794	0.1248	0.7579	-0.15%

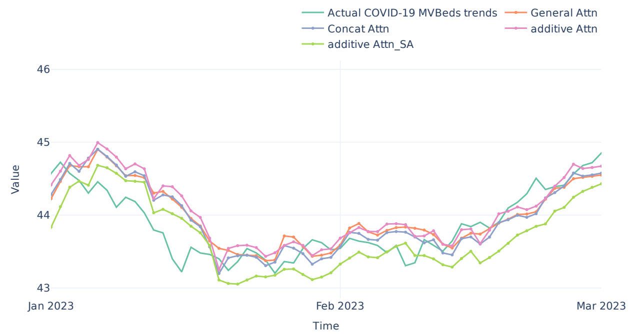

Fig. 5. Comparative analysis of the actual vs forecasted trends from January 2023 till March 2023 for MVbeds demands multi-step models

The results of the hyperparameter-tuned models were good. The LSTM_SA model had a slight improvement on the LSTM model, with a decrease of 2.6% in MAE, 4.4% in MSE, and 2.6% in MASE. However, more complex architectures, such as RNN_RNN and GRU_RNN_SA, despite their intricate architectures, did not perform well, with a considerable decrease in MAE. The MAE of GRU_RNN_SA was 1.1949, which is almost five times that of the basic GRU model. The outcome of the experiments raises a critical question about the efficacy and reliability of combining different architectures for forecasting. In hindsight, the results are relatively positive within the context of this research, serving to determine the forecastability of the data set.

V. Conclusion

This study conducted a thorough evaluation of various deep learning architectures to forecast the short-term demand for hospitalisation resources during the COVID-19 pandemic in England. The forecast horizon used in training these models was 1 day and 14 days using data available since 2020. The LSTM model, particularly LSTM_SA, achieved the best results. The impact of attention mechanisms and the simulated annealing algorithm for hyperparameter tuning was also examined, with varying degrees of success. This work makes an important contribution to the field of healthcare analytics, demonstrating the usefulness of deep learning models for resource allocation in pandemics. Short-term forecasting for a few days or weeks is a useful insight to better manage planning during future pandemics, leading to a reduction in mortality rates and an increase in quality of life. Future research should consider incorporating external variables, such as vaccination rates and policy changes, which makes the data multivariate, as well as explore other hyperparameter optimisation techniques to enhance forecast accuracy.

Acknowledgment

The authors would like to thank the Centre for Computational Science and Mathematical Modelling, Coventry University, for providing Michael Ajao-Olarinoye a full scholarship and supporting this work.

References

[1] M. A. Rguibi, N. Moussa, A. Madani, A. Aaroud, and K. Zine-Dine, “Forecasting covid-19 transmission with arima and lstm techniques in morocco,” *SN Computer Science*, vol. 3, no. 2, p. 133, 2022.

[2] R. Somyanonthanakul, K. Warin, W. Amasiri, K. Mairiang, C. Mingmalairak, W. Panichkitkosolkul, K. Silanun, T. Theeramunkong, S. Nitikraipot, and S. Suebnukarn, “Forecasting covid-19 cases using time series modeling and association rule mining,” *BMC Medical Research Methodology*, vol. 22, no. 1, p. 281, 2022.

[3] S. Berkhahn and M. Ehrhardt, “A physics-informed neural network to model covid-19 infection and hospitalization scenarios,” *Advances in Continuous and Discrete Models*, vol. 2022, no. 1, p. 61, 2022.

[4] D. Painuli, D. Mishra, S. Bhardwaj, and M. Aggarwal, “Forecast and prediction of covid-19 using machine learning,” in *Data Science for COVID-19*. Elsevier, 2021, pp. 381–397.

[5] M. O. Alassafi, M. Jarrah, and R. Alotaibi, “Time series predicting of covid-19 based on deep learning,” *Neurocomputing*, vol. 468, pp. 335–344, 2022.

[6] R. C. Staudemeyer and E. R. Morris, “Understanding lstm–a tutorial into long short-term memory recurrent neural networks,” *arXiv preprint arXiv:1909.09586*, 2019.

[7] F. Kamalov, K. Rajab, A. Cherukuri, A. Elnagar, and M. Safaraliev, “Deep learning for covid-19 forecasting: state-of-the-art review.” *Neurocomputing*, 2022.

[8] M. Goic, M. S. Bozanic-Leal, M. Badal, and L. J. Basso, “Covid-19: Short-term forecast of icu beds in times of crisis,” *Plos one*, vol. 16, no. 1, p. e0245272, 2021.

[9] M. Tello, E. S. Reich, J. Puckey, R. Maff, A. Garcia-Arce, B. S. Bhattacharya, and F. Feijoo, “Machine learning based forecast for the prediction of inpatient bed demand,” *BMC medical informatics and decision making*, vol. 22, no. 1, p. 55, 2022.

[10] D. Borges and M. C. Nascimento, “Covid-19 icu demand forecasting: A two-stage prophet-lstm approach,” *Applied Soft Computing*, vol. 125, p. 109181, 2022.

[11] A. Zeroual, F. Harrou, A. Dairi, and Y. Sun, “Deep learning methods for forecasting covid-19 time-series data: A comparative study,” *Chaos, solitons & fractals*, vol. 140, p. 110121, 2020.

[12] C. E. Overton, L. Pellis, H. B. Stage, F. Scarabel, J. Burton, C. Fraser, I. Hall, T. A. House, C. Jewell, A. Nurtay *et al.*, “Epibeds: Data informed modelling of the covid-19 hospital burden in england,” *PLoS computational biology*, vol. 18, no. 9, p. e1010406, 2022.

[13] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[14] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[15] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[16] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, “On the properties of neural machine translation: Encoder-decoder approaches,” *arXiv preprint arXiv:1409.1259*, 2014.

[17] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” *Advances in neural information processing systems*, vol. 27, 2014.

[18] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *arXiv preprint arXiv:1409.0473*, 2014.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.

[20] N. England. (2023) Covid-19 hospital activity. [accesse: 20-07-2023]. [Online]. Available: https://www.england.nhs.uk/statistics/statistical-work-areas/covid-19-hospital-activity/

[21] M. Joseph, *Modern Time Series Forecasting with Python: Explore industry-ready time series forecasting using modern machine learning and deep learning*. Packt Publishing Ltd, 2022.

[22] P. J. Van Laarhoven, E. H. Aarts, P. J. van Laarhoven, and E. H. Aarts, *Simulated annealing*. Springer, 1987.

[23] Y. Yoo, “Hyperparameter optimization of deep neural network using univariate dynamic encoding algorithm for searches,” *Knowledge-Based Systems*, vol. 178, pp. 74–83, 2019.