Predicting machine's performance data using the stacked long short-term memory (LSTM) neural networks

Jianrong Dai (Dai_jianrong@cicams.ac.cn)

Research

Keywords: long short-term memory networks (LSTM), daily QA, machine performance check (MPC), predictive time series, radiotherapy

Posted Date: October 11th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-961567/v1

Predicting machine's performance data using the stacked long short-term memory (LSTM) neural networks

Min Ma[1], Jianrong Dai[1#]

[1] Department of Radiation Oncology, National Cancer Center/National Clinical Research Center for Cancer/Cancer Hospital, Chinese Academy of Medical Sciences and Peking Union Medical College, Beijing, 100021, China

#Corresponding author

Jianrong Dai, PhD

Department of Radiation Oncology

National Cancer Center/National Clinical Research Center for Cancer/Cancer Hospital, Chinese Academy of Medical Sciences and Peking Union Medical College

Beijing 100021, China

Tel: 86-10-8778-8893

Email: dai_jianrong@cicams.ac.cn

Abstract

Purpose: Machine Performance Check (MPC) is a daily quality assurance (QA) tool for Varian machines. The daily QA data based on MPC tests show machine performance patterns and potentially provide warning messages for preventive actions. This study developed a neural network model that could predict the trend of data variations quantitively.

Methods and materials: MPC data used were collected daily for 3 years. The stacked long short-term memory (LSTM)model was used to develop the neural work model. To compare the stacked LSTM, the autoregressive integrated moving average model (ARIMA) was developed on the same data set. Cubic interpolation was used to double the amount of data to enhance prediction accuracy. After then, the data were divided into 3 groups: 70% for training, 15% for validation, and 15% for testing. The training set and the validation set were used to train the stacked LSTM with different hyperparameters to find the optimal hyperparameter. Furthermore, a greedy coordinate descent method was employed to combinate different hyperparameter sets. The testing set was used to assess the performance of the model with the optimal hyperparameter combination. The accuracy of the model was quantified by the mean absolute error (MAE), root-mean-square error (RMSE), and coefficient of determination (R^2).

Results: A total of 867 data were collected to predict the data for the next 5 days. The mean MAE, RMSE, and R^2 with all MPC tests was 0.013, 0.020, and 0.853 in LSTM, while 0.021, 0.030, and 0.618 in ARIMA, respectively. The results show that the LSTM outperforms the ARIMA.

Conclusions: In this study, the stacked LSTM model can accurately predict the daily QA data based on MPC tests. Predicting future performance data based on MPC tests will foresee possible machine failure, allowing early machine maintenance and reducing unscheduled machine downtime.
Keywords: long short-term memory networks (LSTM), daily QA, machine performance check (MPC), predictive time series, radiotherapy

1. Introduction

Linear accelerators (linacs) undergo daily quality assurance (QA) testing to ensure that radiation treatments are delivered safely and accurately, and that they meet the quality and safety criteria of AAPM TG 142 [1]. Daily QA testing are normally performed by the radiotherapist using a conventional QA instrument/ phantom.

The Machine Performance Check (MPC) is used for daily QA on the Varian linacs (Varian Medical Systems, Inc., Palo Alto, CA). MPC is a fully integrated KV and MV image-based tool to assess the performance of linac [2]. There are 2 types of MPC tests: beam constancy checks and geometric testing. First, beam constancy checks assess the output, beam center, and uniformity constancy against a user–defined baseline. Second, geometric tests evaluate the radiation isocenter size, the coincidence of MV and KV isocenters, collimator and gantry angle accuracy, jaw and multi-leaf collimator leaf accuracy positions, and couch pitch and roll accuracy. All tests are highly automated, and the user only should set up the IsoCal phantom and bracket on the treatment couch, then beam on for the predetermined energy. MPC application has been evaluated as a linac daily QA tool by some investigators [2-6].

The daily MPC result is used to track on the linac's long-term stability when processing large quantities of data. With those data, medical physicists could calibrate data baseline and monitor the linac's state to predict variation cycles and take preventive actions. To understand the underlying structure and functions that produce the observed tests in MPC, an appropriate modeling tool is needed to extract and analyze the longitudinal daily MPC data and predict future trends.

These sequential sets of daily MPC data measured over successive days can be considered time series. Therefore, in the context of time-series predictive modeling, the question of predicting the future trend and variations has been raised [7]. Traditionally, statistical modeling techniques like to autoregressive integrated moving average model (ARIMA) [8] and their variations (AR, MA, ARMA) [9-11], which only capture the linear elements of the time series and may not be sufficient for the daily QA data. Non-linear time series are best analyzed using recurrent neural network (RNN). However, RNN is difficult to deal with long time series [12]. Therefore, the long short-term memory (LSTM) network is proposed to tackle the forgetting problem [13], a type of RNN. LSTM has shown good performance in various fields (finance, public transportation, astronomy, environmental science, and medicine) [12]. In the previous studies of the predictive model development about daily QA in linac, Li and chan (2017) [7] used artificial neural networks (ANN), and

ARMA on 5-year daily beam QA data, which showed ANN had better prediction performance than ARMA. Puyati et al. (2020) [14] used statistical process control and ARIMA to forecast QA. However, there is poor performance to predict QA data in linac or trends and exist time lags in the predictive model.

A generalized LSTM model was developed to predict daily QA data/trend based on MPC tests in this study. Additionally, this study emphasized discovering the common behaviors of the linac performance so that physicists could be more confident in predicting the machine's future behavior and taking action in a planned way before the tolerance level is reached. Finally, to compare and provide context for our results, we also developed a prediction model with ARIMA on the same data set.

2. Methods and material

2.1. Data acquisition

MPC is designed to examine and evaluate the machine's performance about 5 min before starting the routine treatment. 24 MPC tests were run, including isocenter, collimation, gantry, and couch tests. Daily MPC data were collected at our institution using Varian Edge (Varian Medical Systems, Palo Alto, CA) for more than 3 years, from August 2017 to October 2020. We presented results for beam data predictive modeling in this study.

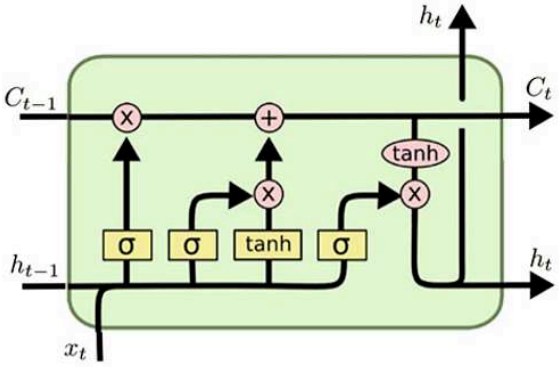
2.2. Data pre-processing

Pre-processing data is a significant step before building a model. In this study, performing data pre-processing included cleaning, interpolation, normalization, and data split. The duplicate data was deleted at the starting point. Cubic interpolation was used to double the amount of data to improve the prediction accuracy. The data was normalized for the model, ranging from −1 to 1. The data was divided into 3 sets: 70% for training, 15% for validation, and 15% for testing. The training set and the validation set was used to train the model with different hyperparameter combinations (see section 2.3.2). The testing set was used to assess the performance of the model with the optimal hyperparameter combination.

2.3. Building LSTM network

2.3.1 LSTM network

LSTM is very powerful in solving sequence prediction problems because it can store previous information [15], which is essential to predict the future data/trend of MPC daily QA data. Through the standard recurrent layer, self-loops, and the internal unique gate structure, the LSTM network effectively improves the forgetting and gradient vanishing problem existing in the traditional RNN [13]. Besides, LSTM can learn to make a one-shot multi-step prediction useful for predicting the time series. An LSTM neural network unit combines 4 gates: an input gate, a cell state, a forgotten gate, and an output gate (Fig. 1) [16].



$$i_t = \sigma(U_i x_t + W_i h_{t-1} + b_i)$$
$$\tilde{c_t} = \tanh(U_c x_t + W_c h_{t-1} + b_c)$$
$$c_t = g_t * c_{t-1} + i_t * \tilde{c_t}$$
$$f_t = \sigma(c x_t + W_g h_{t-1} + b_f)$$
$$o_t = \sigma(U_o x_t + W_o h_{t-1} + b_o)$$
$$h_t = o_t * \tanh(c_t)$$

Figure 1. The structure of LSTM as described by Varsamopoulos (2018) [18]. input gate (i_t), input module gate ($\tilde{c_t}$), forget gate (f_t), and output gate (o_t). b is bias vectors, c_t is cell state, h_t is the hidden state, and σ is the sigmoid activation function. All these controllers determine how much information to receive from the last loop, and how much to pass to the new state.

The forget gate is used to determine which messages pass through the cell, then enter the input gate, which decides how many new messages to add to the cell state, and finally decide the output message through the output gate [17].

The original LSTM model is comprised of a single hidden LSTM layer followed by a standard feedforward output layer. The stacked LSTM is an extension to this model that has multiple hidden LSTM layers where each layer contains multiple memory cells [12]. The stacked LSTM hidden layers make the model deeper, more accurately earning the description as a deep learning technique. It is the depth of neural networks that are attributed to the approach's success on various challenging prediction problems [19]. The stacked LSTM is now a stable technique for challenging sequence prediction problems [20]. An LSTM model with many LSTM layers is a stacked LSTM architecture (Fig. 2) [21]. An LSTM layer above provides a sequence output rather than a single value output to the LSTM layer below. Specifically, one output per input time step is one output time step for all input time steps. Therefore, in this study, the stacked LSTM was selected.

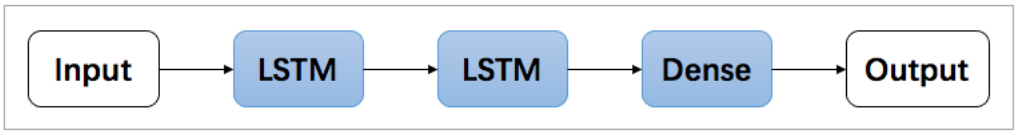


Figure 2. The stacked LSTM architecture.

For the ARIMA, there are 3 critical parameters in ARIMA: p (the past value used to predict the next value), q (past prediction error used to predict future values), and d (order of differencing) [22, 23]. ARIMA parameter optimization requires much time. Therefore, in this study, ARIMA selects the combination (5, 1, 0).

2.3.2. Model training

The experiment's LSTM model is built on the Keras API package (TensorFlow2.0) in Python 3.6 settings (Python Software Foundation, Wilmington, DE). In this study, networks with two LSTM layers were investigated. The loss value was evaluated by the root-mean-square error (RMSE). The activation functions used the rectified linear units (Relu) function. A greedy coordinate descent method [24] was employed to find the optimal hyperparameter of the model.

The length of time lags, the optimizers, the learning rates, the number of epochs, the number of hidden units, and the batch sizes were among the tuning parameters. First, we sought to find the optimal length of time lags when the optimizer was Adam, the learning rate was 0.01, the number of epochs was 150, the number of hidden units was 50, and the batch sizes were 32. Subsequently, we determined the type of optimizer with the optimal length of time lags. Next, the appropriate learning rate was determined by comparing results from various learning rates. Then, we sought to find the optimal number of epochs and hidden units in turn. Lastly, to determine the optimal batch size, a similar comparison was performed. The batch size was adjusted to avoid errors from memory shortage. By testing the parameter values of different combinations, and the model suitable for the data was finally found. The tunning hyperparameters were presented in section 2.3.3.

2.3.3 Hyperparameters optimization

Hyperparameters selection and optimization play an important role in obtaining superior accuracy with the LSTM network [25]. The validation set's mean absolute error (MAE) was used to evaluate the model's performance for each parameter combination. The following hyperparameters were investigated for the stacked LSTM network:

(1) Length of time lags – A time lag refers to a sequence of daily MPC QA data acting as the stacked LSTM model's input. The length of time lag represents the number of intakes to make a prediction, and different lengths of time lag may cause different prediction results.

(2) Optimizer – The optimizer is in charge of minimizing the stacked LSTM model's objective function.

(3) Learning rate – The optimizer's performance is affected by the learning rate. The learning rate determines how much the weight is updated at the end of each batch.

(4) Number of epochs – The number of epochs specifies how many times the stacked LSTM model traverses the whole training dataset. Each sample in the training dataset has the opportunity to update the internal model parameters once every epoch.

(5) Number of hidden units per layer – The number of neurons in a layer controls the representational capacity of the network. The same value for each LSTM layer was assigned.

(6) Batch size – In iterative gradient descent, the batch size refers to the number of patterns that are given to the network before the weights are updated. It is also a training optimization for the network, determining how many patterns to read and keep in memory.

2.4. Evaluation of predictive accuracy

To evaluate the error between the predicted and observed values in the testing set, the RMSE, MAE, and coefficient of determination (R^2) was selected.

2.5. The trend lines

The trend lines were used to analyze the trend of linac operating status and thereby help medical physicists decide whether to take preventive actions. The stacked LSTM model was applied to predict the next-5-day daily MPC results in this study. The trend lines were plotted by the polynomial fit from five-step-head predictive values.

3. Results

3.1. Hyperparameter tuning in LSTM

Figure 3 shows the MAE (in a relative unit) as a function of time lags, optimizers, learning rates, epochs, hidden units, and batch sizes. The optimal hyperparameter value is summarized in Table 1. Among them, the learning rate had the greatest impact on the model. The best performance was set to 0.001 of the learning rates, and the worst was set to 0.1, causing up to 0.039 difference in relative MAE. Furthermore, the type of optimizer had the second greatest impact on the model. In comparison, the length of time lags and the number of hidden units demonstrated only a modest impact on the model's predictive performance. Finally, the number of epochs and the batch size showed little impact on the predictive accuracy.

Table 1. The summary of LSTM hyperparameters investigated in this study, and the recommended configurations, and the impact level of each parameter.

Hyperparameters	Range	Recommended configuration	Impact
Length of time lags	{1, 5, 10, 15, 20, 30, 50}	15	Middle
Optimizer	{Adam, RMSprop, Adagrad, Adadelta}	Adam	High
Learning rate	{0.0001, 0.001, 0.01, 0.1}	0.001	High
Number of epochs	{50, 100, 150, 200, 250, 300}	300	Low
Number of hidden units per layer	{10, 30, 50, 70, 100}	50	Middle
Batch size	{8, 16, 32, 64, 128, 256}	32	Low

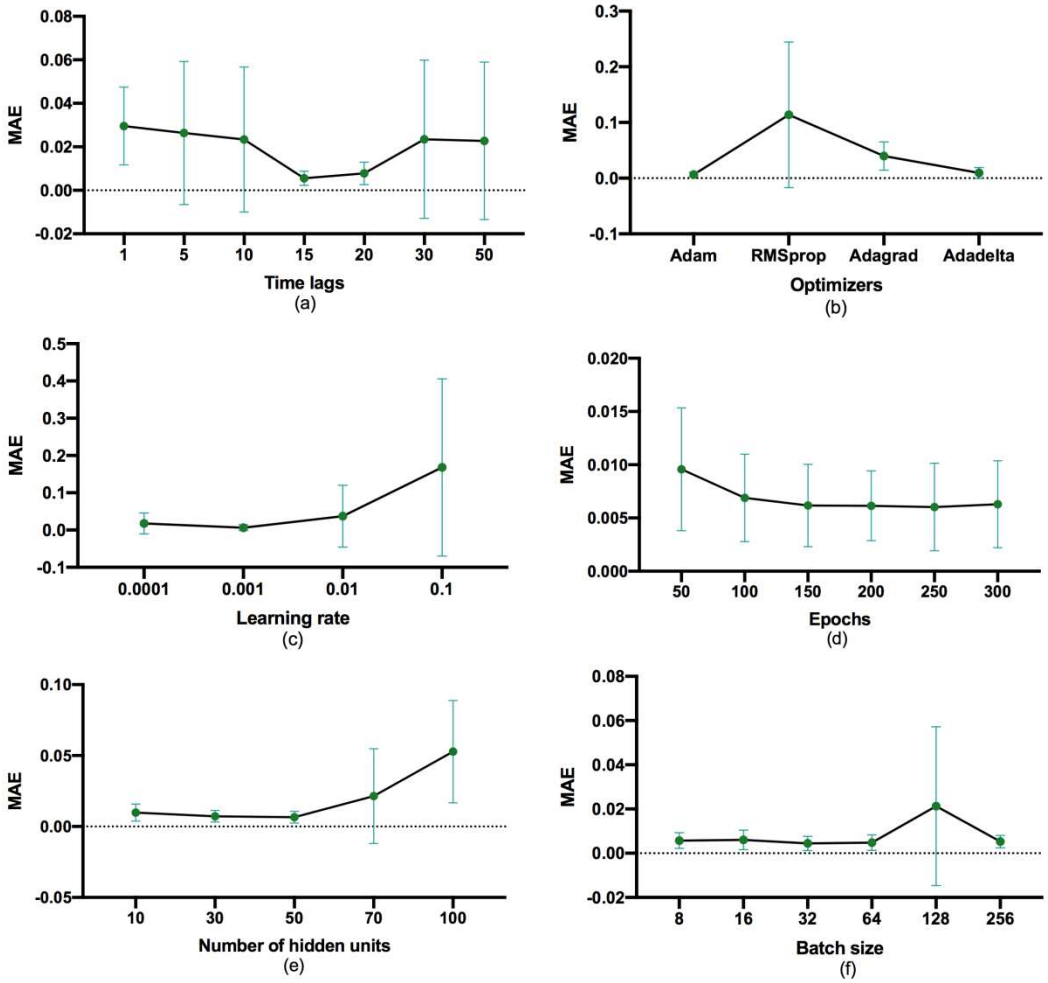


Figure 3. The MAE of predicted data (mean value in green, 95%CI in blue) with different values of different (a) the length of time lags, (b) the optimizers, (c) the learning rates, (d) the number of epochs, (e) the number of hidden units, and (f) the batch sizes. MAE is the mean absolute error.

3.2. Predictive performance evaluation

A total of 867 data is collected to predict the data for the next 5 days. Table 2 shows the performance of the stacked LSTM model in predicting daily MPC tests using the optimal hyperparameter and ARIMA. The mean MAE, RMSE, and R^2 with all MPC tests was 0.013, 0.020, and 0.853 in LSTM, while 0.021, 0.030, and 0.618 in ARIMA, respectively. LSTM performs better than ARIMA in 23 MPC items with the smaller MAE value, smaller RMSE value, and higher R^2, except for gantry relative (LSTM: MAE = 0.006, RMSE = 0.007 and R^2 = 0.095; ARIMA: MAE = 0.004, RMSE = 0.006 and R^2 = 0.383). The best predictive performance of LSTM was couch rotation (MAE = 0.001, RMSE = 0.004 and R^2 = 0.975), but the worst was gantry relative (MAE = 0.006, RMSE = 0.007 and R^2 = 0.095). Additionally, Figure 4 shows the comparison of model performance in terms of the coefficient of

determination (R^2). The R^2 value of LSTM is higher point than ARIMA in Figure 4, except for the R^2 value of gantry relative. In general, LSTM outperforms the ARIMA.

Figure 5 depicts 3 representative cases (beam center shift, beam output change, and beam uniformity change) of the observed versus the predicted curves using the stacked LSTM model with the optimal hyperparameter combination in testing data.

Table 2. Results of the stacked LSTM and ARIMA model evaluation.

MPC test		**MAE**		**RMSE**		**R2**	
Categories	**Test**	LSTM	ARIMA	LSTM	ARIMA	LSTM	ARIMA
Isocenter	Size (mm)	0.002	0.006	0.003	0.008	0.915	0.399
	MV imager projection offset (mm)	0.010	0.013	0.020	0.020	0.906	0.902
	KV imager projection offset (mm)	0.013	0.015	0.018	0.024	0.915	0.859
Collimation	Max offset leaves A (mm)	0.010	0.009	0.012	0.013	0.906	0.901
	Max offset leaves B (mm)	0.008	0.008	0.010	0.012	0.888	0.860
	Mean offset leaves A (mm)	0.007	0.009	0.010	0.013	0.912	0.849
	Mean offset leaves B (mm)	0.006	0.008	0.008	0.011	0.894	0.823
	Jaw X1 (mm)	0.012	0.013	0.018	0.019	0.872	0.847
	Jaw X2 (mm)	0.011	0.013	0.019	0.019	0.828	0.823
	Jaw Y1 (mm)	0.017	0.044	0.022	0.062	0.931	0.465
	Jaw Y2 (mm)	0.017	0.043	0.022	0.057	0.935	0.557
	Rotation offset (°)	0.030	0.047	0.039	0.064	0.732	0.278
Gantry	Absolute (°)	0.003	0.006	0.005	0.008	0.808	0.462
	Relative (°)	0.006	0.004	0.007	0.006	0.095	0.383
Couch	Lateral (mm)	0.005	0.012	0.006	0.016	0.918	0.393
	Longitudinal (mm)	0.004	0.010	0.006	0.014	0.947	0.658
	Pitch (°)	0.001	0.002	0.002	0.003	0.875	0.514
	Roll (°)	0.002	0.004	0.002	0.005	0.877	0.406
	Rotation (°)	0.001	0.002	0.001	0.004	0.975	0.436
	Vertical (mm)	0.008	0.016	0.011	0.021	0.870	0.520
	Rotation-induced couch shift (mm)	0.008	0.017	0.011	0.024	0.909	0.565
Beam	Center shift (mm)	0.014	0.027	0.021	0.037	0.874	0.611
	Beam output change (%)	0.069	0.098	0.120	0.139	0.931	0.907
	Uniformity change (%)	0.054	0.082	0.078	0.129	0.765	0.412
Mean		0.013	0.021	0.020	0.030	0.853	0.618

Abbreviations: MAE = mean absolute error, RMSE = root - mean-square error, R^2 = coefficient of determination.

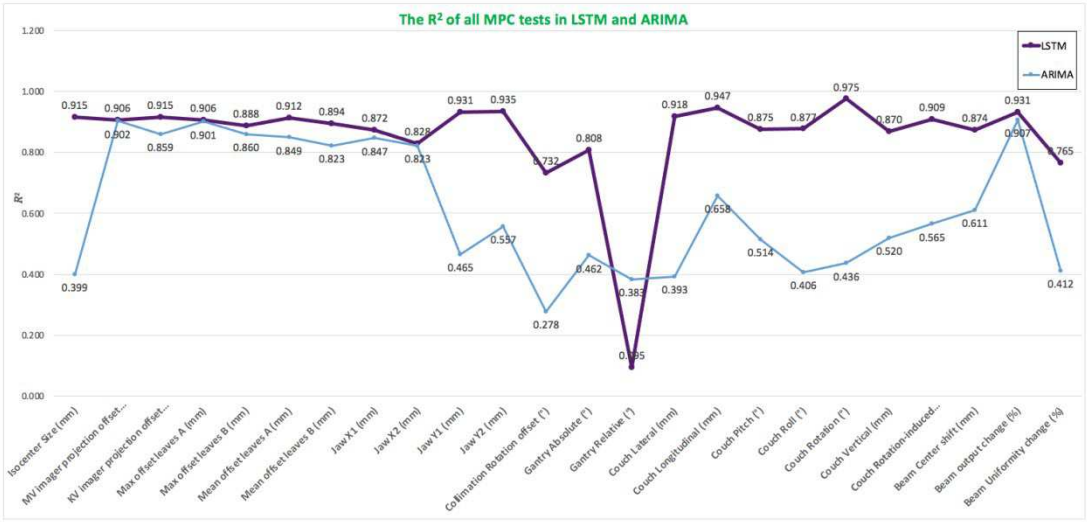


Figure 4. Comparison graph of model performance in the coefficient of determination (R^2). The purple line means LSTM, and the blue line means ARIMA.

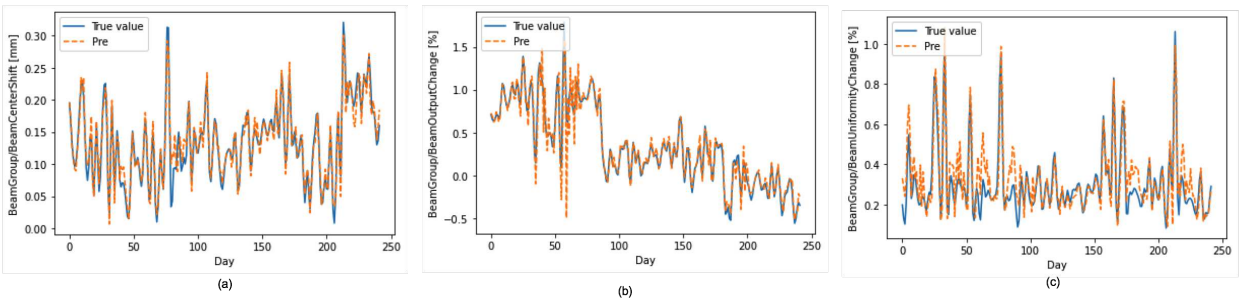


Figure 5. Comparison of predicted and observed beam QA results, including (a) beam center shift, (b) beam output change, and (c) beam uniformity change using the stacked LSTM model with the optimal hyperparameters in testing data.

3.3. The trend lines

The weekly trend line for the beam is shown in Figure 6. All predictive values were within the tolerance. The trend was that the beam center shift drop but remains at normal-stage levels. The trend of the beam output change and beam uniformity change rose, located in the normal range. This provides the opportunity to adjust the machine.

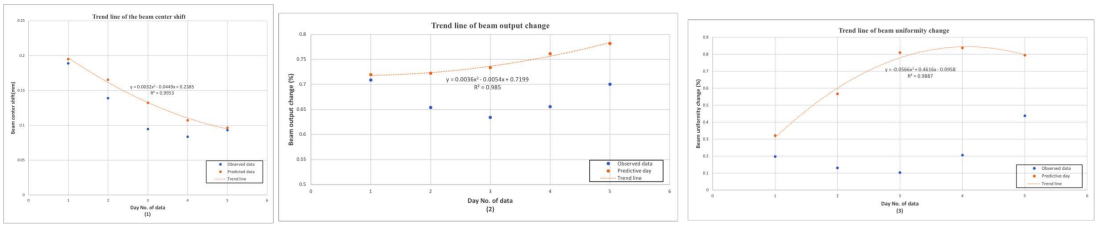


Figure 6. An example of the trend line to detect (a) the beam center shift, (b) beam output change, and (c) beam uniformity change.

4. Discussion

4.1. Impact of hyperparameters

This study demonstrates the need to tunning the hyperparameter using a deep LSTM model for daily MPC testing to obtain good predictive results. The learning rate determines how fast your neural net "learns." If the learning rate is too high, our loss will start jumping all over the place and never converge [26]. If the learning rate is too low, the model will take way too long to converge [26], as illustrated above. The challenge of using a learning rate is that their hyperparameters must be defined in advance, and they depend heavily on the type of model and problem. Adaptive gradient descent algorithms (Adagrad, Adadelta, RMSprop, Adam) provide a heuristic approach without requiring expensive work to manually tuning hyperparameters for the learning rate [27]. According to the MAE value (Fig. 3), Adam and learning rate setting to 0.001 was recommended to use in the stacked LSTM model. Besides, when adjusting the different lengths of time lags, the LSTM predictive effect will be delayed (Fig. 7). R^2 value of the beam center shift is 0.603 (the lengths of time lag = 1), while R^2 value of the beam center shift is 0.874 (the lengths of time lag = 15). Lag observations for a univariate time series can be used as time lags for an LSTM model, which can improve forecast performance.

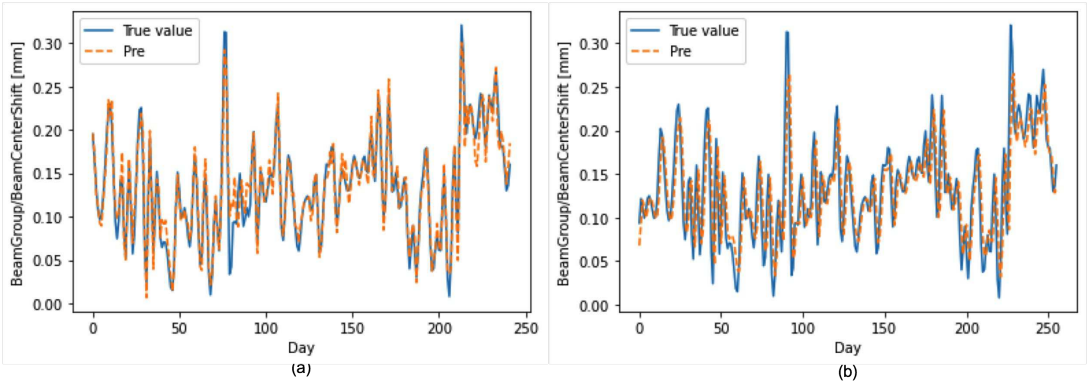


Figure 7. The predictive performance of the beam center shift with (a) the length of time lags= 15, and (b) the length of time lags= 1in the stacked LSTM model.

4.2. Predictive performance

This is the first study to implement a stacked LSTM model for daily MPC data prediction to the best of our knowledge. It is one of the first few attempts to develop and evaluate a single generic stacked LSTM model. The stacked LSTM model allowed connections through time and provided a way to feed the hidden layers from previous steps (long-term and short-term) as additional inputs to the next stage, in contrast to earlier studies that only focused on studying the power of ANN [7]. The stacked LSTM is effective at predicting daily MPC data. However, the generic

stacked LSTM is poor in predicting the data of the gantry relative with 2 times cubic interpolation. In Figure 8(a), the predictive range is significantly shifted up and slightly delayed. According to Wang et al. (2019) study [28], we guess that LSTM predictive performance is related to the signal frequency. Interpolation reduces high-frequency signals and can greatly improve the predictive ability of the stacked LSTM model. Therefore, we try 4 times cubic interpolation and 6 times cubic interpolation in the stacked LSTM model, which significantly improves the accuracy performance (Fig. 8(b) and (c)). The predictive performance of the gantry relative with 6 times cubic interpolation is great (R^2 = 0.978) in the stacked LSTM model.

To illustrate the robustness of the model, we apply LSTM model to the QA data of output dose weekly on the Elekta's linac without changing any hyperparameters chosen for this study. The MAE, RMSE and R^2 is 0.229, 0.283, and 0.750, respectively (Fig. 9). For a clinical routine, it is unnecessary to retrain the neural network each day with the daily acquired MPC data.

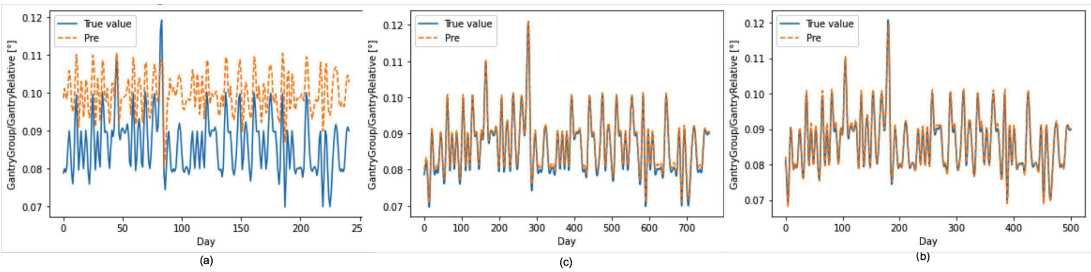
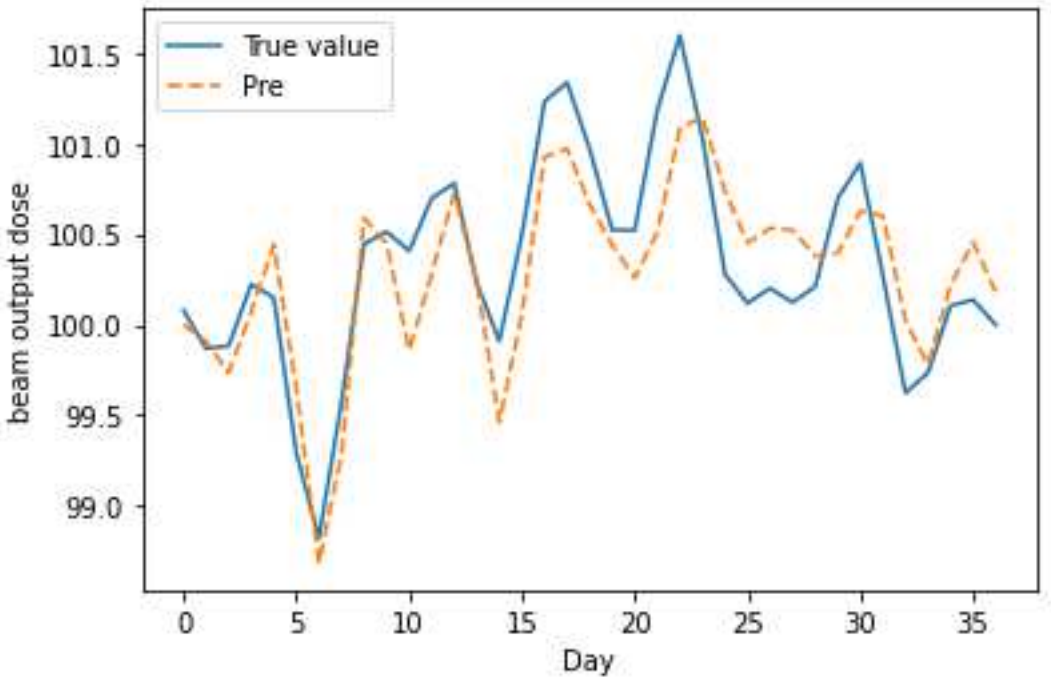


Figure 8. The predictive performance of the gantry relative to (a) 2 times cubic interpolation, (b) 4 times cubic interpolation, and (c) 6 times cubic interpolation in the stacked LSTM model.



4.3. The trend lines

For all daily MPC tests, the predicted data locates within the clinical tolerances (AAPM TG-142) [1], providing a window of opportunity to prevent the performance issue in advance. However, in the actual situation, besides keeping parameters within the tolerance, a clinical physicist should monitor trends in the machine performance [29] and to know when the linac needs to be maintained, thereby reducing the chance of linac downtime. Here, a five-step-ahead prediction is appropriate to provide trends in linac status.

4.4. Limitations and future work

However, there exist some limitations in this study. Considering the time-consuming problem, tuning hyperparameters does not choose between the grid search, the random search, and random search. Furthermore, some hyperparameters correlate with each other and can result in different performances when optimized simultaneously rather than tuning [30]. Due to prediction models being based on large learning-phase datasets, the predictive models are not designed to detect large sudden one-off jumps in data such as might be expected with a linac component failure. Linac interlocks are still required to mitigate treatment delivery errors from such events and routine retrospective QA. Predictive QA is more suited to detecting and predicting gradual drifts and failures that repeat at regular intervals.

The present study results suggest that the approach of predictive QA based on MPC tests is feasible, but additional data on more linacs are required. Therefore, such a study is proposed as future work.

5. Conclusions

This study developed and evaluated a generalized stacked LSTM model for daily MPC prediction. This model has a better performance than the ARIMA model and can reduce the unscheduled linac downtime and allow linac performance parameters to be controlled within tolerances in the clinic.

CONFLICT OF INTEREST

The authors have no conflicts to disclose.

References

[1] Klein EE, Hanley J, Bayouth J, Yin FF, Simon W, Dresser S, et al. Task Group 142 report: Quality assurance of medical accelerators a. Medical physics. 2009;36:4197-212.

[2] Clivio A, Vanetti E, Rose S, Nicolini G, Belosi MF, Cozzi L, et al. Evaluation of the Machine Performance Check application for TrueBeam Linac. Radiat Oncol. 2015;10:97.

[3] Barnes MP, Pomare D, Menk FW, Moraro B, Greer PB. Evaluation of the truebeam machine performance check (MPC): OBI X-ray tube alignment procedure. J Appl Clin Med Phys. 2018;19:68-78.

[4] Barnes MP, Greer PB. Evaluation of the TrueBeam machine performance check (MPC) beam constancy checks for flattened and flattening filter-free (FFF) photon beams. J Appl Clin Med Phys. 2017;18:139-50.

[5] Barnes MP, Greer PB. Evaluation of the truebeam machine performance check (MPC): mechanical and collimation checks. J Appl Clin Med Phys. 2017;18:56-66.

[6] Barnes MP, Greer PB. Evaluation of the truebeam machine performance check (MPC) geometric checks for daily IGRT geometric accuracy quality assurance. J Appl Clin Med Phys. 2017;18:200-6.

[7] Li Q, Chan MF. Predictive time-series modeling using artificial neural networks for Linac beam symmetry: an empirical study. Annals of the New York Academy of Sciences. 2017;1387:84.

[8] Siami-Namini S, Tavakoli N, Siami Namin A. A Comparison of ARIMA and LSTM in Forecasting Time Series. 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)2018. p. 1394-401.

[9] Tu Y-H, Peng C-C. An ARMA-Based Digital Twin for MEMS Gyroscope Drift Dynamics Modeling and Real-Time Compensation. Ieee Sensors Journal. 2021;21:2712-24.

[10] Lee K, Lee C-H, Kwak M-S, Jang EJ. Analysis of multivariate longitudinal data using ARMA Cholesky and hypersphere decompositions. Computational Statistics & Data Analysis. 2021;156.

[11] Chakraborty D, Sanyal SK. Time-series data optimized AR/ARMA model for frugal

spectrum estimation in Cognitive Radio. Physical Communication. 2021;44.
[12] Yu Y, Si X, Hu C, Zhang J. A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. Neural Comput. 2019;31:1235-70.
[13] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput. 1997;9:1735-80.
[14] Puyati W, Khawne A, Barnes M, Zwan B, Greer P, Fuangrod T. Predictive quality assurance of a linear accelerator based on the machine performance check application using statistical process control and ARIMA forecast modeling. Journal of Applied Clinical Medical Physics. 2020;21:73-82.
[15] Gers FA, Schmidhuber J, Cummins F. Learning to forget: Continual prediction with LSTM. Neural Computation. 2000;12:2451-71.
[16] Pascanu R, Mikolov T, Bengio Y. Understanding the exploding gradient problem. ArXiv. 2012;abs/1211.5063.
[17] Elman J. Finding Structure in Time. Cogn Sci. 1990;14:179-211.
[18] Varsamopoulos S, Bertels K, Almudever C. Designing neural network based decoders for surface codes2018.
[19] Anastasi G, Bertholet J, Poulsen P, Roggen T, Garibaldi C, Tilly N, et al. Patterns of practice for adaptive and real-time radiation therapy (POP-ART RT) part I: Intra-fraction breathing motion management. Radiotherapy and Oncology. 2020;153:79-87.
[20] Li Y, Lin Z. Melody Classifier with Stacked-LSTM. ArXiv. 2020;abs/2010.08123.
[21] Fern·ndez S, Graves A, Schmidhuber J. Sequence Labelling in Structured Domains with Hierarchical Recurrent Neural Networks. IJCAI2007.
[22] Zhang G. Time series forecasting using a hybrid ARIMA and neural network model. Neurocomputing. 2003;50:159-75.
[23] Box G, Jenkins G. Time series analysis, forecasting and control. 1970.
[24] Dhillon I, Ravikumar P, Tewari A. Nearest Neighbor based Greedy Coordinate Descent. NIPS2011.
[25] Greff K, Srivastava R, Koutnìk J, Steunebrink BR, Schmidhuber J. LSTM: A Search Space Odyssey. IEEE Transactions on Neural Networks and Learning Systems. 2017;28:2222-32.
[26] Yu C, Qi X, Ma H, He X, Wang C, Zhao Y. LLR: Learning learning rates by LSTM for training neural networks. Neurocomputing. 2020;394:41-50.
[27] Zaheer R, Shaziya H. A Study of the Optimization Algorithms in Deep Learning. 2019 Third International Conference on Inventive Systems and Control (ICISC). 2019:536-9.
[28] Wang S, Li C, Lim A. Why Are the ARIMA and SARIMA not Sufficient. ArXiv. 2019;abs/1904.07632.
[29] Chan MF, Li Q, Tang X, Li X, Li J, Tang G, et al. Visual Analysis of the Daily QA Results of Photon and Electron Beams of a Trilogy Linac over a Five-year Period. International journal of medical physics, clinical engineering and radiation oncology. 2015;4:290-9.
[30] Lin H, Shi C, Wang B, Chan MF, Tang X, Ji W. Towards real-time respiratory motion prediction based on long short-term memory neural networks. Phys Med Biol. 2019;64:085010.