CIGI QUALITA MOSIM 2023

Forecasting product quality using peephole long short term memory

TUAN LE, HAI-CANH VU, AMELIE PONCHET-DURUPT, NASSIM BOUDAOUD, ZOHRA CHERFI-BOULANGER
[1] Roberval laboratory, University of Technology of Compiegne
France
tuan.le@utc.fr

***Abstract* – Predicting the future health of production systems in manufacturing through forecasting is a reliable method. Manufacturers can detect potential issues early and take appropriate actions by using forecasting techniques. While many studies focus on predicting system health using process data such as sound, vibration, and magnetic readings of machines, there are limited studies that use product quality data to forecast the evolution of manufacturing systems. Besides, forecasting the short-term product quality time series in manufacturing systems, which are nonlinear, non-stationary, and nonseasonal, can be challenging. This study explores the use of peephole long short-term memory-based recurrent neural networks to forecast product quality data and address this challenge. Peephole LSTM-based RNN can effectively utilize the long-term dependencies in the product quality time series for more accurate forecasting. Experiments have been conducted to demonstrate that LSTM-based RNN can accurately forecast the complex quality of time series with a long forecasting horizon, and its performance is superior compared to many other forecasting methods.**

***Keywords* –** Time series forecasting, deep learning, and smart manufacturing.

1 INTRODUCTION

Forecasting is used as a tool in manufacturing for predicting the future health of production systems and creating maintenance plans (Kahraman et al., 2010). By using forecasting, manufacturers can predict potential issues with production systems and plan maintenance activities to prevent or minimize downtime (Vu et al., 2021). This proactive approach can result in several benefits for the manufacturing process, including improved efficiency (Mourtzis & Vlachou, 2018), reliability (Mobley, 2002), and cost savings (He et al., 2018).

Many studies in the field of manufacturing focus on using process data, such as sound, vibration, and magnetic readings of machines, to predict system health and schedule maintenance more effectively. For instance, (Janssens et al., 2016) found that by analyzing vibration data, it was possible to detect early signs of bearing wear and predict when maintenance would be needed. (Peng et al., 2010) used sound data to predict the health of a machine's gears and detect early signs of gear wear. Additionally, (Scalabrini Sampaio et al., 2019) used magnetic readings to predict the health of a machine's electrical systems, detecting early signs of electrical issues such as power loss. Furthermore, (Wu et al., 2021) used a combination of vibration and sound data to predict the health of a machine's shafts and detect early signs of shaft wear. (Deutsch & He, 2017) used a feedforward neural network (FNN) to predict the remaining useful life of bearings in a manufacturing system. The study found that the FNN model was able to accurately predict the remaining useful life of the bearings and improve maintenance planning. (Zhu & Chen, 2006) predict the remaining useful life of machines in a manufacturing system. The study found that the CNN model was able to accurately predict the remaining useful life of the machines based on images of the machines and improve maintenance planning.

Although extensive research has been done, accurate product quality forecasting still needs to be improved in manufacturing. Currently, limited studies use product quality data in this way, possibly due to a need for more resources or an understanding of how to utilize the data. As data collection and analysis capabilities improve, it is expected that more studies will be conducted in the future utilizing product-quality data. Using product quality data in manufacturing systems to predict their evolution can be beneficial in several ways. Analyzing this data can help manufacturers identify patterns and trends in their products, as well as potential issues with their processes (Rüßmann et al., 2015).

On the other hand, product quality forecasting is usually a univariate time series forecasting problem more challenging than the corresponding multivariate time series forecasting problem. Because there is no additional information from other data sources that can be utilized for learning (du Preez & Witt, 2003). In addition, compared with linear, stationary, and seasonal time series, short-term product quality time series in manufacturing systems are nonlinear, non-stationary, and nonseasonal, where nonseasonal means without apparent periodicity in time. It is not easy to forecast accurately such a time series in a long time horizon. Therefore, more efforts are needed to develop more effective forecasting methods.

In this study, we propose a short-term product quality forecasting solution by utilizing the Long-Short-Term-Memory (LSTM) method (Hochreiter & Schmidhuber, 1997). LSTM is a specialized Recurrent Neural Network (RNN) that excels in learning and predicting temporal sequences and long-term dependencies more effectively than other methods such as Deep Neural Networks and traditional RNNs (Sak et al., 2014). We introduce a new forecasting approach that utilizes LSTM to accurately predict the complex and unpredictable nature of univariate product quality time series. This approach can effectively forecast over a long-term horizon.

The remainder of this paper is organized as follows. The univariate time series forecasting problem is introduced in Section 2. In Section 3, the product quality forecasting scheme with LSTM-based RNN is presented. Experiments and conclusions are given in Sections 4 and 5, respectively.

2 MULTI-STEP AHEAD TIME SERIES FORECASTING

The goal of multi-step ahead forecasting for a univariate product quality time series is to use past observations, represented by $X_{t_1}, X_{t_2}, \cdots, X_{t_N}$, to predict future observations, represented by $X_{t_N+1}, X_{t_N+2}, \cdots, X_{t_N+H}$, where H is the forecasting horizon. The time horizon for forecasting in product quality control can vary depending on the specific product and industry, it is often determined by the time it takes for a product to go from production to delivery to the customer. For example, perishable goods have a shorter time horizon compared to products with a longer production and delivery cycle such as cars or machinery. Factors such as the level of uncertainty in the forecasting process and the cost of forecasting errors may also influence the time horizon for forecasting. Typically, product quality data are obtained through smart sensors or CPS (Cyber-physical system), if the smart sensors have a sampling interval of 60 minutes, the forecasting horizon H can be for 24 hours (one day) ahead of product quality forecasting. On the other hand, CPS has an even higher sampling frequency with a sampling interval in sub-seconds, and the forecasting horizon can be extremely long.

2.1 *Recursive Strategy*

Three methods are commonly used for multi-step ahead time series forecasting: the recursive strategy, the direct strategy, and the multiple-input and multiple-output (MIMO) strategy (Taieb et al., 2012). The traditional and most simple method is the recursive strategy (Hamzaçebi et al., 2009), which uses a single forecasting model, $f(.)$, to make predictions one step at a time. This is done by using the equation:

$$X_{t+1} = f(X_t, X_{t-1}, \cdots, X_{t-d+1}) + \varepsilon \tag{1}$$

Where t ranges from d to N, d is the dimension of the estimator, ε is the additive noise, $f: \mathbf{R}^d \to \mathbf{R}$ is the estimator, and $\mathbf{R}$ represents the real field. To make predictions H steps ahead, the one-step-ahead prediction X_{t_N+1} is first made using the equation (1). Then, with the forecasted X_{t_N+1} as part of the input time series, the next step is to estimate X_{t_N+2} using the same one-step ahead forecasting model in equation (1), and this process is repeated until X_{t_N+H} has been estimated.

The recursive strategy is simple to use, but it can be affected by the build-up of forecasting errors when the forecasting horizon is large. In this strategy, any errors made in previous predictions are carried forward and accumulated, which can negatively impact the accuracy of future predictions. This problem is more noticeable as the forecasting horizon increases (Taieb et al., 2012).

2.2 *Direct Strategy*

The direct strategy is another method for multi-step ahead forecasting (Cox, 1961). Unlike the recursive strategy, it creates H different forecasting models based on the observed time series data for each forecasting horizon. The equation for this strategy is:

$$X_{t+h} = f_h(X_t, X_{t-1}, \cdots, X_{t-d+1}) + \varepsilon_h \tag{2}$$

Where h ranges from 1 to H, f_h is the h^{th} forecasting model and ε_h is the additive noise associated with the h^{th} model. The direct strategy does not use any forecasted value as input, so it is not affected by accumulated errors. However, since each of the H forecasting models is trained separately and independently, it may result in conditional independence among the H forecasted values (An & Anh, 2015). This independence effect can degrade forecasting performance by not reflecting the statistical dependency among the forecasted data.

2.3 *Multiple-input and Multiple-output (MIMO) Strategy*

Both the recursive and direct strategies are known as single-output strategies, as they use multiple inputs (a vector) to create a single output (a scalar) (Taieb et al., 2012). The MIMO strategy, on the other hand, uses multiple inputs to create multiple outputs (Bontempi, 2008). The result of this strategy is a time series (a vector) instead of a scalar. The equation for this strategy is:

$$X_{t+1}, X_{t+2}, \cdots, X_{t+h} = F(X_t, X_{t-1}, \cdots, X_{t-d+1}) + \varepsilon_H \tag{3}$$

where ε_H is the noise vector, $F: \mathbf{R}^d \longrightarrow \mathbf{R}^H$. Unlike the single-output strategies, the MIMO strategy can handle the conditional independence problem by preserving the temporal statistical dependency in the forecasted time series. However, it has lower flexibility and variability compared to other forecasting strategies as all the data are forecasted using the same model (Taieb et al., 2010).

3 LSTM-BASED RNN FOR PRODUCT QUALITY FORECASTING

3.1 *Recurrent Neural Networks (RNNs)*

Feedforward neural networks, such as MLP (Multilayer Perceptron), DNN (Deep Neural Network), and CNN (Convolution Neural Network), have achieved great success in various supervised or unsupervised machine learning applications. However, their performance is heavily dependent on the independence assumption among training and test data (Lipton et al., 2015). When data in a time series are dependent on each other or the assumption of independence is not met, the learning performance of these networks will degrade due to their lack of capability to model long-term dependencies. Time series forecasting is a prime example of a scenario where current data points are related to previous data points, and long-term dependence is at the core of time series forecasting. Furthermore, feedforward neural networks are limited by the requirement of fixed-length inputs and targets (Sutskever et al., 2014), which also makes them inappropriate for sequence (such as time series) learning.

On the other hand, Recurrent Neural Networks (RNNs) are specifically designed to handle sequential data or time series (Medsker & Jain, 1999). RNNs allow signals to travel both forward and backward through the network by introducing loops, which enable internal connections among hidden units. These internal connections make RNNs more suitable for using information from past data to predict future data. Furthermore, RNNs have the ability to explore temporal relationships among data that are far apart from each other (Pascanu et al., 2013).

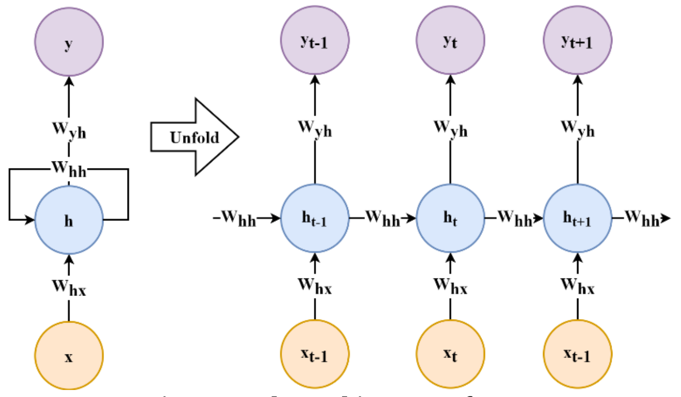

Figure 1. The architecture of RNN

Fig.1 illustrates the structure of an RNN. Given a time series input of $\mathbf{x}=\{x_1,x_2,\cdots,x_T\}$, the RNN repeatedly calculates the hidden state sequence $\mathbf{h}=\{h_1,h_2,\cdots,h_T\}$ and the output sequence $\mathbf{y}=\{y_1,y_2,\cdots,y_T\}$ using the following equations:

$$h_t = f(W_{hx}x_t + W_{hh}h_{t-1} + b_h) \tag{4}$$

$$y_t = g(W_{yh}h_t + b_y) \tag{5}$$

In (4) - (5), W_{hx}, W_{hh} and W_{yh} represent the input-hidden weight matrix, the hidden-hidden weight matrix, and the hidden-output weight matrix respectively. b_h and b_y are the bias of the hidden and output layers respectively. $f(.)$ and $g(.)$ are the activation functions for the hidden and output layers respectively. The RNN uses the hidden state h_t at the time step t to remember the network's state. The hidden state captures all the information from previous time steps.

Multi-step-ahead time series forecasting involves predicting multiple steps into the future, which requires accounting for dependencies between data points. However, as the interval of data dependencies increases, simple RNNs may struggle with the gradient vanishing problem (Bengio et al., 1994). This means that the influence of input data at much earlier times t_e on the forecasted data x_{t+h} decreases rapidly as the time difference between $t+h$ and t_e increases. As a result, simple RNNs may not be well suited for forecasting problems with long-term dependencies.

3.2 LSTM (Long-Short-Term-Memory) Architecture

The LSTM architecture, developed in 1997 by (Hochreiter & Schmidhuber, 1997) and further improved by others (Graves et al., 2013), is an efficient type of Recurrent Neural Network (RNN) designed to address the issue of vanishing gradients in standard RNNs when handling long-term dependencies.

In contrast to the standard RNN, which is composed of a series of simply hidden layers, the hidden layers of LSTM have a more complex structure. This structure includes the implementation of gates and memory cells in each hidden layer. The input, forget, and output gates control the flow of information into and out of the memory cells, while the memory cells themselves store the information over a long period of time, effectively resolving the vanishing gradient problem (Sak et al., 2014). This makes LSTM an ideal architecture for problems that require the examination of long-term dependencies.

Since the gates cannot get any information from the memory cell output when the output gate is closed, the LSTM does not know how long the memory should be for the model. To resolve this problem, peephole connections can be added to the LSTM memory cells. Working as an immediate supervisor, peephole connections make it possible for all the gates to inspect the cell states (Sutskever et al., 2014). Fig. 2 shows the architecture of a general LSTM memory block with peephole connections added.

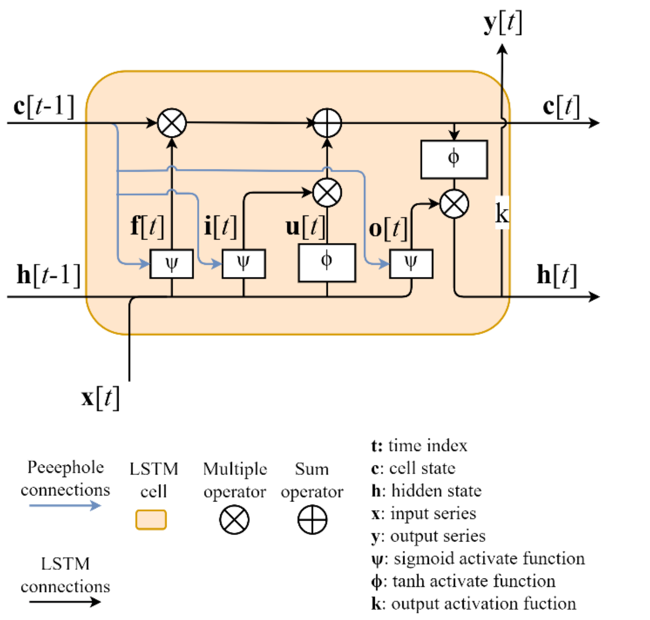

Figure 2. Long-Short Term Memory cell with peephole connections

3.3 LSTM-based RNN Forecasting Scheme

This paper uses an LSTM-based Recurrent Neural Network (RNN) scheme for forecasting product quality time series. LSTM is chosen for its advantages in time series forecasting, and the scheme utilizes peephole connections to enhance the LSTM's performance. The input time series $\mathbf{x}=\{x_1,x_2,\cdots,x_T\}$ is transformed by the LSTM into two output time sequences: hidden state $\mathbf{h}=\{h_1,h_2,\cdots,h_T\}$, and output $\mathbf{y}=\{y_1,y_2,\cdots,y_T\}$ through an iterative process. The hidden state **h** represents the relevant information in the input sequence **x** to make predictions **y**. Specifically, the states of the memory cells are updated using the following procedure.

First, as shown in Fig. 2, the forget gate is applied to help the LSTM to decide what information to throw away from the cell state. The function $\Psi(.)$ is generally a sigmoid used to calculate the activation of the forget gate as

$$f_t = \Psi(W_{fx}x_t + W_{fh}h_{t-1} + W_{fc}c_{t-1} + b_f) \tag{6}$$

The output f_t of equation (6) is a value between 0 and 1 corresponding to the last cell state c_{t-1}. The value 0 means forgetting the last state completely, while the value 1 stands for keeping the last state completely.

Next, we need to let the LSTM know what new information is going to be stored in the new cell state. To begin with, the LSTM uses a sigmoid layer, which is named the input gate layer i_t, where

$$f_t = \Psi(W_{ix}x_t + W_{ih}h_{t-1} + W_{ic}c_{t-1} + b_i) \tag{7}$$

to decide what information to update. The tanh layer $\Phi(.)$

constructs a vector $\mathbf{u}_t$ to store the new candidate values to be added to the new cell state as

$$u_t = \Phi(W_{cx}x_t + W_{ch}h_{t-1} + b_c) \tag{8}$$

Then, the old cell state c_{t-1} is updated to a new cell state c_t with the estimated f_t and u_t. Specifically, the old cell state is multiplied with f_t in order to forget information from the last state. The candidate values are multiplied with the input gate layer to decide how much new information to be updated to the new cell state, which gives

$$c_t = u_t i_t + c_{t-1} f_t \tag{9}$$

Another sigmoid layer $\Psi(.)$ is then used as the output gate to filter and output the cell state as o_t, where

$$o_t = \Psi(W_{ox}x_t + W_{oh}h_{t-1} + W_{oc}c_{t-1} + b_o) \tag{10}$$

Furthermore, a cell output tanh activation function $\Phi(.)$ is applied over the cell state, which is then multiplied by the output o_t to give the desired information

$$h_t = o_t \Phi(c_t) \tag{11}$$

As for the output of the memory block, an output activation function $k(.)$ is used, i.e,

$$y_t = k(W_{yh}h_t + b_y) \tag{12}$$

In (6 - 12), the matrices W_{ix}, W_{fx}, W_{ox}, W_{cx} are the appropriate input weight matrices, W_{ih}, W_{fh}, W_{oh}, W_{ch} are the recurrent weight matrices, W_{yh} represents the hidden output weight matrix, W_{ic}, W_{fc}, W_{oc} denote the weight matrices of peephole connections. The vectors b_i, b_f, b_o, b_c are the corresponding bias vectors.

4 EXPERIMENT EVALUATION

4.1 Experiment setup

In this section, we conduct experiments to evaluate the effectiveness of using LSTM-based RNNs for forecasting product quality. We compare the performance of our proposed LSTM-based RNN scheme against several other methods: the SARIMA model, which is a Seasonal Autoregressive Integrated Moving Average model; the SVR model, which is a widely used model in financial time series forecasting; and CNN, which is a Convolutional Neural Network model. We use two evaluation metrics to measure performance: root mean square error (RMSE) and mean absolute percentage error (MAPE) and R2 score which is calculated by comparing the forecasting results $\hat{y} = \{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_H\}$ to the actual values $y = \{y_1, y_2, \ldots, y_H\}$. Specifically,

$$RMSE = \sqrt{\frac{1}{H}\sum_{1}^{H}(y_i - \hat{y}_i)^2} \tag{13}$$

$$MAPE = \frac{100\%}{H}\sum_{1}^{H}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{14}$$

In order to quantify the proportion of variance in the target that is explained by the forecasting methods, we also consider the R^2 index:

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}, \tag{15}$$

Where H is the forecasting horizon and $\bar{y}$ is the mean value of $\mathbf{y} = y_1, y_2, ..., y_H$. To ensure fairness in the comparison, we evaluated the performance of the LSTM-based RNN scheme against other methods using two different datasets: a product quality dataset sourced from Kaggle and a widely used airline passenger dataset. The airline passenger dataset includes 144 observations spanning 12 years and displays a clear upward trend and strong seasonal fluctuations, making it useful for assessing the scheme's ability to predict short time series with multiple seasonal patterns.

In our research, data was gathered from the quality control process of a roasting machine. The machine is composed of five identical chambers, each with three temperature sensors. The data also includes readings of the height and moisture content of the raw materials at the time of entry into the machine. The raw materials take one hour to pass through the kiln. The quality of the final product is evaluated in the lab by taking samples every hour, which is given a score between 200 and 500, with higher scores indicating better quality. The dataset of product quality has 744 observations over 31 days and exhibits distinct variations in quality characteristics.

4.2 Experiment Results with the Airline Data Set

In this study, we used a forecasting horizon of 12 for experiments on the international airline passenger dataset. From the results shown in Fig. 3 and Fig. 4, it can be observed that all four methods were able to capture the upper trend and seasonal patterns to varying degrees. However, it was found that LSTM, CNN, and SARIMA performed better than SVR, as reflected in their smaller root mean square error (RMSE) and mean absolute percentage error (MAPE) values in Table 1. The CNN model is the best model with the capability to explain the proportion of variance in the target because of the highest R2 score (Shown in Fig. 5).

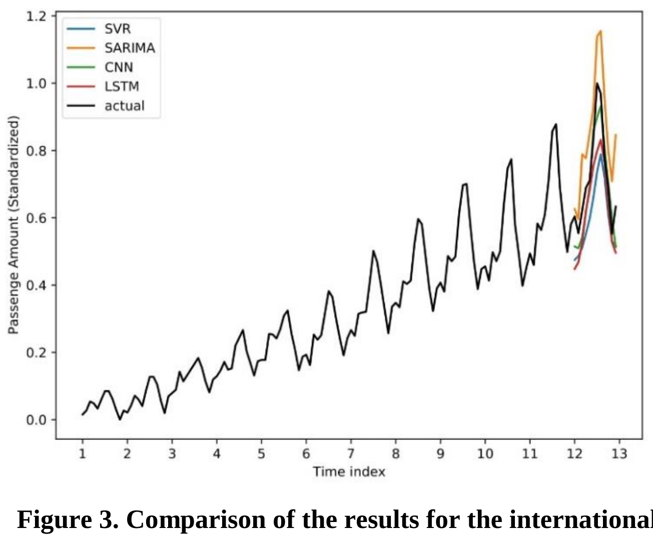

Figure 3. Comparison of the results for the international airline passenger data set both within and outside of the sample

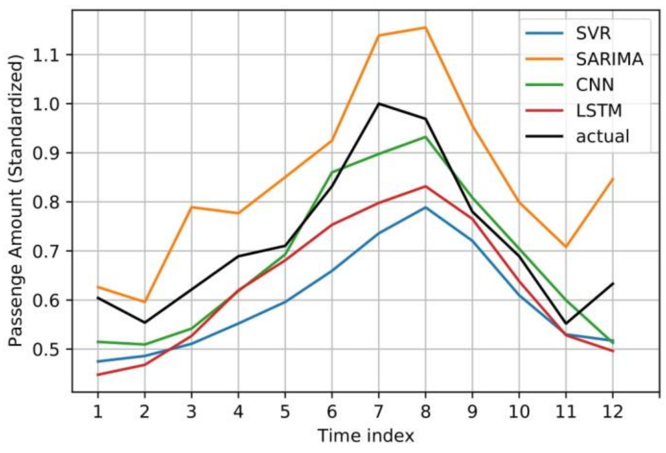

Figure 4. Comparison of forecasting performance for the international airline passenger data set.

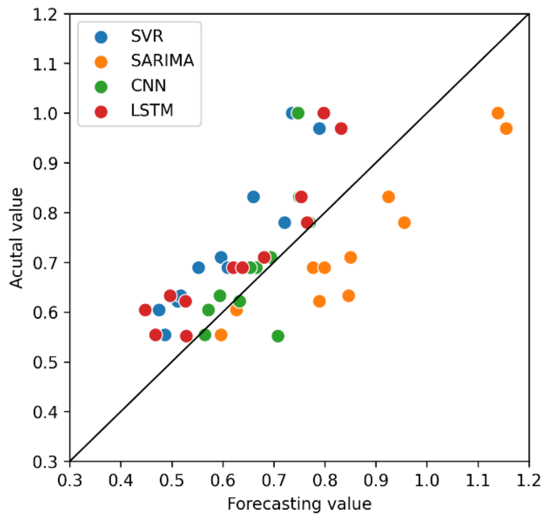

Figure 5. Predicted and actual value of the forecasting methods for international airline passenger data set.

Table 1. Forecasting RMSE, MAPE, and R2 scores for the Airline Passenger Data Set

Forecasting method	RMSE	MAPE	R2
SVR	0.1565	0.1749	0.0970
SARIMA	0.1393	0.1703	0.0499
CNN	0.1659	0.1921	**0.5184**
LSTM	**0.1059**	**0.1509**	0.4508

4.3 Experiment Results with the Product Quality Data Set

In our experiments, we utilized 720 observations of product quality data to predict future quality over a 24-step horizon. This means we used the product quality data of the previous 30 days to forecast the quality of the following day. As shown in Fig. 5, the product-quality-score time series is more complicated compared to the airline passenger time series, lacking any discernible seasonal patterns or trends. This non-stationarity and non-seasonality present difficulties for conventional forecasting methods. Additionally, the longer forecasting horizon of 24 steps makes accurate predictions even more challenging.

Table 2. RMSE, MAPE, and R2 score results for the Product Quality Data Set

Forecasting method	RMSE	MAPE	R2
SVR	0.1660	0.2220	0.2650
SARIMA	0.1918	0.3550	0.1227
CNN	0.1509	**0.1820**	**0.4640**
LSTM	**0.1280**	0.2320	0.2261

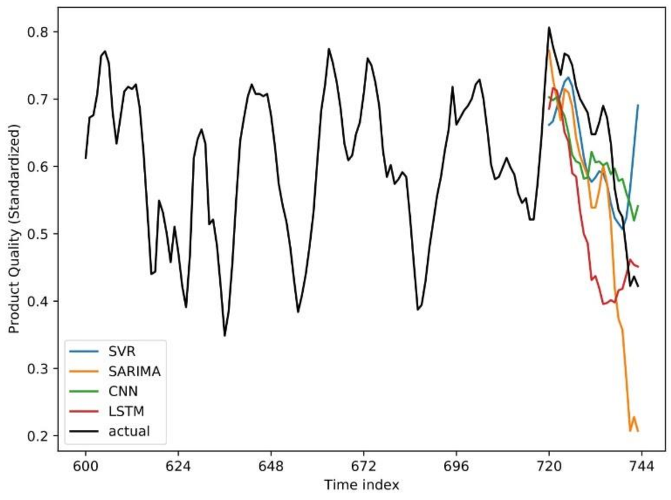

Figure 6. Comparison of in-sample and out-of-sample performance for the product quality dataset

The results of the experiments on the product quality data set, as shown in Fig. 5, indicate that LSTM performed well in comparison to the original time series. Fig. 6 further highlights that LSTM outperformed the other methods by producing the most accurate forecast. Table 2 confirms this, with LSTM having the smallest forecasting errors. In the complex scenario of forecasting product quality, the performance of the other four methods was generally inadequate. The CNN method was able to capture the general trend of the real-time series, but the forecasted results were inaccurate and resulted in large RMSE and MAPE errors. The SARIMA method was not successful in this extended horizon forecasting problem due to the non-stationarity and non-seasonality of the product quality time series. Furthermore, the CNN model is the best model with the capability to explain the proportion of variance in the target because of the highest R2 score (Shown in Fig. 8).

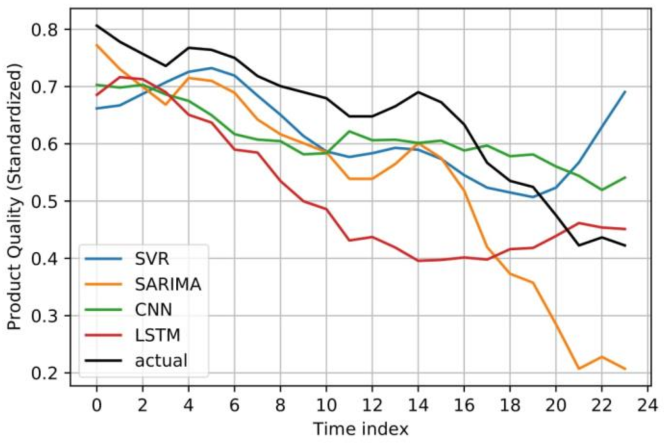

Figure 7. Comparison of forecasting results for the product quality data set.

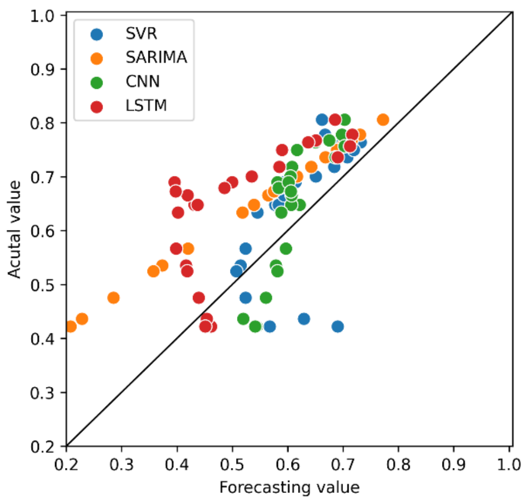

Figure 8. Predicted and actual value of the forecasting methods for international airline passenger data set.

5 Conclusions

In summary, this study suggests the implementation of a Long-Short-Term-Memory (LSTM) based Recurrent Neural Network (RNN) to effectively forecast short-term product quality. By utilizing, the long-term dependencies present in the time series data, LSTM proves to be capable of accurately predicting complex and non-stationary product quality patterns. The proposed method was tested using both a benchmark international airline passenger dataset and a more comprehensive product quality roasting process dataset, with results demonstrating its superiority over traditional forecasting techniques in the challenging task of forecasting short-term product quality.

6 References

An, N. H., & Anh, D. T. (2015). Comparison of strategies for multi-step-ahead prediction of time series using neural network. *2015 International Conference on Advanced Computing and Applications (ACOMP)*, 142–149.

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, *5*(2), 157–166.

Bontempi, G. (2008). Long term time series prediction with multi-input multi-output local learning. *Proc. 2nd ESTSP*, 145–154.

Cox, D. R. (1961). Prediction by exponentially weighted moving averages and related methods. *Journal of the Royal Statistical Society: Series B (Methodological)*, *23*(2), 414–422.

Deutsch, J., & He, D. (2017). Using deep learning-based approach to predict remaining useful life of rotating components. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *48*(1), 11–20.

du Preez, J., & Witt, S. F. (2003). Univariate versus multivariate time series forecasting: an application to international tourism demand. *International Journal of Forecasting*, *19*(3), 435–451.

Graves, A., Jaitly, N., & Mohamed, A. (2013). Hybrid speech recognition with deep bidirectional LSTM. *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, 273–278.

Hamzaçebi, C., Akay, D., & Kutay, F. (2009). Comparison of direct and iterative artificial neural network forecast approaches in multi-periodic time series forecasting. *Expert Systems with Applications*, *36*(2), 3839–3844.

He, Y., Han, X., Gu, C., & Chen, Z. (2018). Cost-oriented predictive maintenance based on mission reliability state for cyber manufacturing systems. *Advances in Mechanical Engineering*, *10*(1), 1687814017751467.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Janssens, O., Slavkovikj, V., Vervisch, B., Stockman, K., Loccufier, M., Verstockt, S., de Walle, R., & van Hoecke, S. (2016). Convolutional neural network based fault detection for rotating machinery. *Journal of Sound and Vibration*, *377*, 331–345.

Kahraman, C., Yavuz, M., & Kaya, \.Ihsan. (2010). Fuzzy and grey forecasting techniques and their applications in production systems. In *Production engineering and management under fuzziness* (pp. 1–24). Springer.

Lipton, Z. C., Berkowitz, J., & Elkan, C. (2015). A critical review of recurrent neural networks for sequence learning. *ArXiv Preprint ArXiv:1506.00019*.

Medsker, L., & Jain, L. C. (1999). *Recurrent neural networks: design and applications*. CRC press.

Mobley, R. K. (2002). *An introduction to predictive maintenance*. Elsevier.

Mourtzis, D., & Vlachou, E. (2018). A cloud-based cyber-physical system for adaptive shop-floor scheduling and condition-based maintenance. *Journal of Manufacturing Systems*, *47*, 179–198.

Pascanu, R., Mikolov, T., & Bengio, Y. (2013). On the difficulty of training recurrent neural networks. *International Conference on Machine Learning*, 1310–1318.

Peng, Y., Dong, M., & Zuo, M. J. (2010). Current status of machine prognostics in condition-based maintenance: a review. *The International Journal of Advanced Manufacturing Technology*, *50*(1), 297–313.

Rüßmann, M., Lorenz, M., Gerbert, P., Waldner, M., Justus, J., Engel, P., & Harnisch, M. (2015). Industry 4.0: The future of productivity and growth in manufacturing industries. *Boston Consulting Group*, *9*(1), 54–89.

Sak, H., Senior, A. W., & Beaufays, F. (2014). *Long short-term memory recurrent neural network architectures for large scale acoustic modeling*.

Scalabrini Sampaio, G., Vallim Filho, A. R. de A., da Silva, L., & da Silva, L. (2019). Prediction of motor failure time using an artificial neural network. *Sensors*, *19*(19), 4342.

Sutskever, I., Vinyals, O., & Le, Q. v. (2014). Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems*, *27*.

Taieb, S. ben, Bontempi, G., Atiya, A. F., & Sorjamaa, A. (2012). A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition. *Expert Systems with Applications*, *39*(8), 7067–7083.

Taieb, S. ben, Sorjamaa, A., & Bontempi, G. (2010). Multiple-output modeling for multi-step-ahead time series forecasting. *Neurocomputing*, *73*(10–12), 1950–1957.

Vu, H.-C., Grall, A., Fouladirad, M., & Huynh, K. T. (2021). A predictive maintenance policy considering the market price volatility for deteriorating systems. *Computers & Industrial Engineering*, *162*, 107686.

Wu, J.-Y., Wu, M., Chen, Z., Li, X., & Yan, R. (2021). A joint classification-regression method for multi-stage remaining useful life prediction. *Journal of Manufacturing Systems*, *58*, 109–119.

Zhu, J., & Chen, J. C. (2006). Fuzzy neural network-based in-process mixed material-caused flash prediction (FNN-IPMFP) in injection molding operations. *The International Journal of Advanced Manufacturing Technology*, *29*(3), 308–316.