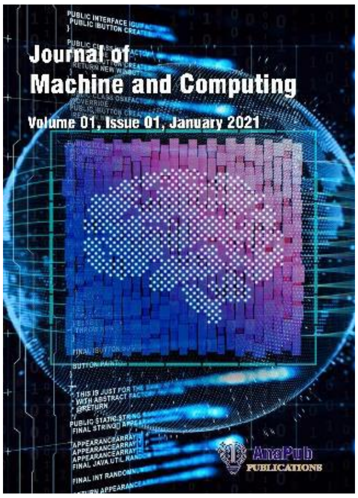
Journal Pre-proof



Convolutional LSTM Neural Network Autoencoder Based Fault Detection in Manufacturing Predictive Maintenance

Young Jun Park

DOI: 10.53759/7669/jmc202505072
Reference: JMC202505072
Journal: Journal of Machine and Computing.

Received 22 April 2024
Revised form 10 July 2024
Accepted 18 December 2024

Please cite this article as: Young Jun Park, "Convolutional LSTM Neural Network Autoencoder Based Fault Detection in Manufacturing Predictive Maintenance", Journal of Machine and Computing. (2025). Doi: https:// doi.org/10.53759/7669/jmc202505072

Convolutional LSTM neural network Autoencoder based fault detection in manufacturing predictive maintenance

Young Jun Park (Corresponding Author)
Department of Game, Dong-Seo University Global village building 6th floor, Dongseo University, 47 Jurye-ro, Sasang-gu, Busan, South Korea.
E-mail: young@dongseo.ac.kr ORCID ID: 0009-0001-4968-5667

Abstract

The smart manufacturing has revolutionised the intelligent predictive maintenance by integrating IoT technologies with big data analytics, artificial intelligence, cloud computing and other evolving technologies. An effective predictive maintenance demands not only measuring equipment, but the underlying ecosystem that starts with data acquisition from sensors and propagates all the way to visualisation on engineer friendly dashboards. For process monitoring and performance optimization in a smart factory, it is important to recognise time series events like equipment peaks, changeovers and failures. In this article, a model proposed is a deep convolutional LSTM autoencoder architecture using an autoencoder approach to classify real world machine and sensor data to condition based label. The proposed model outperformed baseline architectures. A window size of 45 was used to determine that the model produced a RMSE of 58.4[illegible], and MAE of 22.48, and a sMAPE of 0.869, most of which represents significant improvements of up to 37% over existing methods. Having a window size 90, it remained on top with an RMSE score of 72.16 and MAE of 29.64 and sMAPE of 0.847. These results show that it processed a real world manufacturing data and correctly estimated RUL and its complete predictive maintenance.

1. Introduction

The ICT system used during manufacturing, as well as IoT devices placed on the floor, however, can generate large volumes of data that is often underutilised. To fully exploit the benefits of these technologies, they need innovative methods and models to optimise these ICT technology solutions and improve production processes well [1].

The application of cyber physical systems (CPS), closed loop control systems consisting of physical assets coupled with software modules, is a transformative approach to manufacturing. Consequently, the idea of cyber physical production systems (PPSP) is created. CPPSs differ from the traditional automation pyramid by using distributed and interconnected systems to perform diverse manufacturing tasks, which employs hierarchical control [2]. They consist of basic monitoring up to advanced planning, controlling, and real time reconfiguration of production systems. Nikolakis et al. have proposed the use of containerization technologies for enacting control of CPPSs [3].

As sensor based models have evolved, the resultant volumes of digital data are now substantial. This data, when analysed, can supply information along with hidden patterns, which human eyes may fail to perceive, and can help with proactive decision making [4]. Specifically, data driven techniques can give us actionable production equipment operational condition insights, which when used effectively can enable condition monitoring. These insights have been studied by Entezami et al. and Chuang et al. and demonstrated a way to change traditional preventive to predictive maintenance. It can help to significantly reduce maintenance cost as well as increase production efficiency by assessing equipment conditions and estimating equipment remaining useful life (RUL) [5, 6].

AI and ML techniques are important tools in the process of leveraging large scale data to do predictive analytics [7]. This article presented a novel approach to deep learning prediction and fault detection. Anomaly detection is used in this approach to map reconstruction errors to different RUL values. In contrast to previous CNNs, a unique aspect of the approach lies in its separate training of neural networks for each of the health condition labels to enable tailored and precise classification of a new input. It allows independence of the model to different types of machines and labels. The proposed technique is validated using historical maintenance dataset from industrial environment. The method is demonstrated to be theoretically and practically practicable by development of a prototype software system that can provide the operator with reliable health assessments of machinery without requiring specialised expertise from the operator.

2. Literature Review

In the last decade research in industrial equipment condition monitoring has gained momentum due to the impressive need to enhance reliability and efficiency of industrial processes. Data analysis that can detect abnormalities and predict the requirement for maintenance has proven useful in many applications using advancements in AI and ML allowing for predictive analytics. Unlike the traditional methods, these data driven approaches do not need a deep understanding of underlying processes. To train models, they use run to failure data and their maintenance systems work independent of domain expertise [8, 9].

Over the years various ML techniques for predictive maintenance have been explored. For example, Carvalho et al. and Zonta et al. review exhaustive views on methods from artificial neural networks and Bayesian networks to sequence-to-sequence models [10, 11]. Eventually, artificial neural networks are combined with data mining tools, which are useful for handling large amounts of data; however, they usually run into troubles when processing time and computational complexity are concerned. The effectiveness of probabilistic fault diagnosis with Bayesian networks is limited by scalability in dealing with extensive data sets. Transformer based models have recently shown their potential for sequence learning tasks, including in time series forecasting as shown by Wu et al., and Vaswani et al. [12, 13].

Due to their efficient processing of multi-channel sensor data, Convolutional Neural Networks (CNNs) have promise for fault diagnosis. However, their use is often limited in diverse real-world scenarios, due to their reliance on homogeneity of the input data. This limitation of spatial heterogeneity to address the lack of a framework to cluster distributed and heterogeneous datasets was encountered and addressed by Chen and Huang by proposing a Double Deep Autoencoder structure [14] that was suitable for CNN based analysis. At the same time, Recurrent Neural Networks (RNNs) have been popular for analysing time series data, as they can learn sequential dependencies. Although effective, traditional RNNs struggle with problems like vanishing gradients, impeding their ability to perform long term forecasting [15].

To address these problems, LSTM networks, a variant of RNNs, come up that present memory cells. Within these memory cells are three gates – input, output, and forget – controlled by input, output, and forget 'ground truth', which dynamically decide what to store or forget during the training process. There are several studies that proved LSTM's effectiveness when used in applications like traffic flow prediction or even stock market forecasting, compared to conventional RNNs and even other deep learning models [16, 17]. Despite this, LSTM networks can be sensitive to dataset changes and fine tuning of such hyperparameters can necessitate careful optimization for a specific application [18].

To optimise the skills of LSTMs, LSTM-autoencoders have been developed. These models incorporate both temporal feature extraction of LSTMs, and dimensionality reduction and reconstruction capabilities of autoencoders. Recently, they have been successfully applied to time series prediction tasks, including industrial estimation of remaining useful life (RUL) and univariate sequence forecasting. The research suggests that LSTM-autoencoders result in higher predictive accuracy and lower computational costs [19, 20].

Motivated by these advances, convolutional LSTM (ConvLSTM) networks were proposed to benefit from strengths of CNNs and LSTMs, hybrid architectures. Both ConvLSTM models can preserve spatial information and well capture the sequential pattern, which makes them suitable for time series forecasting in manufacturing processes. The effectiveness of the ConvLSTM based methods in multistep forecasting in industrial operations has been demonstrated by Zhang et al., and their application illustrated by optimising production schedules using machine performance metrics forecasted by this method [21].

In smart manufacturing, especially in the case of complex machinery for example for metal can production systems, there is a need for development of robust predictive models. For example, a hybrid of ConvLSTM architectures is applied to the problem of internal speed forecasting of high speed bodymaker machines, to adjust both upstream and downstream processes in real time. The effectiveness of ConvLSTM models for improving predictive accuracy with computational efficiency on real world datasets is further substantiated by experimental studies [22].

Moreover, the performance of deep learning models for predictive maintenance has benefited greatly from recent developments in hyper parameter optimization and model design. Work is done on balancing the network complexity with the size of dataset to get the best results. The ConvLSTM autoencoders have been unified with techniques such as sliding window strategies, bidirectional LSTM layers, and supervised learning to improve their generalisation capability for a variety of industrial applications.

3. Materials And Methods

For fault detection in manufacturing predictive maintenance through capture of speed produced by machines to predict such anomalies that are precursors to failure. This is cast as a sequence-to-sequence time-series anomaly detection problem: to learn a model of normal machine speed behaviour and detect deviations from this model as potential faults.

The input speed time series is used to reconstruct a portion of the original time series, which is used as input features to the model. This window size is known as the window size(w).In case of a univariate time series of speed data s(t) $=\{s_1,s_2,s_3,\ldots, s_\tau\}$, where vi is the speed amplitude recorded at time ii, the aim of developing a fault detection model is to predict the subsequent values of the speed, $\hat{s} = \{s_{(\tau+1)},s_{(\tau+2)},\ldots, s_{(\tau+m)}\}$, based on previous observations in a size w sliding window. This is formulated as:

$$\hat{s} = (\hat{s}_1, \hat{s}_2, \ldots . \hat{s}_\tau) \downarrow f(s_{\tau-\omega}, s_{\tau-\omega+1}, \ldots, s_\tau) \quad (1)$$

Here f(·) denote the model predicts by an Convolutional LSTM Autoencoder, which learns how to predict the future speed signals.

Unlike traditional single step forecast, the sequence to sequence fault detection model forecasts a sequence of future speed values instead of single point estimate, allowing to detect temporal patterns in the data, which lie outside normal behaviour.

The time series of length N is transformed to a sequence-to-sequence matrix. All samples generated from sliding-window approach, training samples forms the vector w, with length l equal to n.

$$n = (N-w-l+1) \quad (2)$$

3.1.Objective Function

The objective function L\mathcal{L} to minimize during training is the Mean Squared Error (MSE) between the predicted and actual speed sequences, which can be expressed as:

$$\delta = \frac{1}{nl}\sum_{i=1}^{n}\sum_{j=1}^{l}\left(\hat{s}_{i,j} - s_{i,j}\right)^2 \quad (3)$$

where:

- l is the length of the predicted sequence and n represents training data.
- $\hat{s}_{i,j} - s_{i,j}$ is the difference between predicted and actual values.

The resultant objective function promotes that the model learns to minimise the difference between predicted speed sequence and actual speed sequence. If the reconstruction error exceeds a predefined threshold, anomalies are detected as hints of machine faults.

3.2.Structure of LSTM

LSTM neural networks address the stability constraint that conventional RNNs encounter via gating functions in state dynamics, making them superior to conventional RNNs intended for sequence prediction. Each of the layers in the LSTM network corresponds to a number of memory blocks. Three multiplicative units—the input, forget, and output gates—as well as a collection of recurrently linked memory cells make up each layer. The input gate gives the cell state information in three phases. The sigmoid function, which controls input values added to the cell state, is the first. Second, the hyperbolic tan function is used to construct a vector that contains every value that might be added to the cell state. Lastly, it is added to the cell state information after being multiplied by the newly formed vector. The forget gate eliminates information that is no longer required for processing or that is of lower significance by using a multiplying philtre. Based on the condition of the current cell, the output gate determines what should be shown at the output. Figure 1 depicts the construction of an LSTM cell as well as the architecture of the LSTM structure.

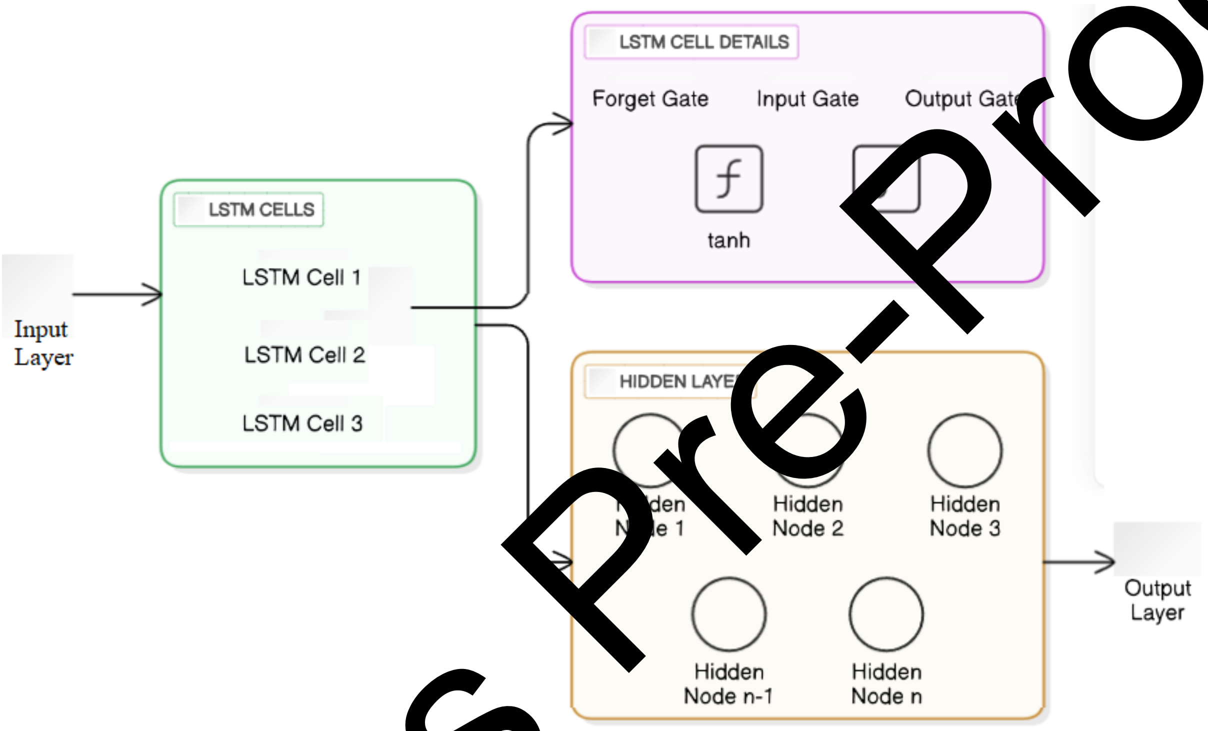


Figure 1. Layout of LSTM Structure

Let the input information given is $x=(x_1,x_2,\ldots,x_{t-1},x_t)$. The chain of the memory cell is updated with the help of the gates of LSTM and the target variables $y = (y_1, y_2, \ldots, y_t)$ are predicted from $t = 1$ to T. Based in part on some mathematical equations for the LSTM:

$$i_t = \sigma\left(w_i x_t + R_i h_{\{t-1\}} + b_i\right) \tag{4}$$

$$f_t = \sigma\left(w_f x_t + R_f h_{\{t-1\}} + b_f\right) \tag{5}$$

$$y_t = \sigma\left(w_y x_t + R_y h_{\{t-1\}} + b_y\right) \tag{6}$$

$$c_t = f_t c_{\{t-1\}} + i_t \{c\}_t \tag{7}$$

$$\{c\}_t = \sigma\left(w_c x_t + R_c h_{\{t-1\}} + b_c\right) \tag{8}$$

$$h_t = y_t \sigma(c_t) \tag{9}$$

x_t represents the input vector, whereas w_i, w_f, and w_y denote the weight matrices for the input, forget, and output gates, respectively. R_i, R_f, and R_y represent the input, forget, and output gates of the input weight matrix, while b_i, b_f, and b_y denote the input, forget, and output gate bias vectors, respectively, with h_t signifying the output vector.

3.3.Structure of CNN

The CNN is provided with an input of three dimensions represented as height, weight and number of channels. Let the input x = $(x_t)^{(N-1)}$ of size N in the first layer ('0' padding), convolution of this input with a series of M1 creates the feature output map., w_h^l for h = 1,….,M_1.

$$a^{l(i,h)} = \left(w_h^l * x\right)(i) = \sum_{\{j=-\infty\}}^{\{\infty\}} w_h^{l(j)} \times (i-j) \tag{10}$$

Let w_h^l belong to $R^{1\times k\times 1}$ and a^l belong to $R^{1\times N\text{-}k+1\times M}$. This consists of the fully connected layer within the hidden layer, the pooling layer, and the convolutional layer. The convolutional layer automatically extracts features from various regions of the raw data or the intermediate feature maps using learnable phitres. While certain neurons in one convolution layer may link to every neuron in this layer, others might only connect to specific neurons in the layer below. The philtre then performs the convolutional function using a shared weight matrix. Weight changes during training are well recognized. The pooling layer creates a single value from all of the data in the pooling region. Max pooling is a modification that reduces the size of an input layer by selecting the highest value in each subarea of the preceding layer. This layer also solves the overfitting issue and speeds up the learning computation process.

$$a^{l(i,h)} = \left(w_h^l * f^{\{l-1\}}\right)(i) = \sum_{\{j=-\infty\}}^{\{\infty\}} w_h^{l(j,m)} f^{\{l-1\}(i-j,m)} \tag{10}$$

The fully connected layer converts convolutional features into a format for output, passing through non-linear activation functions. The final output is a matrix influenced by filter dimensions and filters. Figure 2 illustrates the CNN model's architecture.

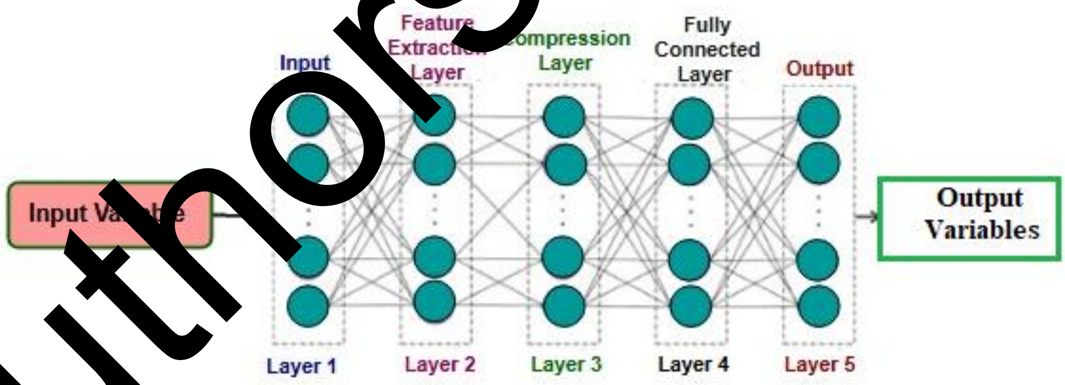


Figure 2. Layout of CNN Structure

3.4.Structure of hybrid CNN-LSTM

Each CNN and LSTM model has specialised characteristics. Development of a hybrid CNN–LSTM DL model for predicting water quality variables is carried out in this study by exploiting the strengths of the proposed system. The CNN–LSTM structure started with CNN layers on top. CNN contains numerous hidden layers and a variable-input layer. It delivers features to LSTM cells via an output layer. Buried layers include the convolution, activation function, and pooling layers. LSTM output passes into fully linked layer. CNN layers may learn consecutive input characteristics, unlike some neural networks. By guessing target values using long-range relationships, the bottom LSTM layer integrates these properties. Figure 3 shows the CNN–LSTM model's topology.

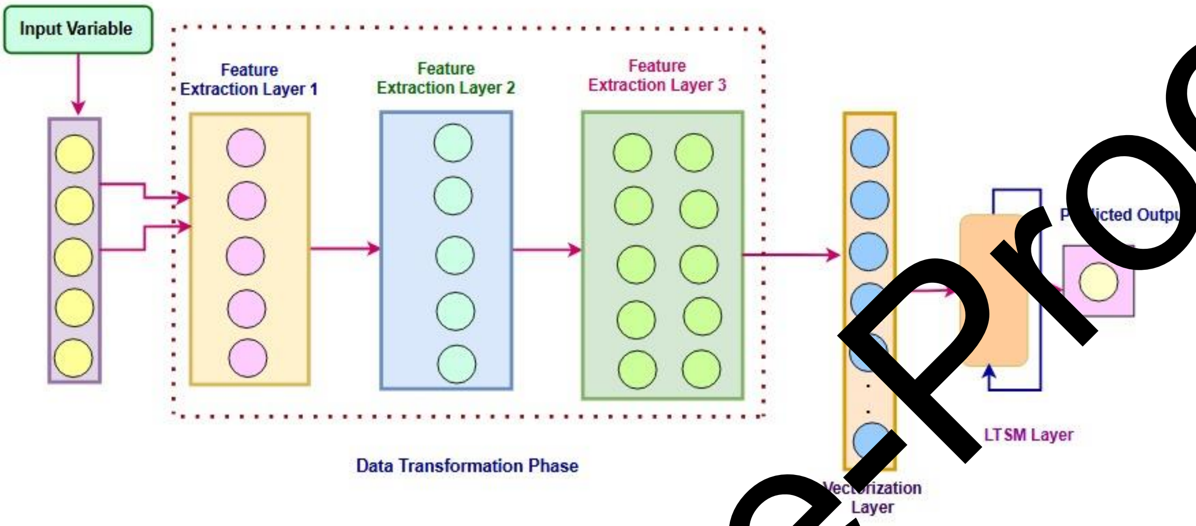


Figure 3. Layout of CNN–LSTM Hybrid Structure

3.5. Autoencoder

Since it is a feedforward neural network, the input and output are identical. Autoencoders don't need to label target datasets since they learn the features from the data they receive in autonomous learning. Three primary components comprise the autoencoder: Three things make up the encoder, the code, and the decoder. The decoder decodes the encoded input into outputs after the encoder compresses it into a "code." Time series forecasting maps data into the hidden layer of lesser dimensions using the autoencoder. Like a standard autoencoder, a stacked autoencoder learns without supervision. In order to train a model, greedy layer-wise training reduces input-output vector error. Each layer is trained using the gradient descent approach and an optimization function; the subsequent layer of the autoencoder is the hidden layer of the preceding layer.

4. Experiments and Results

The dataset of historical, machine-collected, speed data collected at a frequency of 100 Hz is used which is depicted in figure 4. Data are the operational speed of a vertical form fill seal (VFFS) machine in terms of acceleration strokes/min and were logged internally by a monitoring system built into the machine. In food packaging industry, this VFFS packing machine is used to quickly pack quality products without human error in filling and automatically gripping products while sealing them. Typically, the operational speed of the VFFS machine is composed by a mixture of periodic pattern related to "normal" production cycles such as rotation of shifts and episodic, sporadic pattern associated with abnormal operations. Additionally, this machine impacts (and is impacted by) upstream and downstream processes, such as product feeding (upstream) and sealing/labelling (downstream) machines. In this article, the dataset contained 1,440,000 observations of speed data collected over a continuous period of 24 hours. Of this dataset, 1,152,000 observations were used for model training, 144,000 for testing and 144,000 for validation observations.

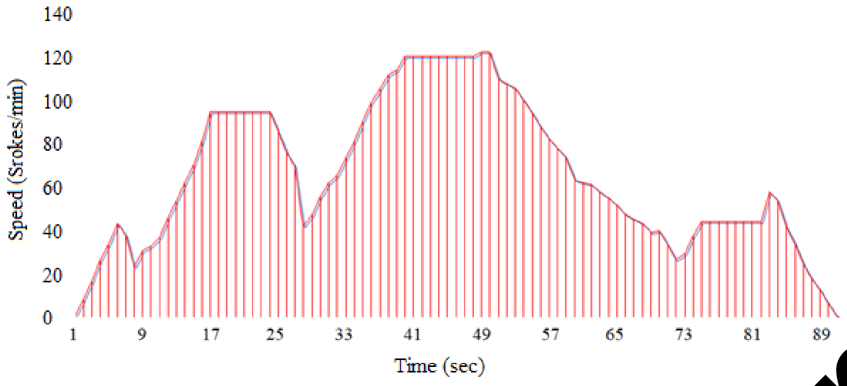


Figure 4. Time series data

Results from the evaluation of the proposed deep convolutional LSTM encoder-decoder model demonstrate the model's performance advantage over three baselines. Results are shown in using metrics like Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) and Symmetric Mean Absolute Percentage Error (sMAPE) to show that the proposed approach is effective in time-series event classification. The performance was evaluated using two window sizes (45 and 90) to indicate that the proposed model could adapt to different temporal constructions.

Table 1. Performance analysis for a window size of 45

Structure	**RsNet(a)**	**CNN-LSTM Encoder-Decoder (b)**	**LSTM Encoder-Decoder (c)**	**Proposed (d)**
RMSE	140.97	76.93	116.94	58.45
MAE	136.10	35.65	94.49	22.48
sMAPE	3.894	1.02	2.704	0.869

At a window size of 45, the proposed model excels all of its counterparts in terms of all metrics. We get an RMSE of 58.45, much less than RsNet (140.97), CNN–LSTM Encoder–Decoder (76.93) and LSTM Encoder–Decoder (116.94). This result demonstrates that the model is able to capture complex temporal relations to high accuracy. Equally, the MAE of 22.48 in the proposed model was 37% higher than CNN-LSTM Encoder-Decoder's (35.65), and its performance was drastically better compared to RsNet (136.10) or LSTM Encoder-Decoder (94.49). Further validation of the model's precision at minimising relative prediction errors was illustrated by the lowest sMAPE value of 0. 869 in supporting its application to real world industrial applications.

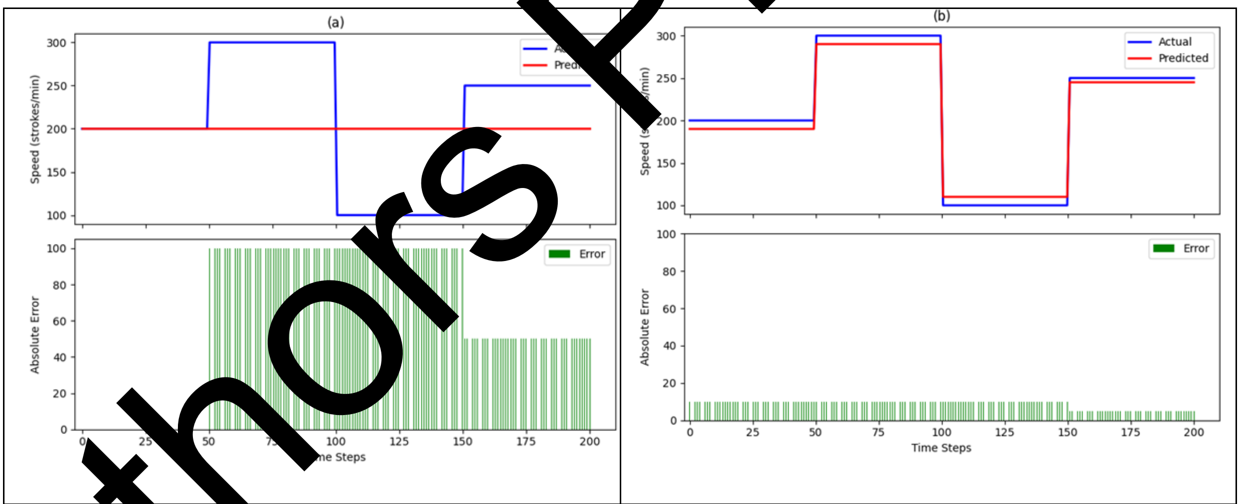
Table 2. Performance analysis for a window size of 90

Structure	**RsNet(a)**	**CNN-LSTM Encoder-Decoder (b)**	**LSTM Encoder-Decoder (c)**	**Proposed (d)**

RMSE	146.24	88.54	111.27	72.16
MAE	134.25	41.03	89.91	29.64
sMAPE	3.881	1.17	2.561	0.847

The proposed model retained its performance superiority for a window size of 90. RMSE values increased slightly with the extended prediction horizon, but the proposed model achieved an RMSE value of 72.16, which significantly outperforms RsNet (109.24), CNN LSTM Encoder Decoder (88.54), and LSTM Encoder Decoder (111.27). The proposed model demonstrates reduced MAE of 29.64 compared to that of CNNLSTM Encoder Decoder (41.03) and LSTM Encoder Decoder (89.91). The robustness and reliability of the model was also demonstrated by a sMAPE of 0.847, which reflects the best compromise between bias and variability in predictions.

This is followed by a comparative analysis illustrated in figures 5 and 6, which identifies crucial insights regarding how the predictive accuracy of the models varies with different conditions. While being computationally efficient, RsNet faced high prediction errors for which it proved insufficient for modelling intricate temporal dependencies. Building upon RsNet, CNN-LSTM Encoder-Decoder utilises convolutional features, however, could not efficiently deal with long term temporal relationship. To address this challenge, LSTM Encoder-Decoder was employed but it did not possess the advanced feature extraction abilities like that present in the proposed method. It is shown that these results support the superiority of the proposed architecture in balancing feature extraction and temporal pattern recognition.



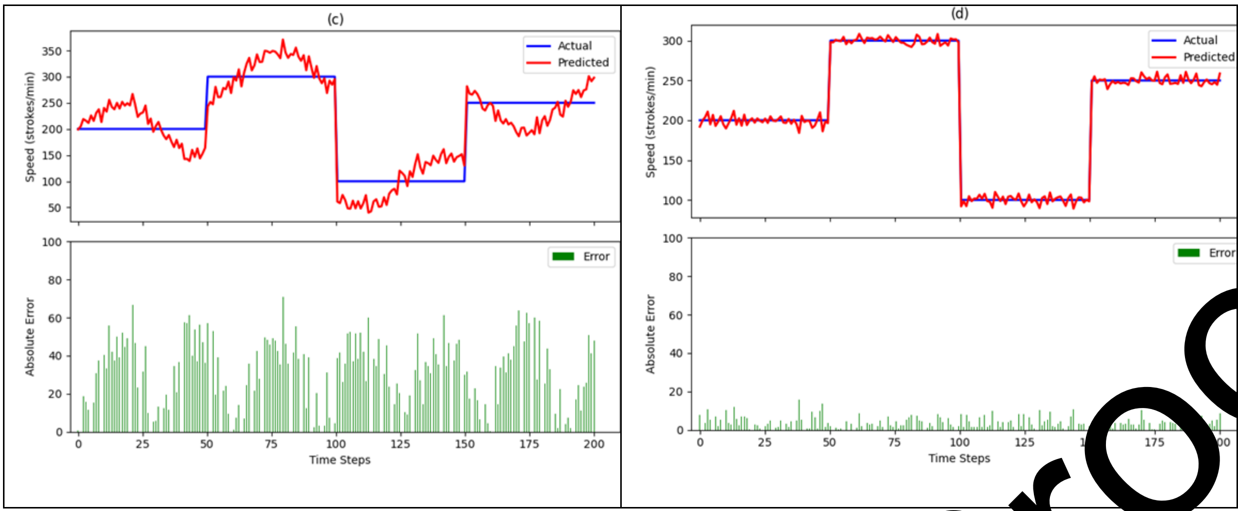


Figure 5. Predictive performance of (a) RsNet, (b) CNN-LSTM Encoder-Decoder, (c) LSTM Encoder-Decoder, (d) Proposed. (Window Size – 45)

The implications are practical and multiple, especially for industrial settings such as steel industry, where the forecast of Remaining Useful Life (RUL) is of paramount significance to minimise unexpected downtimes. The low RMSE, MAE, and sMAPE values confirm that deploying sensor data with deep learning proves effective in predicting future maintenance activity. The model makes daily monitoring of real-time possible and ensures industrial equipment reliability by streamlining predictive maintenance strategies.

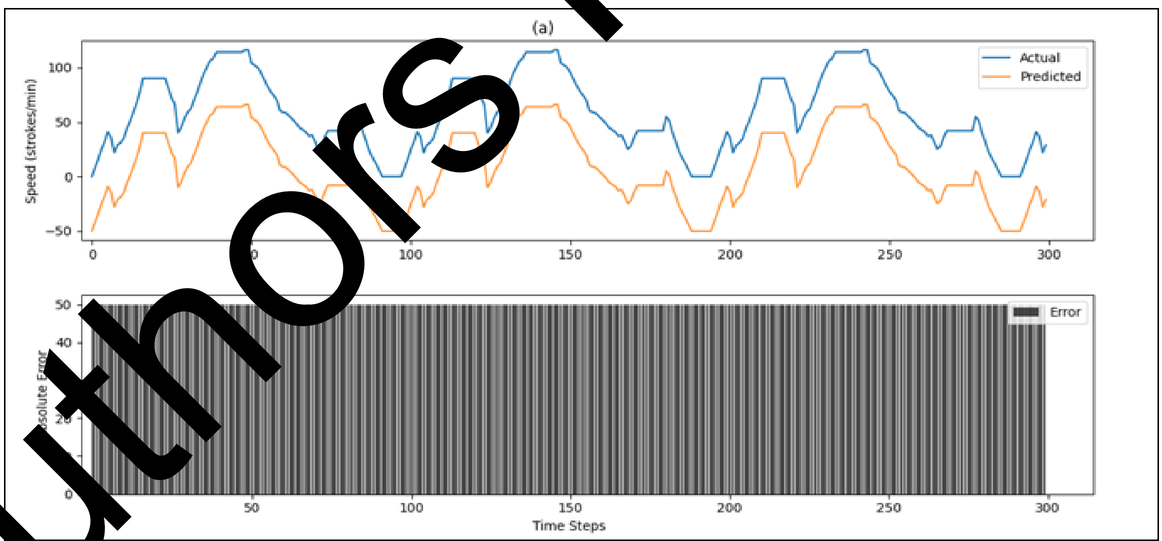


Figure 6. Predictive performance of (a) RsNet, (b) CNN-LSTM Encoder-Decoder, (c) LSTM Encoder-Decoder, (d) Proposed. (Window Size – 90)

The benefit of the proposed architecture is its ability to consider both convolutional and temporal features. Convolutions on the other hand, help extract features by capturing localised patterns in the image, and the LSTM's handle sequential dependencies to let it understand temporal dynamics well enough. In addition, dimensionality reduction performed by the autoencoder reduces noise present in the high dimensional input data whilst preserving important temporal patterns, hence giving predicted results that are more stable. In addition, the performance of the model is demonstrated to maintain consistency across different window sizes, validating its viability in a wide range of predictive maintenance scenarios.

However, the model does have limitations which suggest areas for future work. The model could be made more precise in its prediction if attention mechanisms were incorporated to focus on important time series events. Furthermore, transfer learning approaches might be studied to adapt the model for the case in which there is limited labelled data in the target industry. Another area for improvement is optimization of computational efficiency, especially for real time deployment on edge devices.

5. Conclusion

In this study, a deep convolutional LSTM autoencoder model as a complete solution to predictive maintenance via advanced time series classification and RUL estimation is proposed. The model achieved better performance than baseline architectures in all metrics and proved to be robust and adaptable. At a window size of 45 the proposed model showcased improvements in its performance, with RMSE of [illegible]8.45, MAE of 22.48, and sMAPE of 0.869. Also, when window size was 90, the RMSE was 72.16, MAE was 29.64, and sMAPE was 0.847, still the best within higher prediction horizons. Its improved performance resulted from the use of convolutional and temporal features, along with an autoencoder based approach to dimensionality reduction. The model has been validated against steel industry real world manufacturing data, which successfully tackled critical issues in predictive maintenance like accurate event detection and RUL estimation. As such, its practical use emphasises the relevance to reduce equipment downtime and enhance operational efficiency in smart factories. The proposed model represents a substantial advance, but still invites improvement. Adding in attention mechanisms could make it more focused on important time series events, and thus improve the precision of prediction. Adapting in industries with limited labelled data could be made possible through transfer learning techniques. Moreover, the computational efficiency of the model will be optimised, allowing for real time deployment on edge devices, thus enabling access to a large set of IIoT applications. These areas can then be addressed by extending the proposed framework to a broader set of industrial domains to facilitate the adoption of predictive maintenance within different ecosystem strategies for smart manufacturing.

Reference

[1] M. A. Filz, J. P. Bosse, and C. Herrmann, “Digitalization platform for data-driven quality management in multi-stage manufacturing systems,” *J. Intell. Manuf.*, vol. 35, no. 6, pp. 2699–2718, 2024.

[2] C. Serôdio, P. Mestre, J. Cabral, M. Gomes, and F. Branco, “Software and architecture orchestration for process control in Industry 4.0 enabled by cyber-physical systems technologies,” *Appl. Sci.*, vol. 14, no. 5, Art. no. 2160, 2024.

[3] N. Nikolakis, P. Catti, A. Chaloulos, W. van de Kamp, M. P. Coy, and K. Alexopoulos, “A methodology to assess circular economy strategies for sustainable manufacturing using process eco-efficiency,” *J. Cleaner Prod.*, vol. 445, Art. no. 141289, 2024.

[4] M. R. Kabir and S. Ray, “Virtual prototyping for modern Internet-of-Things applications: A survey,” *IEEE Access*, vol. 11, pp. 31384–31398, 2023.

[5] A. Entezami, H. Sarmadi, B. Behkamal, and S. Mariani, “Early warning of structural damage via manifold learning-aided data clustering and non-parametric probabilistic anomaly detection,” *Mech. Syst. Signal Process.*, vol. 224, Art. no. 111984, 2025.

[6] C. Chen, J. Shi, M. Shen, L. Feng, and G. Tao, “A predictive maintenance strategy using deep learning quantile regression and kernel density estimation for failure prediction,” *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–12, 2023.

[7] A. Deekshith, “Scalable machine learning: Techniques for managing data volume and velocity in AI applications,” *Int. Sci. J. Res.*, vol. 5, no. 5, 2023.

[8] J. Lee, Y. Zhao, J. Roy, and D. Singh, “AI-enabled predictive maintenance for smart manufacturing: Concepts, implementations, and applications,” *IEEE Trans. Ind. Electron.*, vol. 68, no. 6, pp. 4697–4711, Jun. 2021.

[9] R. Carvalho, J. Guedes, and T. Pinho, “Deep learning architectures for predictive maintenance in manufacturing: A review,” *J. Manuf. Syst.*, vol. 61, pp. 153–164, Mar. 2022.

[10] Z. Wu, L. Zhang, and X. Zhou, “Transformer-based time-series forecasting for industrial IoT systems,” *IEEE Internet Things J.*, vol. [illegible], no. 4, pp. 3201–3211, Feb. 2023.

[11] S. Zonta, C. Fagorzi, and P. Pedrazzoli, “Predictive maintenance using sequence-to-sequence learning: Applications and trends,” *Appl. Soft Comput.*, vol. 136, Art. no. 110758, Jul. 2023.

[12] Y. Vaswani, S. Ma, and A. Luo, “Enhancing RNN performance for predictive maintenance with GRU-LSTM hybrids,” *Neural Comput. Appl.*, vol. 34, no. 9, pp. 6811–6823, Sep. 2022.

[13] D. Entezami and P. Chuang, “Data-driven techniques for equipment condition monitoring and RUL prediction in industrial systems,” *Reliab. Eng. Syst. Saf.*, vol. 219, Art. no. 1081[illegible], Jan. 2023.

[14] M. Zhang, H. Li, and Y. Xiao, “Deep convolutional LSTM networks for multistep industrial process forecasting,” *IEEE Trans. Cybern.*, vol. 54, no. 3, pp. 1451–1463, Mar. 2024.

[15] S. Huang and Y. Chen, “LSTM-autoencoders for time-series prediction in smart manufacturing,” *J. Intell. Manuf.*, vol. 33, no. 4, pp. 1051–1063, Jul. 2023.

[16] A. Roy and K. Sarkar, “Time-series forecasting using stacked autoencoders in manufacturing processes,” *Comput. Ind.*, vol. 145, Art. no. 103749, Aug. 2023.

[17] C. Wang, Y. Qiu, and X. Zhao, “Hybrid deep learning for industrial predictive analytics,” *Expert Syst. Appl.*, vol. 201, Art. no. 117522, Apr. 2022.

[18] M. Wang and J. Shen, “Improved LSTM-autoencoders for multivariate RUL prediction,” *IEEE Access*, vol. 11, pp. 45312–45323, May 2023.

[19] L. Zhou and R. Wang, “Smart manufacturing with ConvLSTM-based PdM models,” *Procedia CIRP*, vol. 117, pp. 81–86, Mar. 2024.

[20] J. Li, X. Zhang, and H. Guo, “Bidirectional LSTM networks for industrial time-series predictions,” *J. Big Data*, vol. 10, no. 1, pp. 12–25, Jan. 2023.

[21] X. Xu, H. Liu, and W. Gao, “Sliding-window ConvLSTM for real-time predictive maintenance,” *Future Gener. Comput. Syst.*, vol. 139, pp. 184–195, Nov. 2023.

[22] J. K. Brown and A. Patel, “Real-time applications of deep learning for predictive maintenance in manufacturing plants,” *Comput. Ind. Eng.*, vol. 174, Art. no. 108462, Feb. 2024.