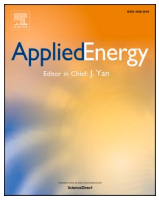
Applied Energy 367 (2024) 123357

Contents lists available at ScienceDirect

Applied Energy

journal homepage: www.elsevier.com/locate/apenergy



Prediction of the temperature of diesel engine oil in railroad locomotives using compressed information-based data fusion method with attention-enhanced CNN-LSTM

Xin Wang [a], Xiang Liu [a], Yun Bai [b,*]

[a] Department of Civil and Environmental Engineering, Rutgers, The State University of New Jersey, Piscataway, NJ, USA
[b] Intelligent Transportation Thrust, Hong Kong University of Science and Technology (Guangzhou), Guangdong, China

HIGHLIGHTS

- A deep learning neural network is developed to predict the temperature of diesel engine oil in locomotives.
- To mitigate data redundancy issue, autoencoder is applied to compress the original inputs into low-dimensional latent space.
- Self-attention is incorporated into the network to enhance the performance of the CNN-LSTM by assigning different weights.

ARTICLE INFO

Keywords:
Engine oil temperature
CNN-LSTM
Compressed information
Attention mechanism

ABSTRACT

Engine oil temperature is a key parameter for ensuring the optimal functioning of diesel engines in locomotives. This paper proposes attention-enhanced CNN-LSTM based on compressed information for predicting engine oil temperature of railroad locomotives using sensor data associated with operational, system-related, and environmental factors. First, the autoencoder was utilized to transform the raw input into latent representation to reduce the computation cost and extract meaningful data patterns. Subsequently, the CNN (convolutional neural network)-LSTM (long short-term memory neural network) was adopted to capture short-term and long-term dependencies within the compressed data by combining the advantages of both CNN and LSTM architectures. CNN is to consider the local correlations, while LSTM is to learn the long-term sequential dependence in the data. Furthermore, the self-attention mechanism was incorporated to enhance the performance of CNN-LSTM by weighing the importance of different parts. An experiment has been conducted using real-world dataset. The results show that our proposed model can make highly accurate predictions with Mean Square Error (MSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) values of 0.335, 0.369, and 0.586%, respectively, which outperform five baselines (i.e., random forests, XGBoost, CNN, LSTM, and CNN-LSTM). This emphasizes the significance of considering both local and long-term correlation and integrating attention mechanism in handling sequence data, contributing to better prediction performance. Overall, the proposed approach provides accurate predictions of engine oil temperature and can be used to support the efficiency and reliability of in-service diesel engines in locomotives by optimizing operating conditions and allowing for proactive measures to prevent overheating before critical temperatures are reached.

1. Introduction

The transportation sector holds significant importance in the context of global energy consumption, accounting for 19% of the world's final energy demand [1]. Rail transportation has historically played a pivotal role in shaping the economic and societal landscapes of regions worldwide. Notably, the energy efficiency of locomotives has positioned rail transport as a more environmentally sustainable option compared to other modes. Ensuring the reliability and efficient operation of locomotive engines can maintain this environmental advantage and enhance the economic competitiveness of rail transportation within the industry.

The temperature of engine oil is a critical parameter to monitor and

* Corresponding author.
E-mail address: yunbai@hkust-gz.edu.cn (Y. Bai).

https://doi.org/10.1016/j.apenergy.2024.123357
Received 4 December 2023; Received in revised form 6 April 2024; Accepted 30 April 2024

control to ensure the energy efficiency, operational power, and longevity of the diesel engine in locomotives [2]. The primary purpose of engine oil is to lubricate moving parts and reduce friction to minimize the wear on engine components for efficient operation of the engine [3,4]. It also provides cooling and cleaning functions by dissipating heat generated during combustion and removing particles like mental fragments that accumulate in the engine over time. However, when the oil temperature becomes too high, engine oil loses its properties and cannot maintain originally intended function [5]. Overheated engine oil not only results in increased friction and wear on engine components but also compromises its effectiveness in cleaning and protecting the engine as excessive heat can break down the oil's additives, leading to decreased engine efficiency and lifespan. Beyond increased friction causing energy loss, diesel engines would transfer to the idling mode due to overheated engine oil, where typical fuel consumption per produced unit of energy is high [2,6]. Furthermore, colloids form due to the oxidation of engine oil by air under high temperatures, which exacerbates diesel engine wear and potentially causes solid cementation, blocking the oil circuit [7,8].

Predicting engine oil temperature holds great significance in the field of predictive maintenance for diesel engines in railroad locomotives, especially in optimizing the performance of in-service diesel engines. First, accurate predictions allow for proactive measures to prevent overheating, which can ensure the reliable operation of railroad locomotives. It can facilitate preventive actions based on engine oil prediction, allowing crews to intervene before critical failures occur. For instance, when the predictive model forecasts an upcoming temperature surge in engine oil, it can trigger an automatic reduction in the engine load or even stop engine operation, minimizing the strain on the cooling system, which can effectively avert overheating scenarios and prevent potential engine damage. Additionally, predicting engine oil temperature can be used for early detection of anomalies. If the predicted temperature deviates significantly from the expected values, it may indicate potential issues with the locomotive engines or its components. To be specific, a sudden increase in engine oil temperature beyond normal operating ranges triggers an alert. Then, maintenance crews would investigate and address potential issues associated with the cooling system, lubrication system, or other components before they escalate.

The temperature of engine oil is subject to various factors that can be mainly divided into three categories: operational, system-related, and environmental factors. Internal combustion engines operate under various conditions such as revolutions per minute (RPM) and speed of the locomotive, leading to different increases in engine oil temperature [9]. Additionally, the engine oil cooling system engages in heat exchange with the diesel engine oil to keep its temperature within the specified range [10]. Thus, multiple parameters associated with the engine oil cooling system like the coolant/water temperature of the intercooler could significantly impact engine oil temperature [11,12]. In addition to operational and system-related factors, environmental factors including ambient temperature and altitude have an important influence on the temperature of engine oil [9]. For instance, cold weather delays oil warming, while hot weather can lead to excessive oil heat.

Due to the interactive effect of multiple impact factors, it is difficult to build a physics-based model to predict the real-time temperature of engines with consideration of the synergy of all factors. Conversely, with the development of big data and artificial intelligence, machine learning approaches have been widely utilized in engineering fields such as structural and mechanical defect prediction [13,14]. They minimize the requirement for domain knowledge by empowering automated data processing and model training. Deep learning (DL), a branch of machine learning, has attracted extensive attention from industry and researchers. It has demonstrated effectiveness in various real-world tasks such as image recognition [15], audio processing [16], natural language processing (NLP) [17,18], and time series analysis [19,20]. In the area of the engine, DL approach has been applied to analyze and predict the critical engine parameters, which include exhaust gas temperature [21], engine output elements (e.g., carbon monoxide and nitrogen oxides) [22,23], and fuel consumption and effective power [24]. While the utilization of DL in predicting engine oil temperature has not been extensively explored, it presents a promising approach for this application due to its capability to capture intricate and non-linear patterns in the data.

Typical neural networks in supervised learning tasks include multilayer perceptron (MLP), convolutional neural network (CNN), and recurrent neural network (RNN). Each of them possesses unique strength in data analysis, allowing them well-suited for various types of datasets. MLPs are effective for feature-based or tabular data, where the model's inputs are not dependent on their spatial or temporal relationships. On the other hand, CNNs are primarily utilized for tasks involving grid-like data such as images and videos via employing convolutional layers to extract hierarchical features and capture spatial dependence in the data [15,25]. To be specific, CNN models such as AlexNet [26], VGG [27], and ResNet [28] have shown state-of-the-art performance in large-scale image classification. Additionally, in the area of object detection, the CNN-based real-time object detection algorithm known as YOLO can detect and locate multiple objects in an image with a single forward pass [29]. Beyond computer vision, 1D-CNNs that apply one-dimensional filter sliding along a single dimension have been leveraged for analyzing the time-series data including sensor data and audio signals. The core of 1D-CNNs is the element-wise multiplication of filter weights with the input data (aka., convolution operation), allowing for identifying and capturing short-term patterns or local fluctuations in the sequential data. In contrast, RNNs have better performance in capturing the long-term dependence in the input data than CNNs by maintaining a hidden state that keeps useful information from previous steps. Long short-term memory (LSTM), an enhanced version of RNN, was designed to address the problem of vanishing and exploding gradients in the training process of traditional RNNs by introducing interacting gates [30,31]. These gates are able to control the flow of information through the cell state, allowing for the storage of information over long sequences. Thus, hybrid neural networks such as CNN-LSTM have been widely utilized in real-world tasks, such as stock analysis [32] and energy consumption prediction [33], to combine the strengths of both CNN and LSTM. In the area of diesel engine analysis, CNN-LSTM models have demonstrated promising results for tasks such as engine fault prediction [34] and engine emission parameters prediction [35]. By leveraging the spatial feature extraction capabilities of CNN and the long-term temporal dependency modeling of LSTM, these models can effectively analyze multivariate time-series sensor data from diesel engines.

Attention mechanism has been integrated into various DL algorithms for its ability to model complex relationships in sequential data [17]. It improves the DL models' ability to effectively capture the most relevant information by dynamically weighing the importance of different parts of the input data. Self-attention mechanism is one of the most influential types of attention, where each element can learn to emphasize relevant information from other elements, regardless of their position in the input sequence [36]. Transformers using self-attention as a key part are able to effectively capture intricate dependencies between input elements in natural language processing (NLP) tasks like machine translation and text summarization tasks [37,38]. Additionally, the integration of self-attention into CNN and/or LSTM has demonstrated its effectiveness in multiple engineering tasks including effluent wastewater quality prediction [39], remaining useful life estimation of mechanical components [40], and traffic analysis [41]. While the basic CNN-LSTM model processes sequential data indiscriminately, the attention-enhanced variant prioritizes informative input sequences while attenuating the impact of noise or irrelevant information. Self-attention mechanism enables the model to focus on critical features, such as sudden changes in engine load or temperature fluctuations, that are indicative of impending variations in engine oil temperature, which could make it a more effective tool for our prediction task.

Nevertheless, due to more complex architecture, neural networks

require more memory storage and computation costs in the training process, especially when dealing with large volumes of sensor data. To avoid information loss in the data collection phase, the data sampling frequency is usually set at a high rate, which introduces a challenge in the form of data redundancy, where some data points are duplicated or in minor variations due to the rapid sampling rate. In the area of artificial intelligence, data redundancy brings about a significant increase in memory requirements and computation costs [42]. Besides, the presence of redundancy makes it more challenging for AI models to explore and capture the underlying data patterns, affecting the generalization performance (prediction accuracy) of the models.

Autoencoder is a type of unsupervised neural network used for data compression, which can be adopted to mitigate the data redundancy issue [43,44]. It significantly reduces the computation requirements for subsequent analysis by compressing the high-dimensional sensor information into lower-dimensional latent space. Autoencoder consists of two parts: encoder and decoder [45,46]. Encoder refers to a compression process that usually has several layers of neurons with a decreasing number in each layer. The bottleneck is the compressed representation generated by the encoder, which contains fewer data dimensions than raw data and includes the most important data patterns. The decoder reconstructs the original data input from the compressed representation with minimal information loss. Therefore, applying compressed information as inputs of the neural networks can both keep the most useful information in the data and reduce the computation requirements.

In summary, predicting the temperature of diesel engine oil is significant for optimizing engine performance to ensure energy efficiency and service life of internal combustion engines in locomotives. Previous research demonstrated the promise of incorporating of attention mechanism into the CNN-LSTM model in analyzing time series tasks [39]. However, it is challenging to have accurate prediction results of engine oil temperature due to the complex interaction of various factors and the high computation requirements caused by data redundancy. These influencing factors encompass a wide array of operational, system-related, and environmental variables, each exerting its own unique impact on engine oil temperature. The complexity of these interactions escalates the computational demands of predictive modeling. Furthermore, this challenge is exacerbated by data redundancy stemming from high-frequency sampling rates, which often leads to duplicated or closely similar data points. Consequently, the processing and analysis of such redundant data impose additional computational overhead, hindering the efficiency and scalability of predictive algorithms. In this research, we propose attention-enhanced CNN-LSTM based on compressed representation for predicting the temperature of diesel engine oil using multiple sensor data in support of efficient and reliable operation of diesel engines in locomotives. By compressing the sensor data, we reduce the redundancy inherent in closely similar or duplicated data points. This compression not only alleviates the computational overhead associated with processing redundant data but also enhances the efficiency and scalability of the predictive model. The main contributions of this paper are summarized as follows.

1. For the first time, a deep learning neural network is developed to predict the temperature of diesel engine oil in locomotives with the consideration of operational, system-related, and environmental factors. CNN is to consider the short-term dependence, and LSTM is to capture the long-term dependence in the data.
2. To mitigate data redundancy issue, autoencoder is applied to compress the original inputs into low-dimensional latent space and keep the meaningful information in the data, which can save memory storage and computation costs.
3. Self-attention is incorporated into the network to enhance the performance of the CNN-LSTM model by assigning different weights to capture informative features and patterns within the data while downplaying the less relevant ones.
4. We conduct the experiment on real-world dataset with over 15.7 million records collected from locomotive diesel engines. The results show that our deep learning outperforms other baselines.

2. Data description

In this study, we collaborated with a rolling stock company to monitor the engine oil temperature and its impact factors by employing multiple sensors in locomotives. The collected factors were associated with operational, system-related, and environmental information, having 55 parameters in total. The related sensors were strategically placed to gather real-time data during locomotive operations. The data collection process adhered to industry standards and best practices to ensure the accuracy and reliability of the gathered information. To be specific, the operating conditions include throttle position, engine speed (i.e., RPM), and locomotive speed. The system-related information contains engine power and load parameters (e.g., the actual power of diesel engine and output power at wheel rim) and engine oil system parameters.

Fig. 1 presents the engine oil circuit system and corresponding sensors that were used to collect related parameters of the engine oil system. Notably, the temperature of engine oil directly from the diesel engine is measured, which is the prediction target in this research. The engine oil with high temperature flows through the engine oil cooler unit. Coolant from the radiator passes through the oil cooler unit to dissipate heat from the engine oil. As the coolant temperature impacts the effectiveness of heat dissipation of engine oil, the temperature information of both the input and output coolant is recorded. The oil filter removes contaminants and impurities from the engine oil. The difference in pressure between the oil entering the filter and exiting the filter indicates the filter's resistance to flow. The greater pressure difference infers that the filter gets clogged with contaminants, which could increase engine oil temperature due to reduced oil flow. Thus, the pressure difference of the oil filter is monitored during locomotive operations for this research. Additionally, temperature and pressure sensors are leveraged to measure the conditions of engine oil entering the engine.

Finally, three environmental parameters were collected to consider environmental impacts on engine oil temperature: ambient temperature, altitude, and atmospheric pressure.

Typically, some operational (e.g., engine speed) and system-related information (e.g., actual engine power) exhibit rapid fluctuations over short time intervals. In contrast, environmental factors like ambient temperature tend to have minor variations within a short timespan. Therefore, the data sampling frequency is set at 5 Hz to avoid potential information loss in the raw data, meaning that the data collection system records related influencing factors every 0.2 s. There are 15.7 million records collected for this research from 36.3 days of locomotive operations.

3. Methodology

Fig. 2 presents the framework of the proposed methodology for predicting the temperature of diesel engine oil, which contains data preprocessing, information compression, and applying attention-enhanced CNN-LSTM for prediction. First, in the data preprocessing phase, we generated the model's input by considering multiple time steps of historical sensor data. The prediction target is defined as the temperature of engine oil given a certain timespan in advance. Then, the autoencoder was leveraged to transfer the raw input into latent representation to reduce the computation requirement. Finally, attention-enhanced CNN-LSTM was developed to predict the temperature of diesel engine oil based on the compressed information generated by the autoencoder.

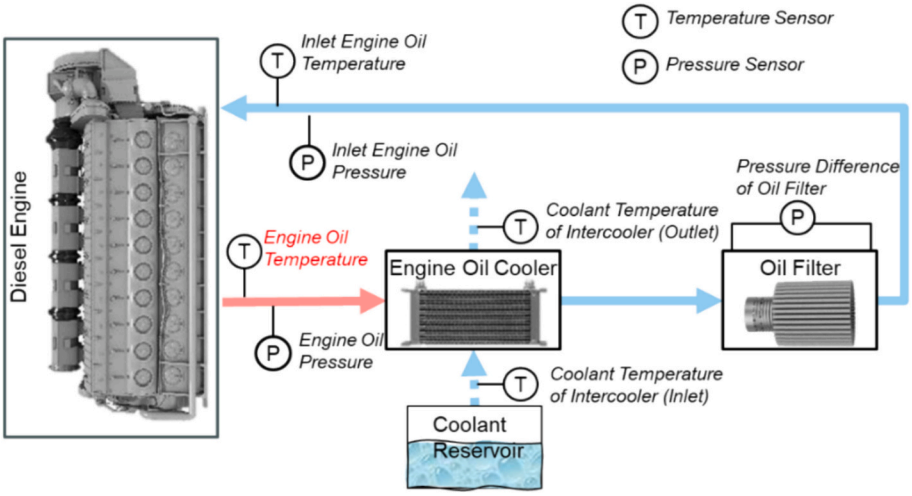


Fig. 1. Schematic Diagram of the Engine Oil Circuit and Sensors.

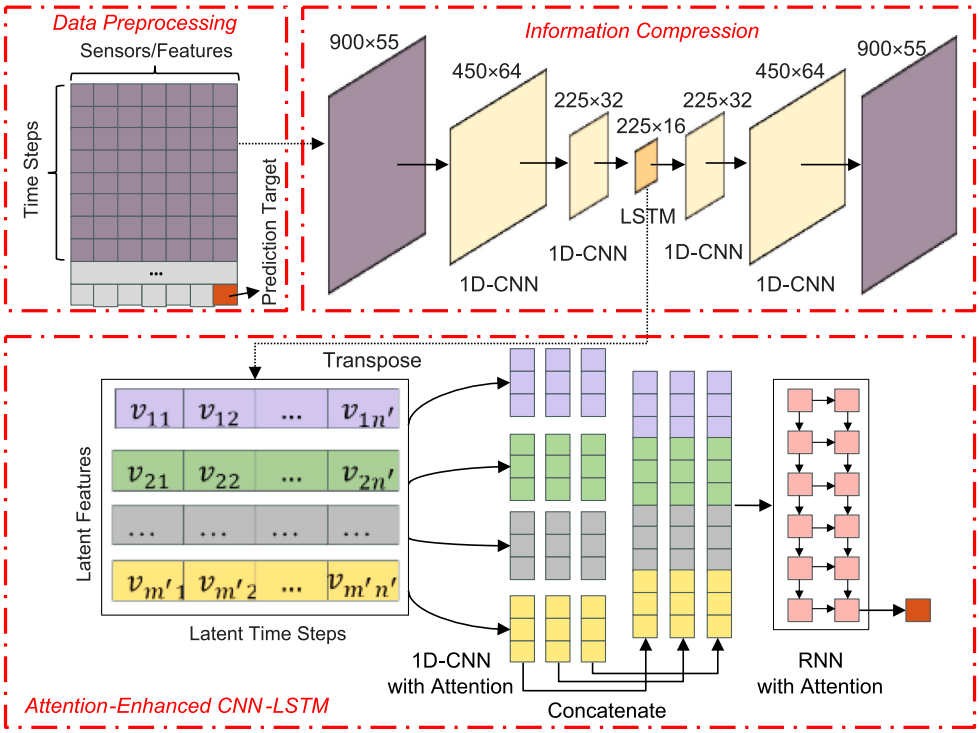


Fig. 2. Methodological Framework for Predicting Temperature of Diesel Engine Oil in Locomotives.

3.1. Data preprocessing

Historical information of sensors can help to identify the data trends and patterns that develop over time, which can improve the performance of the models. In this research, we considered time-series data collected from various sensors as the input of the model to predict the temperature of diesel engine oil given a certain timespan in advance. Specifically, we set the number of time steps (window length) of historical data to 900 and predict temperature 300 time steps in advance. In other words, information of all sensors spanning the past three minutes was utilized to forecast the engine oil temperature one minute ahead.

For a given dataset $D = \{(X_k, Y_k)|k = 1, 2, ..R\}$, Y_k is the prediction target of the model, and X_k represents the inputs of the model having R instances in total, as shown in Eq. (1–2).

$$X_k = \left\{x_{ij}^k | i = 1, 2, \ldots n; j = 1, 2, \ldots m\right\} \tag{1}$$

$$Y_k = y_k \tag{2}$$

where, x_{ij}^k indicates the value of i^{th} row and j^{th} column in k^{th} instance X_k, n denotes the number of different sensors, and m represents the number of time steps (i.e., window length). In this research, $n = 55$ denoting that all the sensor data collected in Section 2 are utilized, and $m = 900$. y_k is a continuous value indicating engine oil temperature one minute in advance. x_{ij} includes the historical information of all sensors; thus, model input X_k ($X_k \in \mathbb{R}^{n \times m}$) is a 2-D matrix.

Notably, the variation of engine oil temperature of two consecutive data points is minor due to high data sampling frequency (5 Hz). Therefore, this research sets the moving steps of the window to 10 s,

equivalent to 50 time steps, and the corresponding number of total instances R equals 0.314 million.

3.2. Information compression

Autoencoder extracts the meaningful data pattern and compresses the data information while minimizing the reconstruction error, where only unlabeled data X_k is involved in the model training process. The input data X_k passes through the encoder network to reduce the dimensionality of the data and generates the compressed (latent) representation $X_k^{'}$. Then, the decoder network reconstructs the original data using compressed representations as input, producing the reconstructed data $\widehat{X}_k$. In this research, the autoencoder structure applied one-dimensional (1-D) convolutional layer and LSTM layer to consider both short-term and long-term dependence in the data [47,48]. The 1-D convolutional layer employs a 1-D convolution kernel to obtain a local feature map Z_k from the input, using Eq. (3) [48].

$$Z_k = f\left(\sum_{k\in M} X_k \times C_k + b\right) \tag{3}$$

where M denotes the feature map. $f(\bullet)$, C_k and b represent the activation function, convolution kernel, and bias term, respectively. This research adopted Rectified Linear Unit (ReLU) function as activation function in the convolution layer, and there are two layers of 1-D CNN in the encoder part.

The convolution layer Z_k passes through a downsampling layer to reduce its dimensions (Fig. 3). The corresponding output is then applied as input of the LSTM layer to capture the long-term dependence in the data and further decrease the dimensionality of the data. LSTM controls the information that can be added or dropped via three gates: input gate, output gate, and forget gate. For the sake of simplicity, the latent representations $X_k^{'}$ of the autoencoder is generated by LSTM layer using Eq. (4).

$$X_k^{'} = LSTM(MaxPool(Z_k, s)) \tag{4}$$

where $MaxPool(\bullet)$ denotes the max-pooling function with pooling size of s. This research utilizes the maximum pooling function and sets $s = 2$. $LSTM(\bullet)$ indicates the function for processing the feature map using the LSTM layer (More details can be found in section 3.3). The reconstructed data $X_k^{'}$ has a reduced dimensionality and can be denoted by Eq. (5).

$$X_k^{'} = \left\{x_{i'j'}^{k} | i' = 1, 2, \ldots n'; j' = 1, 2, \ldots m'\right\}\ n' < n, m' < \mathrm{m} \tag{5}$$

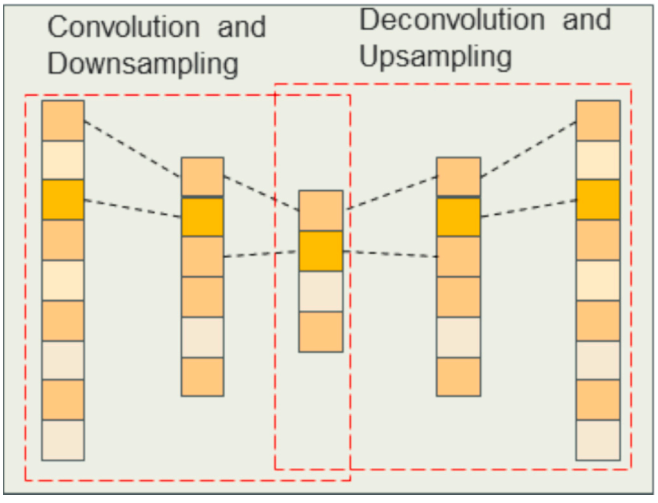


Fig. 3. Schematic Diagram of Convolution, Deconvolution, and Pooling.

where $x_{i'j'}^{k}$ is the element in the i'^{th} row and j'^{th} column of the compressed 2-D matrix $X_k^{'}$, n' and m' are the number of total rows and columns of the $X_k^{'}$, respectively. This research set $n' = 225$ and $m' = 16$.

In the encoder neural network above, the convolution and downsampling produce the latent feature map and reduce dimensions. By taking the latent representation $X_k^{'}$ as input, the decoder neural network leverages the deconvolution operation and upsampling to generate the output $\widehat{X}_k$ that ideally resembles the original input X_k. The decoding process can be calculated using Eq. (6–7).

$$\widehat{X}_k = UpSample\left(Z_k^{'}, s^{'}\right) \tag{6}$$

$$Z_k^{'} = g\left(\sum_{k\in M} X_k^{'} \times C_k^{'} + b^{'}\right) \tag{7}$$

where $UpSample(\bullet)$ is the upsampling function; $Z_k^{'}$ denotes the output of the deconvolution function; $s^{'}$ represents the upsampling size that usually equals the pooling size of downsampling, which means $s^{'} = s = 2$; $g(\bullet)$ indicates the activation function of deconvolution; $X_k^{'}$ is the output of the deconvolution layer after upsampling transformation, $C_k^{'}$ and $b^{'}$ are the deconvolution kernel and bias term for the deconvolution layer, respectively.

The training process of autoencoder is to minimize the difference (information loss) between the input data X_k and reconstructed data $\widehat{X}_k$, given the constraints of the proposed architecture. Thus, although the compressed representation $X_k^{'}$ has smaller dimensionality than raw data X_k, it captures the most informative data patterns and provides minimal information loss. This research adopted mean square error as the loss function to evaluate information loss during model training, which is calculated using Eq. (8).

$$MSE = \frac{1}{K}\sum_{k=1}^{K}(X_k - \widehat{X}_k)^2 = \frac{1}{Kmn}\sum_{k=1}^{K}\sum_{i=1}^{n}\sum_{j=1}^{m}\left(x_{ij}^{k} - \widehat{x}_{ij}^{k}\right)^2 \tag{8}$$

where the x_{ij}^{k} and $\widehat{x}_{ij}^{k}$ denotes the value of the element in the i^{th} row and j^{th} column of the matrix associated with the k^{th} instance X_k and its corresponding reconstructed representation $\widehat{X}_k$, respectively.

3.3. Attention-enhanced CNN-LSTM

The attention-enhanced CNN-LSTM model was proposed to process the compressed information generated by the autoencoder for predicting the temperature of diesel engine oil. It takes advantage of both CNN and LSTM architecture to capture short-term and long-term dependencies within the compressed data. First, the convolutional operation was employed to produce a feature map to consider the local patterns and correlations of the input data. Next, LSTM was applied to learn the long-term sequential dependence in the feature map. Furthermore, the attention mechanism was incorporated into the CNN-LSTM to enhance its ability to focus on more relevant and important features of the input data.

Similar to the CNN applied in the autoencoder, in our proposed model, the convolution operation involves element-wise multiplication of the filter weight with the input data at each position and summing the results to produce a feature map. Here, the convolution operation applies the latent representations $X_k^{'}$ as input, which is presented in Eq. (9–10). Subsequently, output feature map $X_k^{''}$ is transformed into a new feature map F_k with reduced dimension by taking the maximum value from a local region of the feature map.

$$X_k^{'} = f_c\left(\sum_{k\in M} X_k^{'} \times C_k^{'} + b^{'}\right) \tag{9}$$

$$F_k = MaxPool\left(X_k^{'}, s_c\right) \tag{10}$$

where M represents the feature map. $f_c(\bullet)$ is the activation function of the covolution layer in proposed model. $C_k^{'}$ and $b^{'}$ denote the convolution kernel and bias term of the convolution layer, respectively. s_c indicates the pooling size of the maximum pooling function.

Then, the self-attention mechanism is incorporated to weigh the importance of different parts of the local feature map produced by CNN. To this end, for attention head h, the input feature F_k is projected into three different spaces to calculate the Query (Q^h), Key (K^h), and Value (V^h) using Eq. (11–13) [36].

$$Q^h = F_k W_Q^h \tag{11}$$

$$K^h = F_k W_K^h \tag{12}$$

$$V^h = F_k W_V^h \tag{13}$$

where W_Q^h, W_k^h, and W_V^h represent the learnable parameters, Q^h determines the focus of attention, K^h serves as a reference for evaluating dependencies, and V^h contains the specific information associated with each element. Then, the attention information S^h for head h is obtained by Eq. (14).

$$S^h = softmax\left(\frac{Q^h\left(K^h\right)^T}{\sqrt{d_k}}\right)V^h \tag{14}$$

where d_k denotes the dimension of the key vectors, and $softmax(\bullet)$ is softmax function. The final multi-head attention output S is the weighted sum of all the attention head using a weight matrix W_S, as follows.

$$S = \left[S^1, S^2, \ldots, S^h\right] W_S \tag{15}$$

where $[\bullet]$ denotes concatenation operation.

For the sake of simplicity, the computing process of multi-head self-attention (Eq. (11–15)) is denoted using function $Attention(\bullet)$, which is shown in Eq. (16).

$$S = Attention(F_k) \tag{16}$$

Subsequently, the output S is utilized as input of the LSTM to learn and capture the long-term dependence in the feature map by processing the information over multiple time steps. The key component of the LSTM layer, LSTM unit, controls the information flow into and out of the cell state and the hidden state using three gates: the input gate (i_t), forget gate (f_t), and output gate (o_t). The hidden state represents the output of the LSTM layer at the current time step, while the cell state stores information acquired from the previous time step. At time t the gate information can be determined using the hidden state H_{t-1} at the previous time step $t-1$ based on Eq. (17–19).

$$f_t = \sigma\left(W_f[H_{t-1}, S_t] + b_f\right) \tag{17}$$

$$i_t = \sigma(W_i[H_{t-1}, S_t] + b_i) \tag{18}$$

$$o_t = \sigma(W_o[H_{t-1}, S_t] + b_o) \tag{19}$$

where, W_f, W_i, and W_o denote learnable weight parameters. b_f, b_i, and b_o represent learnable bias parameters. S_t is the information of the feature map at time step t. σ is the activation function for computing gate information, which is the sigmoid function in this research. The cell state C_t is calculated using gate information given by Eq. (20).

$$C_t = f_t \odot C_{t-1} + i_t \odot \widetilde{C}_t \tag{20}$$

$$\widetilde{C}_t = tanh\left(W_c\left[H_{t-1}, V_k^t\right] + b_c\right) \tag{21}$$

where $\widetilde{C}_t$ is the input node (aka. Candidate cell state); $tanh(\bullet)$ denotes the tanh activation function; $\odot$ represents the Hadamard product operator; W_c and b_c are learnable weight and bias parameters, respectively. Finally, the hidden state H_t is obtained based on the cell state C_t, as shown in Eq. (22).

$$H_t = o_t \odot tanh(C_t) \tag{22}$$

The forget gate decides what information from the previous cell state should be forgotten, while the input gate controls the flow of new information into the cell state. The output gate considers the cell state and determines the next hidden state. In this research, we adopt two LSTM layers. The final prediction result $\widehat{y}_k$ of the proposed model is obtained by processing the output of the LSTM with attention mechanism and then fully connecting the corresponding output to single unit using the linear activation function $f_o(\bullet)$, which is presented in Eq. (23).

$$\widehat{y}_k = f_o(Attention([H_1, H_2, \ldots, H_L])) \tag{23}$$

where L is the number of LSTM cells in the last LSTM layer.

Mean Squared Error (MSE) is a common loss function used to measure the difference between the predicted values and the actual target values. Therefore, we applied MSE as the loss function in this research, which is calculated by Eq. (24).

$$MSE = \frac{1}{R}\sum_{k=1}^{R}\left(y_k - \widehat{y}_k\right)^2 \tag{24}$$

4. Experiment and results

In this section, we first introduced the evaluation metrics and baselines, followed by implementation details of deploying the model. Then, we showed the results of the proposed model and compared it with different baselines.

4.1. Evaluation metrics and baselines

Evaluation metrics are to statistically indicate the generalization performance of machine learning model. Three metrics were adopted to compare the performance of our proposed model and the following baselines: MSE, Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE), which are presented in Eq. (24–26).

$$MAE = \frac{1}{R}\sum_{k=1}^{R}\left|y_k - \widehat{y}_k\right| \tag{25}$$

$$MAPE = \frac{1}{R}\sum_{k=1}^{R}\left|\frac{y_k - \widehat{y}_k}{y_k}\right| \tag{26}$$

For comparative purposes, four deep learning models: MLP, CNN, LSTM, and CNN-LSTM (without attention mechanism) were applied in this research. Additionally, two machine learning models: random forests and XGBoost were also selected as baselines. Random forests is one of the most popular ensemble methods that aggregates the predictions of multiple decision trees to make more accurate predictions [49]. On the other hand, XGBoost employs gradient boosting framework that builds the final predictive model by sequentially combing the prediction of multiple weak models [50].

Notably, CNN, LSTM, and CNN-LSTM models can utilize the 2-D matrix as input, while the MLP, random forests, and XGBoost are mainly used for feature-based data structures. Therefore, we used the same data structure (time-series data X_k) when applying CNN, LSTM,

and CNN-LSTM models. In contrast, MLP, random forests, and XGBoost only considered the latest record of each sensor as feature-based input in predicting engine oil temperature. While MLP, random forests, and XGBoost are typically not used for time-series tasks, they serve as essential benchmarks in this paper for evaluating the capability of deep learning models to capture the short-term and long-term dependencies in sequence data.

4.2. Implementation details

The proposed model was implemented utilizing Keras version 2.11.0. [51] on a server equipped with an NVIDIA RTX 3070 GPU with an AMD Ryzen 3700× CPU @3.59 GHz and 112GB of system memory. The encoder part of the autoencoder consists of three convolution layers and one LSTM layer. The predictive model proposed in this paper has one convolution layer and two LSTM layers. To ensure the network's effective training, several critical hyperparameters were selected: the batch size was configured to 128, while the learning rate was set to 0.0001. A dropout probability of 0.1 was utilized to mitigate overfitting. The model's parameters were optimized using the Adam optimizer. The patience epoch value was set to 20, which means if the performance metric on the validation set (i.e., validation loss) could not improve for 20 consecutive epochs, the training progress would be stopped to prevent overfitting. In our experiment, we split the whole dataset into training, validation, and testing sets with a ratio of 6:2:2. For a comprehensive understanding of our research, code about both our proposed model and baselines, as well as explanations of input parameters, are available at this repository: https://github.com/DaiJunyan/Engine-Oil-Temperature-Prediction.

4.3. Results

4.3.1. Latent representation of the autoencoder

The proposed autoencoder structure was trained using training and validation sets, and corresponding training and validation loss (MSE) curves are presented in Fig. 4. After 153 epochs, the validation loss cannot be improved for the next 20 consecutive epochs. Thus, the model's training process stopped at the 173th epoch. Then, the t-Distributed Stochastic Neighbor Embedding (t-SNE) was introduced to visualize the relationship between the latent space and prediction target, which provides insights into the complex interactions of their relationships [52]. The t-SNE calculates the pairwise similarity between the data instances in the high-dimensional space based on the Gaussian probability distribution, which is also applied to similarities of data instances in the lower-dimensional space. Correspondingly, the similar data instances in the high-dimensional space correspond to nearby data instances in the lower-dimensional space (2D space).

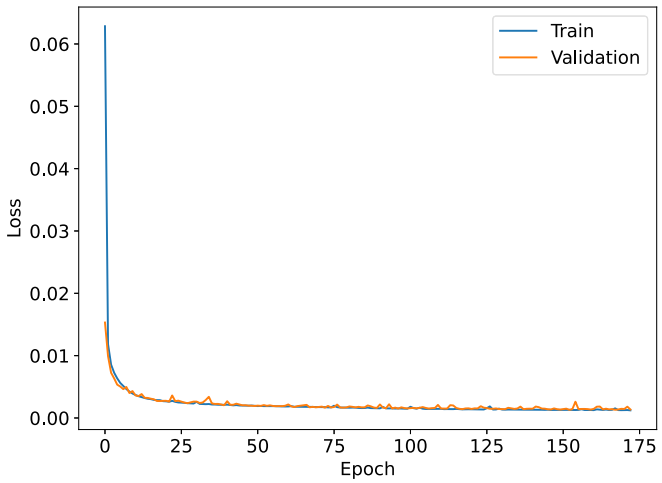


Fig. 4. Training and the Validation Loss Curve of the Autoencoder.

Fig. 5 compares t-SNE plot of the raw data and latent representation, where yellow points refer to data instances with high engine oil temperature, and blue points correspond to data instances with low engine oil temperature. It is found that the autoencoder can capture the underlying patterns in the data and create more structured representations by improving the separation between different categories compared to the raw data. Therefore, the compressed information not only reduces the dimensionality of the data to save computation costs but also provides more informative features for model training.

4.3.2. Performance of the proposed model

The generated latent representation was utilized as input of the proposed attention-enhanced CNN-LSTM to predict the engine oil oil temperature in locomotives. The training and the validation loss curves of the proposed model are presented in Fig. 6. It can be observed that the validation loss remains stable when the training epoch reaches 189 times, and the corresponding time needed for the model to converge is 5481 s.

Subsequently, the trained model was employed to predict the engine oil temperature in the test set to evaluate the generalization performance of the trained model. The MSE for training, validation, and test set are 0.116, 0.170, and 0.335, respectively, which indicates the model has good performance on the training set and generalizes reasonably well to the validation and test set.

The predicted temperature and actual temperature of engine oil in the time domain (partial test set) are presented in Fig. 7 (a). From a partially zoomed view, it can be observed that the prediction results are almost completely the same with minor variation, which indicates the highly accurate prediction results of the proposed model. Additionally, we compared the statistical distribution of the predicted temperature of diesel engine oil and the ground-truth values, which are demonstrated in Fig. 7 (b). For most data points, the predicted temperatures closely align with the actual temperatures. The error of the predicted results is symmetrically distributed. The mean error is approximately 0.063 °C, and the standard deviation is 0.575 °C.

However, there are relatively higher prediction errors at lower temperatures. One potential reason accounting for this phenomenon is the scarcity of data instances in this temperature range. The temporal nature of the recorded data, where sensor readings commence when the train starts and cease when the train stops, presents a unique challenge in capturing the dynamics of low-temperature conditions. Consequently, the model may exhibit relatively greater discrepancies when predicting low-temperature instances.

Finally, three evaluation metrics: MSE, MAE, and MAPE of the proposed model and baselines are summarized in Table 1. Generally, the ensemble learning algorithm, i.e., random forests and XGBoost, perform better than MLP, in terms of all three evaluation metrics. XGBoost has a smaller MSE but slightly greater MAE and MAPE than random forests. In other words, XGBoost is better at reducing large errors compared to random forests. In contrast, random forests provides more accurate predictions in terms of the magnitude of errors judging from its MAE and MAPE.

On the other hand, the CNN and LSTM models take advantage of temporal correlations of the sequence data to provide better performance than three traditional machine learning models, i.e., MLP, XGBoost, and random forests. CNN usually captures the short-term dependence of the data, while LSTM tends to focus on long-term correlations. The hybrid neural network CNN-LSTM combines the strengths of two architectures and achieves better performance than plain CNN or plain LSTM, with lower values for all three evaluation metrics. Additionally, as the integration of the attention mechanism into the CNN-LSTM model, the attention-enhanced CNN-LSTM model exhibits superior performance with the lowest MSE (0.335), MAE (0.369), and MAPE (0.586%). This also emphasizes the significance of the attention

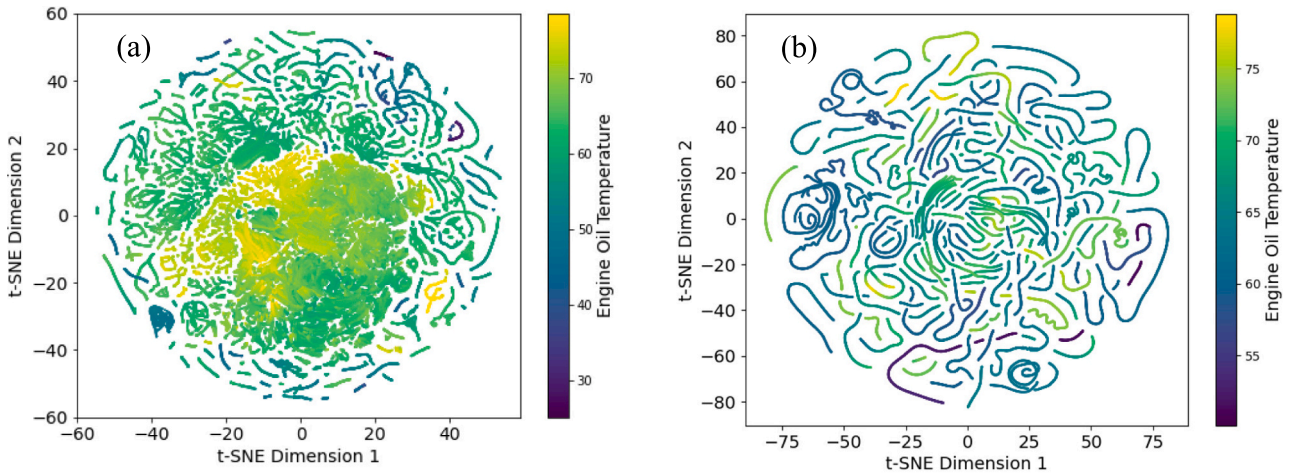


Fig. 5. Low-dimensional Embedding Features (a) Raw Feature, (b) Latent Representation from Autoencoder.

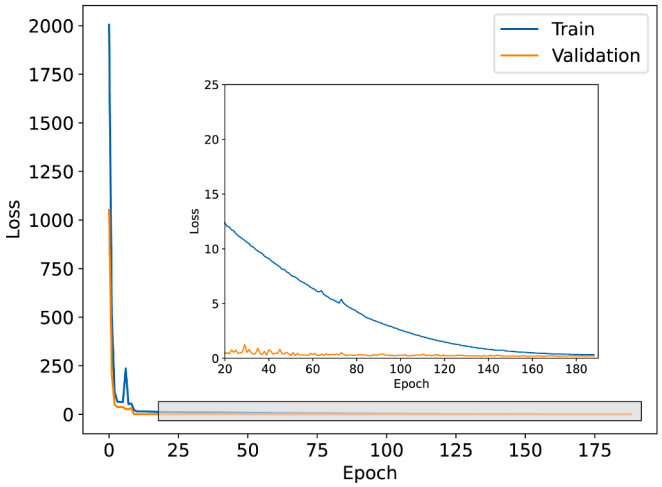


Fig. 6. Training and Validation Loss Curve of the Attention- Enhanced CNN-LSTM.

mechanism in handling sequence data, allowing the model to weigh different time steps more effectively, leading to more accurate predictions.

The attention mechanism incorporated into the CNN-LSTM model enables it to dynamically weigh the importance of different input sequences, effectively highlighting critical features within the sequence data that contribute to accurate predictions of engine oil temperature. This approach aligns with the physical reality of diesel engine operation, where certain factors, such as engine load, ambient temperature, and operating conditions, exert varying degrees of influence on engine oil temperature at different time intervals, such as sudden changes in engine load or temperature fluctuations. Unlike the basic CNN-LSTM model, which processes sequential data without discerning the relative significance of individual elements, the attention-enhanced variant excels at focusing on relevant features while filtering out noise or irrelevant information. This heightened level of discernment not only enhances prediction accuracy but also provides deeper insights into the underlying factors causing engine oil temperature fluctuations. Consequently, the attention-enhanced CNN-LSTM approach surpasses its basic counterpart by offering a more nuanced understanding of the data and a more refined ability to capture subtle yet impactful patterns, ultimately leading to superior predictive performance and more informed decision-making in locomotive operations.

The accurate predictions provided by our proposed model not only prevent overheating but also contribute to fuel efficiency and extend the lifespan of locomotive engines. Operators can make informed decisions about engine operation based on real-time predictions of the engine oil temperature. For example, the proposed model predicts a gradual increase in engine oil temperature and would exceed the green tag (88 °C) or yellow tag (92 °C) during a prolonged heavy-duty operation. With

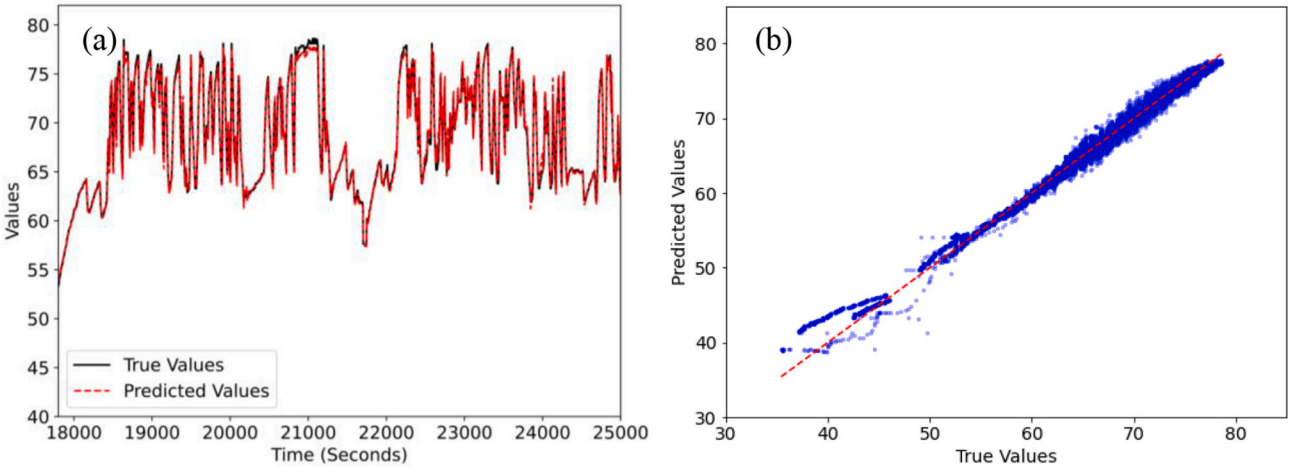


Fig. 7. Performance of Predicted Results (a) Results in the Time Domain (Partial), (b) Statistical Distribution of Predicted Values and True Values.

Table 1
Performance Comparison of Proposed Model with Baselines.

Evaluation Metric	Random Forests	XGBoost	MLP	CNN	LSTM	CNN-LSTM	**Proposed Model in this Paper (Attention-Enhanced CNN-LSTM)**
MSE	8.305	5.402	44.535	0.766	1.187	0.497	**0.335**
MAE (°C)	1.249	1.289	4.084	0.644	0.867	0.436	**0.369**
MAPE	2.537%	2.594%	8.465%	0.969%	1.329%	0.664%	**0.586%**

this information, operators can proactively adjust the engine load, ensuring optimal engine oil temperatures. Furthermore, it can support early detection of anomalies. Furthermore, detecting significant deviations between predicted and expected engine oil temperatures can serve as an early warning system for potential issues within locomotive engines or their components. A simple example could be a 5% temperature difference signaling a potential issue with the cooling system. Crews can inspect and rectify the issue before it escalates, preventing costly breakdowns. Overall, our proposed model can support proactive maintenance measures, ensuring optimal locomotive engine performance and longevity while minimizing the risk of damage or failure.

5. Conclusion and future work

5.1. Conclusion

This study develops a comprehensive deep learning approach for predicting the temperature of diesel engine oil using sensor data collected from locomotives regarding operational, system-related, and environmental factors. It can serve as a critical tool for optimizing diesel engine performance by taking proactive measures to keep the engine oil within optimal ranges, which not only maximizes fuel combustion efficiency but also diminishes the wear and tear on engine components, thereby extending the operational life of the engines.

Autoencoder structure that incorporated 1-D convolutional and LSTM layers was employed to consider local and global correlations in the input based on historical information of the multiple sensors in locomotives. It can effectively keep the most informative data patterns in the compressed latent representation while minimizing the reconstruction error. Furthermore, the CNN-LSTM model was proposed to process the compressed information by capturing both short-term and long-term dependencies. The attention mechanism was incorporated into the model to enhance model's ability to focus more on relevant features by dynamically weighing the importance of different input sequences. The experiment results indicate that plain CNN and plain LSTM have better performance in predicting engine oil temperature than random forests, XGBoost, and MLP, as they consider the time-series information, which contains more underlying patterns and dependencies that are crucial for accurate temperature predictions. Additionally, the hybrid CNN-LSTM outperforms plain CNN and plain LSTM, highlighting the effectiveness of combining these two architectures to harness both short-term and long-term dependencies within the data. With the integration of the attention mechanism, the proposed attention-enhanced CNN-LSTM model obtains the best prediction accuracy with MSE, MAE, and MAPE values of 0.335, 0.369, and 0.586%. This emphasizes the significance of attention mechanism in enabling the model to discern and weigh the importance of different data features, thus providing a superior level of precision and robustness in predicting engine oil temperature.

In summary, our proposed model achieved superior performance compared to the baselines in predicting the temperature of diesel engine oil. Accurate prediction results can be used to better support operational decision-making in industrial settings to mitigate the risk of overheating or excessive cooling, ensuring the optimal functioning of diesel engines. It not only ensures energy efficiency and operational power but also extends the engine's lifespan while reducing maintenance and repair costs.

5.2. Future work

This section discusses the limitations of the current study and future research directions of this study. First, due to data limitations, only one type of locomotive engine is included in this study. Additional experiments with a more extensive and diverse dataset could provide a more comprehensive evaluation of the model's robustness. Therefore, future research would involve the application of the proposed model across multiple locomotive engines to assess its adaptability and performance in diverse operational contexts.

As the temperature of locomotive engine oil is impacted by various factors, it is important to advance our understanding of intricate relationships and enhance model interpretability, which can facilitate informed decision-making, proactive maintenance strategies, and optimal performance management in locomotive engine operations. Therefore, future research will refine the predictive modeling framework to incorporate comprehensive physical interpretations of parameter influences on engine oil temperature. It involves quantifying the relative contributions of each parameter to oil temperature variations, thereby facilitating a deeper understanding of the underlying mechanisms driving temperature fluctuations. Leveraging advanced data analytics techniques, such as sensitivity analysis and feature importance ranking, we aim to identify the most influential factors and their interactions within the engine system.

Furthermore, this study focuses on engine oil temperature prediction in locomotives. It is of significance to extend our proposed model to other critical parameters in the operation of locomotives and other machinery. Exhaust gas temperature, for example, helps assess engine health and performance, optimize fuel efficiency, control emissions, and ensure safe operation. By adapting our model to exhaust gas temperature prediction and other related tasks, we can create a more comprehensive predictive maintenance system that covers multiple vital parameters. This will not only contribute to enhancing the overall reliability and efficiency of locomotives but also lead to more eco-friendly and energy-efficient transportation.

Last but not least, exploring the model's capacity to automatically process a wider array of sensor inputs, while addressing issues of data redundancy, would enhance its versatility. This adaptability is particularly relevant as our model demonstrates proficiency in compressed information through an autoencoder, allowing for streamlined processing of diverse sensor data. This broader implementation could contribute to the model's effectiveness in real-world scenarios where multiple sensor inputs and potential data redundancies are commonplace.

CRediT authorship contribution statement

Xin Wang: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis. **Xiang Liu:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Investigation, Funding acquisition, Conceptualization. **Yun Bai:** Writing – review & editing, Validation, Supervision, Software, Project administration, Investigation, Funding acquisition, Data curation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial

interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

Acknowledgments

Part of the research was conducted while the corresponding author was affiliated with the Rutgers Center for Advanced Infrastructure and Transportation. No agency or institution is accountable for the views and analyses presented in this paper.

References

[1] Khalili S, Rantanen E, Bogdanov D, Breyer C. Global transportation demand development with impacts on the energy demand and greenhouse gas emissions in a climate-constrained world. Energies 2019;12:3870. https://doi.org/10.3390/en12203870.
[2] Holmberg K, Andersson P, Nylund N-O, Mäkelä K, Erdemir A. Global energy consumption due to friction in trucks and buses. Tribol Int 2014;78:94–114. https://doi.org/10.1016/j.triboint.2014.05.004.
[3] Chen X, Wang Z, Pan S, Pan H. Improvement of engine performance and emissions by biomass oil filter in diesel engine. Fuel 2019;235:603–9. https://doi.org/10.1016/j.fuel.2018.08.038.
[4] Hemmat Esfe M, Abbasian Arani AA, Esfandeh S, Afrand M. Proposing new hybrid nano-engine oil for lubrication of internal combustion engines: preventing cold start engine damages and saving energy. Energy 2019;170:228–38. https://doi.org/10.1016/j.energy.2018.12.127.
[5] Hemmat Esfe M, Esfandeh S, Abbasian Arani AA. Proposing a modified engine oil to reduce cold engine start damages and increase safety in high temperature operating conditions. Powder Technol 2019;355:251–63. https://doi.org/10.1016/j.powtec.2019.07.009.
[6] Rahman SMA, Masjuki HH, Kalam MA, Abedin MJ, Sanjid A, Sajjad H. Impact of idling on fuel consumption and exhaust emissions and available idle-reduction technologies for diesel vehicles – a review. Energ Conver Manage 2013;74:171–82. https://doi.org/10.1016/j.enconman.2013.05.019.
[7] Hasannuddin AK, Wira JY, Sarah S, Aqma WMN Wan Syaidatul, Hadi AR Abdul, Hirofumi N, et al. Performance, emissions and lubricant oil analysis of diesel engine running on emulsion fuel. Energ Conver Manage 2016;117:548–57. https://doi.org/10.1016/j.enconman.2016.03.057.
[8] Mohamed Musthafa M. Synthetic lubrication oil influences on performance and emission characteristic of coated diesel engine fuelled by biodiesel blends. Appl Therm Eng 2016;96:607–12. https://doi.org/10.1016/j.applthermaleng.2015.12.011.
[9] Baskov V, Ignatov A, Polotnyanschikov V. Assessing the influence of operating factors on the properties of engine oil and the environmental safety of internal combustion engine. Transp Res Proc 2020;50:37–43. https://doi.org/10.1016/j.trpro.2020.10.005.
[10] Sidik NAC, Yazid MNAWM, Mamat R. A review on the application of nanofluids in vehicle engine cooling system. Int Commun Heat Mass Transf 2015;68:85–90. https://doi.org/10.1016/j.icheatmasstransfer.2015.08.017.
[11] Saeed M, Berrouk AS, AlShehhi MS, AlWahedi YF. Numerical investigation of the thermohydraulic characteristics of microchannel heat sinks using supercritical CO2 as a coolant. J Supercritical Fluids 2021;176:105306. https://doi.org/10.1016/j.supflu.2021.105306.
[12] Nasir S, Berrouk AS, Aamir A, Gul T. Significance of chemical reactions and entropy on darcy-forchheimer flow of H2O and C2H6O2 convening magnetized nanoparticles. Int J Thermofluids 2023;17:100265. https://doi.org/10.1016/j.ijft.2022.100265.
[13] Behera S, Choubey A, Kanani CS, Patel YS, Misra R, Sillitti A. Ensemble trees learning based improved predictive maintenance using IIoT for turbofan engines. In: Proceedings of the 34th ACM/SIGAPP symposium on applied computing. Limassol Cyprus: ACM; 2019. p. 842–50. https://doi.org/10.1145/3297280.3297363.
[14] Wang X, Liu X, Bian Z. A machine learning based methodology for broken rail prediction on freight railroads: a case study in the United States. Construct Build Mater 2022;346:128353. https://doi.org/10.1016/j.conbuildmat.2022.128353.
[15] Deng J, Dong W, Socher R, Li L-J, Li Kai, Fei-Fei Li. ImageNet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. Miami, FL: IEEE; 2009. p. 248–55. https://doi.org/10.1109/CVPR.2009.5206848.
[16] Abdel-Hamid O, Mohamed A, Jiang H, Deng L, Penn G, Yu D. Convolutional neural networks for speech recognition. IEEE/ACM Trans Audio Speech Lang Process 2014;22:1533–45. https://doi.org/10.1109/TASLP.2014.2339736.
[17] Gu Y, Lyu X, Sun W, Li W, Chen S, Li X, et al. Mutual correlation attentive factors in dyadic fusion networks for speech emotion recognition. In: Proceedings of the 27th ACM international conference on multimedia. Nice France: ACM; 2019. p. 157–66. https://doi.org/10.1145/3343031.3351039.
[18] Wang J, Yu L-C, Lai KR, Zhang X. Tree-structured regional CNN-LSTM model for dimensional sentiment analysis. IEEE/ACM Trans Audio Speech Lang Process 2020;28:581–91. https://doi.org/10.1109/TASLP.2019.2959251.
[19] de Bruin T, Verbert K, Babuska R. Railway track circuit fault diagnosis using recurrent neural networks. IEEE Trans Neural Networks Learn Syst 2017;28:523–33. https://doi.org/10.1109/TNNLS.2016.2551940.
[20] Zhong S, Lyu W, Zhang D, Yang Y. Bikecap: Deep spatial-temporal capsule network for multi-step bike demand prediction. In: 2022 IEEE 42nd international conference on distributed computing systems (ICDCS). Bologna, Italy: IEEE; 2022. p. 831–41. https://doi.org/10.1109/ICDCS54860.2022.00085.
[21] Liu J, Huang Q, Ulishney C, Dumitrescu CE. Machine learning assisted prediction of exhaust gas temperature of a heavy-duty natural gas spark ignition engine. Appl Energy 2021;300:117413. https://doi.org/10.1016/j.apenergy.2021.117413.
[22] Mohamed Ismail H, Ng HK, Queck CW, Gan S. Artificial neural networks modelling of engine-out responses for a light-duty diesel engine fuelled with biodiesel blends. Appl Energy 2012;92:769–77. https://doi.org/10.1016/j.apenergy.2011.08.027.
[23] Mebin Samuel P, Gnanamoorthi V, Purushothaman P, Gurusamy A, Devaradjane G. Prediction efficiency of artificial neural network for CRDI engine output parameters. Transp Eng 2021;3:100041. https://doi.org/10.1016/j.treng.2020.100041.
[24] Çay Y, Çiçek A, Kara F, Sağiroğlu S. Prediction of engine performance for an alternative fuel using artificial neural network. Appl Therm Eng 2012;37:217–25. https://doi.org/10.1016/j.applthermaleng.2011.11.019.
[25] Wang X, Bai Y, Liu X. Prediction of railroad track geometry change using a hybrid CNN-LSTM spatial-temporal model. Adv Eng Inform 2023;58:102235. https://doi.org/10.1016/j.aei.2023.102235.
[26] Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Commun ACM 2017;60:84–90. https://doi.org/10.1145/3065386.
[27] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. 2014. https://doi.org/10.48550/ARXIV.1409.1556.
[28] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR). Las Vegas, NV, USA: IEEE; 2016. p. 770–8. https://doi.org/10.1109/CVPR.2016.90.
[29] Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: Unified, real-time object detection. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR). Las Vegas, NV, USA: IEEE; 2016. p. 779–88. https://doi.org/10.1109/CVPR.2016.91.
[30] Bengio Y, Simard P, Frasconi P. Learning long-term dependencies with gradient descent is difficult. IEEE Trans Neural Netw 1994;5:157–66. https://doi.org/10.1109/72.279181.
[31] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput 1997;9:1735–80. https://doi.org/10.1162/neco.1997.9.8.1735.
[32] Liu S, Zhang C, Ma J. CNN-LSTM neural network model for quantitative strategy analysis in stock markets. In: Liu D, Xie S, Li Y, Zhao D, El-Alfy E-SM, editors. Neural information processing. Cham: Springer International Publishing; 2017. p. 198–206. https://doi.org/10.1007/978-3-319-70096-0_21.
[33] Kim T-Y, Cho S-B. Predicting residential energy consumption using CNN-LSTM neural networks. Energy 2019;182:72–81. https://doi.org/10.1016/j.energy.2019.05.230.
[34] Li J, Jia Y, Niu M, Zhu W, Meng F. Remaining useful life prediction of turbofan engines using CNN-LSTM-SAM approach. IEEE Sensors J 2023;23:10241–51. https://doi.org/10.1109/JSEN.2023.3261874.
[35] Aygun H, Dursun OO, Toraman S. Machine learning based approach for forecasting emission parameters of mixed flow turbofan engine at high power modes. Energy 2023;271:127026. https://doi.org/10.1016/j.energy.2023.127026.
[36] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R, editors. Advances in neural information processing systems. Curran Associates, Inc.; 2017. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.
[37] Zeng P, Hu G, Zhou X, Li S, Liu P, Liu S. Muformer: a long sequence time-series forecasting model based on modified multi-head attention. Knowl Based Syst 2022;254:109584. https://doi.org/10.1016/j.knosys.2022.109584.
[38] Shaw P, Uszkoreit J, Vaswani A. Self-attention with relative position representations. 2018. https://doi.org/10.48550/ARXIV.1803.02155.
[39] Li Y, Kong B, Yu W, Zhu X. An attention-based CNN-LSTM method for effluent wastewater quality prediction. Appl Sci 2023;13:7011. https://doi.org/10.3390/app13127011.
[40] Xia J, Feng Y, Lu C, Fei C, Xue X. LSTM-based multi-layer self-attention method for remaining useful life estimation of mechanical systems. Eng Failure Analysis 2021;125:105385. https://doi.org/10.1016/j.engfailanal.2021.105385.
[41] Liu Q, Liu T, Cai Y, Xiong X, Jiang H, Wang H, et al. Explanatory prediction of traffic congestion propagation mode: a self-attention based approach. Physica A: Stat Mech Its Appl 2021;573:125940. https://doi.org/10.1016/j.physa.2021.125940.
[42] Chen J-A, Niu W, Ren B, Wang Y, Shen X. Survey: exploiting data redundancy for optimization of deep learning. ACM Comput Surv 2023;55:1–38. https://doi.org/10.1145/3564663.
[43] Ye Z, Yu J. Health condition monitoring of machines based on long short-term memory convolutional autoencoder. Appl Soft Comput 2021;107:107379. https://doi.org/10.1016/j.asoc.2021.107379.
[44] Yu W, Kim IY, Mechefske C. Remaining useful life estimation using a bidirectional recurrent neural network based autoencoder scheme. Mech Syst Signal Process 2019;129:764–80. https://doi.org/10.1016/j.ymssp.2019.05.005.

[45] Cheng Y, Hu K, Wu J, Zhu H, Shao X. Autoencoder quasi-recurrent neural networks for remaining useful life prediction of engineering systems. IEEE/ASME Trans Mechatron 2022;27:1081–92. https://doi.org/10.1109/TMECH.2021.3079729.
[46] Han T, Pang J, Tan ACC. Remaining useful life prediction of bearing based on stacked autoencoder and recurrent neural network. J Manuf Syst 2021;61:576–91. https://doi.org/10.1016/j.jmsy.2021.10.011.
[47] Chen S, Yu J, Wang S. One-dimensional convolutional auto-encoder-based feature learning for fault diagnosis of multivariate processes. J Process Control 2020;87:54–67. https://doi.org/10.1016/j.jprocont.2020.01.004.
[48] Masci J, Meier U, Cireşan D, Schmidhuber J. Stacked convolutional auto-encoders for hierarchical feature extraction. In: Honkela T, Duch W, Girolami M, Kaski S, editors. Artificial neural networks and machine learning – ICANN 2011. Berlin Heidelberg, Berlin, Heidelberg: Springer; 2011. p. 52–9. https://doi.org/10.1007/978-3-642-21735-7_7.
[49] Ho Tin Kam. Random decision forests. In: Proceedings of 3rd international conference on document analysis and recognition. Montreal, Que., Canada: IEEE Comput. Soc. Press; 1995. p. 278–82. https://doi.org/10.1109/ICDAR.1995.598994.
[50] Chen T, Guestrin C. Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd Acm Sigkdd international conference on knowledge discovery and data mining; 2016. p. 785–94. https://doi.org/10.1145/2939672.2939785.
[51] Gulli A, Pal S. Deep learning with Keras. Packt Publishing Ltd; 2017.
[52] van der Maaten L, Hinton G. Visualizing data using t-SNE. J Mach Learn Res 2008;9:2579–605. http://jmlr.org/papers/v9/vandermaaten08a.html.