1 **LSTM+MA: A Modified LSTM-based Model for Predicting the Pavement IRI**
2

3 **Tianjie Zhang**
4 Department of Computer Science
5 Boise State University, Boise, Idaho, 83706
6 Email: tjzhang@u.boisestate.edu
7
8 **Alex Smith**
9 Department of Computer Science
10 Boise State University, Boise, Idaho, 83706
11 Email: alexsmith194@u.boisestate.edu
12
13 **Yang Lu**
14 Associate Professor
15 Department of Civil Engineering
16 Boise State University, Boise, Idaho, 83706
17 Email: yanglufrank@boisestate.edu
18
19 Word Count: 6277 words + 3 table (250 words per table) = 7027 words
20
21
22 *Submitted [7/30/2023]*
23

ABSTRACT

Effective prediction of pavement performance is essential for transportation agencies to appropriately plan the maintenance and rehabilitation of roads. The International Roughness Index (IRI) is one of the most important pavement performances. However, the research on IRI prediction is mainly focused on linear regression and traditional machine learning, which cannot take the time-accumulated influence of related features into account. In this work, a Long Short-Term Memory (LSTM)-based model, LSTM+MA, is proposed to predict the IRI of pavements based on the time-series data extracted from the Long-Term Pavement Performance (LTPP) dataset. Some preprocessing methods and hyperparameter fine-tuning are conducted to improve the accuracy of the model. The performance of the LSTM+MA is compared with other state-of-the-art models, including Logistic Regressor (LR), Support Vector Regressor (SVR), Random Forest (RF), K-nearest-neighbor Regressor (KNR), Fully connected Neural Network (FNN), XGBoost (XGB), Recurrent Neural Network (RNN) and LSTM. The results show that the preprocessing methods can help the model learn quickly from the data and reach high accuracy with small epochs. Also, it shows that the proposed LSTM+MA model significantly outperforms other models, with coefficients of determination (R^2) of 0.965 and Mean Square Error (MSE) of 0.030 in the test datasets. Moreover, an overfitting score is proposed in this work to represent the severity degree of the overfitting problem and it shows that our proposed model does not suffer severely from overfitting.

Keywords: International Roughness Index, LSTM, Deep Learning, LTPP

INTRODUCTION

Pavement roughness is defined as the deviation of a pavement surface from an actual surface with characteristic dimensions that affect vehicle dynamics, ride quality, dynamic loads, and drainage (1). It is considered one of the most significant indicators of pavement condition as it dramatically affects driving comfort and safety, especially for high-speed-limit roads (2). Various factors including construction quality, maintenance, distress, climate, and traffic all affect pavement roughness (3). As pavements deteriorate over time due to these factors, it is essential to accurately predict their roughness condition for effective maintenance and rehabilitation planning. International Roughness Index (IRI) is widely recognized as a key indicator of pavement functional condition and ride quality, which is extensively utilized for evaluating road conditions (4). It is costly for transportation agencies to manually survey and monitor the pavement roughness (5). Therefore, predicting the IRI during pavement life becomes essential for decision-making and rehabilitation planning in transportation infrastructure management.

A lot of research and attempts have been made to estimate the roughness based on pavement and traffic conditions. Traditionally, many researchers have tried to model IRI using a linear regression method. However, the relationship between IRI and other parameters is highly nonlinear (6). Machine learning-based approaches are more advanced and become more popular to solve nonlinear problem. For example, Deng et al. (7) modified a feedforward network with particle swarm optimization to predict the rutting performance of asphalt pavement using material properties, structure parameters, and traffic conditions from collecting data from Idaho State. Wang et al.(8) developed a hybrid grey relation analysis and Support Vector Regressor (SVR) to predict pavement performance, obtaining an Root Mean Square Error (RMSE) of 0.298. However, overfitting becomes a major limitation of machine learning-based approaches. The model can easily learn from the training data too well, leading to a poor performance in unseen data. Ensemble models, which combine predictions from multiple individual models to make a final prediction, become popular in IRI evaluation because it takes various decisions from several approaches which can efficiently ease the overfitting issue (9). The idea behind ensemble learning is to leverage the diversity and collective intelligence of each individual model to improve overall predictive accuracy and robustness (10). In recent years, there have been numerous research works using ensemble learning in predicting pavement performance. For example, Wang et al. (11) developed an Adaboost regression model to improve the performance of predicting IRI by using the inputs including pavement thickness, service age, average annual daily truck traffic (AADTT), and cracks from Long-Term Pavement Performance (LTPP) dataset. The proposed model got an R^2 of 0.95 and Mean Square Error (MSE) of 0.0088 far outperforming linear regression models. Gong et al. (2) used a Random Forest Regression model to estimate the IRI value of flexible pavement by considering the distress, traffic, maintenance and structure data from LTPP. It outperformed the Linear Regression model (R^2 is 0.62) and got an R^2 more than 0.95 in both training and testing datasets. Damirchilo et al. (6) used XGboost to avoid overfitting of the training and handle the missing value in LTPP datasets. It obtained a R^2 of 0.7 outperforming the Random Forest (RF) and SVR. Song et al. (5) proposed a ThunderGBM-based ensemble learning model coupled with the Shapley Additive Explanation method to predict the IRI of asphalt pavements. 2699 observations were extracted from the LTPP database to train the model and the model achieved a satisfactory result with a R^2 value of 0.88 and RMSE of 0.08.

However, these machine learning-based methods did not consider the time-related characteristics of pavement roughness. The pavement would get its initial IRI based on the quality of construction. After construction, the IRI of the pavement changes with the pavement condition, climate, and traffic every day. In other words, IRI is a time-accumulated index that every previous factor would impact. Thus, models considering the time effect become the most popular direction. Recurrent Neural Networks (RNNs) are a class of neural networks designed explicitly for sequential data processing, making them well-suited for time series prediction tasks (12). Unlike other deep learning algorithms like Convolutional Neural Networks (13) or traditional Feed-forward Neural Networks (14), RNNs can capture dependencies and patterns in the temporal dynamics of the data, allowing them to model the evolving behavior of pavements over time. By utilizing the historical IRI measurements and other relevant variables in the LTPP dataset, we can train an RNN model to learn the underlying patterns and make accurate predictions

of future IRI values. Some research works have been done using an RNN to predict pavement performance. For example, Zhou et al. (1) proposed an RNN-based model to predict the IRI on the asphalt pavement based on the LTPP datasets. The loads, temperature, precipitation, evaporation, rutting and cracking are considered as input parameters for the analysis. The results showed that the presented RNN model reached a R^2 of 0.93. Han et al. (15) proposed a modified RNN model for falling weight deflectometer (FWD) back calculation. The model showed a stronger generalization ability than traditional Artificial Neural Networks.

However, a traditional RNN model may suffer from the vanishing gradient problem. It happens during the backpropagation through time when gradients diminish exponentially (16). This makes it challenging for traditional RNNs to capture long-term dependencies in the LTPP data, which are critical in time series prediction tasks. Moreover, standard RNNs have a simple memory mechanism that cannot selectively remember or forget information over longer sequences. This limitation can hinder their ability to effectively model complex patterns in the data and may not fully leverage historical information when predicting IRI values.

Therefore, in this work, we proposed a Long Short-Term Memory (LSTM) based model, LSTM+MA, which combines two LSTM blocks and a multi-head attention layer to predict the IRI performance of the pavement. In this model, one LSTM network is utilized as an encoder to capture important information from the entire sequence. Then, the cell state and hidden state from this LSTM model are transferred to the other LSTM network which works as a decoder to generate a sequence. The output layer from the encoder and decoder are combined together in the multi-head attention mechanism to focus on different parts of the encoded input during the decoding process. Data preprocessing and hyperparameter fine-tuning are used in this work to improve the models' performance. In addition, the presented LSTM+MA model is compared with other state-of-the-art models, including Logistic Regressor (LR), SVR, RF, K-nearest Neighbor Regressor (KNR), Fully-connected Neural Network (FNN), XGBoost (XGB), RNN and LSTM. The results show that our model outperforms other models as it gains the highest R^2 among all the tested models.

METHODS
Data engineering
To train the models, 25,167 samples are collected from the LTPP dataset. This dataset is a comprehensive and widely used resource in the field of transportation engineering. It is a collection of data gathered from pavement sections across the United States and Canada. The primary objective of the LTPP program is to monitor and evaluate the long-term performance of various pavement types under different conditions. In this work, predictors in traffic, pavement construction and climate are obtained from LTPP and utilized for training (**Table 1**). Climate-related factors, including precipitation, temperature, and Freeze-Thaw cycles, are considered in the training procedure. Pavement construction factors include initial IRI value, age, maintenance type, and transverse crack length. While for the traffic, the Average Annual Daily Traffic (AADT) is included in the train process.

TABLE 1 Predictor utilized in the prediction and its description

Predictor	Description
Traffic	
AADT	Average Annual Daily Traffic
Pavement construction	
IRI_0	The initial IRI value after the pavement was built.
Age	Years since the road was constructed.
M_{type}	Different types of maintenance interventions, such as crack sealing or overlay.
L_{tc}	Length of transverse crack.

T_s	For the collected samples, the surface type is either asphalt or concrete.
Climate	
T_{ave}	Average annual temperature
FT_c	Freeze-thaw cycle: number of days in a year when the maximum air temperature is greater than 0°C (32°F) and minimum air temperature is less than 0°C (32°F) on the same day.
P_{ave}	Average annual precipitation.
S	Amount of snow that will be received in an average year.
H_{ave}	Average annual humidity.

A higher volume of traffic can accelerate pavement deterioration, leading to increased roughness. By incorporating AADT as a predictor, the model can capture the influence of traffic on IRI. The initial IRI (IRI_0) is the most important parameter as it serves as a baseline and can influence the future evolution of the pavement condition (17). Pavement age since the start of measuring IRI is a critical factor in understanding the long-term performance of the pavement. It helps quantify the effects of ageing and deterioration processes on the roughness of the pavement. Cracking is one of the most common distresses observed in pavements. Including transverse crack length as a predictor helps the model consider the extent of cracking as a factor affecting roughness. Longer cracks may result in more significant roughness development over time. Maintenance activities can greatly influence pavement condition and roughness. Different types of maintenance interventions, such as crack sealing or overlay, can have varying effects on IRI. By incorporating maintenance type as a predictor, the model can learn how different maintenance actions impact the roughness of the pavement. Climate-related factors have a significant impact on the pavement's performance and condition. Changes in temperature, precipitation, humidity, snowfall, and freeze-thaw cycles can lead to cracking and deterioration.

Data preprocessing

We need to preprocess our data before inputting it into the model. We generated sequences that were between 1 and n historical samples. For example, if a road had 5 IRI measurements (through 2018 to 2022), four sequences would be generated with 1, 2, 3, and 4 historical elements with the goals of predicting this road's 2nd, 3rd, 4th, and 5th year's IRI value, respectively. These sequences contain the initial measurements and are continuous. However, as many sequences are subsets of other sequences and in order to prevent data leakage, we separated our train and test datasets by which roads were in them, not by randomly splitting the samples. The samples are arranged in order of least recent to most recent. Every sample has a feature representing the number of months since the road was constructed to when the measurement was made. There is also an empty sample appended to the end of every sequence that contains only the number of months since the construction to predict the IRI.

We also transposed the original input matrix to increase the training speed and accuracy of our models **(Figure 1)**.
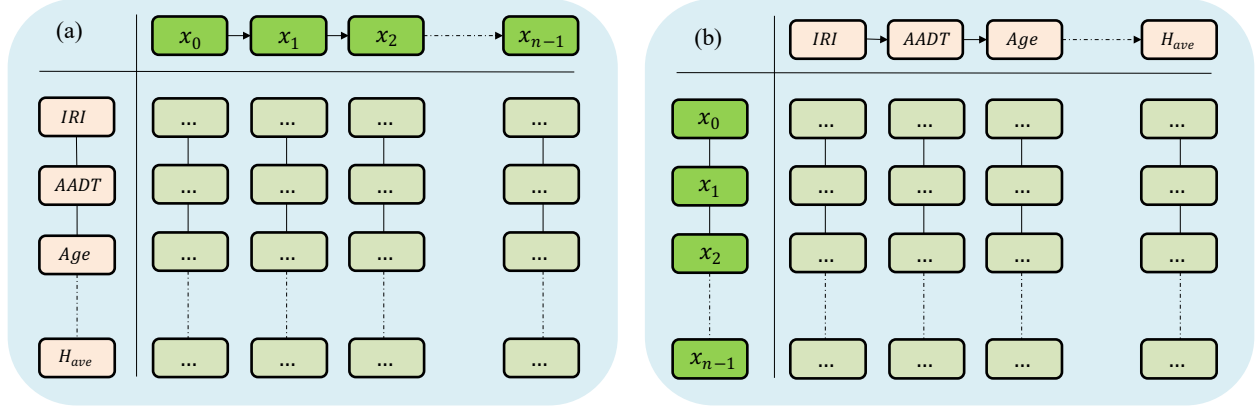
Figure 1. Dataset shape visualization: (a) Conventional time-series data; (b) Transposed time-series data.

As one of the main problems with processing and predicting this kind of data is retaining the impact of inputs from earlier in the input sequence, transposing the input matrix allows the model to process the entire series of events at the same time. In the transposed data, the model is not given a sequence of events. It is given a series of properties that have only an implied sequence and an explicit temporal order for all the data points. This greatly reduces the requirements on the model to remember and helps both models like the RNN which have problems with vanishing gradients and models like LSTMs and the LSTM+MA which already have other measures in place to help.

At the same time, some other preprocessing was applied to the data. The mean value of the left and right wheel path IRI was used as the IRI value of a road. Min-max scaling was applied to each parameter, which is particularly useful when the features have different scales or ranges, as it helps to normalize the data and bring all features to the same scale (0 to 1). It is commonly employed in machine learning algorithms that are sensitive to the scale of the input features.

LSTM+MA

The proposed LSTM+MA model is a sequence-to-sequence architecture that combines LSTMs and attention mechanisms. It is designed for predicting the IRI of pavement based on various predictors from traffic, pavement construction, and climate data (**Figure 2**).
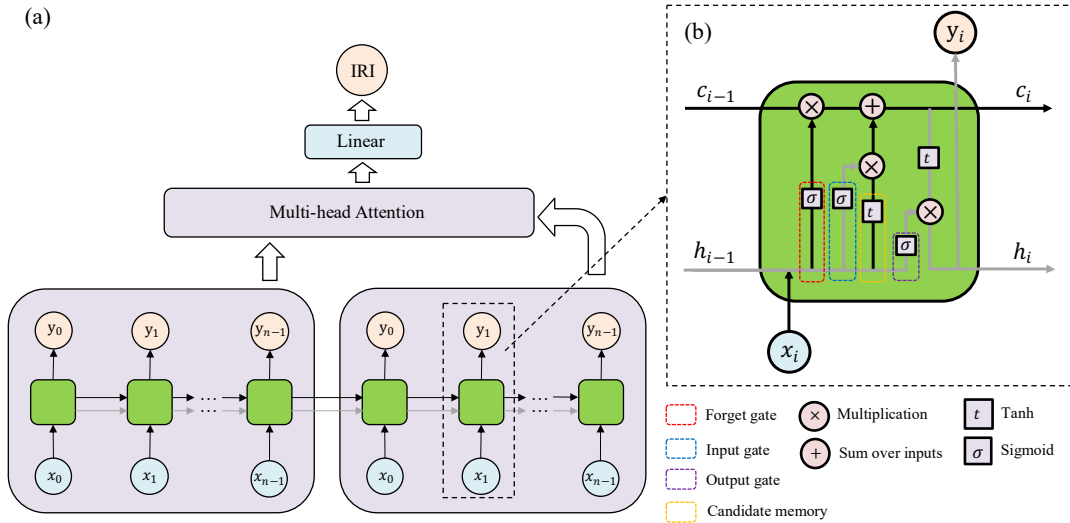

6

Figure 2. (a) The architecture of the proposed LSTM+MA model. x_i stands for the input vector and y_i is the output vector. (b) The structure of the LSTM layer. c_i is the cell state and h_i is the hidden state.

The presented LSTM+MA follows an encoder-decoder architecture, which is commonly used for sequence-to-sequence tasks (**Figure 2 (a)**). The encoder processes the input sequence $(x_0, x_1, \dots, x_{n-1})$ and captures its temporal information, while the decoder generates the output sequence $(y_0, y, \dots, y_{n-1})$ based on the information provided by the cell state (C_i) and hidden state (h_i) from encoder. LSTM is a type of RNN that addresses the vanishing gradient problem and allow for capturing long-term dependencies in sequential data (*18*). The encoder and decoder both utilize LSTM layers, enabling the model to handle variable-length input and output sequences effectively. The LSTM component can be broken down into a few specific components that are responsible for different parts of the model behavior. The input gate, forget gate, output gate and candidate for cell state update in the LSTM component can be explained using **Equation 1, 2, 3 and 4**.

$$i_i = \sigma(W_{ii}x_i + b_{ii} + W_{hi}h_{i-1} + b_{hi}) \qquad (1)$$
$$f_i = \sigma(W_{fi}x_i + b_{fi} + W_{hf}h_{i-1} + b_{hf}) \qquad (2)$$
$$o_i = \sigma(W_{oi}x_i + b_{oi} + W_{ho}h_{i-1} + b_{ho}) \qquad (3)$$
$$g_i = \tanh(W_{gi}x_i + b_{gi} + W_{hg}h_{i-1} + b_{hg}) \qquad (4)$$

Where σ is the sigmoid function, tanh stands for the tangent hyperbolic function, i_i is the input gate, f_i is the forget gate, o_i is the output gate, g_i is the candidate memory, W is the weight matrix and b is the bias. The input, forget, and the candidate memory allow the model to learn what data to retain in the cell matrix as it processes the sequence. The output gate updates the hidden state of the model out provides that to the next iteration. These components are then combined to generate the hidden and cell states as shown in **Equation 5 and 6**.

$$h_i = o_i \odot tanh(c_i) \qquad (5)$$
$$c_i = f_i \odot c_{i-1} + i_i \odot g_i \qquad (6)$$

Where h_i is the output of the LSTM layer and the hidden layer for the next LSTM layer in the sequence. It can be calculated as the Hadamard product, or element wise product of the output gate and the current cell state. c_i is the current cell state and is the sum of the Hadamard product of the forget gate and the last cell state and the product of the input gate and the candidate gate. This step forgets old information and stores the new information that the model has.

The attention mechanism enhances the model's ability to focus on relevant parts of the input sequence when generating the output. This is especially useful in long sequences, as it helps the model to attend to the most informative time steps and ignore noise or less relevant information.

Multi-head attention (*19*) is used as the attention mechanism in this work because it allows the model to attend to different positions within the input sequence simultaneously. It computes multiple attention heads, each capturing different patterns and relationships in the data. Moreover, it helps the model capture complex dependencies and relationships between predictors, which can be valuable in pavement IRI prediction, where the performance is influenced by multiple factors. The attention mechanism embedded improves the model's interpretability by highlighting the importance of different parts of the input sequence during the prediction process. After applying the attention mechanism, the model uses a linear layer to transform the output to the desired prediction format (IRI values). The output is reshaped to match the appropriate dimensions for the final prediction.

There are some advantages of our proposed LSTM+MA model: (1) The LSTM layer is a specialized variant of RNNs, designed to address the vanishing gradient problem and facilitate the modeling of long-term dependencies in sequential data. By incorporating memory cells and gating mechanisms, LSTMs can effectively retain and update information over extended time steps, making

them particularly well-suited for time series prediction tasks (**Figure 2 (b)**). The cell state (C_i) is controlled by the forget gate, input gate, and candidate memory for each time step. By applying element-wise multiplication to the forget gate and the previous cell state, the LSTM forgets irrelevant information. The input gate and the candidate memory determine which new information is incorporated into the updated cell state. The resulting cell state retains valuable historical context while selectively integrating new information, facilitating the modeling of long-term dependencies in the data. The hidden state (h_i) is the LSTM layer's output for the current time step, representing the processed and summarized information that is passed to the next time step. It is calculated by applying element-wise multiplication to the cell state and the output of the output gate. The hidden state captures relevant information from the current input and retains valuable historical context from previous time steps, enabling the LSTM to carry forward essential features as it processes the sequence. (2) The attention mechanism in the model allows it to focus on the entire context of the encoded and generated sequence to predict the future state more accurately as well as make the model be able to consider a much larger context. (3) The model's ability to process variable-length sequences makes it more flexible and adaptable to real-world pavement data, where the length of historical data may vary for different road sections. The multiheaded attention layer further enhances the model's capacity to capture diverse patterns and relationships between predictors, which can lead to better predictive performance.

Evaluation Metrics

The performance of models is evaluated by MSE and R^2. MSE measures the average of the squared differences between the predicted results and actual values. It can provide a measure of how well the model's predictions align with the true values, with higher values indicating greater errors, as shown in **Equation 7**.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i) \qquad (7)$$

Where n is the number of total evaluated data, y_i is the i-th data point, $\hat{y}_i$ is the prediction of the i-th data.

R^2 stands for the coefficient of determination. It measures the proportion of the total variation in the prediction that can be explained by the model. It is used to measure the accuracy of a model's fit and can be calculated using **Equation 8**.

$$R^2 = \frac{SSR}{SST} = \frac{\sum_{i=1}^{n}(\hat{y}_i - \bar{y})}{\sum_{i=1}^{n}(y_i - \bar{y})} \qquad (8)$$

Where SSR is the sum of squares regression, SST stands for the sum of squares total, $\bar{y}$ is the average value of the data.

To evaluate and compare the performance between different models statistically, each model is trained and tested three times on randomly split train and test datasets. The average value and standard deviation would be calculated based on the tests. By doing this, it can show the general accuracy of the model as well as the robustness of the model.

The overfitting problems are also considered in this work. Overfitting occurs when the regression models learn the training data too well, resulting in poor generalization to unseen data. The difference between the training and testing performance is considered as an overfitting score in this work as shown in **Equation 9**. It is utilized as a metric to quantify the overfitting issue in the selected approach.

$$S_o = R^2_{train} - R^2_{test} \qquad (9)$$

Where S_o stands for the overfitting score, R^2_{train} is the R^2 value when model applying in the train set, R^2_{test} is the R^2 value when model applying in the test set. It can provide a measure of the overfitting degree in the model, with higher values indicating greater overfitting problem.

Overall Procedure

The aim of the work is to propose a model that can predict the IRI value with higher accuracy based on the predictors extracted from the LTPP dataset. First, data engineering is applied on the LTPP data to seek some useful and related predictors for the IRI prediction. Then, data preprocessing is applied to process the data. After a random train-test split, the proposed model would be compared with other state-of-the-art models to show its high performance (**Figure 3**).
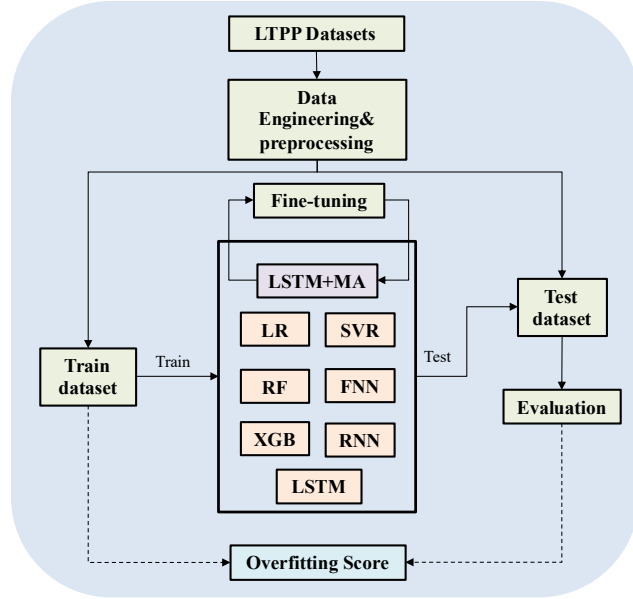

Figure 3. The overall procedure of the work.

Seven models, which are popular used in other research, are trained and tested on this dataset and compared with our proposed method. Models include LR, SVR, RF, KNR, FNN, XGB, RNN and LSTM. The details of these models are shown below (**TABLE 2**). Each model would be run three times based on different train-test split. By doing this, the average and standard deviation of these three running could be calculated which can make the experiments statistically meaningful. A hyperparameter fine-tuning process would be conducted in the LSTM+MA model to find the best performance it can reach. In this approach the possible parameters, including the input sequence length, the learning rates, the leaning-rate decay rate, the LSTM layer dropout, the number of LSTM layers, the batch size, and the number of training epochs, are selected to determine the optimal permutation of input parameters for the model's performance.

TABLE 2 The description of all the models used in this work to compare with. SL stands for supervised learning, EL stands for ensemble learning and ANN stands for artificial neural network.

Model	Type	Description
LR	SL	Logistic Regression models the probability of an input belonging to a particular class using a logistic (sigmoid) function. It is a simple and interpretable algorithm that can be used for both binary and multi-class classification problems.
SVR	SL	It is an extension of Support Vector Machines (SVM) for regression, where it tries to find a hyperplane that best fits the data while minimizing the error.

RF	EL	Random Forest is based on the concept of constructing multiple decision trees during training and then combining their predictions to make a final prediction. Each decision tree is trained on a random subset of the data and a random subset of features, which helps reduce overfitting and improve generalization.
KNR	SL	KNN Regressor is a non-parametric supervised learning algorithm used for regression tasks. It makes predictions based on the average (or weighted average) of the k-nearest data points in the training set to the new input data.
FNN	ANN	A Feedforward Neural Network is an artificial neural network that consists of a series of linear layers. To make predictions data flows from the beginning of the network through its layers, each of which contains densely connected neurons or nodes that process the data.
XGB	EL	It is a variant of gradient boosting that builds multiple weak learners (often decision trees) sequentially, where each new learner corrects the errors made by the previous ones.
RNN	ANN	RNN is a type of neural network designed to handle sequential data, making it suitable for time-series analysis. Unlike traditional feedforward neural networks, RNNs have connections that form cycles, allowing them to maintain hidden states and process sequences of data. RNNs are especially useful for tasks where the current output depends on previous inputs and the order of the data matters.

Machine learning models are implemented using scikit-learn and pytorch, two machine learning libraries for Python. The data processing methods and models are all implemented in Python and computed under the following machine speculations: Ubuntu 22.04.2 LTS, AMD Ryzen 9 5900, NVIDIA RTX 3080 with 16GB RAM, 16GB of system RAM.

RESULTS

Figure 4 shows the difference of MSE and R^2 score during the training and testing procedure when using original dataset and the transposed dataset for the model LSTM+MA.
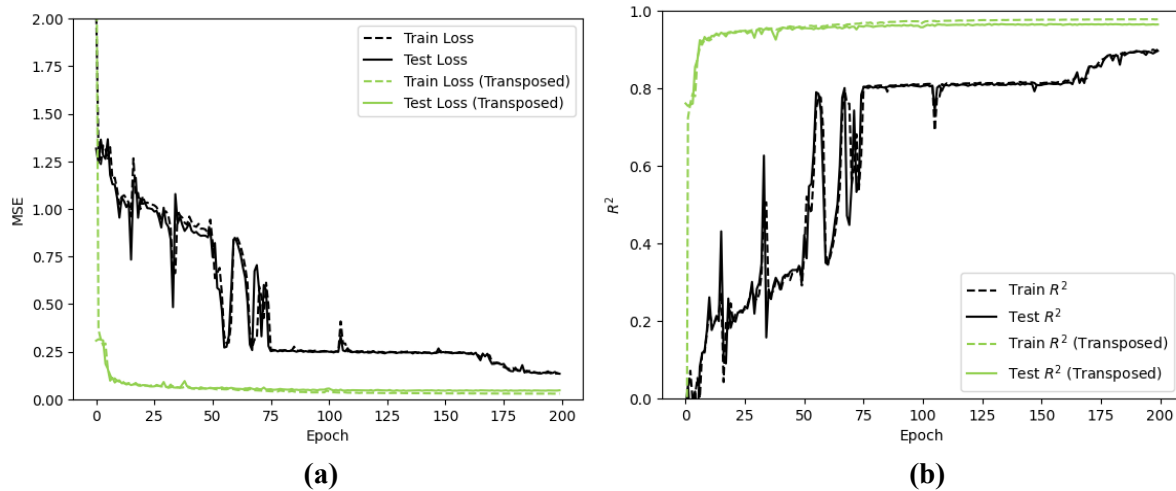

(a)　　　　　　　　　　　　　　　　　(b)

Figure 4. Graphs that compare the MSE and R^2 scores in the training and testing procedure with the origonal dataset and the transposed input dataset for the LSTM+MA model: (a) MSE; (b) R^2 scores.

It shows that no matter using original data or the transposed dataset, the predicting MSE is decreasing with the increasing of running epochs and the R^2 score increases with more running iterations. However, when the transposed and origonal time-series datasets are comapred by training models with each dataset and looking at the MSE and R^2 over the training sequence, it can be seen that the transposed

10

data leads to a significant improvement in training time and accuracy (**Figure 4**). By utilizing the transposed data, the MSE drops dramatically and quickly to a very low state around epoch 25 which is even lower than the model in epoch 200 when using original data. In other words, the model learns from the data much quicker when the LTPP data is transposed. Moreover, the method with transposed data as preprocessing performs much smoother in MSE and R^2. The model trained on the original data shows a sawtooth shape on the relationship between MSE (or R^2) with epochs. It means the model cannot learn very stable from the original data.

It is interesting to find that the test data closely mirrored the performance of the models on the training data no matter using original dataset or processed dataset, which means that the model demonstrates robustness to noise and generalize well to unseen data.

After preprocessing the data, a grid-search approach is utilized to determine the optimal parameters for the LSTM+MA model. The input sequence length, the learning rates, the leaning-rate decay rate, the LSTM layer dropout, the number of LSTM layers, the batch size, and the number of training epochs were selected to be optimized. We discover that the optimal sequence length for the LSTM-MA model is the last 5 elements and the optimal number of layers for the encoder and decoder is 3. The optimal parameters from the hyperparameter tuning are shown in **Table 3**.

TABLE 3 Hyperparameter tuning results

Parameter Name	Value
Learning Rate	0.01
Number of LSTM Layers	3
Input Sequence Length	5
Learning rate decay	0.8
Batch size	512
Epochs	200
Dropout	0.25

After determining the best performance of our proposed model, we compare this model to other state-of-the-art models including LR, SVR, RF, KNR, FNN, XGB, RNN, and LSTM. **Figure 5** shows the R^2 and MSE values from the models' comparison results.
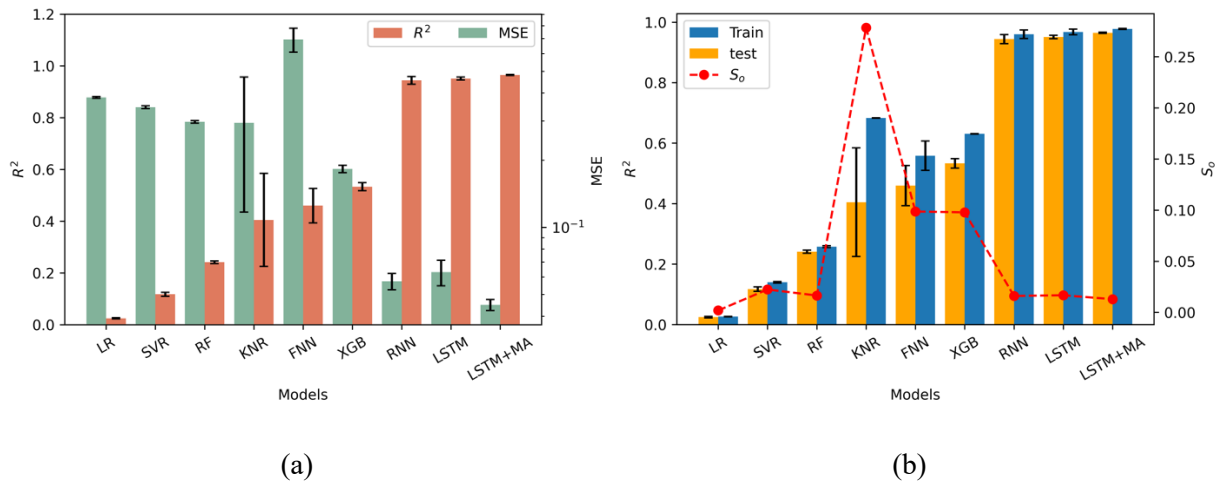

(a)   (b)

Figure 5. The performance of all the models: (a) The R^2 and MSE value of all the models in test dataset. The models are organized in ascending order based on the R^2 which is achieved on the test dataset. The red bar stands for the R^2 and the green bar stands for the MSE. The black error bar stands for the standard deviation; (b) The model performance on the train dataset and test dataset. The blue bar is the R^2 value of the model on the train dataset while the yellow bar is the R^2 score on test dataset. The red dash line is the overfitting score.

11

Figure 5(a) is organized in ascending order by R^2 value of each model achieved on the test dataset. It shows that our proposed LSTM+MA model outperforms other models in the LTPP dataset as it gets the highest R^2 value (0.965) and the lowest MSE (0.030). It is noteworthy that the models which consider the time-related impacts, including RNN, LSTM and LSTM+MA, have higher accuracy than the models which don't consider the time-accumulated influence. The black error bar is the standard deviation when running in randomly split train-test datasets. A small standard deviation means the model performs stable in different data inputs. In other words, it can represent the robustness of the model. It is interesting to find that the KNR gets the apparent larger standard deviation than other models both in R^2 and MSE. This is because the KNR algorithm is predicted based on the K nearest points which means it is highly related to the interaction among the data points. Our proposed model gets the lowest standard deviation (0.0026) in R^2 value, which means it obtains a stable performance and is much more robust than other models. Moreover, the high performance of RNN, LSTM and LSTM+MA proves that the pavement IRI is a time-related factor and using time-series model can perform better on IRI prediction.

Overfitting is a serious problem in the long-term pavement performance prediction. It occurs when the model learns the training data too well, resulting in poor generalization. Therefore, to obtain the performance difference on the train and test dataset of each model, the R^2 score from the training and testing performance are considered and shown in **Figure 5(b)**. The red dash line represents the overfitting score (S_o) among all the models. It shows that KNR is the most overfitted model compared to others. Although LR obtains the lowest overfitting problem with a S_o of 0.0018, the performance of LR model is the worst among all the models. The LSTM+MA gets the second lowest S_o (0.0128) except LR and at the same time, it obtains the highest R^2 value. In other words, our proposed model keeps high accuracy as well as a low overfitting problem in predicting the IRI value based on the LTPP dataset.

DISCUSSION

The proposed LSTM+MA model adjusting the LSTM to make it combine with multi-head attention reaches the highest R^2 value compared to other models. It leads to a higher performance than using a plain LSTM, which means that the embedded attention mechanism does improve the model information extraction and interpretation capacity.

The model that can handle time-series data shows a dominant performance than other models. This is because the pavement IRI, as well as friction, rutting and deflection, are all time-series data, which means the traffic, climate and other factors would have an accumulated influence on these performances with time. In other words, our presented model is not only limited in predicting the pavement IRI, but also can be used to predict other pavement performance including friction, deflection and transverse profile as they are all time-series data. Moreover, it can also be used in other fields like stoke price predicting or weather prediction where the data is highly time related.

Although there are important improvements revealed by this study, there are also limitations. First, the choosing of predictors are not considered thoroughly. We must point out that we do not discuss the dependency between each predictor which might introduce redundancy and overfitting into the model. However, these problems could be solved if we consider using a feature/component ablation section to choose the most related predictors. Also, we could apply Principal Component Analysis (PCA) to reduce the dimensionality of data to avoid noise and redundancy.

CONCLUSIONS

A LSTM based method LSTM+MA is proposed in this work to predict the pavement IRI based on time-series data extracted from the LTPP database. Traffic, Pavement construction and Climate are considered the most important factors to the contribution of IRI and are utilized as predictors to train the models. Our proposed approach combines two LSTM layers as an encoder and decoder to extract the time-related information from the dataset. A multi-head attention layer is connected with the LSTM layer to allow the model to attend to different positions within the input sequence simultaneously, which can

improve the model's interpretability by highlighting the importance of different parts of the input sequence. Some data preprocessing method like data transpose and min-max scale, as well as hyperparameter fine-tuning are utilized to improve the performance of models. It demonstrates that transposing the data matrix can apparently accelerate the convergence speed and helps the model to learn from data more quickly and adequately.

State-of-the-art models including LR, SVR, RF, KNR, FNN, XGB, RNN and LSTM are implemented and compared with our presented method. Results show that the proposed LSTM+MA achieves a higher accuracy than other models as it obtains the highest R^2 value (0.965) and the lowest MSE (0.030). In addition, the overfitting problem is studied and an overfitting score is proposed in this work to quantify the degree of overfitting of models. It shows that the LSTM+MA model obtains the second lowest overfitting score (0.0128) except LR. The proposed method keeps high accuracy and low overfitting degree in predicting the IRI value based on the LTPP dataset.

AUTHOR CONTRIBUTIONS

The authors confirm contribution to the paper as follows: study conception and design: Tianjie Zhang, Yang Lu; data collection: Alex Smith; analysis and interpretation of results: Tianjie Zhang, Alex Smith; draft manuscript preparation: Tianjie Zhang, Alex Smith. All authors reviewed the results and approved the final version of the manuscript.

REFERENCES

[1] Zhou, Q., E. Okte, and I. L. Al-Qadi. Predicting pavement roughness using deep learning algorithms. *Transportation Research Record,* Vol. 2675, No. 11, 2021, pp. 1062-1072.

[2] Gong, H., Y. Sun, X. Shu, and B. Huang. Use of random forests regression for predicting IRI of asphalt pavements. *Construction and Building Materials,* Vol. 189, 2018, pp. 890-897.

[3] Mazari, M., and D. D. Rodriguez. Prediction of pavement roughness using a hybrid gene expression programming-neural network technique. *Journal of Traffic and Transportation Engineering (English Edition),* Vol. 3, No. 5, 2016, pp. 448-455.

[4] Kaloop, M. R., S. M. El-Badawy, J. W. Hu, and R. T. Abd El-Hakim. International Roughness Index prediction for flexible pavements using novel machine learning techniques. *Engineering Applications of Artificial Intelligence,* Vol. 122, 2023, p. 106007.

[5] Song, Y., Y. D. Wang, X. Hu, and J. Liu. An efficient and explainable ensemble learning model for asphalt pavement condition prediction based on LTPP dataset. *Ieee Transactions on Intelligent Transportation Systems,* Vol. 23, No. 11, 2022, pp. 22084-22093.

[6] Damirchilo, F., A. Hosseini, M. Mellat Parast, and E. H. Fini. Machine learning approach to predict international roughness index using long-term pavement performance data. *Journal of Transportation Engineering, Part B: Pavements,* Vol. 147, No. 4, 2021, p. 04021058.

[7] Deng, Y., and X. Shi. An accurate, reproducible and robust model to predict the rutting of asphalt pavement: neural networks coupled with particle swarm optimization. *Ieee Transactions on Intelligent Transportation Systems,* Vol. 23, No. 11, 2022, pp. 22063-22072.

[8] Wang, X., J. Zhao, Q. Li, N. Fang, P. Wang, L. Ding, and S. Li. A hybrid model for prediction in asphalt pavement performance based on support vector machine and grey relation analysis. *Journal of advanced transportation,* Vol. 2020, 2020, pp. 1-14.

[9] Ikeagwuani, C. C., and D. C. Nwonu. Statistical analysis and prediction of spatial resilient modulus of coarse-grained soils for pavement subbase and base layers using MLR, ANN and Ensemble techniques. *Innovative Infrastructure Solutions,* Vol. 7, No. 4, 2022, p. 273.

[10] Webb, G. I., and Z. Zheng. Multistrategy ensemble learning: Reducing error by combining ensemble learning techniques. *IEEE Transactions on Knowledge and Data Engineering,* Vol. 16, No. 8, 2004, pp. 980-991.

[11] Wang, C., S. Xu, and J. Yang. Adaboost algorithm in artificial intelligence for optimizing the IRI prediction accuracy of asphalt concrete pavement. *Sensors,* Vol. 21, No. 17, 2021, p. 5682.

[12] Sherstinsky, A. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D: Nonlinear Phenomena,* Vol. 404, 2020, p. 132306.

[13] Zhang, T., D. Wang, A. Mullins, and Y. Lu. Integrated APC-GAN and AttuNet Framework for Automated Pavement Crack Pixel-Level Segmentation: A New Solution to Small Training Datasets. *Ieee Transactions on Intelligent Transportation Systems,* Vol. 24, No. 4, 2023, pp. 4474-4481.

[14] Abd El-Hakim, R., and S. El-Badawy. International roughness index prediction for rigid pavements: an artificial neural network application. *Advanced Materials Research,* Vol. 723, 2013, pp. 854-860.

[15] Han, C., T. Ma, S. Chen, and J. Fan. Application of a hybrid neural network structure for FWD backcalculation based on LTPP database. *International Journal of Pavement Engineering,* Vol. 23, No. 9, 2022, pp. 3099-3112.

[16] Dong, Y., Y. Shao, X. Li, S. Li, L. Quan, W. Zhang, and J. Du. Forecasting pavement performance with a feature fusion LSTM-BPNN model.In *Proceedings of the 28th ACM international conference on information and knowledge management*, 2019. pp. 1953-1962.

[17] Chen, X., H. Zhu, Q. Dong, and B. Huang. Optimal thresholds for pavement preventive maintenance treatments using LTPP data. *Journal of Transportation Engineering, Part A: Systems,* Vol. 143, No. 6, 2017, p. 04017018.

[18] Sak, H., A. Senior, and F. Beaufays. Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. *arXiv preprint arXiv:1402.1128*, 2014.

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