The Automatic Driving of Tunnel Boring Machine

Jinpu Cao (06439552)
Civil and Environmental Engineering
Stanford University

***Abstract*—A huge volume of data can be collected by a Tunnel Boring Machine (TBM) during tunneling. The collected data enables the possibility of a data-driven prediction model of TBM performance. This study develops models for predicting the TBM operation parameters based on the Long Short-Term Memory (LSTM). Given the initial operating parameters for a short period of time, the model can predict some future interpolating points of these parameters in the driving cycle. The automatic driving of TBM can be achieved by updating the predictions with the most recent parameter series.**

Keywords—Tunnel Boring Machine, Time Series, Long Short-Term Memory, Sequence to Sequence

I. Introduction

Since my undergraduate major is underground construction and engineering, various tunneling techniques, such as TBM, are familiar to me. Tunnel Boring Machine (TBM), which known as a "mole", is a machine used to excavate tunnels with a circular cross section through a variety of soil and rock strata. Fig. 1. shows the construction of a tunnel by using TBM. More than one hundred sensors installed on the TBM record parameters every second. Among all these parameters, rotate speed of cutterhead N, advance speed V, total thrust F and cutterhead torque T are most important. Generally, in a driving cycle, the operator makes adjustment about the state of surrounding rock mass and TBM according to F and T. Then, N and V will be adjusted accordingly based on the operator's experience.

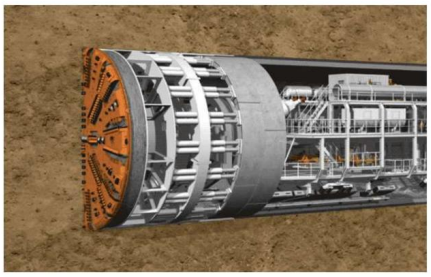

Fig. 1. Tunnel Boring Machine (TBM) Tunneling

It is a challenge for anyone to operate a TBM. Machine learning makes it possible to use huge amounts of historical data to aid operators of TBM. The study is about to drawing the whole curves of three parameters V, F and T (the change in N is abrupt) with their initial series given. The main model in the study takes sequence and returns sequence. Specifically, the inputs are V, F and T series ($timestep \times 3$) in the initial period and the outputs are V, F and T series ($timestep2 \times 3$) in the future.

II. Related Work

There have been some works on the prediction of TBMs' operating parameters. In general, the methods for TBM's load prediction can be grouped into three categories: empirical methods (combined with experiments) [1,2], rock-soil mechanics methods [3,4]and numerical simulation methods [5]. The empirical methods, like many engineering standards, can only guide the operation of TBM in a general direction. The rock-soil mechanics methods need certain geological survey data which is not available everywhere. Compared to traditional numerical simulation methods like multiple regression model, models armed with machine learning techniques can give better prediction of TBM operation data, like [6] uses a support vector regression model for predicting tunnel boring machine penetration rates. For prediction related to time series such as the report discusses, deep learning, a subfield of machine learning can play to its strengths. [7] established a model based on LSTM to predict tunneling parameters in the steady phase based on the data in the rising phase. [8] established a global attention mechanism based long-short-term-memory (LSTM) network to model the cyclic TBM construction data and make predictions of lithology at the tunnel face. [9] used three kinds of recurrent neural networks (RNNs), including traditional RNNs, long-short term memory (LSTM) networks and gated recurrent unit (GRU) networks, to deal with the real-time prediction of TBM operating parameters based on TBM in-situ operating data. [7,8,9] show RNN and its variants are powerful in dealing with the TBM operation data. The object of the study is like [9]. The difference is that [9] used the parameters of the first 5 seconds to predict the parameters of the 6th second and update, however, the report uses the parameters of the first 30 seconds to predict the whole series of parameters in the future, which is more helpful to TBM operators.

III. Data and Features

In my junior year (2019), I am lucky to have access to a TBM dataset. China Railway Engineering Equipment Group Co., Ltd. The dataset is collected in the Jilin Yinsong Water Diversion Project, China. The TBM performed effective tunneling for 728 days, and TBM data were collected at a frequency of 1 Hz. In total, 120 GB of monitoring data were collected. The data included 199 parameters, such as the cutterhead torque and total thrust. The TBM tunneling time series data are usually stored in the form of continuous cycle with the on-off of the TBM machine.

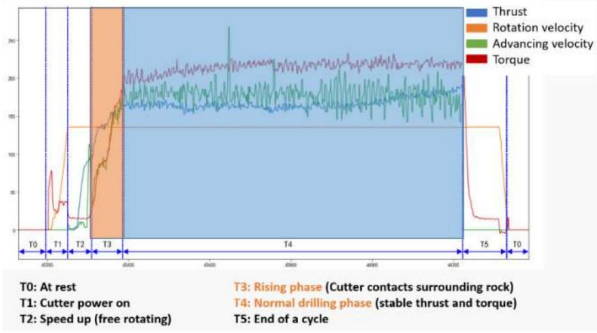

Fig. 2. A complete Cycle of TBM Tunneling

Generally, a cycle can be divided into three phases roughly (Fig.2.). In the first phase (T1+T2), the cutter powers on and the TBM advances freely. In the second phase (rising phase, T3), the cutter contacts the surrounding rock. So, this is a transition period, and many parameters will change violently. This stage will continue no more than 150 seconds. In the third phase (steady phase, T4), the TBM sustains a stable relationship with surrounding rocks. This phase, lasting several minutes, is a focus in tunnel construction. At this phase, many parameters have a stable value.

A. Preprocess

Raw data is stored in TXT files in days (728). The daily file contains a varying number of tunneling cycles (Fig.3.). The driving cycles can be achieved from the daily TXT files based on the condition that none of the key parameters (N, V, F, T) are zero. At the same time, according to the rated values of the TBM (N_{rated}=7.6 rmp, V_{rated} =120 mm/min, F_{rated}=23260 kN, T_{rated}=8410kN·m), exclude outliers.

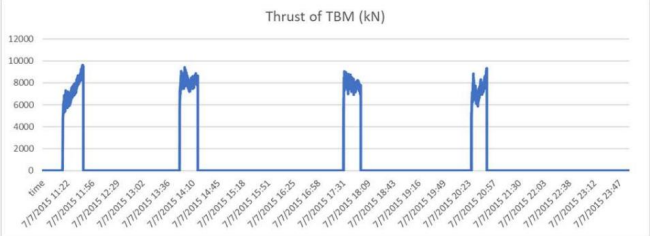

Fig. 3. Several Cycles in a Daily Record

Under ideal conditions, the drilling footage of a complete TBM tunneling cycle is 1.8 m (more than 1000 seconds). However, the footage of each cycle considerably varies due to unexpected events in the tunneling process, such as unfavorable geological conditions. This study takes the period of a single tunneling cycle greater than 600 seconds as the extraction condition to avoid inaccurate data collection caused by the short TBM tunneling period. In the end, a total of 10256 tunneling cycles were obtained to validate lithology prediction. In the experiment, only 1067 tunneling cycles are used due to time constraint.

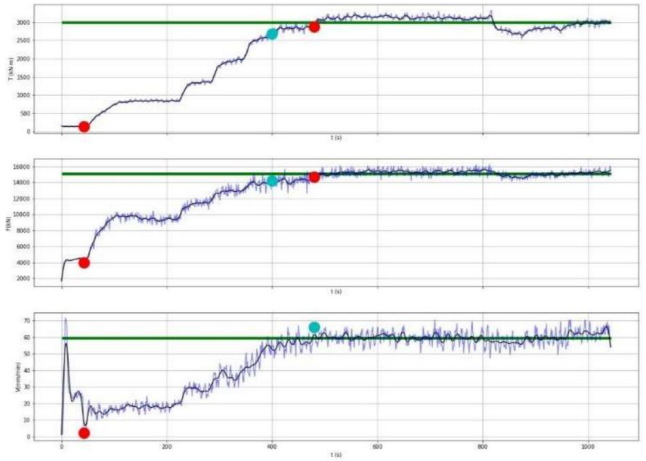

Fig. 4. The Result after Preprocessing

Basically, the operator will accelerate TBM at the very beginning. The velocity will rapidly fall when the cutterhead starts to interact with the tunnel face closely and then the key parameters start to raise gradually. It is generally believed that the data starting from this time, i.e. in raising phase and steady phase, can reflect the interaction relationship between operator, machine and rock mass. For the purpose of the study, i.e. predicting the whole curve with the initial sequences given, it is a pivotal preprocess to find the starting time of raising phase t_{raise} and stable phase t_{stable}, and also the steady values of key parameters V_s, T_s and F_s. In the research, the following algorithm is implemented to find these.

Algorithm: Extract Raising and Steady Phase from a Cycle
1: Calculate the stable values of V_s, T_s and F_s based on the fact that in a cycle most values fluctuate around the stable value. The frequency distribution histogram is used here.
2: The start time of stable phase t_{stable} is identified when the total thrust F begins to be stable (the gradient is less than some tolerance).
3: In velocity sequence, find corresponding time t_0 of the peak with the greatest prominence from the beginning to stable phase. Here, scipy.signal.find_peaks in python library is used. Find the corresponding time of minimum of velocity from t_0 to t_{stable} as the start time of raising time t_{raise}.

In order to improve the accuracy of the prediction, Savitzky–Golay filter is used here. A Savitzky–Golay filter is a digital filter that can be applied to a set of digital data points for the purpose of smoothing the data, that is, to increase the precision of the data without distorting the signal tendency [11]. Scipy.signal.savgol_filter in python library is imported when implementing. The length of the filter window is 81 and the order of the polynomial used to fit the samples is 4 based on the multiple attempts.

One example after preprocessing is shown in Fig.4. There are 3 subplots representing torque $(T, kN \cdot m)$, total thrust (F, kN) and velocity $(V, mm/min)$ versus time $(t, second)$, respectively. In each subplot, the fluctuating purple line is the original cycle data split from raw dataset. The stable black line is the data smoothed. The left red point is the start point of raising phase and the right cyan point is the start point of stable phase. The bolded green line is the stable value of each parameter. Normalize all parameters via dividing by their rated values, i.e., substitute V/V_{rated}, F/F_{rated} and T/T_{rated} for V, F and T. The V, F and T sequences in the raising phase and their stable values after preprocessing will be the dataset for the deep learning later. The inputs and outputs will be extended in the METHOD part.

IV. Method

A. LSTM Networks

The long short-term memory (LSTM) network, which was proposed in 1997 [10], is a special variant of recurrent neural networks (RNNs). RNNs provide a powerful tool for the tasks of time-series prediction, because of the existence of loop structure where the previous outputs of hidden nodes are used to generate the response for the current input [9]. Fig. 5. Shows the structure of RNNs. In this figure, $\vec{x}_t$ is the input vector at t time. $\vec{h}_t$ is the hidden output vector at t-time. $\vec{o}_t$ is the output vector at t time. W_t is the weight matrix between the hidden output vector at $t-1$ time and the hidden output vector at t time. U_t is the weight matrix between the input vector at t time and the hidden output vector at t time. V_t is the weight matrix between the hidden output vector at t time and the output vector at t time.

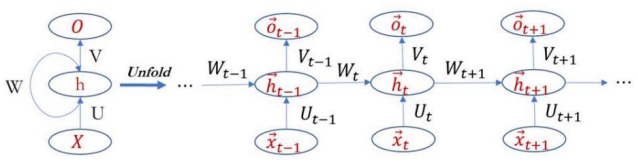

Fig. 5. The Structure of RNNs

Traditional RNNs exist vanishing gradient problems since the multiple matrix multiplications involved. LSTM can solve the vanishing gradient problems and get better performance in long-term time series prediction. The diagram in Fig. 6. is a typical LSTM unit. The key to LSTM is the cell state ($\vec{C}_t$). In the diagram, vector concatenation is represented by the merging arrows (+), while vector duplication is represented by the forking arrows (×).

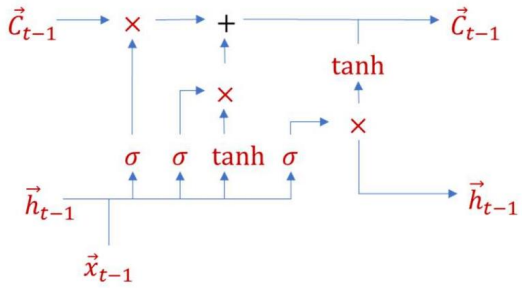

Fig. 6. The Structure of LSTM

The gates in LSTM have been designed to remove or add information to the cell state. They are a composition of a sigmoid layer and multiplication operations. The sigmoid layer output is a value between 0 and 1, which indicates the weight of information flow. The LSTM achieves the control and protection of the cell state through these three gates. The forget gate decides what information will be disposed of from the cell state. This transfer can be defined as:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (1)$$

where σ is the sigmoid activation function, W_f is the weight of the connections between neurons, h_{t-1} is the output of the last neuron, x_t is the input of the current neuron, and b_f is the bias of neuron. The forget gate determines the effect of the input on the current cell state, and the preservation and discarding of the previous cell state.

The input gate determines how much new information will be stored in the current cell state. This transfer can be defined as:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (3)$$

where tanh is the activation function, and b_i and b_C are the biases of the neural network. The sigmoid layer decides which value will be updated, while the tanh layer creates a new vector $\tilde{C}$ which can be added to the cell state. After completing the above steps, the updated cell state can be defined as:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (4)$$

where $*$ is an element-wise production operation. In the updating procedure of the cell state, the old cell state C_{t-1} is multiplied by f_t to forget some information, and then, the new candidate value $i_t * \tilde{C}_t$ from the input gate is added. After the cell state has been updated, the output gate will output the current cell state. This transfer can be defined as:

$$\sigma_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (5)$$

$$h_t = \sigma_t * \tanh C_t \quad (6)$$

In summary, the above three gates are composed of sigmoid and tanh neural network layers, which help in the selection of effective information. The above description mainly refers [7].

B. Prediction Method

The object of the study is to achieve predicting the whole curve of V, T and F in a tunneling cycle with the initial sequence given and updating the curve with TBM advancing. Define τ as the length of sequence of inputs, which is always known as the time steps of inputs, n as the length of the whole sequence. An intuitive way to do this is setting inputs to initial sequence and outputs to the remaining/future sequence and training the mass samples with LSTM or other edge tools. One training sample packing inputs and outputs together can be written as follows:

$$(\{x_1, x_2, \dots, x_\tau\}, \{x_{\tau+1}, x_{\tau+2}, \dots, x_n\})$$

Where $x_i (\, i = 1, \cdots, n)$ means the parameters vector at i time, i.e., $x_i = [V_i, T_i, F_i]^T$. This method is feasible, but the results of prediction are not satisfying. Another method is predicting step-by-step like [9]. Training samples for an RNN-based predictor are formed as follows:

$$(\{x_1, x_2, \dots, x_\tau\}, x_{\tau+1});$$

$$(\{x_2, x_3, \dots, x_{\tau+1}\}, x_{\tau+2});$$

$$\vdots$$

$$(\{x_{n-\tau}, x_{n-\tau+1}, \dots, x_{n-1}\}, x_n);$$

The performance of this method is good, but it can only make a prediction for next time (or a certain time in the further). Operators cannot see the long-term trends with this model.

1) Future interpolating points prediction model

To solve the problem, new outputs for an RNN-based predictor are explored in the study. The training samples can be written as follows, which is the same as common samples in sequence-to-sequence (seq2seq) problem:

$$\left(\{x_1, x_2, \dots, x_\tau\}, \{x_{t_1^{(1)}}, x_{t_2^{(1)}}, \cdots, x_{t_{stable}^{(1)}}\}\right);$$

$$\left(\{x_2, x_3, \dots, x_{\tau+1}\}, \{x_{t_1^{(2)}}, x_{t_2^{(2)}}, \cdots, x_{t_{stable}^{(2)}}\}\right);$$

$$\vdots$$

Where $\{x_{t_1^{(1)}}, x_{t_2^{(1)}}, \cdots, x_{t_{stable}^{(1)}}\}$ is a sequence formed by several points from τ-time to the start time of stable phases t_{stable} in a tunneling circle. The corner mark (1) in the sequence means it is the first sample. Define the number of elements in the sequence as η. The corresponding time vector of the sequence $[t_1^{(1)}, t_2^{(1)}, \cdots, t_{stable}^{(1)}]^T$ can be formed by evenly inserting η points between τ and $t_{stable}^{(1)}$ ($t_{stable}^{(1)}$ is excluded since it will be predicted separately in the following model) and the sequence can be established accordingly. The number of training samples that can be extracted from one cycle depends on τ, $t_{stable}^{(1)}$ and η. The input of each sample in the model is a $\tau - by - 3$ matrix (each column is composed by corresponding T, F, V). The output is a $\eta - by - 3$ matrix (each column is composed by corresponding T, F, V) The model can be easily interpreted from the interpolation perspective. With some points given, one can see the trend of a curve by linking the points. The model just predicts some interpolating points in the future.

2) Stable points prediction model

Another model used in the study is predicting the start time of the stable phase t_{stable} and the stable values V_s, T_s and F_s of a tunneling cycle. For one tunneling cycle, the corresponding training samples can be written as follows:

$$\left(\{\tau_1^T; x_1^T; x_2^T; \dots; x_\tau^T\}, \{t_{stable}^{(1)}, x_{t_{stable}^{(1)}}^T\}\right);$$

$$\left(\{\tau_2^T; x_2^T; x_3^T; \dots; x_{\tau+1}^T\}, \left\{t_{stable}^{(2)}, x_{t_{stable}^{(2)}}^T\right\}\right);$$

$$\vdots$$

Where $x_{t_{stable}^{(1)}}^T = \left[T_s^{(1)}, F_s^{(1)}, V_s^{(1)}\right]$, $\tau_i^T = [t_i, t_{i+1} \cdots, t_{i+\tau-1}]$. The reason importing time index τ_i^T of inputs here is to improve the regression performance of the start time of stable phases $t_{stable}^{(i)T}$, which is proven by trails in experiments. So, the input of each sample in the first model is a $\tau - by - 4$ matrix (each column is composed by t, T_t, F_t, V_t). The output is a $1 - by - 4$ row vector $(t_{stable}, T_s, F_s, V_s)$. With t_{stable} and $x_{t_{stable}}$ predicted, the time vector $\left[t_1^{(1)}, t_2^{(1)}, \cdots, t_{stable}^{(1)}\right]^T$ can be calculated. Then use the model mentioned before to predict $\{x_{t_1^{(1)}}, x_{t_2^{(1)}}, \cdots, x_{t_{stable}^{(1)}}\}$. Thus, the whole future curves of V, F, T parameters can be plotted.

V. EXPERIMENTS

In this section, the experimental results of the proposed LSTM predictors for TBM operating data are shown. All experiments were processed by using Keras in a computer with AMD Ryzen 9 5900HX with Radeon Graphics 3.30 GHz, 32 GB RAM. Two kinds of metrics: the root of mean square error (RMSE) and the mean absolute percentage error (MAPE) are used to evaluate the predicting results. RMSE and MAPE can be expressed as follow respectively:

$$MAPE = \frac{100}{T}\sum_{t=1}^{T}\frac{|\hat{y}_t - y_t|}{y_t}\% \quad (7)$$

$$RMSE = \sqrt{\frac{1}{T}\sum_{t=1}^{T}\|\hat{y}_t - y_t\|_2^2} \quad (8)$$

where $\hat{y}_t$ and y_t stand for the prediction and the real values for input x_t.

Divide 1067 tunneling cycles into 747 training cycles, 160 validation cycles and 160 test circles (75%:15%:15%). The training samples, validation samples and test samples can be extracted from the corresponding cycles according to the method in the last section. After trying many trials with different parameters, the sampling parameters are determined. The length of raising phase/time steps of input τ equals to 30. The number of future interpolating points η equals to 9. Since the start time of stable phase of each cycle is different, the number of samples extracted from each cycle is different, set the max number of samples extracted from each cycle equal to 100 to make sampling more uniform. 51962 samples, including 36770 training samples (71%), 8246 validation samples (16%) and 6946 test samples (13%), are extracted for experiments.

A. Stable points prediction

After trying many trials with different hyper parameters, the structure of regression model of predicting the start time of stable phase t_{stable} and the stable values $x_{t_{stable}}$ is determined, which is shown in Fig. 7. The model is established by stack one LSTM layer with 50 neurons, one LSTM layer with 50 neurons, one Dense (fully connected neural networks mentioned in our lectures) with 50 neurons and one Dense with 4 neurons one by one. The loss function is mean square error, and the metrics are $RMSE$ and $MAPE$. The optimizer is Adam with 0.01 learning rate, which works well in practice and compares favorably to other stochastic optimization methods [12]. Before fitting the model, the time index of the input and the start time of stable phase are normalized via dividing by 500 to decrease their weights in loss function.

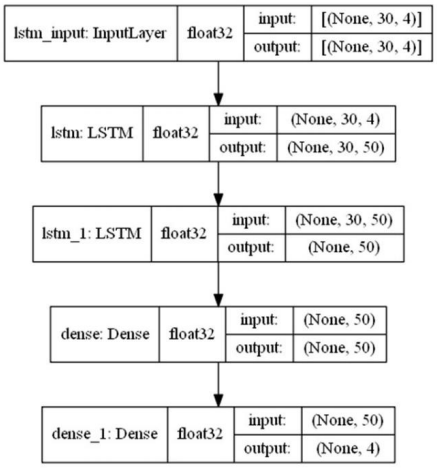

Fig. 7. The Structure of Stable Points Prediction Model

Fit the model with 500 of batch size. TABLE I. shows the evaluating results on testing set after 15 epochs. Comparing to the benchmark which means regarding the average value of outputs of training set as the outputs of testing set, it can be found that all output parameters outperform than benchmark. The error of the stable value of thrust F_s is relatively small, which makes sense since we choose the start time of stable phase of thrust as t_{stable} at first based on its intuitive stability.

TABLE I. EVALUATING RESULTS OF MODEL I

	t_{stable}	T_s	F_s	V_s
Unit	-	$kN \cdot m$	kN	mm/min
Threshold	500	8410	23260	120
$RMSE_{mean}$	111.5	481.0	26515	11.5
$RMSE$	74.4	392.5	1255.0	8.0
$MAPE_{mean}$ (%)	47.4	14.6	16.9	15.6
$MAPE$ (%)	26.1	11.2	6.9	10.4

B. Future interpolating points prediction

The next step is to establish the future interpolating points prediction model. The structure of regression model of predicting the future interpolating points $\{x_{t_1^{(1)}}, x_{t_2^{(1)}}, \cdots, x_{t_{stable}^{(1)}}\}$ is determined, which is shown in Fig. 8. In this model, an encoder-decoder LSTM is used, which is a model comprised of two sub-models: one called the encoder that reads the input sequences and compresses it to a fixed-length internal representation, and an output model called the decoder that interprets the internal representation and uses it to predict the output sequence [13]. The model is established by referring [13] too.

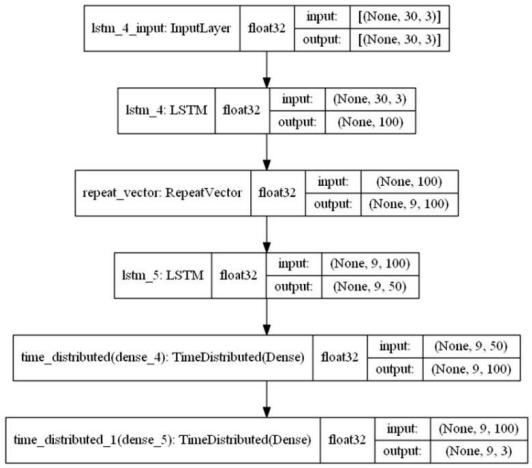

Fig. 8. The Structure of Future Interpolating Points Prediction Model

Fit the model with 500 of batch size. TABLE II. shows the evaluating results on testing set after 15 epochs. Similarly, use the mean values of outputs of training set as benchmark. We can find the performance is acceptable from the from an engineering application perspective.

TABLE II. EVALUATING RESULTS OF MODEL II

	$\overline{T}$	$\overline{F}$	$\overline{V}$
Unit	$kN \cdot m$	kN	mm/min
Threshold	8410	23260	120
$RMSE_{mean}$	716.489	2583.985	15.76
$RMSE$	415.467	1073.065	9.074
$MAPE_{mean}$ (%)	0.375	0.194	0.361
$MAPE$ (%)	0.144	0.07	0.187

In this model, future interpolating points η is an important parameter. Fig. 9. shows the reason $\eta = 9$. Select different numbers of future interpolating points and rerun the model. Other hyperparameters can be selected in the same way.

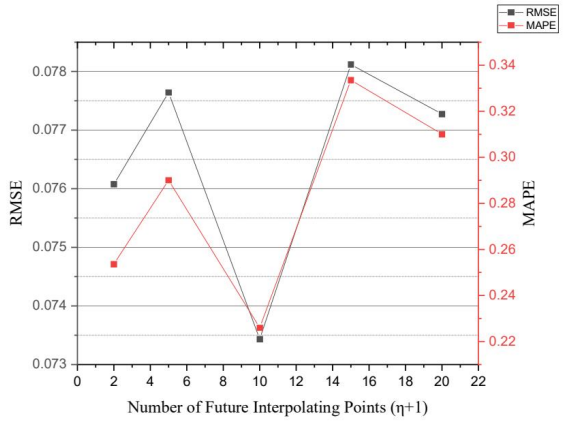

Fig. 9. Select the Number of Interpolating Points

C. *Whole future curve prediction*

Having the start time of stable phase t_{stable} , the stable values of parameters T_s, F_s, V_s (from Model I) and the coordinates of future interpolating points (from Model II), the next step is to draw the whole circle curve and update it real time. Take the prediction of thrust from test set for example. Fig.10. shows one result of thrust sequence prediction in the beginning. The green curve represents the original 30 seconds thrust sequence. The red curve represents the future trends predicted based on the original sequence.

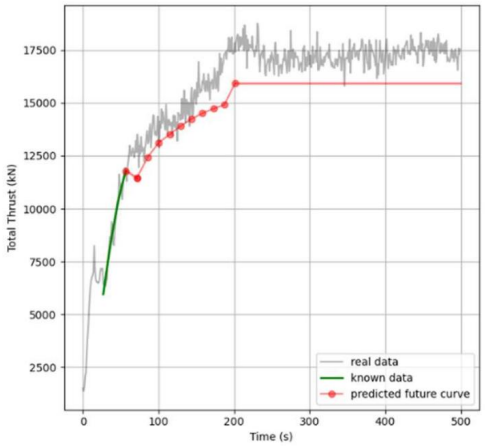

Fig. 10. The Result of Thrust Sequence Prediction (Start)

Fig.11. shows the prediction results of whole thrust curve when we update it lively. From these two figures we can see that for this test sample, the method in the study can give relatively good prediction of the whole curve intuitively.

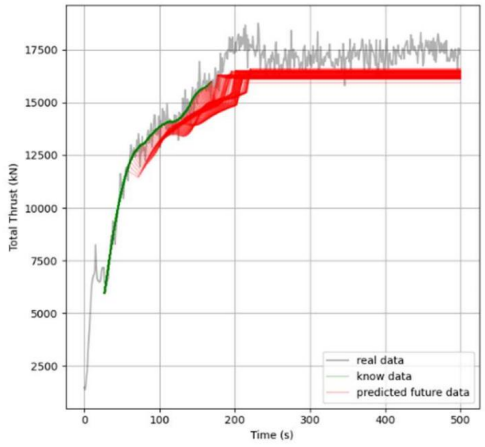

Fig. 11. The Result of Thrust Sequence Prediction (real-time update)

VI. CONCLUSION AND FUTURE WORK

The report proposes a method for predicting the future trends of sequences in the framework of deep learning (LSTM) and implements the method based on TBM operation data. The method can give relatively accurate results for some future points quantitatively (Experiments Section part A) and predict the future trends well for large part of test samples qualitatively (Experiments Section part B). Some algorithms in the report like extracting raising and steady phase from a cycle can also been used for reference when dealing with some periodic data.

Long-short term memory (LSTM) networks gives moderate prediction results when deal with sequences prediction in the study. More deep learning method can be tried based on this, such as Gated Recurrent Units (GRU), CNN-LSTM, etc. The method proposed (achieve auto-drive) can only drive as an operator at most. In the tunnel construction field, there is no uniform standard to evaluate what is the best driving cycle, thus, no label for driving cycles. Some unsupervised learning tools or reinforcement learning might need be used for the breakthrough in the subject.

VII. Appendices

Code is submitted in gradescope.

The elaborated edition codes (updating):

https://github.com/J-i-n-p-u/CS229---project---code.git

VIII. Contributions

Thanks for China Railway Engineering Equipment Group Co., Ltd. providing me the valuable TBM operation data. Thanks for advisor Ian gives me inspiration for my project. Thanks for the contribution of instructors and assistants in CS229.

IX. References

[1] R. Mikaeil, M.Z. Naghadehi, F. Sereshki, Multifactorial fuzzy approach to the penetrability classification of TBM in hard rock conditions, Tunn. Undergr. Space Technol. 24 (5) (2009) 500–505, https://doi.org/10.1016/j.tust.2008.12.007.

[2] S. Yagiz, New equations for predicting the field penetration index of tunnel boring machines in fractured rock mass, Arab. J. Geosci. 10 (2) (2017) 33, https://doi.org/ 10.1007/s12517-016-2811-1.

[3] R. Gertsch, L. Gertsch, J. Rostami, Disc cutting tests in Colorado Red Granite: implications for TBM performance prediction, Int. J. Rock Mech. Min. Sci. 44 (2) (2007) 238–246, https://doi.org/10.1016/j.ijrmms.2006.07.007.

[4] M. Entacher, S. Lorenz, R. Galler, Tunnel boring machine performance prediction with scaled rock cutting tests, Int. J. Rock Mech. Min. Sci. 70 (2014) 450–459, https://doi.org/10.1016/j.ijrmms.2014.04.021.

[5] C. Su, Y. Wang, H. Zhao, P. Su, C. Qu, Y. Kang, T. Huang, Z. Cai, L. Wang, Analysis of mechanical properties of two typical kinds of cutterheads of shield machine, Adv. Sci. Lett. 4 (6-7) (2011) 2049–2053, https://doi.org/10.1166/asl.2011.1545.

[6] Satar Mahdevari, Kourosh Shahriar, Saffet Yagiz, Mohsen Akbarpour Shirazi, A support vector regression model for predicting tunnel boring machine penetration rates, International Journal of Rock Mechanics and Mining Sciences, Volume 72, 2014, Pages 214-229, ISSN 1365-1609, https://doi.org/10.1016/j.ijrmms.2014.09.012.

[7] H. Chen, C. Xiao, Z. Yao, H. Jiang, T. Zhang and Y. Guan, "Prediction of TBM Tunneling Parameters through an LSTM Neural Network," 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO), 2019, pp. 702-707, doi: 10.1109/ROBIO49542.2019.8961809.

[8] Zaobao Liu, Long Li, Xingli Fang, Wenbiao Qi, Jimei Shen, Hongyuan Zhou, Yulong Zhang, Hard-rock tunnel lithology prediction with TBM construction big data using a global-attention mechanism-based LSTM network, Automation in Construction, Volume 125, 2021, 103647, ISSN 0926-5805, https://doi.org/10.1016/j.autcon.2021.103647.

[9] Xianjie Gao, Maolin Shi, Xueguan Song, Chao Zhang, Hongwei Zhang, Recurrent neural networks for real-time prediction of TBM operating parameters, Automation in Construction, Volume 98, 2019, Pages 225-235, ISSN 0926-5805, https://doi.org/10.1016/j.autcon.2018.11.013.

[10] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[11] Savitzky, Abraham, and Marcel JE Golay. "Smoothing and differentiation of data by simplified least squares procedures." Analytical chemistry 36.8 (1964): 1627-1639.

[12] Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).

[13] Brownlee, J. (2021, January 6). Multi-Step LSTM Time Series Forecasting Models for Power Usage. Machine Learning Mastery. https://machinelearningmastery.com/how-to-develop-lstm-models-for-multi-step-time-series-forecasting-of-household-power-consumption/.