Estimation of Plasma Vertical Position by Long-Short Term Memory Network with Time2Vec in a Small Tokamak Device PHiX

Sejung JANG and Hiroaki TSUTSUI

Tokyo Institute of Technology, Tokyo 152-8550, Japan

(Received 22 February 2024 / Accepted 2 June 2024)

Time evolution of plasma vertical position is estimated by using long-short term memory networks (LSTM) with Time2Vec technique which incorporates temporal information into a neural network. Since many tokamak devices have elongated cross-section in achieving high performance whereas accurate vertical position feedback control is required in order to avoid vertical displacement events (VDEs). Our data-driven model, using experimental data obtained from a small tokamak device PHiX in Tokyo Institute of Technology, can estimate the plasma vertical displacement by incorporating operational scenario coils current data. The model achieved high performance by combining Time2Vec with LSTM. We can also interpret the weights extracted from a trained, data-driven model by comparing the model's predictions.

Keywords: tokamak, vertical instability, LSTM, machine learning, Time2Vec

DOI: 10.1585/pfr.19.1403023

1. Introduction

Although tokamak is one of the leading a candidate for a fusion power reactor, there remain many problems to be solved for commercial application. Since tokamak plasma with a elongated cross-section have demonstrated high performance [1, 2], many tokamak devices use elongated plasma. The vulnerability of elongated plasmas to vertical instability emphasizes the importance of estimation to minimize and analyze the occurrence of vertical displacement events (VDEs) [3]. Estimating plasma vertical position using operational data is crucial for safely controlling elongated plasma and mitigating disruptions linked to VDEs, which lead to the influx of impurities and wall damage due to plasma interactions with the wall. In order to solve this problem, we utilized machine learning techniques to develop models which can estimate plasma vertical position. So, we create a data-driven model for multivariable regression of VDEs by utilizing the neural network [4]. Plasma discharge experimental data are obtained as time series data such as densities, temperature and magnetic field, etc. These data are important resources for solving the many problems, but there are limitations as humans cannot analyze the enormous time series data. However, the neural network can easily deal with the enormous amount of data. The recent works have shown the effectiveness of using machine learning for analyzing radiation collapse [5] and plasma vertical position [6] using support vector machines (SVM). These SVM models, specifically designed to distinguish between 'disruptive' and 'non-disruptive' states, showed excellent predictive performance. However, SVM has the disadvantage of being sensitive to noise, and it also requires trying multiple combinations of parameters. Previous studies show that classification models can distinguish between true and false, but they cannot demonstrate how the values change. Conversely, we developed a deep learning-based regression model using noisy data and used this model to predict the vertical position of the plasma in real time. The recurrent neural network (RNN)-based models [7] such as long short-term memory (LSTM) [8] have been conventionally used to estimate plasma disruptions [9–11] using time-series data. In contrast to standard feedforward neural networks, RNNs have an internal loop, a kind of memory function, which allows them to retain and apply information through a series of data points. The results of these RNN-based models suggest that data-driven models for plasma disruption can be utilized to mitigate and prevent such disruptions. Despite the advancements, they face limitations such as frequent false alarms, compounded by the challenge of interpreting LSTM networks due to their complexity and non-linearity. Therefore, LSTM alone falls short in delivering sufficient performance. Hence, there's a necessity for an some approach to supplement the LSTM and improve its efficacy.

In this study, we apply the LSTM, which is an advanced version of the RNN model, and the Time2Vec [12] is a novel approach that incorporates time-related features for multivariate regression to estimate VDEs using plasma discharge experimental data from PHiX at Tokyo Institute of Technology. By utilizing Time2Vec, we are able

author's e-mail: jang.sejung@torus.nr.titech.ac.jp

to interpret the model by extracting the weight parameters of the neural networks from the data-driven model. Time2Vec is advantageous in its flexibility, easily integrating with various time series datasets for machine learning without altering existing models. Combining the strengths of LSTM and Time2Vec, we can also enhance the model's regression ability than conventional LSTM model. This article is organized as follows. First, we present LSTM with Time2Vec model in Sec. 2. We present the result of its application and discuss the evaluation in Sec. 3. Finally, our conclusions are presented in Sec. 4.

2. LSTM with Time2Vec

In this work, we enhanced LSTM by integrating Time2Vec as shown in Fig. 1. The LSTM, an advanced type of RNN designed to mitigate the vanishing gradient problem, is commonly employed for handling time-series data. The LSTM system consists of four gates and two internal states as shown in Fig. 2 and Eqs. (1)-(6),

$$\boldsymbol{f}_t = \sigma(\mathsf{W}_\mathrm{f}[\boldsymbol{h}_{t-1}, \bar{\boldsymbol{x}}_t] + \boldsymbol{B}_\mathrm{f}), \tag{1}$$

$$\boldsymbol{i}_t = \sigma(\mathsf{W}_\mathrm{i}[\boldsymbol{h}_{t-1}, \bar{\boldsymbol{x}}_t] + \boldsymbol{B}_\mathrm{i}), \tag{2}$$

$$\boldsymbol{o}_t = \sigma(\mathsf{W}_\mathrm{o}[\boldsymbol{h}_{t-1}, \bar{\boldsymbol{x}}_t] + \boldsymbol{B}_\mathrm{o}), \tag{3}$$

$$\boldsymbol{g}_t = \tanh(\mathsf{W}_\mathrm{g}[\boldsymbol{h}_{t-1}, \bar{\boldsymbol{x}}_t] + \boldsymbol{B}_\mathrm{g}), \tag{4}$$

$$\boldsymbol{C}_t = \boldsymbol{f}_t * \boldsymbol{C}_{t-1} + \boldsymbol{i}_t * \boldsymbol{g}_t, \tag{5}$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t * \tanh(\boldsymbol{C}_t). \tag{6}$$

At a time step t, the vectors $\boldsymbol{f}_t$, $\boldsymbol{i}_t$, $\boldsymbol{o}_t$ and $\boldsymbol{g}_t$ represent the activations of the forget, input, output, and cell gates, respectively, while $\boldsymbol{h}_{t-1}$ and $\boldsymbol{C}_{t-1}$ signifies the hidden state and cell state from the previous time step $(t-1)$; W_f, W_i, W_o and W_g are the respective weight matrices for the gates, and $\boldsymbol{B}_\mathrm{f}$, $\boldsymbol{B}_\mathrm{i}$, $\boldsymbol{B}_\mathrm{o}$ and $\boldsymbol{B}_\mathrm{g}$ are the bias vectors associated with the gates; σ stands for the sigmoid activation function, [ ] and $*$ represent concatenation of vectors and Hadamard product; $\bar{\boldsymbol{x}}_t$ represents the input data at the current time step. The forget gate's decision process, involving values ranging from 0 (forget) to 1 (remember), determines the transfer rate to the next cell, impacting memory retention. In an LSTM network, the output at time step t is computed considering the information stored in its internal states $\boldsymbol{C}_{t-1}$ and $\boldsymbol{h}_{t-1}$ from previous time steps. In our implementation, we set $\boldsymbol{h}_0$ and $\boldsymbol{C}_0$ as random values that follow a normal distribution with mean of 0 and standard deviation of 1. Passing the prior hidden state and current input through the sigmoid function computes the forget gate's values, which directly influence the cell state ,the output gate is the result of the LSTM block. We can set the dimension of the LSTM vectors including gate to d_LSTM. The output $\boldsymbol{h}_t$ from the LSTM is fed into a dense layer without activation as is shown in Fig. 1,

$$\hat{y}_t = \boldsymbol{W}_\mathrm{dense} \cdot \boldsymbol{h}_t + b_\mathrm{dense}, \tag{7}$$

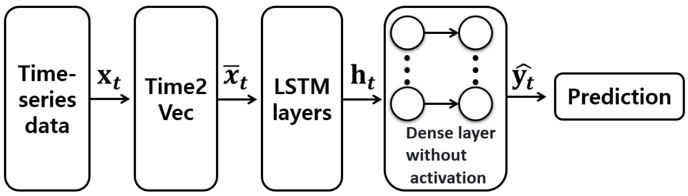

Fig. 1 The model architecture of LSTM with Time2Vec.

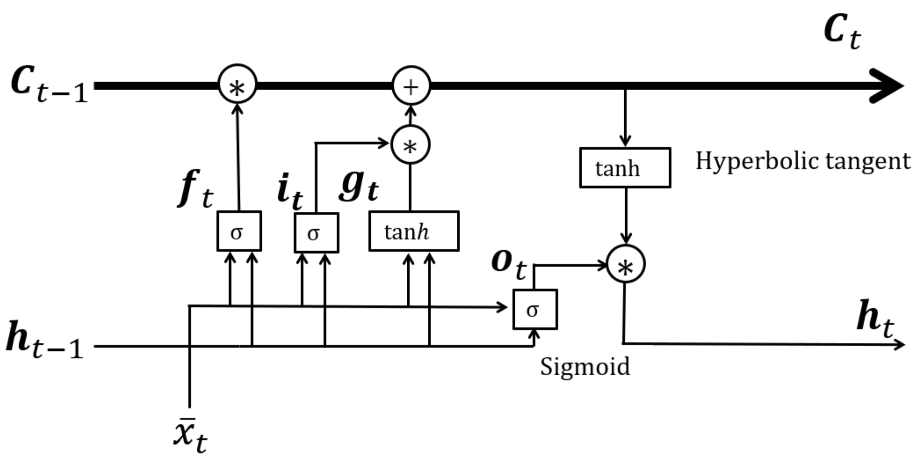

Fig. 2 The input gate $\boldsymbol{i}_t$, output gate $\boldsymbol{o}_t$, forget gate $\boldsymbol{f}_t$, cell gate $\boldsymbol{g}_t$, hidden state $\boldsymbol{h}_t$ and cell state $\boldsymbol{C}_t$ in LSTM systems with Time2Vec, where ⊕ represents the sum operation and ⊛ represents the Hadamard product operation.

Table 1 The parameters of the LSTM+Time2Vec model.

Parameter	Explanation	Value
T	Number of time steps	750
n	Number of input parameters	8
k	Number of Frequencies in Time2Vec	200
d_{LSTM}	Dimension of LSTM gates and states	128

where $\boldsymbol{W}_{\mathrm{dense}}$ and b_{dense} are learnable parameters and $\hat{y}_t$ is the predicted value. We utilized the loss function, which is mean squear error,

$$\mathrm{Loss} = \frac{1}{T}\sum_{t=1}^{T}(y_t - \hat{y}_t)^2, \tag{8}$$

where y_t is the actual experimental data.

Original LSTM models do not treat time itself as a feature assuming that inputs are synchronous. When time is known to be a relevant feature, it is fed in as yet another input dimension. Time2Vec [12] is a learnable vector representation which is embeded in stead of the time τ,

$$\mathbf{v}_\tau[0] = w_0\tau + b_0, \tag{9}$$

$$\mathbf{v}_\tau[i] = \sin(w_i\tau + b_i) \quad 1 \leq i \leq k. \tag{10}$$

Here the parameters i and k represent a index and the number of frequency used in machine learning respectively, while w_i and b_i are learnable parameters trained through backpropagation based on the loss (8). The time τ is related to the time index t as shown in

$$\tau = \tau_0 + (t-1)\Delta\tau, \tag{11}$$

where τ_0 is the initial value of τ at $t = 0$ and $\Delta\tau$ represents the change in τ for each unit time increment. The result of the Time2Vec function $\mathbf{v}_\tau$ is combined with other inputs $\boldsymbol{x}_\tau$ in the LSTM model to determine the network's behavior at each time τ. In the LSTM, the input $\bar{\boldsymbol{x}}_t$ is originally equal to $\boldsymbol{x}_t$, but in the Time2Vec process, it has transformed to $\bar{\boldsymbol{x}}_t = [\boldsymbol{x}_t, \mathbf{v}_t]$ as is shown in Fig. 1. The advantages of using Time2Vec include capturing temporal patterns and handling irregular time intervals. It helps the neural network understand time-related data by providing vectors with learnable parameters. The parameters for d_{LSTM} and k were chosen as listed in Table 1. We employed the Adam optimizer [13] to guide the training process and the code in this work was developed using the PyTorch framework [14].

3. Application to Small Tokamak

The PHiX [15, 16] is a small tokamak device in Tokyo Institute of Technology. It is designed to research the plasma vertical stability by helical magnetic field [17, 18] created by saddle coils [15, 19]. We extracted data from PHiX to create a data-driven model, as shown in Table 2 and Fig. 3. Figure 3 shows typical discharges used in this work, where the vertical position of the plasma is determined using the modified correlation coefficient method [15, 20]. The dataset listed in Table 2 includes measured plasma parameters, control actuator values as inputs including currents of poloidal field coils whose positions are shown in Fig. 4. PHiX features a non-axisymmetric magnetic field, supported by saddle coils (SCs) that are installed along the upper, lower, and outer peripheries of the torus, which helps in maintaining the stability of the plasma's vertical position. The coils of PF1 and PF6 are connected in series in same directions to create a vertical magnetic field, while the coils of PF2 and PF5 are connected in seriesin opposite directions to create a horizontal magnetic field. Although the coil currents PF1 and PF6 (PF16), PF2 and PF5 (PF25) are not independent as is indicated previous sentences, they were used as independent in this work. Despite the series connection, the values of PF1 and PF2 currents are smooth, while those of PF5 and PF6 have fast oscillations. The reason for this is that the measurements of PF1 and PF2 were done via an analog filter. In these experiments, PF25 currents are pre-programed as is shown in Fig. 3 (f). Since the plasma is vertically unstable in these magnetic field configurations, plasma moves downward without feedback control.

Table 2 Experimental data of PHiX used in this work. Units in parentheses are those used in the experimental data.

Parameter
Input $\boldsymbol{x}_t$
Plasma Current (A)
Loop Voltage (V)
Poloidal Field Coils Currents (PF16, PF25) (A)
Ohmic Heating Coil Current (A)
Saddle coil current (A)
Output y_t
Plasma Vertical Position (m) whose spatial resolution is 1 mm.
Time interval (10^{-5}s) $\Delta\tau$

In the model, all training data are scaled to a range with a minimum value of 0 and a maximum value of 1. We have created a dataset where the training data consists of 55 shots, and the test data consists of 4 shots. Using these data, the model performs multivariate regression to estimate plasma vertical displacement. All the data in the dataset, except for the plasma vertical displacement, is used as input data, with the plasma vertical displacement as the output as is shown in Table 2. Used data in this work satisfy that the plasma current exceeds 1 kA.

In our study, we evaluated the performance of the LSTM model against the Time2Vec+LSTM model in estimating plasma vertical displacement. All the trainable parameters were determined to minimize loss of Eq. (8) on

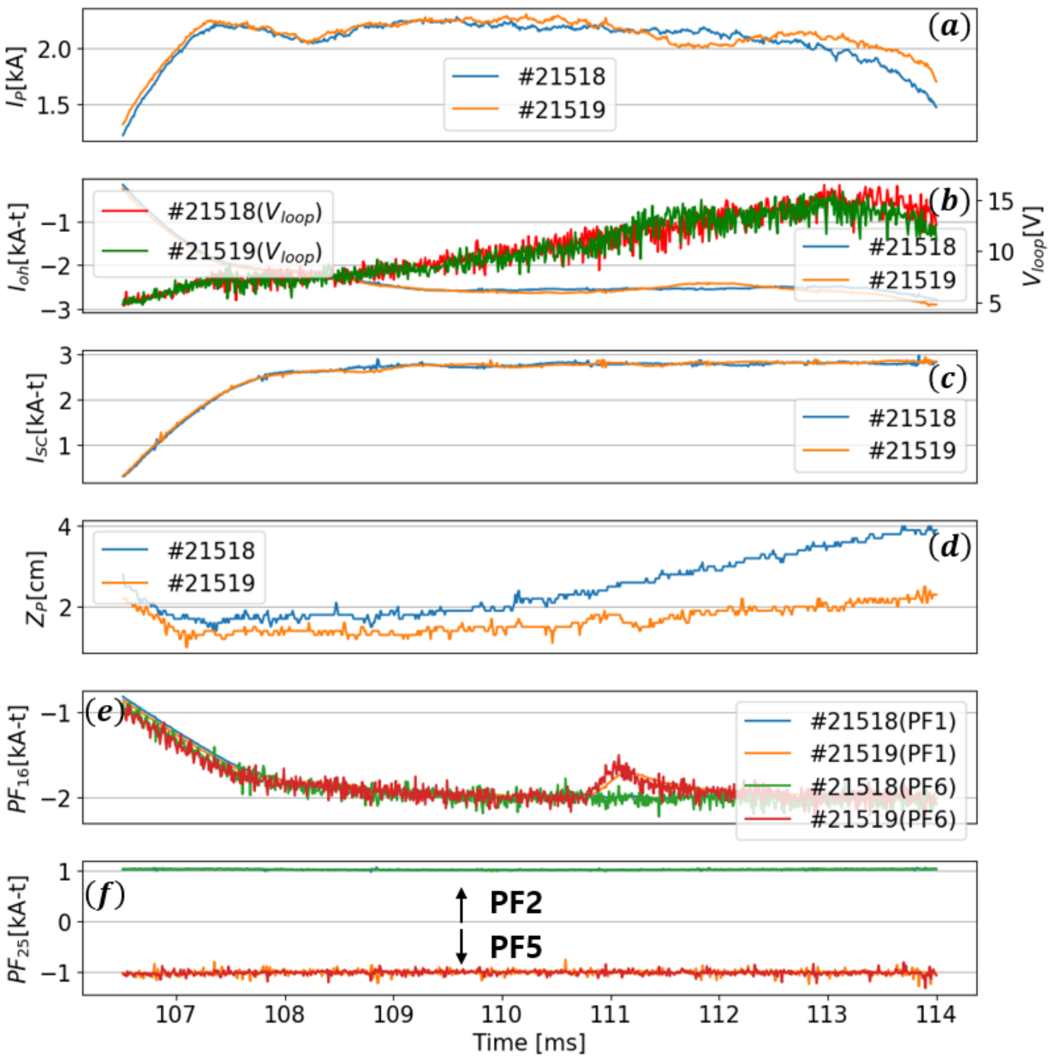

Fig. 3 Typical temporal evolutions in PHiX experiments. (a) Plasma current I_p, (b) Ohmic coil current I_{oh} and Loop voltage V_{loop}, (c) Saddle coil current I_{SC}, (d) Vertical position of the plasmas Z_p, (e) and (f) Poloidal field coil currents.

Table 3 Comparison of Training Loss and Test Loss.

Model	Training Loss	Test Loss
LSTM	0.101	0.103
LSTM+Time2Vec	0.027	0.042

Table 4 Comparison of Test Loss.

	LSTM	LSTM+Time2Vec
#21518	0.068	0.046
#21519	0.090	0.018
#21521	0.028	0.011
#21525	0.22	0.09

the training and test data.

The integration of Time2Vec with the LSTM model demonstrated better performance over the standard LSTM model, as evidenced by the reduced loss evaluated by Eq. (8), detailed in Table 3 and depicted in Fig. 5 in which observed data change stepwise because the resolution of the correlation method is chosen to be 1 mm. Table 4 further reveals that Time2Vec achieved a lower test loss compared to LSTM without Time2Vec.

In order to investigate the superiority of Time2Vec, we applied power spectrum analysis to validate that the Time2Vec layer is capable of capturing temporal information. Since the output from Time2Vec described in Eqs. (9) and (10) is fed into an LSTM in my model, the output $\boldsymbol{v}$ of Time2Vec is assigned to the sigmoid and hyperbolic tangent functions as arguments in the form $Wv+B$ as is shown in Eqs. (1)-(4). According to Eq. (10), the argument includes a term of $W\sin(\omega\tau + b)$ which is similar to that of discrete Fourier transform. Although the output $\hat{y}$ in Eq. (7) is a nonlinear function of $\boldsymbol{v}$, the temporal characteristic of $\hat{y}$ is captured to the weight parameters W and ω after training. Before the training, the initial weights in Eqs. (1)-(4) are

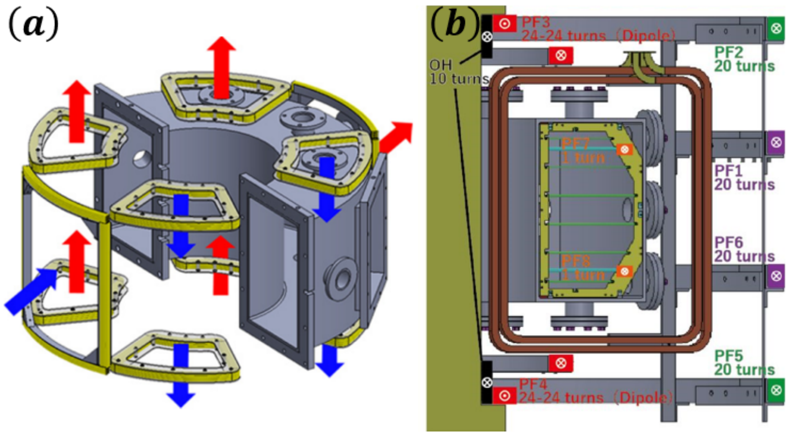

Fig. 4 The positioning of multiple coils in the PHiX [19]. (a) Configuration of Saddle Coils on PHiX. (b) Poloidal cross section and coils positions of PHiX . The brown square is the TF coil, and OH coils are wound on an iron core (grass green).

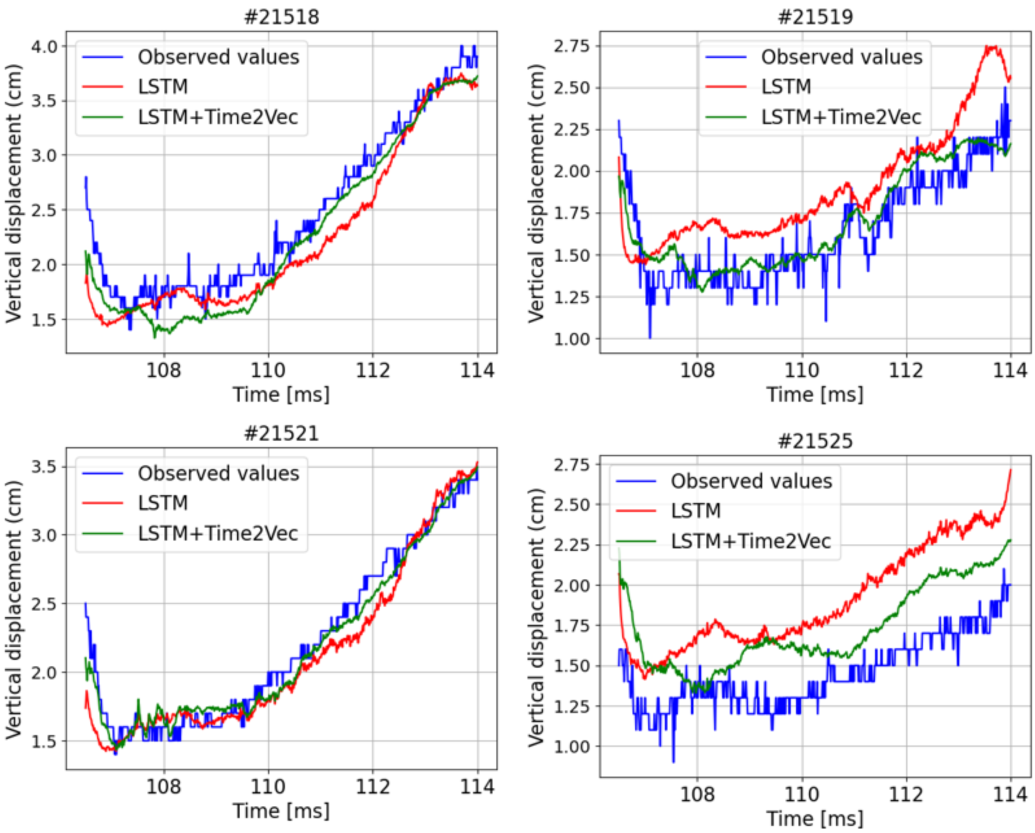

Fig. 5 The estimation results from model LSTM and LSTM+Time2Vec using test data. The start time is set at 106.5 milliseconds after the discharge.

randomly set within the range of −0.1 to 0.1 and the initial weights w_0 and bias b_i are set with random values between ±0.05, while the frequency w_i of $1 \le i \le k$ are configured to have a distribution ranging from 0 to 50 kHz which is the maximum frequency determined by the sampling time $\Delta\tau$ in Table 2. After the training, W^2's of $\sin(w\tau + b)$ are obtained as a function of w as depicted in Fig. 6. These figures show large weights at a low frequency region. In order to check the efficiency of Time2Vec, we compared the distribution of W^2 in Fig. 6 and the power spectra of plasma vertical positions, PF5 currents, and loop voltage in the training data are depicted in Fig. 7. It shows significant weight changes at 10 kHz and 24 kHz. When comparing the power spectra of multiple inputs and output, no special frequencies were found. Regarding the reasons for this phenomenon, it is currently under investigation and

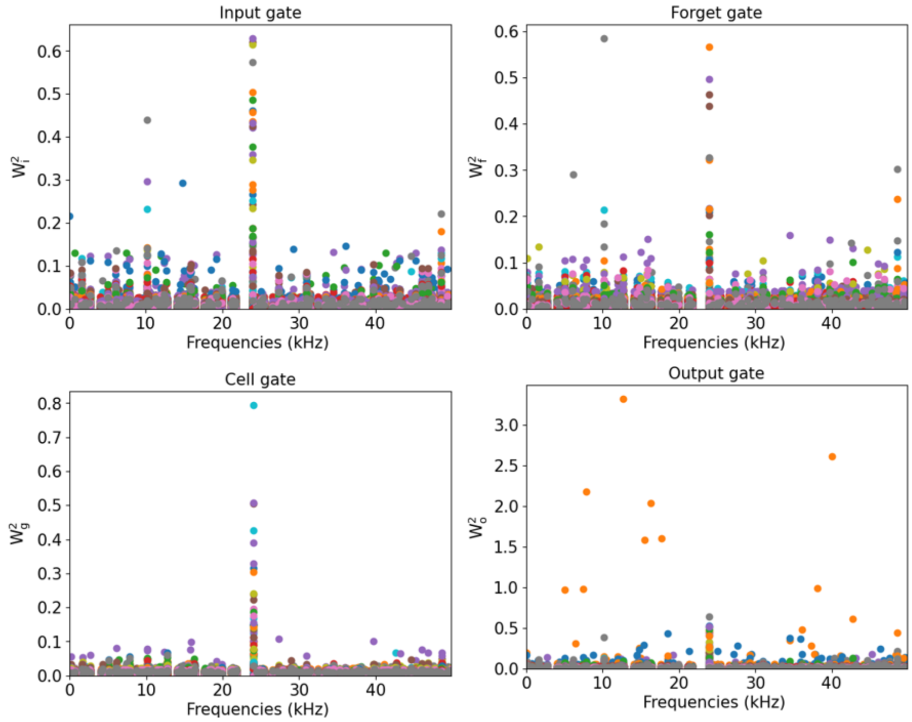

Fig. 6 Learned weights and frequencies of Time2Vec model. The axis of abscissas represents the frequency w_i in Eq. (10), while the axes of ordinates represent the squared weights W_f^2, W_i^2, W_o^2 and W_g^2 in Eqs. (1)-(4). corresponding to the frequency w_i.

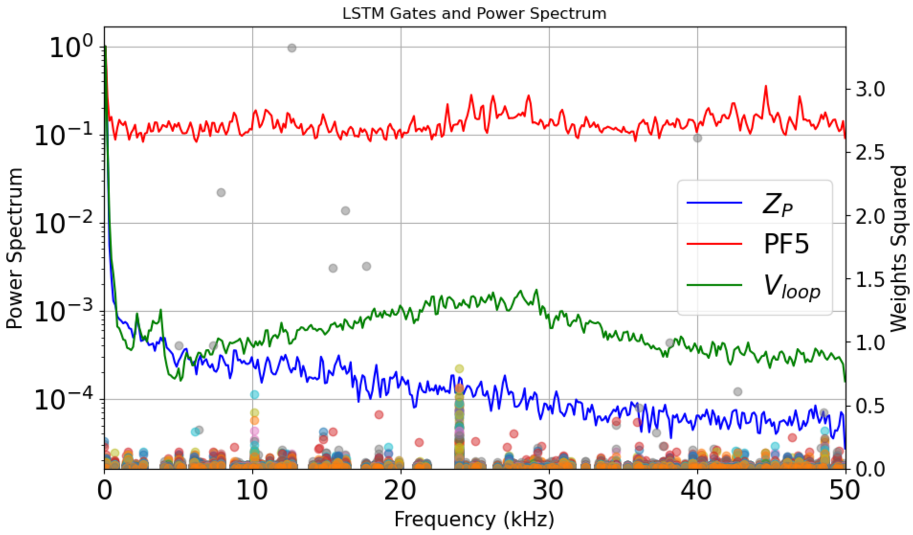

Fig. 7 A comparison of the power spectra of the experimental data and the learned parameters in Time2Vec. The solid lines represent the average power spectra, which are normalized to have a value of unity at frequency =0, of the plasma vertical position, PF5 currents, and loop voltage in the training data. The dots, which are chosen from Fig. 6, are squared weights of the trained model in Eqs. (1)-(4) as functions of w_i in Eq. (10).

remains unknown. The original paper on Time2Vec [12] demonstrates the extraction of characteristic frequencies. However, the experimental data used in this study do not have inherent characteristic frequencies according to the power spectra.

4. Conclusions

This study aims to analyze plasma VDEs by estimating the plasma vertical displacement. This data-driven model with accumulated data from a specific tokamak demonstrates the capability to estimate plasma movements thereby enabling us to design a operational scenario. We observed that the model employing Time2Vec yielded better performance in terms of test loss compared to LSTM without Time2Vec. Additionally, we attempted to confirm how Time2Vec affects the model's predictions by extracting weight parameters from Time2Vec, and two special frequencies are identified. However the experimental data used in this study do not have corresponding frequency components. Further research is, therefore, needed regarding the significant changes in weight at specific frequencies.

Acknowledgments

The authors would like to thank Dr. S. Naito for useful advices with experimental data. This work was supported by JSPS KAKENHI Grant Number 21H01061.

[1] F. Troyon, R. Gruber, H. Saurenmann, S. Semenzato and S. Succi, Plasma Phys. Control. Fusion **26**(1)A, 209 (1984).
[2] R.J. Goldston, Plasma Phys. Control. Fusion **26**(1)A, 87 (1984).
[3] Y. Nakamura, R. Yoshino, N. Pomphrey and S.C. Jardin, J. Nucl. Sci. Technol. **33**(8), 609 (1996).
[4] S. Weidman, *Deep Learning from Scratch* (O'Reilly Media, Sebastopol, CA, 2019), Chap. 2.
[5] T. Yokoyama, H. Yamada, S. Masuzaki, B.J. Peterson, R. Sakamoto, M. Goto *et al.*, Plasma Fusion Res. **17**, 2402042 (2022).
[6] S. Inoue, Y. Miyata, H. Urano and T. Suzuki, Nucl. Fusion **62**(8), 086007 (2022).
[7] D.E. Rumelhart, G.E. Hinton and R.J. Williams, Nature **323**(6088), 533 (1986).
[8] S. Hochreiter and J. Schmidhuber, Neural Comput. **9**(8), 1735 (1997).
[9] J. Kates-Harbeck, A. Svyatkovskiy and W. Tang, Nature **568**, 526 (2019).
[10] A. Pau, A. Fanni, S. Carcangiu, B. Cannas, G. Sias, A. Murari and F. Rimini, Nucl. Fusion **59**, 106017 (2019).
[11] A. Agarwal, A. Mishra, P. Sharma, S. Jain, S. Ranjan and R. Manchanda, Plasma Phys. Control. Fusion **63**(11), 115004 (2021).
[12] S.M. Kazemi, R. Goel, S. Eghbali, J. Ramanan, J. Sahota, S. Thakur *et al.*, arXiv preprint arXiv:1907.05321 (2019).
[13] D.P. Kingma and J. Ba, arXiv preprint arXiv:1412.6980 (2014).
[14] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan *et al.*, Advances in Neural Information Processing Systems **32**, 7994 (2019).
[15] S. Naito, M. Murayama, S. Hatakeyama, D. Kuwahara, Y. Suzuki, H. Tsutsui and S. Tsuji-Iio, Nucl. Fusion **61**(11), 116035 (2021).
[16] K. Munechika, H. Tsutsui and S. Tsuji-Iio, Plasma Fusion Res. **16**, 2402033 (2021).
[17] A. Reiman, Phys. Rev. **99**, 135007 (2007).
[18] K. Yasuda, T. Fujita, A. Okamoto, H. Arimoto, S. Kimata, K. Kado and K. Tsunoda, Phys. Plasmas **28**, 082108 (2021).
[19] S. Naito, Y. Suzuki and H. Tsutsui, Plasma Fusion Res. **17**, 2403069 (2022).
[20] K. Kurihara, Y. Kawamata, M. Sueoka, H. Hosoyama, I. Yonekawa, T. Suzuki, T. Oikawa, S. Ide and JT-60 Team, Fusion Eng. Des. **74**(1-4), 527 (2005).