

PAPER • OPEN ACCESS

# EAST discharge prediction without integrating simulation results

To cite this article: Chenguang Wan *et al* 2022 *Nucl. Fusion* **62** 126060

View the [article online](#) for updates and enhancements.

## You may also like

- [Disruption prediction using a full convolutional neural network on EAST](#)  
B H Guo, B Shen, D L Chen et al.
- [A dimensionality reduction algorithm for mapping tokamak operational regimes using a variational autoencoder \(VAE\) neural network](#)  
Y. Wei, J.P. Levesque, C.J. Hansen et al.
- [Breakdown prediction of ion source at EAST-NBI using neural network](#)  
Jinxin Wang, Na Wang, Zhimin Liu et al.

# Corrigendum: EAST discharge prediction without integrating simulation results (2022 *Nucl. Fusion* **62** 126060)

Chenguang Wan<sup>1,2,\*</sup> , Zhi Yu<sup>1</sup> , Alessandro Pau<sup>3</sup> , Xiaojuan Liu<sup>1</sup>  and Jiangang Li<sup>1,2,\*</sup>

<sup>1</sup> Institute of Plasma Physics, Hefei Institutes of Physical Science, Chinese Academy of Sciences, Hefei 230031, China

<sup>2</sup> University of Science and Technology of China, Hefei 230026, China

<sup>3</sup> Ecole Polytechnique Fédérale de Lausanne (EPFL), Swiss Plasma Center (SPC), Lausanne, CH-1015, Switzerland

E-mail: [chenguang.wan@ipp.ac.cn](mailto:chenguang.wan@ipp.ac.cn) and [j\\_li@ipp.ac.cn](mailto:j_li@ipp.ac.cn)

Received 18 May 2023

Accepted for publication 1 June 2023

Published 13 June 2023



## Abstract

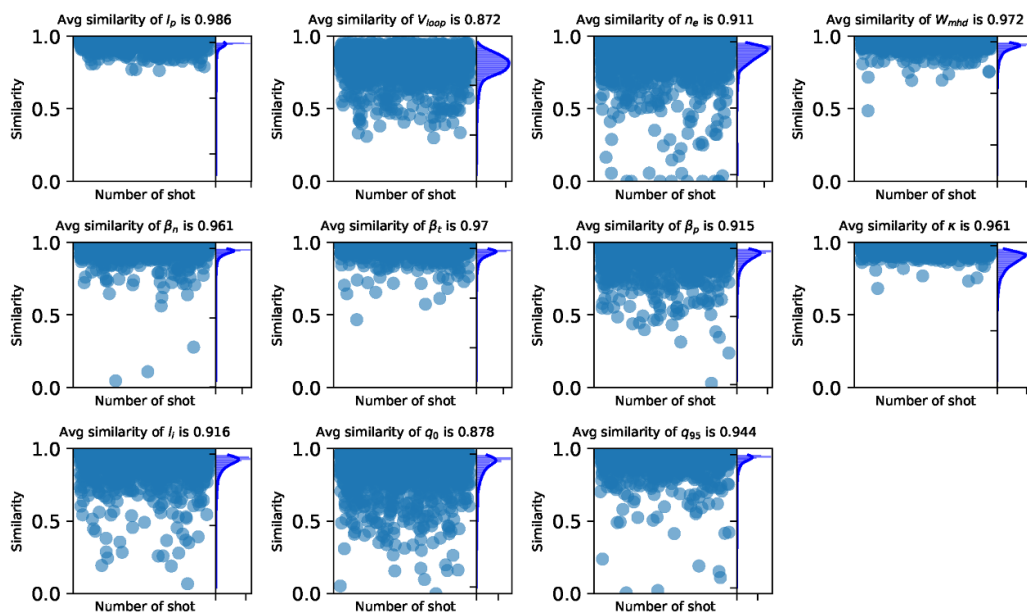
In the published paper titled ‘EAST discharge prediction without integrating simulation results’ (2022 *Nucl. Fusion* **62** 126060), figure 5 used the wrong plot, and is not matched with table 3. We used the plot discussed with referees, the case when we do not use IC 1 as a proxy. This corrigendum provides the correct plot for figure 5.

(Some figures may appear in colour only in the online journal)

\* Authors to whom any correspondence should be addressed.



Original Content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](https://creativecommons.org/licenses/by/4.0/). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.



**Figure 5.** The similarity distribution and average similarity in the whole test set. The figure shows the similarity distributions of the output signals (see table 1 in the original paper).

## Acknowledgments

The authors would like to thank all the members of EAST Team for providing such a large quantity of past experimental data. The authors sincerely thank Prof. Qiping Yuan, Dr Ruirui Zhang, and Prof. Jinping Qian for explaining the experimental data.

This work was supported by the National Key R&D project under Contract No. Y65GZ10593, the National MCF Energy R&D Program under Contract No. 2018YFE0304100, and the Comprehensive Research Facility for Fusion Technology

Program of China under Contract No. 2018-000052-73-01-001228.

## ORCID iDs

Chenguang Wan <https://orcid.org/0000-0002-6005-4460>  
 Zhi Yu <https://orcid.org/0000-0003-0000-8750>  
 Alessandro Pau <https://orcid.org/0000-0002-7122-3346>  
 Xiaojuan Liu <https://orcid.org/0000-0002-0331-8730>

# EAST discharge prediction without integrating simulation results

Chenguang Wan<sup>1,2,\*</sup> , Zhi Yu<sup>1</sup> , Alessandro Pau<sup>3</sup> , Xiaojuan Liu<sup>1</sup>   
and Jiangang Li<sup>1,2,\*</sup>

<sup>1</sup> Institute of Plasma Physics, Hefei Institutes of Physical Science, Chinese Academy of Sciences, Hefei 230031, China

<sup>2</sup> University of Science and Technology of China, Hefei, 230026, China

<sup>3</sup> École Polytechnique Fédérale de Lausanne (EPFL), Swiss Plasma Center (SPC), CH-1015 Lausanne, Switzerland

E-mail: [chenguang.wan@ipp.ac.cn](mailto:chenguang.wan@ipp.ac.cn) and [j\\_li@ipp.ac.cn](mailto:j_li@ipp.ac.cn)

Received 7 June 2022, revised 15 October 2022

Accepted for publication 20 October 2022

Published 10 November 2022



## Abstract

In this work, a purely data-driven discharge prediction model was developed and tested without integrating any data or results from simulations. The model was developed based on the experimental data from the Experimental Advanced Superconducting Tokamak (EAST) campaign 2010–2020 discharges and can predict the actual plasma current  $I_p$ , normalized beta  $\beta_n$ , toroidal beta  $\beta_t$ , beta poloidal  $\beta_p$ , electron density  $n_e$ , stored energy  $W_{mhd}$ , loop voltage  $V_{loop}$ , elongation at plasma boundary  $\kappa$ , internal inductance  $l_i$ ,  $q$  at magnetic axis  $q_0$ , and  $q$  at 95% flux surface  $q_{95}$ . The average similarities of all the selected key diagnostic signals between prediction results and the experimental data are greater than 90%, except for the  $V_{loop}$  and  $q_0$ . Before a tokamak experiment, the values of actuator signals are set in the discharge proposal stage, with the model allowing to check the consistency of expected diagnostic signals. The model can give the estimated values of the diagnostic signals to check the reasonableness of the tokamak experimental proposal.

Keywords: discharge prediction, machine learning, tokamak

(Some figures may appear in colour only in the online journal)

## 1. Introduction

The entire prediction of a plasma discharge in a tokamak is a complicated and critical task, which needs to be enhanced beyond the current capabilities of available simulation codes. It is commonly used to check the consistency of the modeled diagnostic signals, assist the experimental data analysis phase, validate theoretical models, control technology R & D, and provide references for the design of an experiment. In the framework of conventional discharge prediction, from a physics point of view, the primary method

is ‘integrated modeling’ [1] derived from first principles. ‘Integrated modeling’ involves lots of different physical processes in tokamaks. Its accuracy depends on the completeness and consistency of the tokamak’s physics derivations at the base of the model itself. High-fidelity and fast simulation of the entire tokamak discharge is still an open problem because of the high non-linearity, multi-spatial-temporal scales and multi-physics nature of tokamak plasmas [2].

A neural network based method is an alternative approach for tokamak discharge prediction without integrating complex physical modeling. The method has been adopted in magnetic fusion research to solve a variety of problems, including disruption prediction [3–9], electron temperature profile estimation from multi-energy SXR diagnostics [10], radiated power estimation [11], filament detection [12], simulation

\* Authors to whom any correspondence should be addressed.



Original content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](https://creativecommons.org/licenses/by/4.0/). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

acceleration [13–15], classifying confinement regimes [16], plasma tomography [17], identification of instabilities [18], estimation of neutral beam effects [19], determination of scaling laws [20, 21] coil current prediction with the heat load pattern [22], equilibrium reconstruction [10, 23–27], and equilibrium solver [28], control plasma [29–34], physics-informed machine learning [35]. Additionally, a method mixed neural-network with simulation code for discharge prediction [36] was noted. In that work [36], only the backward (past) observation information was used without considering the forward (future) information. While there are tasks such as disruptions prediction, which must inherently satisfy online causality settings, other tasks like discharge predictive modeling can take advantage of a wider context as far as offline analysis is concerned. In the offline discharge prediction, where backward and forward information is in principle equally important to describe the dynamic of the system. Another limitation of that work is that it allows predicting only a restricted set of signals (i.e., stored energy  $W_{\text{mhd}}$ , loop voltage  $V_{\text{loop}}$  and electron density  $n_e$ ). The evolution of a tokamak discharge is a complex process characterized by many global parameters, such as equilibrium and kinetic quantities. The restricted set of quantities namely  $W_{\text{mhd}}$ ,  $V_{\text{loop}}$ ,  $n_e$ , is still not enough for the predictive modeling of a tokamak discharge. Moreover, the model requires integrating a physical simulation code to estimate the actual  $I_p$  as an input signal. The total inference time of that model would be long and inefficient. The tokamak simulation code (TSC) [37, 38], as used in that paper to simulate the entire discharge plasma current  $I_p$  typically takes several hours if multiple models (auxiliary-heating, current-drive, alpha-heating, pellet-injection, etc) are included. In contrast, our model's typical total inference time is  $\sim 1$  s. In the present work, a new machine learning architecture was designed to consider wider contextual information, predicted eleven key signals of tokamak discharge, and values of all the input signals that can be directly available or given by the machine learning model without integrating any physical simulation code results.

In the present work, we trained the bidirectional long short-term memory (LSTM) models using large-scale data from the Experimental Advanced Superconducting Tokamak (EAST) tokamak [39–41] coming from 2010–2020 campaigns. Bidirectional LSTM [42–44] connects two hidden layers with the information propagating in opposite directions to the same output. The output layer can obtain information from backward and forward states simultaneously with these designations. With the 96 actuator signals (introduce further in table 1) as input, the model is able to reproduce the whole tokamak discharge time evolution of eleven key diagnostic signals, that are actual plasma current  $I_p$ , normalized beta  $\beta_n$ , toroidal beta  $\beta_t$ , beta poloidal  $\beta_p$ , electron density  $n_e$ , stored energy  $W_{\text{mhd}}$ , loop voltage  $V_{\text{loop}}$ , elongation at plasma boundary  $\kappa$ , internal inductance  $l_i$ ,  $q$  at magnetic axis  $q_0$ , and  $q$  at 95% flux surface  $q_{95}$ .

This paper is organized as follows. First, section 2 details the deep learning model architecture. Then, section 3 describes the data preprocessing and selection criteria. Section 4 shows

the model training process. Next, section 5 presents the model results, and a depth analysis is given. Finally, we make a brief discussion and conclusion in section 6.

## 2. Model

In this work, a bidirectional recurrent neural network (RNN) [42–44] was developed, and contextual information was taken into account. In this paper, our deep learning architecture stacked four bidirectional LSTM cells.

Theoretically, the bidirectional LSTM network simultaneously minimizes the objective function for the forward and the backward passes. Because both future and past information is available to the model during inference, the prediction of the model does not depend on an individually defined delay parameter [43]. We consider the tokamak discharge evolution as a sequence-to-sequence process, and as in a language translation task, the bidirectional LSTM can utilize past and future information for current prediction. In this work, the proposed model is designed for the task of offline discharge modeling, and it exploits a wider contextual information the future information utilization is equivalent to relaxing the causality constraint to obtain greater contextual information. It has been shown in several works [45, 46] that a bidirectional LSTM is often able to model more efficiently and robustly. Bidirectional long-term dependencies between time steps of time series or sequence data are particularly useful for regression tasks. The network, having access to the complete time series at each time step, will be more robust to the noise in the reconstruction of the tokamak discharge, improving the similarity of the reconstructed parameters.

Figure 1 shows the deep learning model architecture stacked four bidirectional LSTMs. For any time step  $t$ , we define the mini-batch input as  $\mathbf{X}_t \in \mathbb{R}^{n \times d}$  (number of examples:  $n$ , number of input features in each example:  $d$ ). In per layer bidirectional LSTM, the forward and backward hidden states for this time step are  $\vec{\mathbf{H}}_t \in \mathbb{R}^{n \times h}$  and  $\overleftarrow{\mathbf{H}}_t \in \mathbb{R}^{n \times h}$ , respectively, where  $h$  is the hidden units number.  $\vec{\mathbf{H}}_t \in \mathbb{R}^{n \times h}$  and  $\overleftarrow{\mathbf{H}}_t \in \mathbb{R}^{n \times h}$  are updated with standard bidirectional LSTM layer operations [43].

Next, the forward and backward hidden states  $\vec{\mathbf{H}}_t$  and  $\overleftarrow{\mathbf{H}}_t$  are concatenated to get the hidden state  $\mathbf{H}_t \in \mathbb{R}^{n \times 2h}$  and fed it into the next layer. When we consider dropout, the hidden state  $\mathbf{H}_t \in \mathbb{R}^{n \times 2h}$  will be randomly masked as zeros with a dropout rate  $\delta$  of 0.1 at each step during the *training* phase. The dropout rate, as well as the other hyperparameters, have been optimized maxing the performance of the model on the validation set, according to the usual ‘hyperparameter tuning’ procedure. The dropout can easily help prevent overfitting [47].

Last, output  $\mathbf{O}_t \in \mathbb{R}^{n \times q}$  (number of outputs:  $q$ ) is computed as follows:

$$\mathbf{O}_t = \text{activation}(\mathbf{H}_t \mathbf{W}_{hq} + \mathbf{b}_q). \quad (1)$$

Here, the weight matrix  $\mathbf{W}_{hq} \in \mathbb{R}^{2h \times q}$  and the bias  $\mathbf{b}_q \in \mathbb{R}^{1 \times q}$  are the model parameters of the output layer.

**Table 1.** The list of signals. The ‘raw signal’ means the original electrical signal, and these could be converted to signals with physical meaning. The IC1 was estimated using a machine learning model and then fed to the diagnostic signals (output) reconstruction model as input.

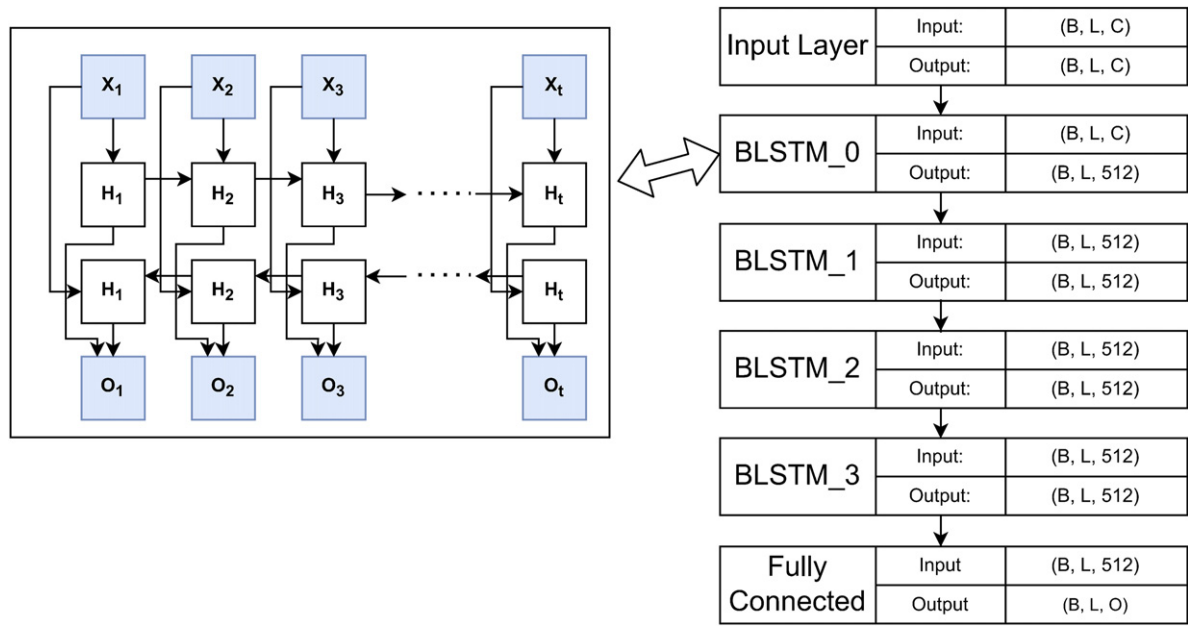
Signals	Physics meanings	Unit	Number of channels	Sampling rate	Meaning of channels
Output signals			12		
$Act.I_p$	Actual plasma current	A	1	1 kHz	Actual plasma current
$n_e$	Electron density	$10^{19} \text{ m}^{-3}$	1	1 kHz	Electron density
$W_{mhd}$	Plasma stored energy	J	1	20 Hz	Plasma stored energy
$V_{loop}$	Loop voltage	V	1	1 kHz	Loop voltage
$\beta_n$	Normalized beta	Dimensionless	1	15 Hz	Normalized beta
$\beta_t$	Toroidal beta	Dimensionless	1	15 Hz	Toroidal beta
$\beta_p$	Beta poloidal	Dimensionless	1	15 Hz	Beta poloidal
$\kappa$	Elongation at plasma boundary	Dimensionless	1	15 Hz	Elongation at plasma boundary
$l_i$	Internal inductance	Dimensionless	1	15 Hz	Internal inductance
$q_0$	$q$ at magnetic axis	Dimensionless	1	15 Hz	$q$ at magnetic axis
$q_{95}$	$q$ at 95% flux surface	Dimensionless	1	15 Hz	$q$ at 95% flux surface
Feedback signal			1		
IC1	In-vessel coil No. 1 current	A	1	1 kHz	In-vessel Rogowski coil No. 1 current
Input signals			95		
Ref. $I_p$	Reference plasma current	A	1	1 kHz	Reference plasma current
PF	Current of PF coils	A	14	1 kHz	PF 0–14 current
$B_{t0}$	Toroidal magnetic field	T	1	1 kHz	Toroidal field at magnetic axis
LHW	Power of lower hybrid wave current drive and heating system	kW	4	20 kHz	2.45 GHz LHW, and 4.6 GHz LHW
NBI	Neutral beam injection system	Raw signal	8	5 kHz	Acceleration voltage and beam current, of No. 1–2 left/right ion source
ICRH	Ion cyclotron resonance heating system	Raw signal	16	5 kHz	Output of detector for rejected power of No. 1–16 transmitter
ECRH/ ECCD	Electron cyclotron resonance heating/current drive system	Raw signal	4	50 kHz	Output of detector for injected power measurement No. 1–4 gyrotron
GPS	Gas puffing system	Raw signal	12	10 kHz	Horizontal ports J, K, D, B; upper port O; lower ports O, C, H
SMBI	Supersonic molecular beam injection	Raw signal	3	10 kHz	3 ports of SMBI
PIS	Pellet injection system	Raw signal	1	10kHz	1 injection line for pellet injection
Ref. Shape	Shape reference	Raw signal	31	1 kHz	20 groups of control points

One of the key features of a bidirectional LSTM is that information from both directions of the sequence is used to estimate the current time step. The architecture uses future (forward) and past (backward) information to perform the inference. The final layer activation function is a linear function since we are dealing with a regression task and do not want to constrain output parameters values.

### 3. Dataset

The whole EAST’s data system stores more than 3000 raw channel data and thousands of processed physical analysis data [48], which records the whole process of the EAST discharge. All the tokamak data was divided into three categories: configuration parameters, actuator signals, and diagnostic signals.





**Figure 1.** The architecture of bidirectional LSTM. The subfigure on the left represents a bidirectional LSTM cell. Here, B, L, C, O are batch size, the longest sequence length of a batch, feature size, and the number of output signal channels.

The configuration parameters describe constants related to the tokamak such as magnetic probe positions. The actuator signals such as the power of ECRH, etc, are actively controlled quantities. The diagnostic signals are observable parameters passively measured from internal plasma such as loop voltage  $V_{loop}$  etc. The configuration parameters are fixed during the tokamak experiments. The discharge prediction task can be essentially reduced to mapping actuator (input) signals to diagnostic (output) signals.

In the present work, the output signals include all important 0D EAST diagnostic signals routinely available after a discharge. The input signals include all the quantities that may affect the output. Table 1 contains detailed information about input and output signals. The input signal ‘Ref. Shape’, in a particular, requires a more in depth discussion. According to the tokamak magnetic control system workflow [49], the shape references affect the in-vessel Rogowski coil current (IC1). So when estimating IC1, such shape references data is required. On EAST, shape feedback control is realized using the isoflux control scheme [50]. As the key concept of isoflux control, the plasma target shape (Ref. Shape) is interpreted as a set of control points defining the desired plasma boundary, and the flux at each control point is regulated to be equal. Usually, one of the control points is chosen as a reference point, which is typically a point on the limiter in a limited plasma configuration or the X point in a diverted configuration. Thus the plasma boundary is controlled by adjusting the poloidal field (PF) coil currents to eliminate the flux error between the reference points and other boundary control points. Some of the selected signals are processed signals with clear physical meaning, and others are unprocessed raw electrical signals. Since some signals of the actual EAST experimental diagnostic system are not processed, we directly selected some unprocessed raw electrical signals.

As long as the input signals include information to determine the output, the unprocessed signals will not affect the modeling result.

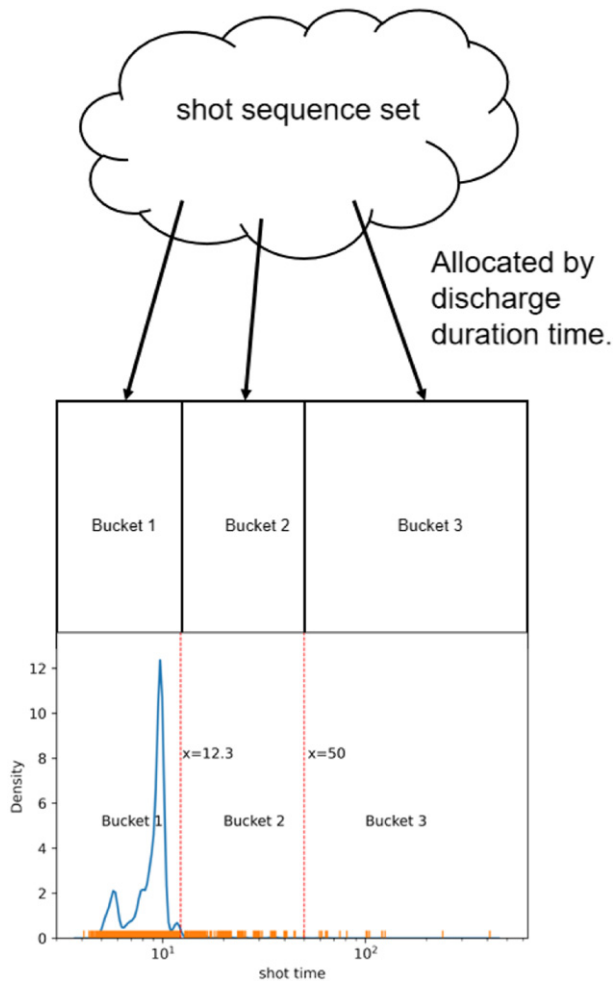
Tokamak discharge evolution is a multi-spatial-temporal scales, non-linear, multi-physics quantities coupling process. There is no simple function to determine the relationship between the actuator signals and the diagnostic signals. Different diagnostic parameters are determined by different inputs. In the present work, the input signals cannot simply and directly determine all the output signals. In our model training experiments,  $I_i$ ,  $\beta_p$  cannot be accurately estimated by the input signals that do not include IC1. When the input signals include the actual IC1, all output signals of the present work can be estimated. Since IC1 values cannot be obtained through direct measurement, they have been indirectly estimated through a machine learning model.

#### 4. Training

This section shows details of model training and data processing. The training model is divided into four steps as follows:

- (a) Obtaining and resampling the data of the 108 data channels (including input, feedback, and output signals as shown in table 1) from the EAST source database, then storing it in HDF5<sup>4</sup> file with each HDF5 file data properties (standard deviation  $\delta$ , mean  $\mu$ , etc) stored in MongoDB.

<sup>4</sup> Hierarchical data format (HDF) is a set of file formats (HDF4, HDF5) designed to store and organize large amounts of data.



**Figure 2.** The shots during time distribution of the input discharge sequences. The number of buckets  $B = 3$ . The y-axis represents distribution density, normalized as a probability density function.

- (b) Standardizing the data with z-scores<sup>5</sup> (also known as standard scores).
- (c) Using bucketing (explained further in section 4.3) batch training for the deep learning model to be reconstructed IC1.
- (d) Integrating the estimated IC1 as input for the eleven key diagnostic signals (table 1 output signals) reconstruction model training.

#### 4.1. Obtaining and resampling

The dataset is selected from EAST tokamak during the 2010–2020 campaigns, and the discharge shot number is in the range #14866–88283 [39–41]. A total of 26230 normal shots were selected. ‘Normal shot’ means that the plasma current is safely landed without disrupting and the flat-top duration is of at least two seconds. Moreover all the key quantities (i.e., magnetic field  $B_t$ , PF) describing the magnetic configuration as well as the actual plasma current are available for the

entire duration of the shots. If there is not a certain magnetic field configuration, it is impossible to constrain (or control) the plasma. Furthermore, during a tokamak experiment, the actual  $I_p$  is a key physical quantity, the experiment is generally considered a failure if there is no actual  $I_p$ . In model usage, actual  $I_p$  is a output signal. Additionally, from a pure technical point of view, the training as well as the testing of the deep learning model cannot be performed if unless at least one output signal is available. Three different data sets are needed while developing the bidirectional LSTM model for discharge modeling. A training set is required for training the model. A validation set is needed for hyper-parameters tuning, whereas a test set is used to measure the final accuracy of the deep learning model, as well as its generalization capability. The training set should use the earliest data of the selected EAST campaign to simulate the new data faced in practice, as it always happens in practice. It is worth mentioning that, despite of the chronologically order, validation and test sets do not contain more recent pulse scheduling/piloting techniques with respect to what is not included in the training set. The capability of more advanced operations and control schemes as well as the target plasma evolve over the years according to the experimental program. For the sake of consistency, completely new control techniques and piloting schemes have not been considered for the evaluation of the current models. The test set and validation set are assumed to be statistically similar, so we can expect the best performance on the test set by optimizing the accuracy of the validation set. For each epoch, the model must input every shot in the validation set to get the performance metric, and the validation set is not used to update the model parameters. So relatively small validation set can accelerate the model training. The shot range of the validation set and the test set should not have any intersection to ensure the objectivity of the test set. The test set should only be used once, that is for assessing the real accuracy of the model. In other words, the test set should be used only for model testing and should not be present during the model training or tuning phase. These three data sets have been carefully selected to meet these strict requirements. Shot #14866–74999 in the EAST database is selected as the training set. 21192 normal shots are reserved at last. The validation set has shot numbers in the range #75000–77000. Shots #77000–88283 are in the test set.

All data are resampled with the same sample rate 1 kHz. Although we used a relatively low sampling rate, the original resampled data set still contains 55 GB of data. Therefore for each shot, the data is saved to an HDF5 file, not in the database server for quick and robust training. The metadata is stored in MongoDB for double-checking the data validity and availability by the human and program.

We align our data with the same time axis by linearly interpolating to up-sample and a simple moving average (SMA) to down-sample. The SMA is the unweighted mean of the  $k$  data-points and works like a low-pass filter. Our SMA used the information from the later time. Since we use a bidirectional LSTM as architecture for the model and we aim to offline discharge modeling, we can remove the causality constraint for the sequences fed to the model for training. From another

<sup>5</sup> z-score is calculated by  $z = (x - \mu) / \sigma$  where  $\mu$  is the mean of the population.  $\sigma$  is the standard deviation of the population.



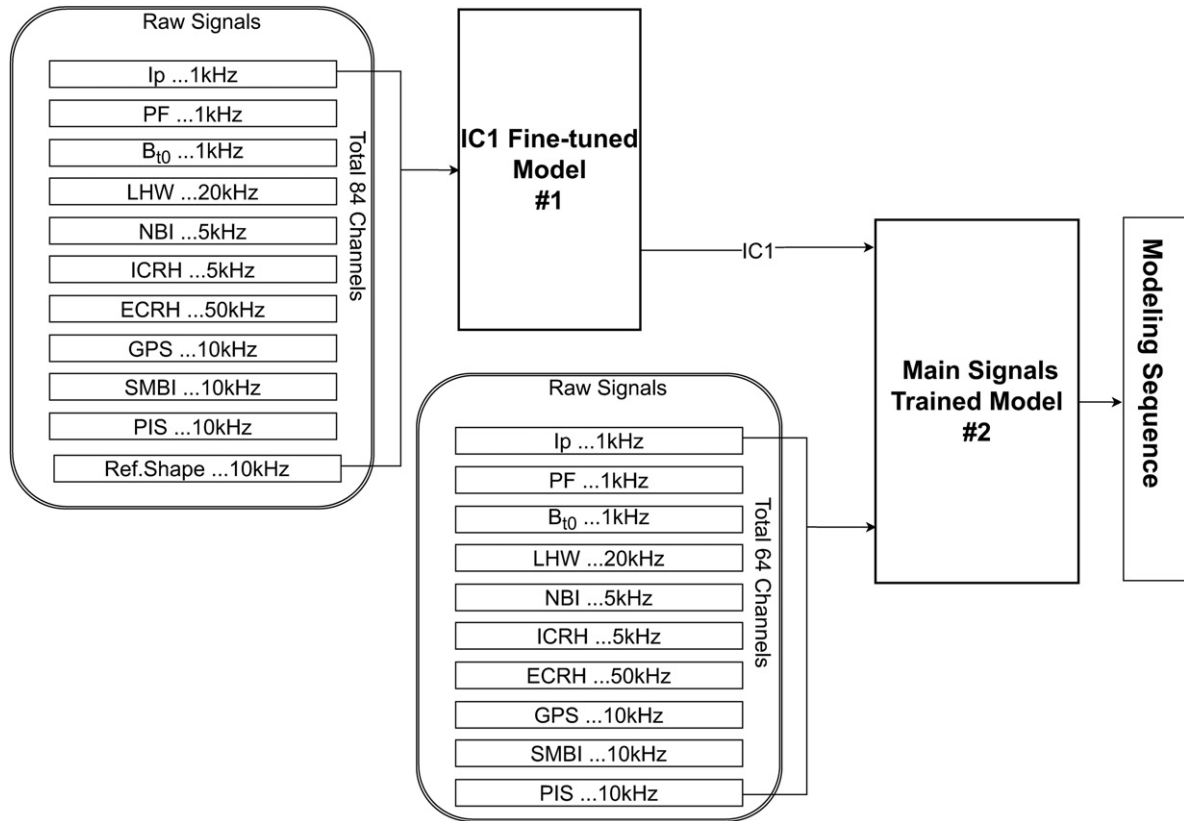


Figure 3. Using the trained model.

Table 2. Our model Hyperparameters.

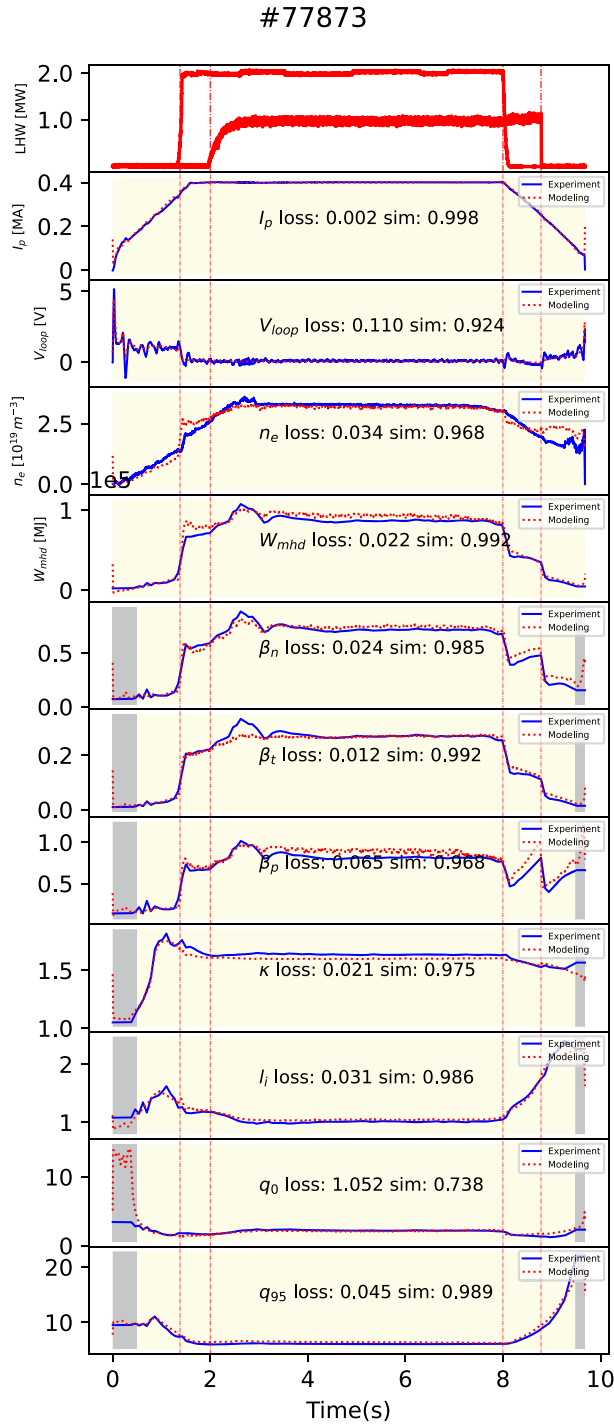
Hyperparameter	Explanation	Best value
$\eta$	Normal learning rate	$3 \times 10^{-3}$ or $5 \times 10^{-3}$
$\eta_f$	Fine-tuning learning rate	$1 \times 10^{-4}$
$\gamma$	Momentum factor	0.5
L2	L2 regularization rate	0.01
Loss function	Loss function type	Masked MSE
Optimizer	Optimization scheme	RMSprop
Dropout	Dropout probability	0.1
dt	Time step	1 ms
Batch_size	Batch size	8, 4, 1
RNN_type	Type of RNN	Bidirectional LSTM
$n_{\text{rnn}}$	Number of RNN stacked	4
$H_{\text{rnn}}$	Hidden size of RNN	512
$n_{\text{encoder}}$	Number of BiLSTMs stacked in encoder	2
$n_{\text{decoder}}$	Number of BiLSTMs stacked in decoder	2

perspective, since high-frequency fluctuations are not a relevant outcome of the experimental proposal stage, we process the data by filtering out high-frequency fluctuations.

#### 4.2. Data standardization

Firstly, all source data was saved in the HDF5 file. Then, discharge duration time and every signal mean, variance, and existence flag will be saved to the MongoDB database shot-by-shot. Saving the mean and variance of each signal of each shot data is necessary not only to double-check the presence of outliers in the signals but also for calculating the global means and

standard deviations for the huge dataset by MapReduce [51]. If a signal in a shot has outliers, then the signal values in this shot will not be used to calculate the signal's global mean and standard deviation. MapReduce is a programming model and an associated implementation for processing and generating big data sets with a parallel, distributed algorithm on a cluster. The reasons for using MapReduce are that if the global mean and standard deviation were calculated directly, the calculation would be overflowed. When the source data means and standard deviations have been calculated, the  $z$ -scores will be applied for standardization. In this step, if the input



**Figure 4.** Comparison of the modeling result and EAST experiment data. ‘Sim’ is similarity (equation (6)) and ‘loss’ is MSE (equation (7)). Shot #77873 have two LHW injections during discharge. No experimental data are available for the gray area, so the modeling results for this area are unreliable. During model training, we used the start value and end value to fill these areas for time axis alignment.

data have NaN (invalid value) or Inf (infinite value) will be replaced by a linear interpolation value and  $3.2 \times 10^{32}$  (it is not the maximum value of the float 32 type, but it is large enough. And it can still be calculated without overflowing), respectively.

#### 4.3. Bucketing batchwise training

Mini-batching gradient descent [52, 53] is a valuable technique for enhancing GPU performance [54] and accelerating the training convergence of deep-learning models. The parameter loss gradients are computed for several examples in parallel and then averaged. However, this operation requires all examples have the same length. Therefore, training RNN or its variants on a large amount of data with different lengths is a quite challenging problem. Bucketing [55] was used to solve this problem in the present work.

The bucketing method can be reduced to a partition problem. Let  $S = \{s_1, s_2, \dots, s_n\}$  be sequences set and the length of sequence  $s_i$  is  $l_i = |s_i|$ . Each GPU processes sequences in a mini-batch in a synchronized parallel manner, so processing time of a mini-batch  $I_{\text{batch}} = \{s_1, s_2, \dots, s_k\}$  is proportional to  $O(\max_{i \in 1, \dots, k} l_i)$  and processing time of the whole set is expressed as:

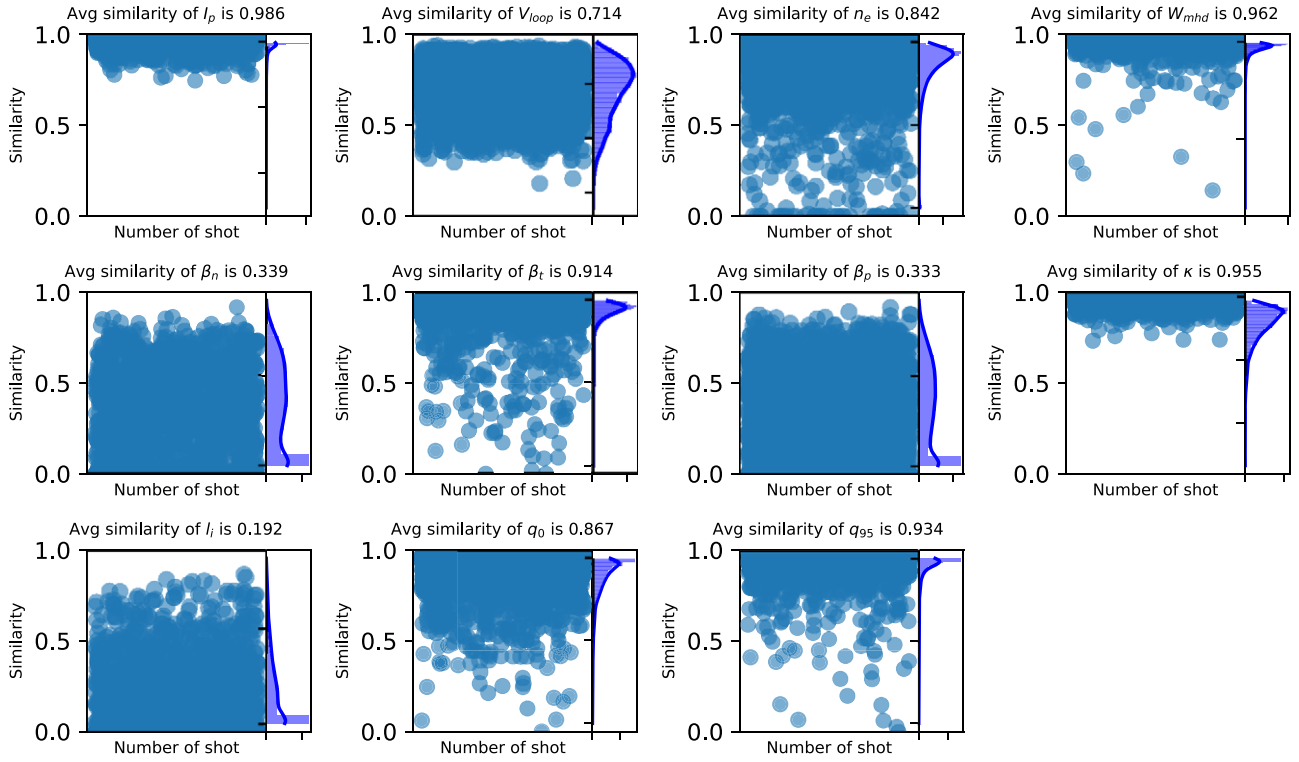
$$T(S) = O(n/k * \max_{i \in 1, \dots, k} l_i). \quad (2)$$

If the dataset sequences were shuffled randomly before splitting, the minimum and maximum sequence lengths in the mini-batch would be very different. As a result, the GPU would do additional work for processing the meaningless tails of shorter sequences. Additionally, the too-long sequence with a non-suitable batch size will be overflowed due to the GPU memory capacity limit. Specifically, if we use the same batch size for long sequences input as for short sequences input, it will take up more GPU memory which will easily cause overflows. We used customized bucketing to optimize the batch training to overcome this flaw and reduce training time. As shown in figure 2, according to length, all sequences are partitioned to  $B$  buckets. Let  $S_i = \{s_{j_1}, s_{j_2}, \dots, s_{j_{k_i}}\}$ . For each bucket, we execute the mini-batch training with different batch sizes. The processing time of the whole set is expressed as:

$$T(S) = \sum_{i=1}^B O(T(S_i)). \quad (3)$$

We manually partitioned the sequence length set in the present work because the different sequence length sets will use different batch sizes. The result of the partition is shown in figure 2. Bucket 1, 2, 3 are in intervals  $[2, 12.3]$ ,  $(12.3, 50]$ , and  $(50, 412]$  respectively. Because of the GPU memory capacity limit, the batch size of the three buckets is set at 8, 4, 1, respectively. These batch sizes can control GPU memory overflow and easily allocate the input tensor for each GPU.

The sequences within every bucket were shuffled randomly. And then, the sequences were generated batch-by-batch. To train batchwise with a batch size  $M$ , we need  $M$  independent shot discharge sequences of the same bucket to feed to the GPU. The different length discharge sequences were padded by zeros to the same length. We did this by using  $M$  processes



**Figure 5.** The similarity distribution and average similarity in the whole test set. The figure shows the similarity distributions of the output signals (see table 1).

**Table 3.** The average similarity and MSE of the eleven diagnostic signals.

Output signals	Average similarity	Average MSE
Act. $I_p$	0.986	0.0015
$n_e$	0.911	0.341
$W_{mhd}$	0.972	0.077
$V_{loop}$	0.872	0.354
$\beta_n$	0.961	0.062
$\beta_t$	0.970	0.130
$\beta_p$	0.915	0.179
$\kappa$	0.961	0.0019
$I_i$	0.916	0.127
$q_0$	0.878	0.129
$q_{95}$	0.944	0.093

to read sequence data in parallel. The  $M$  sequences were fed to a buffer first to solve the problem of GPU and CPU speed mismatch since data from HDF5 files are read through a CPU.

#### 4.4. Model training

IC1 is an in-vessel Rogowski coil current [49, 56], and this current does not have a direct reference signal. In-vessel current distribution play a key role in the accuracy of reconstruction through a free boundary plasma equilibrium solver. The estimation of such in-vessel current distribution through simulation codes is very expensive and given the large number of discharges used to train and assess the performance of the model described in this paper, we decided to rely on

the IC1 measured current, which acts basically as a proxy for the in-vessel current distribution. Although this might potentially introduce errors in the reconstruction of equilibrium quantities, it was found that not including IC1 in the set of input parameters heavily affected the accuracy of the reconstruction of quantities such as  $\beta_n$ ,  $\beta_p$  and  $I_i$ . The IC1 signal cannot be programmed or manually set in the experiment proposal stage. Therefore, first, we train a machine learning model (same model architect as depicted in figure 1) to reconstruct IC1. Secondly, we connected the trained IC1 reconstruction model and the diagnostic parameters reconstruction model for training, where the IC1 reconstruction part was fine-tuned using a minimal learning rate. For this reason, the output of the IC1 reconstruction model and the input IC1 to the deep learning model are different.

To facilitate fine-tuning the IC1 part, the model for reconstructing IC1 and the diagnostic parameters has the same architecture, initializer, optimizer, training set, validation set, test set, etc, in which only the input and output are different. The input of the IC1 reconstruction includes all input signals in table 1. The input of the diagnostic parameters reconstruction includes the output of the IC1 reconstruction model and does not use the shape reference signals. The training process is similar to using the trained model, as shown in figure 3.

The deep learning model uses end-to-end training executed on  $8 \times$  Nvidia P100 GPUs with PyTorch [57] on the CentOS7 operating system. The weight initialization scheme for the deep learning model is Xavier initialization [58], bias initializer is zeros, and optimizer is RMSprop [59]. The loss function of this training should be noted. The function is

masked MSELoss, which has some improvements for mean squared error (MSE) loss function. The masked MSELoss can be described as:

$$l(\mathbf{x}, \mathbf{y}) = L = \frac{\sum_{i=0}^{i=N} \{l_1, l_2, \dots, l_N\}}{N}, \quad (4)$$

$$l_i = \sum_{j=0}^{j=\text{len}} (\mathbf{x}_j^i - \mathbf{y}_j^i)^2, \quad (5)$$

where  $N$  is the batch size,  $\mathbf{x}$  and  $\mathbf{y}$  are the batch experimental sequence and batch predicted sequence,  $\mathbf{x}_j^i$ ,  $\mathbf{y}_j^i$  are the  $j$ th point values of the  $i$ th experimental sequence and predicted sequence. The subtlety of this work is ' $j = \text{len}$ ', where ' $\text{len}$ ' is the length of the  $i$ th sequence. So the masked MSE function can prevent useless training of the zeros padding section of the sequence. In training our model, multiple sets of hyperparameters were tried. It was a trial-and-error approach on different sets of hyperparameters. The best hyperparameter set was selected based on the best performances of the validation set. Table 2 shows the best hyperparameter set.

## 5. Results

The model was tested on unseen data (test set), as shown in figure 3. In the present work, models are used in sequence. The #1 deep learning model is used first to get the IC1 estimated values and then the #2 deep learning model to estimate the tokamak's main diagnostic signals. The preprocessing steps on new data are the same as before, keeping the training set parameters (the mean  $\mu$  and the standard deviation  $\sigma$ ) for standardization.

This section details the results and analyzes. The results include representative modeling results and eleven diagnostic signal similarity distributions of the test set. Additionally, the present work uses similarity and means square error (MSE) as quantitative measurements of the accuracy of the modeling results.

$$S(\mathbf{x}, \mathbf{y}) = \max\left(\frac{\Sigma(\mathbf{x} - \bar{\mathbf{x}})(\mathbf{y} - \bar{\mathbf{y}})}{\sqrt{\Sigma(\mathbf{x} - \bar{\mathbf{x}})^2 \Sigma(\mathbf{y} - \bar{\mathbf{y}})^2}}, 0\right), \quad (6)$$

$$\text{MSE}(\mathbf{x}, \mathbf{y}) = \frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \mathbf{y}_i)^2, \quad (7)$$

where  $\mathbf{x}$  is experimental data,  $\mathbf{y}$  is modeling result,  $\bar{\mathbf{x}}$ ,  $\bar{\mathbf{y}}$  are the means of the vector  $\mathbf{x}$  and vector  $\mathbf{y}$ ,  $\mathbf{x}_i$ ,  $\mathbf{y}_i$  are the point values of the vector  $\mathbf{x}$ ,  $\mathbf{y}$ . The MSE is the mean ( $\frac{1}{n} \sum_{i=1}^n$ ) of the squares of the errors  $(\mathbf{x}_i - \mathbf{y}_i)^2$ . MSE will be affected by the outlier but it can more accurately measure the difference of values. Similarity can only measure whether the trend is consistent, but it cannot measure the difference in value.

A typical entire process discharge prediction of our model is shown in figure 4. In the present work, the trained model can reproduce the eleven diagnostic signals during experimental proposal stage, from ramp-up to ramp-down, without relying on any physical codes.

Figure 4 shows our model can accurately reproduces the slopes of the ramp-up and ramp-down and the amplitude of the

flat-top. The model can also reflect the external auxiliary system signal impact on the diagnostic signal. The vertical dash-dot lines of figure 4 indicate the rising and falling edges of the external auxiliary system signals and the plasma response.

The whole test data set of shot range 77000–88283 is used to quantitatively evaluate the reliability of the modeling results of the eleven signals. The statistical results of the similarity and MSE between modeling results and experimental data are shown in figure 5 and table 3. Except for  $q_0$  and  $V_{\text{loop}}$ , the average similarity of other key signals is greater than 90%. And the similarity distribution is concentrated above 90%. The average similarity of  $q_0$  is greater than 85%. This quantity has a poor similarity, because if the equilibrium reconstruction is not properly constrained with pressure profiles and kinetic measurements,  $q_0$  might be unreliable. Since it might suffer from large variance, as the model is struggling mostly for the reconstruction of that parameter. Most of the errors in the reconstruction of  $V_{\text{loop}}$  are related to the plasma start-up and shut-down phases. To sum up, all selected key diagnostic signals, excluding  $V_{\text{loop}}$  are regarded as almost wholly predicted.

## 6. Conclusion and discussion

In the present work, we propose a tokamak discharge prediction method based on bidirectional LSTM. The bidirectional LSTM was developed to introduce the wider contextual information of discharge sequence to achieve a more efficient model. The model was trained on the EAST experimental dataset in shot range #14866–88283. This model can use the actuator signals to reproduce the normal discharge evolution process of the eleven key signals (i.e., actual plasma current  $I_p$ , normalized beta  $\beta_n$ , toroidal beta  $\beta_t$ , beta poloidal  $\beta_p$ , electron density  $n_e$ , stored energy  $W_{\text{mhd}}$ , loop voltage  $V_{\text{loop}}$ , elongation at plasma boundary  $\kappa$ , internal inductance  $l_i$ ,  $q$  at magnetic axis  $q_0$ , and  $q$  at 95% flux surface  $q_{95}$ .) without having as input any quantity derived by physical models. Bidirectional LSTM architecture is robust to outliers. The average similarities of all the selected key diagnostic signals between modeling results and the experimental data are greater than 90%, except for the  $V_{\text{loop}}$  and  $q_0$ . The results presented in this paper demonstrate the effectiveness of using a purely data-driven model to assist the validation of the experimental proposal for a tokamak discharge.

The present work demonstrates that the model can easily be extended to more diagnostic signals. Compared with physical models, experimental data-driven models have proven to be very efficient computationally. Once the machine learning model has been trained, a run of the trained machine learning model is faster by orders of magnitude with respect to the whole process of tokamak discharge modeling. The total inference time of our model for an entire discharge prediction is about 1 s. The present work shows very promising results exploiting experimental data-driven modeling as a supplement to physical-driven modeling tokamak. Another important point which is worth mentioning is that our model currently uses a smoothed version of the measured actuator signals as input and not directly the



corresponding programmed plasma control system (PCS) trajectories. A complete understanding of the actuator trajectories prior to a shot is challenging. The feasibility of using programmed PCS trajectories as model input to predict the evolution of a tokamak discharge will be the object of future work. Besides, we want to integrate the model into the PCS to automatically check the control strategy. 1D and 2D plasma profiles (kinetic quantities, radiation distribution) are also particularly important for tokamak discharge modeling, since they can support scenario development with particular reference to operational limits in high-performance scenarios [60].

## Acknowledgments

The authors would like to thank all the members of EAST Team for providing such a large quantity of past experimental data. The authors sincerely thank Professor Qiping Yuan, Dr Ruirui Zhang, and Professor Jinping Qian for explaining the experimental data. This work was supported by the National Key R & D project under Contract No. Y65GZ10593, the National MCF Energy R & D Program under Contract No. 2018YFE0304100, and the Comprehensive Research Facility for Fusion Technology Program of China under Contract No. 2018-000052-73-01-001228. This work was also supported in part by the Swiss National Science Foundation.

## ORCID iDs

Chenguang Wan  <https://orcid.org/0000-0002-6005-4460>

Zhi Yu  <https://orcid.org/0000-0003-0000-8750>

Alessandro Pau  <https://orcid.org/0000-0002-7122-3346>

Xiaojuan Liu  <https://orcid.org/0000-0002-0331-8730>

## References

- [1] Falchetto G.L. *et al* 2014 The European Integrated Tokamak Modelling (ITM) effort: achievements and first physics results *Nucl. Fusion* **54** 043018
- [2] Bonoli P. *et al* 2015 Report of the workshop on integrated simulations for magnetic fusion energy sciences *Technical Report* (Massachusetts Institute of Technology)
- [3] Kates-Harbeck J., Svyatkovskiy A. and Tang W. 2019 Predicting disruptive instabilities in controlled fusion plasmas through deep learning *Nature* **568** 526–31
- [4] Hu W.H.H. *et al* 2021 Real-time prediction of high-density EAST disruptions using random forest *Nucl. Fusion* **61** 066034
- [5] Guo B.H. *et al* 2021 Disruption prediction on EAST tokamak using a deep learning algorithm *Plasma Phys. Control. Fusion* **63** 115007
- [6] Cannas B., Fanni A., Sonato P. and Zedda M.K. 2007 A prediction tool for real-time application in the disruption protection system at JET *Nucl. Fusion* **47** 1559–69
- [7] Cannas B., Delogu R.S., Fanni A., Sonato P. and Zedda M.K. 2007 Support vector machines for disruption prediction and novelty detection at JET *Fusion Eng. Des.* **82** 1124–30
- [8] Cannas B., Fanni A., Pautasso G., Sias G. and Sonato P. 2010 An adaptive real-time disruption predictor for ASDEX Upgrade *Nucl. Fusion* **50** 075004
- [9] Yoshino R. 2003 Neural-net disruption predictor in JT-60U *Nucl. Fusion* **43** 1771–86
- [10] Clayton D.J., Tritz K., Stutman D., Bell R.E., Diallo A., LeBlanc B.P. and Podestà M. 2013 Electron temperature profile reconstructions from multi-energy SXR measurements using neural networks *Plasma Phys. Control. Fusion* **55** 095015
- [11] Barana O., Murari A., Franz P., Ingesson L.C. and Manduchi G. 2002 Neural networks for real time determination of radiated power in JET *Rev. Sci. Instrum.* **73** 2038–43
- [12] Cannas B., Carcangiu S., Fanni A., Farley T., Militello F., Montisci A., Pisano F., Sias G. and Walkden N. 2019 Towards an automatic filament detector with a Faster R-CNN on MAST-U *Fusion Eng. Des.* **146** 374–7
- [13] Honda M. and Narita E. 2019 Machine-learning assisted steady-state profile predictions using global optimization techniques *Phys. Plasmas* **26** 102307
- [14] Meneghini O. *et al* 2017 Self-consistent core-pedestal transport simulations with neural network accelerated models *Nucl. Fusion* **57** 086034
- [15] Meneghini O. *et al* 2021 Neural-network accelerated coupled core-pedestal simulations with self-consistent transport of impurities and compatible with ITER IMAS *Nucl. Fusion* **61** 026006
- [16] Murari A., Mazon D., Martin N., Vagliasindi G. and Gelfusa M. 2012 Exploratory data analysis techniques to determine the dimensionality of complex nonlinear phenomena: the L-to-H transition at JET as a case study *IEEE Trans. Plasma Sci.* **40** 1386–94
- [17] Ferreira D.R. and Carvalho P.J. 2020 Deep learning for plasma tomography in nuclear fusion *NeurIPS 2020 Deep Inverse Workshop Poster* pp 1–5 (<https://openreview.net/forum?id=fuB3vZdTh0>)
- [18] Murari A., Arena P., Buscarino A., Fortuna L. and Iachello M. 2013 On the identification of instabilities with neural networks on JET *Nucl. Instrum. Methods Phys. Res. A* **720** 2–6
- [19] Boyer M.D., Kaye S. and Erickson K. 2019 Real-time capable modeling of neutral beam injection on NSTX-U using neural networks *Nucl. Fusion* **59** 056008
- [20] Murari A. *et al* 2010 Machine learning for the identification of scaling laws and dynamical systems directly from data in fusion *Nucl. Instrum. Methods Phys. Res. A* **623** 850–4
- [21] Gaudio P., Murari A., Gelfusa M., Lupelli I. and Vega J. 2014 An alternative approach to the determination of scaling law expressions for the L–H transition in tokamaks utilizing classification tools instead of regression *Plasma Phys. Control. Fusion* **56** 114002
- [22] Böckenhoff D., Blatzheim M., Hölbe H., Niemann H., Pisano F., Labahn R. and Pedersen T.S. 2018 Reconstruction of magnetic configurations in W7-X using artificial neural networks *Nucl. Fusion* **58** 056009
- [23] Coccione E., Morabito C. and Martone R. 1994 Identification of noncircular plasma equilibria using a neural network approach *Nucl. Fusion* **34** 1349–63
- [24] Bishop C.M., Haynes P.S., Smith M.E.U., Todd T.N. and Trotman D.L. 1994 Fast feedback control of a high temperature fusion plasma *Neural Comput. Appl.* **2** 148–59
- [25] Jeon Y.-M., Na Na, Kim M.-R. and Hwang Y.S. 2001 Newly developed double neural network concept for reliable fast plasma position control *Rev. Sci. Instrum.* **72** 513–6
- [26] Wang S.Y., Chen Z.Y., Huang D.W., Tong R.H., Yan W., Wei Y.N., Ma T.K., Zhang M. and Zhuang G. 2016 Prediction of density limit disruptions on the J-TEXT tokamak *Plasma Phys. Control. Fusion* **58** 055014
- [27] Joung S. *et al* 2020 Deep neural network Grad–Shafranov solver constrained with measured magnetic signals *Nucl. Fusion* **60** 016034

- [28] van Milligen B.P., Tribaldos V. and Jiménez J.A. 1995 Neural network differential equation and plasma equilibrium solver *Phys. Rev. Lett.* **75** 3594–7
- [29] Bishop C.M., Haynes P.S., Smith M.E.U., Todd T.N. and Trotman D.L. 1995 Real-time control of a tokamak plasma using neural networks *Neural Comput.* **7** 206–17
- [30] Yang B., Liu Z., Song X. and Li X. 2020 Design of HL-2A plasma position predictive model based on deep learning *Plasma Phys. Control. Fusion* **62** 125022
- [31] Wakatsuki T., Suzuki T., Hayashi N., Oyama N. and Ide S. 2019 Safety factor profile control with reduced central solenoid flux consumption during plasma current ramp-up phase using a reinforcement learning technique *Nucl. Fusion* **59** 066022
- [32] Rasouli H., Rasouli C. and Koohi A. 2013 Identification and control of plasma vertical position using neural network in Damavand tokamak *Rev. Sci. Instrum.* **84** 023504
- [33] Yang B., Liu Z., Song X., Li X. and Li Y. 2020 Modeling of the HL-2A plasma vertical displacement control system based on deep learning and its controller design *Plasma Phys. Control. Fusion* **62** 075004
- [34] Seo J., Na Y.-S., Kim B., Lee C.Y., Park M.S., Park S.J. and Lee Y.H. 2021 Feedforward beta control in the KSTAR tokamak by deep reinforcement learning *Nucl. Fusion* **61** 106010
- [35] Mathews A., Francisquez M., Hughes J.W., Hatch D.R., Zhu B. and Rogers B.N. 2021 Uncovering turbulent plasma dynamics via deep learning from partial observations *Phys. Rev. E* **104** 025205
- [36] Wan C., Yu Z., Wang F., Liu X. and Li J. 2021 Experiment data-driven modeling of tokamak discharge in EAST *Nucl. Fusion* **61** 066015
- [37] Jardin S.C., Bell M.G. and Pomphrey N. 1993 TSC simulation of ohmic discharges in TFTR *Nucl. Fusion* **33** 371–82
- [38] Jardin S., Pomphrey N. and Delucia J. 1986 Dynamic modeling of transport and positional control of tokamaks *J. Comput. Phys.* **66** 481–507
- [39] Wan B., Li J., Guo H., Liang Y., Xu G., Wang L. and Gong X. 2015 Advances in H-mode physics for long-pulse operation on EAST *Nucl. Fusion* **55** 104015
- [40] Wan B., Li J., Guo H., Liang Y. and Xu G. oct 2013 Progress of long pulse and H-mode experiments in EAST *Nucl. Fusion* **53** 104006
- [41] Li J. and Wan B. 2011 Recent progress in RF heating and long-pulse experiments on EAST *Nucl. Fusion* **51** 094007
- [42] Thireou T. and Reczko M. 2007 Bidirectional long short-term memory networks for predicting the subcellular localization of eukaryotic proteins *IEEE/ACM Trans. Comput. Biol. Bioinform.* **4** 441–6
- [43] Schuster M. and Paliwal K.K. 1997 Bidirectional recurrent neural networks *IEEE Trans. Signal Process.* **45** 2673–81
- [44] Graves A. and Schmidhuber J. 2005 Framewise phoneme classification with bidirectional LSTM and other neural network architectures *Neural Netw.* **18** 602–10
- [45] Abduljabbar R.L., Dia H. and Tsai P.-W. 2021 Development and evaluation of bidirectional LSTM freeway traffic forecasting models using simulation data *Sci. Rep.* **11** 23899
- [46] Siami-Namini S., Tavakoli N. and Namin A.S. 2019 The performance of LSTM and BiLSTM in forecasting time series 2019 IEEE Int. Conf. Big Data (Big Data) (Los Angeles 09–12 December 2019) pp 3285–92 (<https://ieeexplore.ieee.org/document/9005997>)
- [47] Srivastava N., Hinton G., Krizhevsky A., Sutskever I. and Salakhutdinov R. 2014 Dropout: a simple way to prevent neural networks from overfitting *J. Mach. Learn. Res.* **15** 1929–58
- [48] Wang F., Wang Y., Chen Y., Li Li and Yang F. 2018 Study of web-based management for EAST MDSplus data system *Fusion Eng. Des.* **129** 88–93
- [49] De Tommasi G. 2019 Plasma magnetic control in tokamak devices *J. Fusion Energy* **38** 406–36
- [50] Anand H., Humphreys D., Eldon D., Leonard A., Hyatt A., Sammuli B. and Welander A. 2021 Plasma flux expansion control on the DIII-D tokamak *Plasma Phys. Control. Fusion* **63** 015006
- [51] Dean J. and Ghemawat S. 2008 MapReduce *Commun. ACM* **51** 107–13
- [52] Dean J. et al 2012 Large scale distributed deep networks *Advances in Neural Information Processing Systems* (Harrahs and Harveys, Lake Tahoe 03–08 December 2012) vol 25 pp 1223–31 (<https://papers.nips.cc/paper/2012/hash/6aca97005c68f1206823815f66102863-Abstract.html>)
- [53] Huang Z., Zweig G., Levit M., Benoit D., Oguz B. and Chang S. 2013 Accelerating recurrent neural network training via two stage classes and parallelization 2013 *IEEE Workshop on Automatic Speech Recognition and Understanding* (Olomouc 08–12 December 2013) pp 326–31 (<https://ieeexplore.ieee.org/abstract/document/6707751>)
- [54] Chetlur S., Woolley C., Vandermersch P., Cohen J., Tran J., Catanzaro B. and Shelhamer E. 2014 cuDNN: efficient primitives for deep learning (arXiv:1410.0759)
- [55] Khomenko V., Shyshkov O., Radyvonenko O. and Bokhan K. 2016 Accelerating recurrent neural network training using sequence bucketing and multi-GPU data parallelization 2016 *IEEE 1st Int. Conf. Data Stream Mining & Processing (DSMP)* (Lviv 23–27 August 2016) pp 100–3 (<https://ieeexplore.ieee.org/document/7583516>)
- [56] Xiao B.J. et al 2008 EAST plasma control system *Fusion Eng. Des.* **83** 181–7
- [57] Adam P. et al 2019 PyTorch: an imperative style, high-performance deep learning library *Advances in Neural Information Processing Systems* (Vancouver 8–14 December 2019) vol 32 ed H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox and R. Garnett pp 8024–35 (<https://papers.nips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html>)
- [58] Glorot X. and Bengio Y. 2010 Understanding the difficulty of training deep feedforward neural networks *J. Mach. Learn. Res.* **9** 249–56
- [59] Graves A. 2013 Generating sequences with recurrent neural networks (arXiv:1308.0850)
- [60] Pau A., Fanni A., Carcangiu S., Cannas B., Sias G., Murari A. and Rimini F. 2019 A machine learning approach based on generative topographic mapping for disruption prevention and avoidance at JET *Nucl. Fusion* **59** 106017