Prediction of high frequency resistance in polymer electrolyte membrane fuel cells using Long Short Term Memory based model

Tong Lin[a], Leiming Hu[a], Willetta Wisely[a], Xin Gu[b], Jun Cai[b], Shawn Litster[a], Levent Burak Kara[a,1]

[a]Carnegie Mellon University, Pittsburgh, PA 15213, USA [b]Shanghai Hydrogen Propulsion Technology Co.Ltd, Innovation Park, Lane 56, Antuo Rd, Jiading, Shanghai 201800, P.R.China

---

Abstract

High-frequency resistance (HFR) is a critical quantity strongly related to a fuel cell system's performance. It is beneficial to estimate the fuel cell system's HFR from the measurable operating conditions without resorting to costly HFR measurement devices. In this study, we propose a data-driven approach for a real-time prediction of HFR. Specifically, we use a long short-term memory (LSTM) based machine learning model that takes into account both the current and past states of the fuel cell, as characterized through a set of sensors. These sensor signals form the input to the LSTM. The data is experimentally collected from a vehicle lab that operates a 100 kW automotive fuel cell stack running on an automotive-scale test station. Our current results indicate that our

[1] Corresponding author
Email address: lkara@cmu.edu (Levent Burak Kara)

Accepted to Energy and AI *October 22, 2020*

prediction model achieves high accuracy HFR predictions and outperforms other frequently used regression models. We also study the effect of the extracted features generated by our LSTM model. Our study finds that only very few dimensions of the extracted feature are influential in HFR prediction. The study highlights the potential to monitor HFR condition accurately and timely on a car.

Keywords:

PEM fuel cell, High frequency resistance, Dynamic system prediction, Machine learning, LSTM

---

1. Introduction

Polymer electrolyte membrane fuel cells (PEMFCs) are a promising power source alternative for medium- and heavy-duty (MDH) vehicles [1]. Compared to internal combustion engines, PEMFCs utilize hydrogen as the fuel and produce much less greenhouse gas emissions. Additionally, PEMFC systems exhibit high power density and high energy efficiency [2]. However, challenges exist that hinder PEMFC's wide-scale adoption and commercialization [3, 4, 5].

A PEMFC system usually contains hundreds of membrane electrode assemblies (MEAs) stacked in series. Figure 1 shows the simplified structure of one single MEA. It contains the anode and the cathode, which are separated by the polymer electrolyte membrane (PEM), usually Nafion. During normal full cell operations, the anode is supplied with hydrogen (H_2) and the cathode is supplied with air. Hydrogen oxidation reaction (HOR)

as shown in Eq. (1) at the anode convert H_2 to protons and electrons. Electrons pass through the external circuit to generate electricity. Protons migrate through the Nafion membrane and lead to oxygen reduction reaction (ORR) at the cathode, as shown in Eq. (2).

$$2H_2 \rightarrow 4H^+ + 4e^- \quad (1)$$

$$O_2 + 4H^+ + 4e^- \rightarrow 2H_2O \quad (2)$$

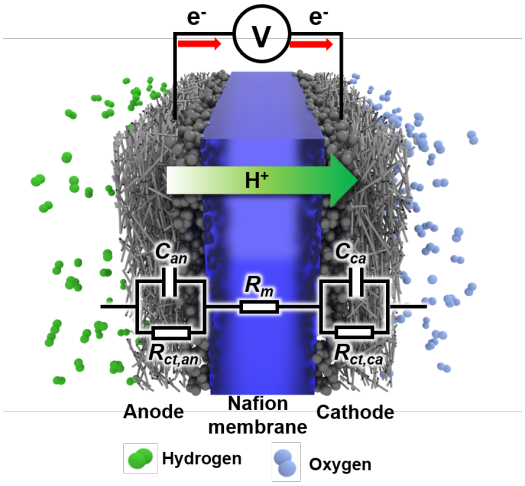


Figure 1: Configuration of the membrane electrode assembly and the corresponding equivalent circuit.

During lab-scale electrochemical impedance spectroscopy (EIS) characterizations, a fuel cell MEA can usually be modelled as an equivalent resistor-capacitor (R_c) circuit using membrane ionic resistance (R_m), anode/cathode charge transfer resistance (R_{an} and R_{ca}) and double layer capacitance (C_{an} and C_{ca}) [6, 7, 8, 9, 10]. The high-frequency resistance (HFR) is the value obtained at high frequencies (usually > 1 kHz)

and mainly represents the membrane ionic resistance. HFR is a good indicator of the water content of the membrane due to its first component dependency on water [11, 12, 13, 14, 15]. It is commonly used for the state-of-health monitoring of a fuel cell stack and is a critical indicator for dry out (i.e., low water content of the polymer electrolyte) [9, 10] or flooding (i.e., too much water inside fuel cell electrodes) [16, 17] conditions inside the fuel cell stack. HFR is also an important indicator of certain types of degradation in the fuel cell [18, 19, 20]. It affects the performance of the fuel cell, especially at high current densities as it is closely related to the ohmic voltage loss ($\Delta V_{ohm} = I \times R$).

Though obtaining HFR in real time for vehicle applications is very beneficial for the vehicle's state-of-health monitoring and control strategy optimization, an on-board measuring device is usually cost-prohibitive [21] to integrate into the existing fuel cell vehicle. Fuel cell systems involve highly nonlinear multiple physics, with a large number of interrelated variables. Especially for a full-scale fuel cell system, building a computational multi-physics model to predict the HFR value in the fuel cell becomes computationally prohibitive and impractical to implement on the on-board control systems for operational vehicles [22]. While computationally tractable physics have been proposed [23, 24], these models may not adequately capture the complex phenomena in full-scale fuel cell stacks. In recent years, machine learning techniques have proven to be very effective in estimating complex coupled transport and electrochemical process outcomes occurring in PEMFC operation [25]. For

such cases, a data driven HFR prediction model trained using an experimental system can prove effective for a real-time, high-accuracy prediction of the behaviors of interest.

In this work, we present a data-driven approach for real-time HFR prediction in PEMFCs. At the heart of our approach is a type of recurrent neural network (RNN), called long short-term memory networks (LSTMs), that incorporates multiple operational inputs of the fuel cell systems at different time steps into the HFR predictions. The model repeats itself over time, receives the fuel cell system signals as its inputs at each time step, automatically analyzes the relationship of the inputs between the two consecutive time steps, and finally generates a feature vector that encodes the inputs occurred in the time period for the HFR prediction. A distinguishing feature of such model is that it can potentially take account of the sequential effect of the system inputs. For example, controlling the same current levels with different orders can result in different HFR signals and such observation can be found by analyzing the current signals and the resulting HFR value.

Analyzing many types of fuel cell system inputs at different time steps means large number of input features to a model, which is usually a difficult task for many data driven models. Due to the high model complexity, LSTMs are able to fit the training samples produced by highly nonlinear multiple physics system, giving us the chance to accurately predict HFR in a full-scale fuel cell stack. In fact, RNNs and LSTMs have achieved great

success in analysis of sequential data produced by highly non-linear systems in other fields.

For discrete time sequences, RNN have emerged to be powerful methods for classification and regression. RNNs are connectionist models that can capture the dynamics of the time signals through the model rollout. Recent models have achieved significant success in video analysis, musical information retrieval [26]. LSTMs are a type of RNN and alleviate the vanishing gradient problem caused by long roll-out of RNN models [27]. In recent years, LSTMs have been widely utilized in the context of transport phenomena as a means for surrogate and reduced order modeling (ROM) of conventional computational fluid dynamics (CFD) [28]. LSTM models are used to predict the temporal turbulent flow properties such as streamwise velocity and Reynolds shear stress and achieves high accuracy on test dataset [29, 30, 31]. In high fidelity fluid dynamics, LSTM models are also used as a substitute for high-order and non-differential models [32, 33]. In wind energy prediction problems, a large body of work utilizes the extracted features of LSTMs to predict wind speed or wind turbine signals [34, 35, 36]. Although the LSTM model achieves high success rates in forecasting sequential data in various domains, it has not received significant attention in fuel cell dynamics modeling. There are several data-driven approaches to create accurate surrogate models for PEMFC simulations [37,38]. However, data-driven models to analyze and predict fuel cell degradation has not been widely adopted, largely due to the significant costs associated with obtaining large amounts of

experimental data. In a seminal work, based on experiment data from a short stack fuel cell, Xie et al.[39] creates a fusion structure using particle filter and RNN for a fuel cell stack prognostic mechanism. Another work that is most closely related to ours, Ma et al. [40] use an LSTM model to predict voltage drop across fuel cells. The model uses the fuel cell voltage observed in the previous time steps to predict the voltage in the future. In our work, we build on their novel work to include various measured physical parameters relevant to the fuel cell at the current time, which can greatly affect the degradation process. Similarly, Chen et al. [41] use a feed-forward multi-layer perceptron (MLP) model, where the inputs to the fuel cell are used as inputs to the machine learning model, including relative humidity, load current, and several others measured at the current time instance. Our work extends this MLP model through the use of a recurrent neural network (LSTM) that takes advantage of the signals' current values as well as their dynamically evolving past values for an enhanced prediction model.

Our main contributions are,

- A new LSTM-based model that incorporates a sequence of multiple operational states as inputs for online prediction of HFR in fuel cells.
- A novel approach to analyze the model parameters and study the feature dimension of LSTM and its relation to HFR dynamics.
- An ablation study to reveal how informative each of the model inputs is to the HFR estimation for a large multi-stack fuel cell system.

2. Experiment details

The experiment data was collected from a commercially available multi-stack fuel cell system (PROME® P390 Fuel Cell System, Shanghai Hydrogen Propulsion Technology (SHPT), Shanghai, China). Figure 2b is an image of the fuel cell system. It has a system power of 92 kW, with a peak efficiency of 60%. The working temperature of the fuel cell system is 90 ~ 95°C. Elevated operating temperature can improve the cathode reaction kinetics and simplify the water and heat management [42]. The fuel cell system can achieve self-humidification without an external humidifier, which is a state-of-the-art design for commercial automotive fuel cell systems [43, 44]. Self-humidification has the advantage of simplifying the system design and make the system more compact and lower cost. Self-humidification is usually achieved by changing the gas flow pattern so the water generated downstream of the cathode will be recirculated internally to the upstream, which is the approach adopted by Toyota [45], Hyundai [46], Ballard [47] and Honda [48]. During testing, the main control variable was the total current of the fuel cell system. Other parameters like cathode side air back pressure, air flow rate, anode side hydrogen pressure, etc., were controlled automatically with onboard pre-optimized control logic and were dependent on the system's total current. Figure 2a shows the fuel cell stack lab-scale testing system configuration and the sensor arrangement. For the anode side, the hydrogen gas was recirculated during the experiment, with the inlet and outlet hydrogen pressure and the outlet hydrogen gas temperature monitored. For the cathode side, the air flow

rate, inlet and outlet air temperature and inlet air pressure were monitored during the experiment. The fuel cell stack temperature was controlled through a water cooling system. The cooling water inlet and outlet temperature and pressure were recorded during the experiment. The HFR was measured using an EIS measuring device (AVL THDA™ Fuel cell monitoring unit, AVL List GmbH, Graz, Austria), and served as the ground truth for the model training and validation. Experiments were conducted in a controlled environment with the ambient temperature monitored and controlled as shown in Figure 2c.

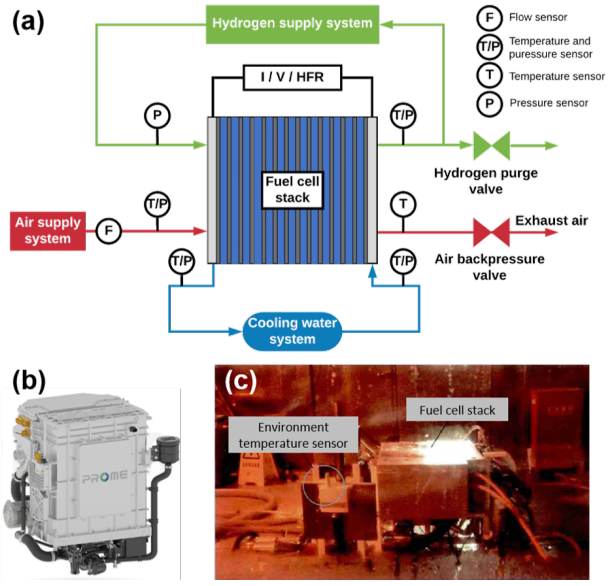
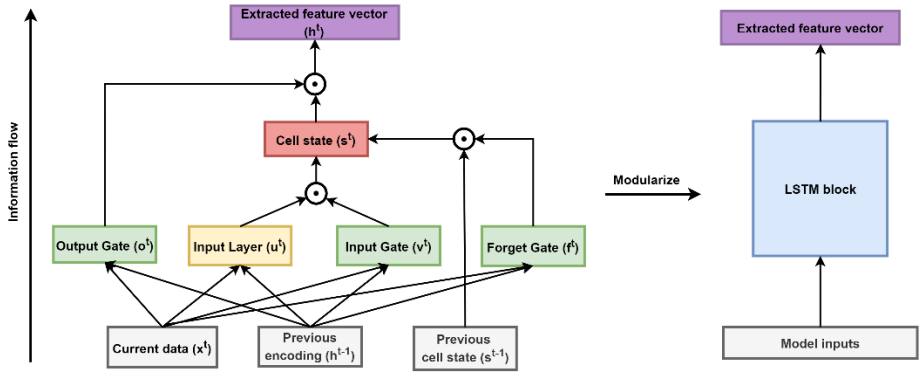


Figure 2: (a) Schematic of the fuel cell stack testing system and sensor arrangement. (b) Commercial fuel cell stack PROME® P390 [49]. (c) Fuel cell testing station.

3. HFR prediction model

3.1. *LSTM structure*

An LSTM model is an instance of recurrent neural networks (RNN). An RNN can be decomposed into two components, a time-slice model and a connection method between the two time-slice models. Once the two components are defined, the time-slice model will roll out itself according to the connection method. Both components of the LSTM are uniquely designed to target the diminishing information flow across long time windows, which is the major shortcoming of RNNs. For a conventional RNN, depletion of information prevents a proper backpropagation of the gradient throughout the network, which leads to the vanishing gradient problem (see Section 2 in supplementary material for further details).



(a)

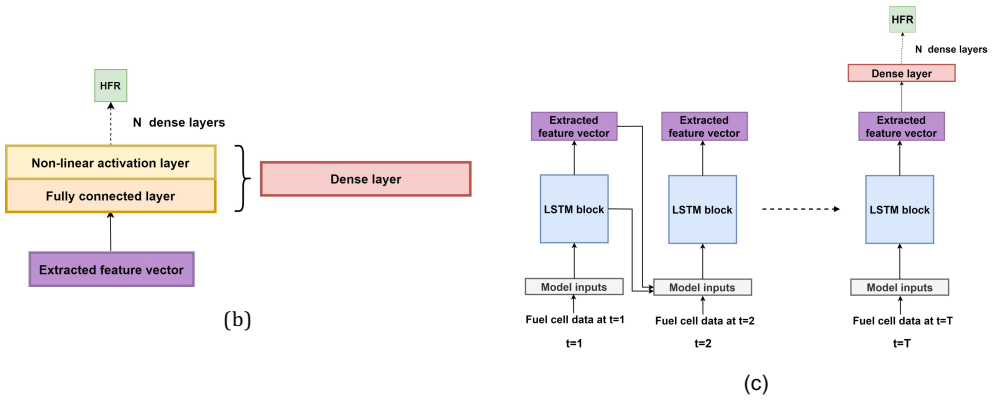


Figure 3: Illustration of LSTM-MLP model: (a) One time slice of LSTM model and its simplified module. (b) Multiple layer perceptron model. (c) LSTM-MLP model.

LSTM enciphers all information up to the current step [50]. As shown in Figure 3a, the inputs to an LSTM time-slice model consists of two parts. The first is the current information received from the external environment. The second part is the outputs of the previous time-slice model, which carries information from the previous time steps. The output of the time-slice model extracts these inputs. The modulization is shown on the right figure.

Cell state and gate mechanisms are at the heart of this model to ensure smooth information flow. Cell state is an embedding that stores information up to the current time. It receives information processed from the current external data and the previous model outputs. Meanwhile, it rolls out itself to preserve the previously stored information. Gate mechanism controls the amount of information to be stored and to be passed. There are three gate mechanisms, input gate, forget gate and output gate. They are all activated by a sigmoid function so that the output values are squashed between 0 and 1. The value of the gate are learned from data so that they

can control the information flow based on the current data and the previous model result. The input gate controls the current information flow sent to the cell state. The forget gate controls the information flow inherited from the previous cell state. The output gate controls how much information to exploit for encryption of the current information. The equations for each gate mechanism are shown in Eq. (3) - Eq. (5).

$$v_i^t = S\left(b_i^v + \sum_{j=1}^{n} T_{i,j}^v \, x_j^t + \sum_{j=1}^{m} W_{i,j}^v \, h_j^{t-1} \right) \tag{3}$$

where v_i^t is the i^{th} element of input gate at time step *t*. x_j^t and h_j^{t-1} are the j^{th} element of data input at time step *t* and extracted feature at time step *t-1* respectively. b_i^v is the i^{th} bias term for input gate v. $T_{i,j}^v$ is the i^{th} row and the j^{th} column of the input weights for input gate. $W_{i,j}^v$ is the i^{th} row and the j^{th} column of the input weights for the previous extracted feature. S stands for sigmoid function. Eq. (3) can also be written in a matrix form. If v and h have the same dimension, the matrix form is, $v_{m\times1}^t = S(b_{m\times1}^v + T_{m\times n} x_{n\times1}^t + W_{m\times m} h_{m\times1}^{t-1})$. In Eq. (3), b, T, W are trainable parameters.

The formulas for the forget and output gates are identical to the input gate, as shown in Eq. (4) and in Eq. (5):

$$f_i^t = S\left(b_i^f + \sum_{j=1}^{n} T_{i,j}^f \, x_j^t + \sum_{j=1}^{m} W_{i,j}^f \, h_j^{t-1} \right) \tag{4}$$

$$o_i^t = S\left(b_i^o + \sum_{j=1}^{n} T_{i,j}^o \, x_j^t + \sum_{j=1}^{m} W_{i,j}^o \, h_j^{t-1}\right) \quad (5)$$

Besides the gates, the current data x^t and the previous extracted feature h^{t-1} are fused to form a new input layer for the current time slice model. The formula of the fusion is similar to that of the gate mechanism except that the activation function is *tanh*. The formula is shown in Eq. (6).

$$v_i^t = \tanh\left(b_i^u + \sum_{j=1}^{n} T_{i,j}^u \, x_j^t + \sum_{j=1}^{m} W_{i,j}^u \, h_j^{t-1}\right) \quad (6)$$

The cell state is an accumulation of gated previous information and the gated current information. An additional operation is needed to represent such accumulation. Thus, the formula for cell state is:

$$s_i^t = f_i^t s_i^{t-1} + v_i^t u_i^t \quad (7)$$

where s_i^t and s_i^{t-1} are the i^{th} cell state at time step t and time step t-1.

The cell state is activated by tanh to limit the value between -1 to 1. The final extracted feature is the gated version of this representation:

$$h_i^t = \tanh(s_i^t) \, o_i^t \quad (8)$$

In order to unroll the model, a connection method between two onetime-slice models needs to be defined. From the above equations, it can be observed that the cell state and the output feature at time step *t-1* are connected to the model at time step *t* as the inputs. The reason to select the two values is that we want to transfer information during unrolling. The two values can be viewed as the summary of the previous information

storage. In the context of LSTM, the connection between the cell states are called internal loops since the representation of information are kept internal and are not yet gated. The connection between the extracted features are called the external loop since the outputs of the LSTM are connected.

3.2. *Model structure concatenation*

The output of LSTM is the extracted feature vector of the previously acquired information $\{x^1,x^2,x^3,..,x^T\}$. We utilize such a feature vector to predict the current HFR value. We add a multiple layer perceptron (MLP) model on top of the last output of LSTM model. The MLP model consists of multiple dense layers of neural networks. Figure 3b shows the structure of such a model. The final output of the model is the current HFR prediction.

Since our MLP model is concatenated directly on top of the last LSTM extracted feature, we train the two models simultaneously. We call this model LSTM-MLP model. Our final deployed model structure is shown in Figure 3c.

3.3. *Model implementation*

3.3.1. *Data organization*

We down-sample the raw sensor data, originally collected at a high sampling rate of 10 Hz, to reduce the computational burden and training time. We down-sample the sensor data at a new sampling rate of 0.5 Hz without introducing data distortion.

For testing, we keep the original sensor readouts from the test sets. As the current is the main controllable input for our fuel cell system and is positively related to the heat and water generation inside the fuel cell, it has a major influence on HFR. We deliberately choose data sets that contain three different patterns of current variation as our test sets. The three pattern types are uniform (constant) current, upramp (increasing) current and cyclic current. The test sets described in Section 4.1.2 are based on this categorization. It should be noted that other operating parameters like air flow rate, cooling water flow rate, hydrogen re-circulation, etc. were controlled by the on-board control logic unit and were dependent on the total current. Experiment data was collected within this pre-optimized parameter space. Operating conditions outside of this parameter space may lead to stack damage and is unlikely to happen during normal fuel cell operations. For example, if the hydrogen flow rate at the anode is too low, water accumulation and hydrogen starvation will occur at the anode, which will result in irreversible fuel cell performance degradation [18, 20].

For a true performance assessment, test sets that are too similar to training sets are not desirable. For this, we compare the current in the training set against that in the test set. This comparison is shown in Figure 4. Most of the current sets in the test group are highly dissimilar to those in training set. The *trend* of the current in test group are also not similar to those in the training set. It should be noted that the operating current of the fuel cell stack can be changed almost instantly during the experiment, but

the HFR value, which is closely related to the membrane water content in the membrane, reflect the cumulative effect of the operation over a much longer time-scale.

3.3.2. *Input data and its format*

The inputs consist of several important signals from the fuel cell system and intrinsic model parameters. The quantities of the fuel cell system are

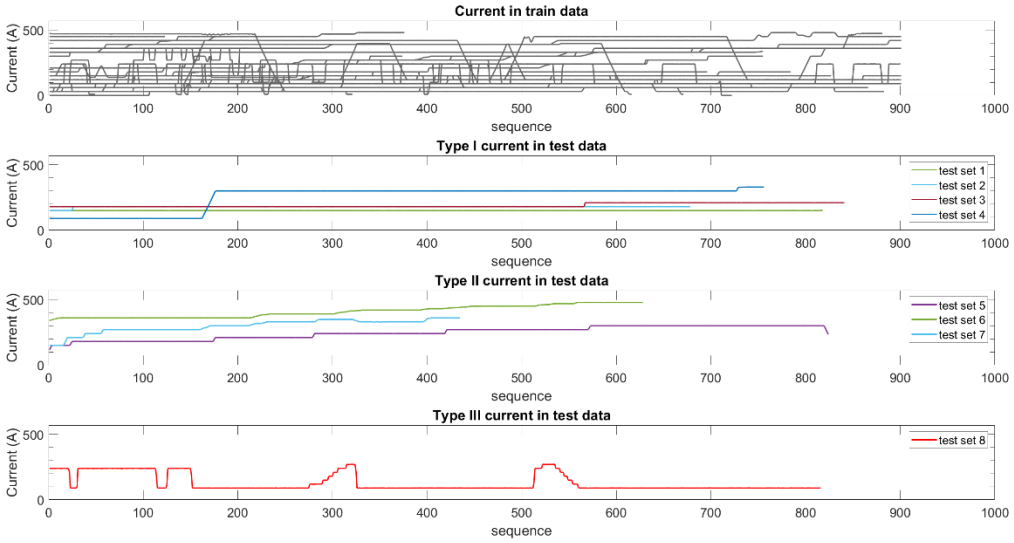


Figure 4: The inputs required for the model to predict a single HFR value.

selected because they can affect temperature or water content in the fuel cell, thus, can also influence HFR. For example, the current is selected as an important input for the HFR prediction model. The fuel cell stack current has a strong influence on the HFR value. The total current of the fuel cell stack is proportional to the electrochemical reaction rate, and thus the amount of water generated inside the stack, as governed by Eq. (9).

$$m_{water} = \frac{M_{water} \times I}{2F} \tag{9}$$

where m_{water} is the rate of water generated inside the fuel cell, *I* is the total current, $F = 96485.33\ C/mol$ is the Faraday constant, and $M_{water} = 18.02\ g/mol$ is the molar weight of water. The amount of water in the fuel cell will directly affect the hydration level of the Nafion membrane and thus the HFR. Depending on the water production rate and the concentration gradient within the membrane, the dissolved water flux can be expressed by Eq.(10) [51]

$$\dot{n}_{water,diff} = -D_{DW}^{eff} \nabla c_{water} \tag{10}$$

where the $\dot{n}_{water,diff}$ is the dissolved water molar flux inside the polymer membrane, D_{DW}^{eff} is the diffusion coefficient of the dissolved water and the ∇c_{water} is the gradient of the water concentration in the polymer membrane.

The current density will also affect the water transport related to the electro-osmotic drag effect, as can be expressed by Eq.(11) [52].

$$\dot{n}_{water,drag} = -\nabla(n_d I)/F \tag{11}$$

where the $\dot{n}_{water,drag}$ is the water molar flux due to electro-osmotic drag, n_d is the electro-osmotic drag coefficient.

The fuel cell stack voltage is also a critical input for the HFR prediction model. The combination of the cell voltage and current reflects the operating point of the fuel cell, the later of which will determine the amount of irreversible losses and heat generation inside the system [53, 54]. 47Specifically, the heat generation inside the fuel cell can be approximated by Eq. (12).

$$Q = \left(\frac{HHV}{2F} - V_{cell}\right) I \quad (12)$$

where Q is the heat generation, HHV is the higher heating value of the fuel and V_{cell} is the fuel cell operating voltage. Higher heat generation will elevate the temperature of the fuel cell, increase the water saturation pressure, accelerate the drying out process of the fuel cell membrane, and lead to the increase of the HFR.

Gas flow rates and gas temperature values at the anode and the cathode will determine the amount of water vapor brought in and out of the fuel cell system and can also affect the temperature inside the system [55]. As for the cathode, the air flow rate is monitored by the flow meter and is directly used for the HFR prediction model. The anode hydrogen flow rate was not directly measured. Instead, the anode hydrogen inlet pressure, and the hydrogen pressure difference, which is roughly proportional with the gas flow rate, are used as the model input. It is worth mentioning that the anode hydrogen gas is cycled in a closed system. At certain conditions, liquid droplet may exist in the anode hydrogen gas cycling loop, making it hard to directly measure the anode gas flow rate. Because of that, the system doesn't have an anode gas flow rate sensor, and anode gas flow rate data was not used as the model input. Due to cost and system complexity concerns, relative humidity sensors were not integrated in the PROME® P390 system. Because of that, relative humidity data for the anode and cathode were also not used as a HFR prediction model input. The water-cooling system is used to bring waste heat out of the fuel cell stack. As mentioned before, higher temperature will accelerate the drying-

out process of the membrane and leads to increased HFR. The water inlet pressure and pressure difference from the water-cooling system are related to the water flow rate. The water inlet temperature and water temperature difference determines the amount of heat transferred. The cooling capacity Q_{cool} of the water cooling system can be approximated with Eq. (13)

$$Q_{cool} = \dot{m}_{cool} C_{p,water} (T_{in} - T_{out}) \quad (13)$$

where $\dot{m}_{cool}$ is the water flow rate in the water cooling system, $C_{p,water}$ is the specific heat capacity of water and the $T_{in} - T_{out}$ is the inlet and outlet water temperature difference. Due to the lower heating capacities, the cathode and anode gas flows have significantly lower cooling capacities. We note that this data driven approach is designed specifically for automotive fuel cell systems with no external humidifiers and recirculation for the anode. For fuel cell systems with external humidifiers, extra parameters like the humidifier temperature need to be included in the model inputs in order to generate accurate prediction.

The organization of the model inputs is shown in Figure 5. A single column in the figure corresponds to the current data block in Figure 3a. In the same column, the other two signals from the model correspond to the previously extracted feature block and the previous cell state block in Figure 3a. For a model length of *T*, which is shown in Figure 3c, *T* such columns are required to predict a single HFR value at time step *T*.

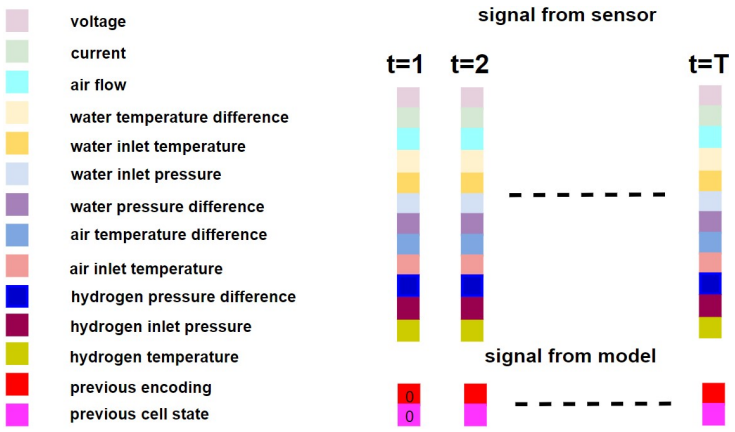


Figure 5: The inputs required for the model to predict a single HFR value.

3.3.3. *Model parameters and training parameters*

In LSTM model, the model length decides how much past data points to look back to predict the current quantity. For the LSTM model, a model length of T = 10 is used. The dimension of the cell state, which always aligns with the extracted feature vector, is set as 512. To be noted, this parameter choice can be changed depending on the application environment. We choose 512 to illustrate that the LSTM-MLP model can have low test error. It is also a convenient choice for our other analysis. However, the cell state dimension is an important hyper parameter for LSTM. We further discuss its effect in the next section. For MLP, a common hyper parameter setup is used. In this work, one dense layer with ReLU activation is used, as shown in Eq. (14). We set the number of hidden units in the dense layer as 256. For the complete model, we select the mean square error (MSE) as the loss. Adam optimizer is used so that

the learned gradient is adaptively corrected during the training [56]. The algorithm is trained on an Intel i7 CPU and an Nvidia 1080 GPU.

$$ReLU(x) = \begin{cases} x, & for\ x \geq 0 \\ 0, & for\ x < 0 \end{cases} \quad (14)$$

For other comparison algorithms, please see Section 1 in supplementary material for further details.

4. Results and discussion

4.1. *Accuracy and algorithm comparison*

We compare our LSTM-MLP model against several other commonly used regression models including linear support vector regression (L-SVR), support vector regression with Gaussian kernel (GK-SVR) and artificial neural networks (ANN).

4.1.1. *Test criteria*

We use three test criteria to test the algorithm performance: (1) root mean squared error (RMSE), (2) mean absolute error (MAE), and (3) mean absolute percentage error (MAPE), Eq. (15). The mean square error is the loss function for our training process. Thus, RMSE is selected as the test criteria for our test data. Nonetheless, a more natural error measure is to directly compute the absolute difference between the predicted HFR value and the true HFR value, and calculate how much this error accounts for in the ground truth. Thus, we also choose MAE and MAPE as the test criteria.

Table 1: Prediction results of different algorithms for three current types: constant, increasing, and cyclic.

	RMSE		
	Current type I	**Current type II**	**Current type III**
LSTM-MLP	3.77	4.70	3.51
L-SVR	6.34	5.47	5.58
GK-SVR	4.09	7.48	4.89
ANN	4.76	8.79	4.69

$$RMSE = \sqrt{\frac{\sum_{n=1}^{N}(y_n - \hat{y}_n)^2}{N}}$$

$$MAE = \frac{\sum_{n=1}^{N}|y_n - \hat{y}_n|}{N} \tag{15}$$

$$MAPE = \frac{100\%}{N}\sum_{n=1}^{N}\left|\frac{y_n - \hat{y}_n}{\hat{y}_n}\right|$$

In the above equations, y_n is the n^{th} prediction, $\hat{y}_n$ is the corresponding ground truth. N is the total number of the prediction points.

4.1.2. *Prediction for fuel cell system under different current patterns*

We now evaluate our model on test datasets involving different current patterns, which is the main controllable factor for our fuel system, and compare the model with other frequently used regression algorithms. The results show that LSTM-MLP model outperforms the other regression algorithms under all conditions.

Figure 6a shows HFR prediction results controlled by a constant current (type I). We see that HFR changes smoothly within the data set. LSTMMLP and GK-SVR make accurate predictions for most of the part. ANN has a small prediction noise for the test set 2 and test set 4. L-SVR has a large noise when predicting test set 1 and test set 4. This is very likely caused by the noisy inputs in the test set. The RMSE error is shown in Table 1. For this type of current, LSTM-MLP has the lowest RMSE error while L-SVR performs the worst mainly due to its limited smoothing effect on test set 1 and test set 4.

Figure 6b shows the prediction under the upramp current (type II), where HFR measurement can exhibit a sudden change. It can be observed that all algorithms capture the general trend for test set 5. However, for test set 7, only LSTM-MLP captures the decreasing change and the prediction has an offset from the ground truth. For test set 6, GK-SVR, LSTM-MLP and ANN manage to keep the predictions in a reasonable range but fail to predict the trend. However, L-SVR captures the HFR changes for this data set. In Table 1, we see that L-SVR has a lower RMSE compared to that of GK-SVR and ANN due to its strong performance for test set 6. However, LSTM-MLP has the lowest overall RMSE. This is mainly due to its strong performance for test set 5 and test set 7 and above average performance for test set 6.

Prediction results of a fuel cell system controlled by cyclic current is shown in Figure 6c, where HFR exhibits the most dynamic change. All of the algorithms can capture the general trend. However, LSTM-MLP has

the best result. In comparison, ANN is susceptible to input noise. L-SVR and GK-SVR fail to predict the large HFR change at around the 400^{th} sequence. L-SVR also has a large prediction shift in the initial sequences. As shown in Table 1, LSTM-MLP performs the best while L-SVR performs the worst.

Table 2: Summary of algorithm performances on all test sets.

	All test sets		
	RMSE	**MAE**	**MAPE**
LSTM-MLP	4.06	3.21	2.82%
L-SVR	5.96	4.65	4.11%
GK-SVR	5.51	3.95	3.48%
ANN	6.35	4.65	4.14%

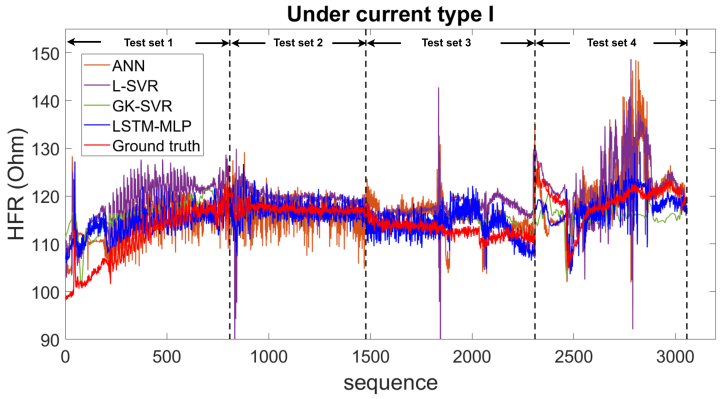
4.1.3. *Summary of algorithm performance*

A summary of the results for all test sets is shown in Table 2. LSTMMLP outperforms all the other three algorithms. It has the lowest average MAPE of 2.82%. It is known that Gaussian kernel has good smoothing effect due to its infinity filter bandwidth. LSTM-MLP's better performance over GK-SVR also indicates that the model is even more robust to input noise than the GK-SVR. The results indicate that LSTM-MLP has a strong ability to accurately capture HFR dynamics.

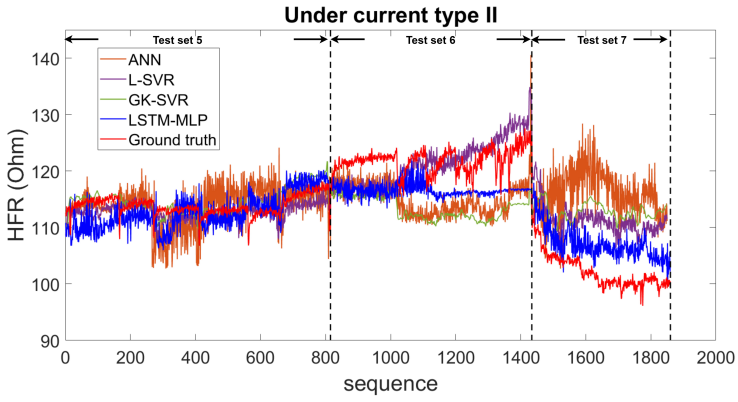
4.2. *Prediction results analysis*

In this section, we give an attempt to reveal the output of LSTM model. The output of LSTM is the extracted feature of previous information. In

Section 3.3.3, we state that we set it to 512 dimensions. However, we find that only few values in the feature vector play an important role for the dynamic change of HFR. The majority of the values in the feature vector don't change their values during the prediction course.



(a)



(b)

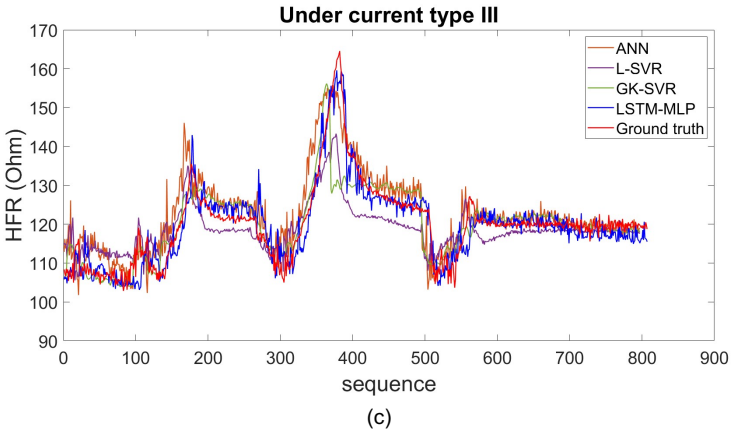


Figure 6: Comparisons of different algorithms under (a) constant current. (b) increasing current. (c) cyclic current.

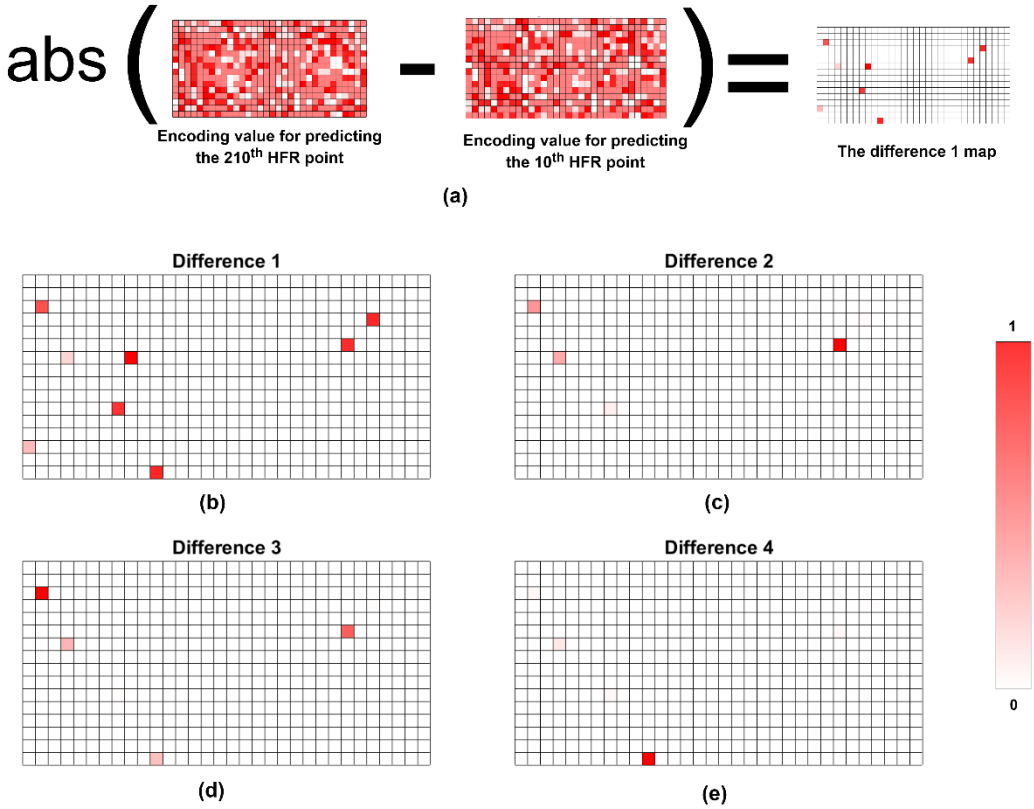


Figure 7: (a). The process of generating the difference map. (b-e) The difference between the two last extracted features of 210^{th} point and 10^{th} point, 410^{th} point and 210^{th} point, 610^{th} point and 410^{th} point, and 810^{th} point and 610^{th} point respectively.

For the setting of 512 feature dimensions, we plot a difference map to indicate the change. The difference map is a 16 × 32 image of the difference between two features. The plot is column major. For example, the difference value of the second dimension of the feature vector is plotted in the second row and the first column of the difference map. The redder a pixel is, the larger the difference of that dimension. A pure red pixel corresponds to a value of 1 and a pure white pixel corresponds to a value of 0. For this experiment, we use the test set with cyclic current. Its HFR value shows the most dynamic change among all the test sets. The data set has 826 points in total. We plot the difference map of the last feature vector of two consecutive predictions across the data set. We find that even though the feature dimension is 512, only a few dimensions show noticeable change. To concretely show that most of the dimensions do not change, we take the last feature vector at 10^{th},210^{th},410^{th},610^{th} and 810^{th} data points and plot the difference of the two consecutive sampled features. Figure 7a shows how a difference map is generated. There are four difference maps in total from the four sampled points. Their results are shown in Figure 7b to Figure 7e. If we look at each of the images, it is clear that most of the dimension values do not change even after 200 points. If we compare each of the difference maps, it is also apparent that the dimensions that are changing are usually the same.

Table 3: Frequency of the change of the feature vector for different dimensions.

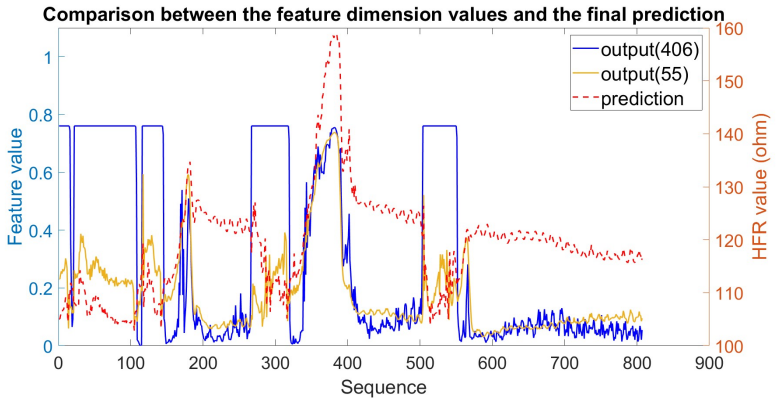
Dimension number	55	406	123	19	436	176	161	135	172	14	45	191
Frequency	773	573	557	530	434	409	245	139	34	20	1	1

Table 3 shows the dimensions that change its value frequently across the data set. The difference of the model's last feature of two consecutive predictions is calculated and normalized. The threshold of 0.01 is used to decide whether the value of a dimension is changed. Among 512 dimensions, only 12 dimensions are changing across the data set under such criteria.

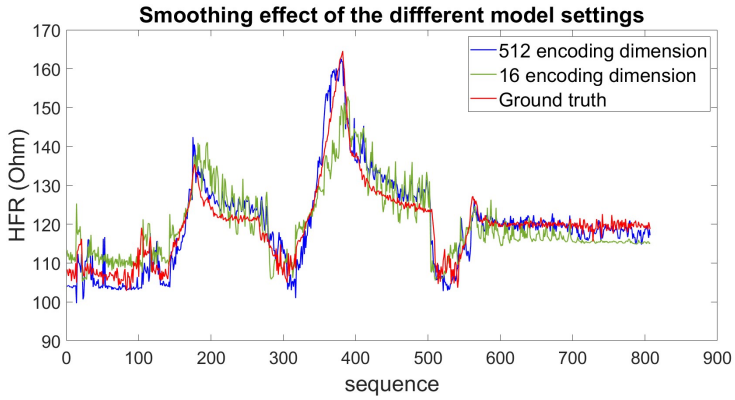
Because the weights for the MLP model are fixed after training, unchangeable dimensions do not bring any variation into the prediction. Even the true HFR value is changing in some regions. Thus, it is reasonable to infer that the prediction of the dynamics of HFR are taken care of only by a few changing dimensions. The first two frequently changed dimensions are plotted in Figure 8a and is demonstrated in our inference. From the figure, we can tell that 55^{th} dimension almost forms the same shape as the final prediction. 406^{th} dimension has large responses at peaks and valleys of HFR. From our work, we find that among all changing dimensions, those whose frequencies are above 400 have obvious responses at the peak or the valley of the prediction. For the other dimensions with low frequencies, the responses either have partial response at some extreme value regions of the prediction or are completely noise. Figures of values of different dimensions of the last feature vector whose frequencies are higher than 400 is attached in Section 3 in supplementary material.

The above observation leads us to a conclusion that only a very small portion of the feature dimensions influence the HFR dynamics change

while most of the dimensions remain constant. To better study its effect, we perform a grid search on the feature dimensions. We successively reduce the dimension of the extracted feature vector to 128, 64 and 16. We then re-train our algorithm and test it on the same three types of test sets. We provide the results along with the frequently used GK-SVR in Table 4. RMSE values tend to become slightly larger as the extracted feature dimensions become smaller. However, the RMSE is still smaller than GK-SVR. From the prediction results we observe that LSTM-MLPs with small extracted feature dimensions have HFR predictions results similar to their large-dimension counterparts but with worsening smoothing effect (Figure 8b). It can be seen that LSTMMLP with 512 extracted features has a smaller variation than that of the 16 extracted feature dimension.



(a)



(b)

Figure 8: (a) The final prediction and the values of frequently changed dimensions of the feature vector. (b) Smoothing effect comparison for different encoding dimensions.

The runtime of the algorithms over a dataset is also tested. We test each algorithm on an i7 cpu restricted to a single thread. We let the algorithm forward pass the entire test set once and record the time. For each algorithm, we repeat the test 100 times and retrieve the averaged run time. As shown in Table 4, the time only increases by 27.4% from 512 dimension to 16 dimensions although the number of parameters is reduced from 1206785 to 6465. However, we expect the positive gain in runtime to be further amplified on an on-board device that is less powerful. Although the runtime increases by reducing the number of extracted dimensions, we find that GK-SVR is marginally faster than all the LSTM-MLP algorithms.

Table 4: Performance comparison of different feature dimension setups.

LSTM-MLP encoding dimension comparison							**GK-SVR**
Encoding dimension	16	32	64	128	256	512	
RMSE	4.61	4.47	4.52	5.49	3.86	4.06	5.51
Speed (sec)	0.82	0.81	0.84	1.02	1.08	1.13	0.67

4.3. *The effect of the inputs on the model's prediction*

While all the selected sensor inputs to our algorithm can have an effect on the HFR of the fuel cell membrane, whether they are actually informative for HFR prediction are *a priori* unknown. As such, we aim to quantify the influence of each model input on the HFR prediction. We use the LSTM-MLP model with 512 extracted feature dimensions. We check the input impact by excluding the input and retraining the same model as before. We compare the RMSE value on all tests with the RMSE value of LSTM-MLP in Table 2. The result is shown in Table 5. From the table, we see that for most of the inputs, RMSE increases after the exclusion of an input. This demonstrates that most of the inputs have an effect in predicting HFR, albeit by different amounts. For example, the air is directly fed into the fuel cell. Therefore, air inlet temperature is a major factor that can affect the inlet water saturation pressure and the fuel cell internal temperature, thus the HFR value. We see that in the table this quantity has a large effect in predicting the HFR. Another example is cooling water temperature difference, which represents the amount of heat brought out of the system and is a good indicator of the fuel cell system internal temperature change and the water saturation process, and thus the relative humidity and the HFR change. We see in the table that the cooling water temperature also has a large influence in predicting the HFR value. In addition, voltage also exhibits significant effect because it reflects the internal heat generation at a given current density, and thus affects the water saturation pressure, the water vapor transport process, and the HFR. The current of the fuel cell

also has a significant influence in the prediction, as it is proportional to the oxygen reduction reaction rate and water generation rate at the cathode. However, not all inputs have a significant impact on the prediction. For instance, when the air temperature difference is excluded, the RMSE decrease is 1.23%, which indicates this signal's minor influence on the prediction model.

4.4 *Prediction stability*

The proposed LSTM-MLP model does not use predicted HFR as an input for the next-step prediction. Thus, unlike many recursive algorithms, any error from the prediction does not accumulate over time. However, the proposed LSTM-MLP model does not guarantee the stability due to the fact that its final layer is fully connected and we impose no bounds on the resulting predictions. The noise from the sensor readings could damage the prediction results. Thus, the sensor readouts need to be smoothed before fed into the model to enhance stability.

The stability guarantee of data-driven methods in dynamical system control is a challenging topic and currently an open research area. For practical use, a lower bound and upper bound can be set on the prediction result to ensure the stability. For a more comprehensive discussion of this issue, we refer the readers to the studies focused on stability issues in the data-drive methods [57][58].

5. Conclusion

This work presents a deep learning method for predicting HFR in fuel cells. Our LSTM- and MLP- based model, can take the previous and current information of the sensor data for a real-time prediction of HFR in operational, deep-stack fuel cells. The fuel cell system is controlled by an applied current, while the training set and test set are collected across different current patterns. The model's prediction results have been studied on test sets that contain three different current patterns. The performance of the algorithm is analyzed. The following conclusions are drawn:

- The presented model achieves high prediction accuracy under varied current patterns. It has an overall RMSE of 4.06 and MAPE of 2.82%, which outperforms other frequently used regression models.
- We find that the LSTM part of the model only uses a small amount of parameters to predict HFR value change and, thus The model can achieve runtime performance with high accuracy by reducing the model size; however, it is also observed that such an LSTM based model will have an decreased smoothing effect, which results in slightly increased prediction errors.
- We demonstrate that most of the sensor data used as model input are informative to the HFR prediction. Air inlet temperature, hydrogen temperature, fuel cell voltage and cooling water temperature difference plays important role for the HFR prediction.

However, we also identify a few theoretically critical factors that did not seem to have a major impact on our predictions. Specifically, in our studies, we find that the difference between the inlet and outlet air temperature does not significantly affect the results and the air flow rate has only a minimal effect on the HFR prediction accuracy.

The model is designed for computationally efficient real-time predictions. Once the model is trained off-line, it can be executed very fast on board. The speed can be further increased by reducing the model size while the accuracy only slightly decreases. Considering the high prediction accuracy and runtime performance, the model may prove promising for fuel cell vehicles, which usually control their fuel cell system using embedded controllers.

Table 5: Model input effect on model prediction performance.

	Algorithm performance after an input is excluded					
Excluded input	Voltage	Current	Air flow	Water inlet temperature	Water temperature difference	Water inlet pressure
RMSE	4.51	4.36	4.15	4.29	4.48	4.43
RMSE increase	11.36%	7.65%	2.47%	5.93%	10.62%	9.38%
Excluded input	Water pressure difference	Air inlet temperature	Air temperature difference	Hydrogen inlet pressure	Hydrogen pressure difference	Hydrogen temperature
RMSE	4.31	4.66	4.00	4.44	4.26	4.58
RMSE increase	6.42%	15.06%	-1.23%	9.63%	5.19%	13.09%

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the results in the paper.

CRediT author statement

Tong Lin: Conceptualization, Methodology, Software, Validation, Formal analysis, Writing - Original Draft, Writing - Review & Editing, Visualization **Leiming Hu**: Validation, Resources, Writing - Original Draft, Writing - Review & Editing, Visualization **Willetta Wisely**: Formal analysis, Writing - Review & Editing **Xin Gu**: Investigation **Jun Cai**: Project administration, Funding acquisition **Shawn Litster**: Conceptualization, Writing - Review & Editing, Supervision, Project administration **Levent Burak Kara**: Conceptualization, Writing - Review & Editing, Supervision, Project administration

Acknowledgment

This research is completed through the collaboration with Shanghai Hydrogen Propulsion Technology (SHPT). We would also like to thank for the financial support from SHPT.

References

[1] D.-Y. Lee, A. Elgowainy, A. Kotz, R. Vijayagopal, J. Marcinkoski, Lifecycle implications of hydrogen fuel cell electric vehicle technology for medium-and heavy-duty trucks, Journal of Power Sources 393 (2018) 217–229.

[2] A. Kongkanand, M. F. Mathias, The priority and challenge of highpower performance of low-platinum proton-exchange membrane fuel cells, The journal of physical chemistry letters 7 (7) (2016) 1127–1137.

[3] J. Wang, H. Wang, Y. Fan, Techno-economic challenges of fuel cell commercialization, Engineering 4 (3) (2018) 352–360.

[4] K. Epping Martin, J. P. Kopasz, K. W. McMurphy, Status of fuel cells and the challenges facing fuel cell technology today, in: Fuel cell chemistry and operation, ACS Publications, 2010, pp. 1–13.

[5] K. Sopian, W. R. W. Daud, Challenges and future developments in proton exchange membrane fuel cells, Renewable energy 31 (5) (2006) 719–727.

[6] J. J. Giner-Sanz, E. Ortega, V. P´erez-Herranz, Mechanistic equivalent circuit modelling of a commercial polymer electrolyte membrane fuel cell, Journal of Power Sources 379 (2018) 328–337.

[7] J. Kim, I. Lee, Y. Tak, B. Cho, Impedance-based diagnosis of polymer electrolyte membrane fuel cell failures associated with a low frequency ripple current, Renewable energy 51 (2013) 302–309.

[8] X. Yan, M. Hou, L. Sun, D. Liang, Q. Shen, H. Xu, P. Ming, B. Yi, Ac impedance characteristics of a 2 kw pem fuel cell stack under different operating conditions and load changes, International Journal of Hydrogen Energy 32 (17) (2007) 4358–4364.

[9] X. Yuan, J. C. Sun, M. Blanco, H. Wang, J. Zhang, D. P. Wilkinson, Ac impedance diagnosis of a 500 w pem fuel cell stack: Part i: Stack impedance, Journal of Power Sources 161 (2) (2006) 920–928.

[10] X. Yuan, J. C. Sun, H. Wang, J. Zhang, Ac impedance diagnosis of a 500 w pem fuel cell stack: Part ii: Individual cell impedance, Journal of Power Sources 161 (2) (2006) 929–937.

[11] R. L. Edwards, A. Demuren, Interface model of pem fuel cell membrane steady-state behavior, International Journal of Energy and Environmental Engineering 10 (1) (2019) 85–106.

[12] M. Zhu, X. Xie, K. Wu, K. Jiao, et al., Experimental investigation of the effect of membrane water content on pem fuel cell cold start, Energy Procedia 158 (2019) 1724–1729.

[13] T. Van Nguyen, A. Ahosseini, X. Wang, V. Yarlagadda, A. Kwong, A. Z. Weber, P. Deevanhxay, S. Tsushima, S. Hirai, Hydrophobic gas-

diffusion media for polymer-electrolyte fuel cells by direct fluorination, Journal of The Electrochemical Society 162 (14) (2015) F1451–F1460.

[14] X. Ye, C.-Y. Wang, Measurement of water transport properties through membrane-electrode assemblies i. membranes, Journal of The Electrochemical Society 154 (7) (2007) B676–B682.

[15] S. G. Kandlikar, Z. Lu, N. Rao, J. Sergi, C. Rath, C. McDade, T. Trabold, J. Owejan, J. Gagliardo, J. Allen, et al., Visualization of fuel cell water transport and performance characterization under freezing conditions, Tech. rep., Rochester Institute of Technology, Rochester, NY (United States) (2010)

[16] H. Nara, T. Momma, T. Osaka, Impedance analysis of the effect of flooding in the cathode catalyst layer of the polymer electrolyte fuel cell, Electrochimica Acta 113 (2013) 720–729.

[17] T. Kadyk, R. Hanke-Rauschenbach, K. Sundmacher, Nonlinear frequency response analysis of pem fuel cells for diagnosis of dehydration, flooding and co-poisoning, Journal of Electroanalytical Chemistry 630 (1-2) (2009) 19–27.

[18] L. Hu, B. K. Hong, J.-G. Oh, S. Litster, Robust operation of fuel cell systems in subfreezing conditions: A material-based solution to achieve better anode durability, ACS Applied Energy Materials 2 (10) (2019) 7152–7161.

[19] B. K. Hong, P. Mandal, J.-G. Oh, S. Litster, On the impact of water activity on reversal tolerant fuel cell anode performance and durability, Journal of Power Sources 328 (2016) 280–288.

[20] P. Mandal, B. K. Hong, J.-G. Oh, S. Litster, Understanding the voltage reversal behavior of automotive fuel cells, Journal of Power Sources 397 (2018) 397–404.

[21] M. A. Danzer, E. P. Hofer, Electrochemical parameter identification—an efficient method for fuel cell impedance characterisation, Journal of Power Sources 183 (1) (2008) 55 – 61. doi:https://doi.org/10.1016/j.jpowsour.2008.04.071.URLhttp://www.sciencedirect.com/science/article/pii/S0378775308009403

[22] J. Nam, P. Chippar, W. Kim, H. Ju, Numerical analysis of gas crossover effects in polymer electrolyte fuel cells (pefcs), Applied energy 87 (12) (2010) 3699–3709.

[23] Y.-S. Chen, C.-W. Yang, J.-Y. Lee, Implementation and evaluation for anode purging of a fuel cell based on nitrogen concentration, Applied energy 113 (2014) 1519–1524.

[24] C.-W. Yang, Y.-S. Chen, A mathematical model to study the performance of a proton exchange membrane fuel cell in a dead-ended anode mode, Applied energy 130 (2014) 113–121.

[25] Y. Wang, B. Seo, B. Wang, N. Zamel, K. Jiao, X. C. Adroher, Fundamentals, materials, and machine learning of polymer electrolytemembrane fuel cell technology, Energy and AI 1 (2020) 100014.doi:https://doi.org/10.1016/j.egyai.2020.100014.

[26] Z. C. Lipton, A critical review of recurrent neural networks for sequence learning, CoRR abs/1506.00019 (2015). arXiv:1506.00019.

[27] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (8) (1997) 1735–1780.

[28] A. T. Mohan, D. V. Gaitonde, A deep learning based approach to reduced order modeling for turbulent flow control using lstm neural networks (2018). arXiv:1804.09269.

[29] P. A. Srinivasan, L. Guastoni, H. Azizpour, P. Schlatter, R. Vinuesa, Predictions of turbulent shear flows using deep neural networks, Physical Review Fluids 4 (5) (May 2019). doi:10.1103/physrevfluids.4.054603.
URL http://dx.doi.org/10.1103/physrevfluids.4.054603

[30] V. Rajendran, K. Y. Kelly, E. Leonardi, K. Menzies, Vortex detection on unsteady cfd simulations using recurrent neural networks, 2018.

[31] Z. Deng, Y. Chen, Y. Liu, K. Kim, Time-resolved turbulent velocity field reconstruction using a long short-term memory (lstm)based artificial intelligence framework, Physics of Fluids 31 (07 2019). doi:10.1063/1.5111558.

[32] A. Mohan, D. Daniel, M. Chertkov, D. Livescu, Compressed convolutional lstm: An efficient deep learning framework to model high fidelity 3d turbulence (2019). arXiv:1903.00033.

[33] C. Schenck, D. Fox, Spnets: Differentiable fluid dynamics for deep neural networks (2018). arXiv:1806.06094.

[34] J. Wang, Y. Li, Multi-step ahead wind speed prediction based on optimal feature extraction, long short term memory neural network and error correction strategy, Applied Energy 230 (2018) 429–443. doi:10.1016/j.apenergy.2018.08.114.

[35] Y. Qin, K. Li, Z. Liang, B. Lee, F. Zhang, Y. Gu, L. Zhang, F. Wu, D. Rodriguez, Hybrid forecasting model based on long short term memory network and deep learning neural network for wind signal, Applied Energy 235 (C) (2019) 262-272. doi: 0.1016/j.apenergy.2018.1.

[36] Z. Zhang, L. Ye, H. Qin, Y. Liu, C. Wang, X. Yu, Y. Xingli, J. Li, Wind speed prediction method using shared weight long short-term memory network and gaussian process regression, Applied Energy 247 (2019) 270–284. doi:10.1016/j.apenergy.2019.04.04.

[37] B. Wang, G. Zhang, H. Wang, J. Xuan, K. Jiao, Multi-physics-resolved digital twin of proton exchange membrane fuel cells witha data-driven surrogate model, Energy and AI 1 (2020) 100004.doi:https://doi.org/10.1016/j.egyai.2020.100004.

[38] B. Wang, B. Xie, J. Xuan, K. Jiao, Ai-based optimization of pemfuel cell catalyst layers for maximum power density via data-driven surrogate modeling, Energy Conversion and Management 205(2020)33112460.doi:https://doi.org/10.1016/j.enconman.2019.112460..

[39] R. Xie, R. Ma, S. Pu, L. Xu, D. Zhao, Y. Huangfu, Prognostic for fuel cell based on particle filter and recurrent neu-ral network fusion structure, Energy and AI 2 (2020) 100017.doi:https://doi.org/10.1016/j.egyai.2020.100017.

[40] R. Ma, T. Yang, E. Breaz, Z. Li, P. Briois, F. Gao, Data-driven proton exchange membrane fuel cell degradation predicationthrough deep learning method, Applied Energy 231 (2018) 102–115.doi:10.1016/j.apenergy.2018.09.111.

[41] K. Chen, S. Laghrouche, A. Djerdir, Degradation model of proton exchange membrane fuel cell based on a novel hybrid method, Applied Energy 252 (2019) 113439. doi: 10.1016/j.apenergy.2019.113439.

[42] H. Zhang, D. Yang, K. Tao, X. Zheng, J. Ma, Investigation of pemfc stack operating at elevated temperature, World Electric Vehicle Journal 4 (3) (2010) 481–486.

[43] K. Kojima, K. Fukazawa, Current status and future outlook of fuel cell vehicle development in toyota, ECS Transactions 69 (17) (2015) 213.

[44] T. Suzuki, Fuel cell stack technology of toyota, ECS Transactions 75 (14) (2016) 423.

[45] Y. Hiroyuki, K. Mikio, A. Hisao. Toyota Fuel Cell System (TFCS). World Electric Vehicle Journal. 2015; 7(1):85-92.

[46] D.H. Lee. Separator for fuel cell having channels for self-humidification: U.S. Patent 7,846,607[P]. 2010

[47] H.H. Voss, R.H . Barton, B.W. Wellset, J.A. Ronne, H.A. Nigsch. Solid polymer fuel cell system and method for humidifying and adjusting the temperature of a reactant stream: U.S. Patent 6,106,964[P]. 2000.

[48] S. Hiroshi, K. Toshikatsu, K. Yoshio, S. Motohiro. Fuel cell system and humidification method: U.S. Patent 7,087,328[P]. 2006..

[49] Figure of the prome p390 fuel cell stack. http://www.shpt.com/pc/product p390.html.

[50] I. Sutskever, O. Vinyals, Q. V. Le, Sequence to sequence learning with neural networks, in: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, NIPS'14, MIT Press, Cambridge, MA, USA, 2014, p. 3104–3112.

[51] S. K. Babu, H. T. Chung, P. Zelenay, S. Litster, , Journal of The Electro-chemical Society 164 (9) (2017) F1037–F1049. doi:10.1149/2.0041712jes.

[52] S. Um, C.-Y. Wang, Computational study of water transport in protonexchange membrane fuel cells, Journal of Power Sources 156 (2) (2006)211 – 223. doi:https://doi.org/10.1016/j.jpowsour.2005.05.095.

[53] M. Ay, A. Midilli, I. Dincer, Exergetic performance analysis of a pem fuel cell, International journal of energy research 30 (5) (2006) 307–321.

[54] H. Ju, H. Meng, C.-Y. Wang, A single-phase, non-isothermal model for pem fuel cells, International Journal of Heat and Mass Transfer 48 (7) (2005) 1303–1315.

[55] S. G. Kandlikar, Z. Lu, Thermal management issues in a pemfc stack–a brief review of current status, Applied Thermal Engineering 29 (7) (2009) 1276–1280.

[56] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, CoRR abs/1412.6980 (2014).

[57] Berberich, J. et al. “Data-Driven Model Predictive Control with Stability and Robustness Guarantees.” ArXiv abs/1906.04679 (2019): n. pag.

[58] Hou, Zhong-Sheng and Z. Wang. “From model-based control to data-driven control: Survey, classification and perspective.” Inf. Sci. 235 (2013): 3-35.

Supplemental material

Prediction of high frequency resistance in polymer electrolyte membrane fuel cells using Long Short Term Memory based model

Tong Lin[a], Leiming Hu[a], Willetta Wisely[a], Xin Gu[b], Jun Cai[b], Shawn Litster[a], Levent Burak Kara[a,*]

[a]*Carnegie Mellon University, Pittsburgh, PA 15213, USA*
[b]*Shanghai Hydrogen Propulsion Technology Co.Ltd, Innovation Park, Lane 56, Antuo Rd, Jiading, Shanghai 201800, P.R.China*

1. Comparison of Algorithms

Various other regression models are selected to compare with the results obtained from LSTM. The models include linear support vector regression (L-SVR), Gaussian kernel support vector machine (GK-SVR) and artificial neural network (ANN). An important reason to choose these models instead of a typical time series model such as auto regressive model is that these models can readily incorporate other physical quantities of interest besides the mere historical values of HFR.

*Corresponding author
Email address: lkara@cmu.edu (Levent Burak Kara)

1.1. Support vector regression

Support vector regression (SVR) is a frequently used regression model. An illustration is shown in Figure S.1. The optimized SVR model can be viewed as a hyperplane. The idea of SVR is similar to the support vector machine classifier. Before optimization, we can assume that there exists a hyperplane with margin support of ϵ to the hyperplane (support region). The equation for the hyperplane is shown in Eq. (S.1).

$$y = \sum_{j=1}^{M} w_j x_j + b \;\; y, b \in R, \; x, w \in R^M \tag{S.1}$$

where M is the total number of dimensions, x_j is the j^{th} feature, w_j is the corresponding weight and b is the bias.

The training points lying within the margin do not incur any loss. The goal is to minimize the L_2 norm of the weights w and the error produced by the points outside the margin. A slack variable is introduced to accommodate the training points that are outside of the support region. The objective function is shown in Eq.(S.2). Here, C is a hyperparameter that controls how tolerant the model is to the points outside the tube. This constrained convex optimization function can be solved using Karush-Kuhn-Tucker (KKT) conditions and Lagrange multipliers [1].

$$
\begin{aligned}
\underset{w}{\text{minimize}} \quad & \frac{1}{2}||w||^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) \\
\text{subject to} \quad & y_i - w^T x_i \leq \epsilon + \xi_i^* \\
& w^T x_i - y_i \leq \epsilon + \xi_i \\
& \xi_i^* \xi_i \geq 0 \\
& i \in 1, 2, 3, \ldots, N
\end{aligned}
\tag{S.2}
$$

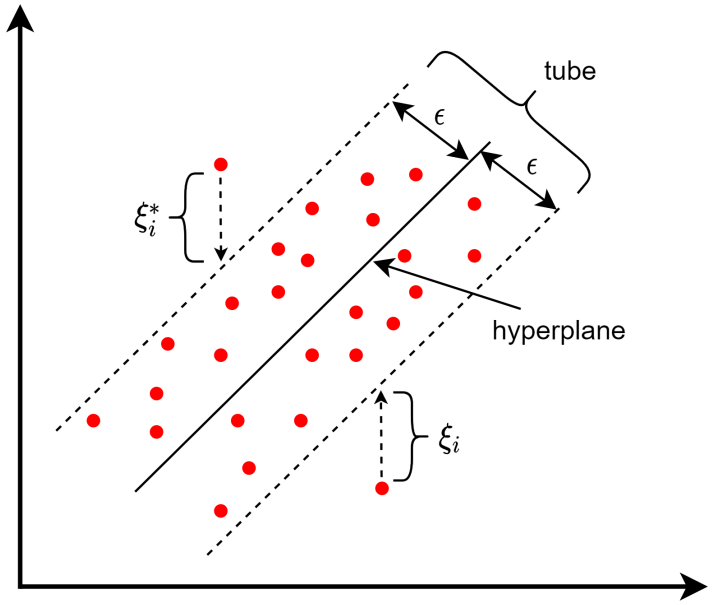


Figure S.1: Illustration of SVR

The optimized model is shown in Eq.(S.3). N_{sv} is the number of support vectors. $\alpha*$ and α are the Lagrange multipliers that corresponds to the support vectors x_i. b is the bias. Note that α_i or α_i^* has to be nonzero and they cannot both be zero. Since the inner product takes the function inputs and the support vectors as inputs, the kernel trick can be used to change the

dimension and the form of the feature vectors without re-optimizing again. If the kernel function is linear, the SVR is called linear SVR (L-SVR). If the kernel function is Gaussian, which is a frequently used kernel function for infinite feature dimension approximation, the SVR is called a Gaussian kernel SVR (GK-SVR). These two different types of SVRs are used in this paper as comparison algorithms.

$$
\begin{aligned}
f(x) &= \sum_{i=1}^{N_{SV}} (\alpha_i^* - \alpha_i) < x_i, x > + b \\
b &= \begin{cases} y_i - \sum_{i=1}^{N_{SV}} (\alpha_i^* - \alpha_i) < x_i, x_i > - \epsilon, & \text{for } \alpha_i \neq 0 \\ -y_i + \sum_{i=1}^{N_{SV}} (\alpha_i^* - \alpha_i) < x_i, x_i > - \epsilon, & \text{for } \alpha_i^* \neq 0 \end{cases}
\end{aligned}
\tag{S.3}
$$

1.2. Artificial neural network

Artificial neural network (ANN) is a common model for regression. Its structure is similar to a one-time slice model of LSTM. The structure is shown in Figure S.2. The difference between ANN and LSTM is that ANN does not explicitly model the dependency of a data sequence. Thus, there is no recursion and gate mechanism in the model. In this paper, we use ANN to compare with LSTM to see whether the explicit modelling of the sequence dependency can effectively improve the prediction of HFR, which is a time series data set.

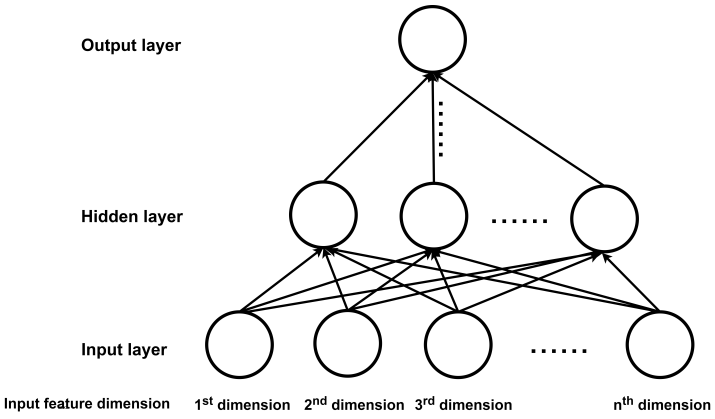


Figure S.2: The structure of an artificial neural network

1.3. Data input format for the comparison algorithms

We use 12 features along with their historical values as the inputs for the comparison models. We select 10 as the window size for the historical values. This is to keep alignment with the choice in the LSTM model. Thus, 120 values are used in total to predict a single HFR value.

2. Vanishing/exploding gradient problem of RNN

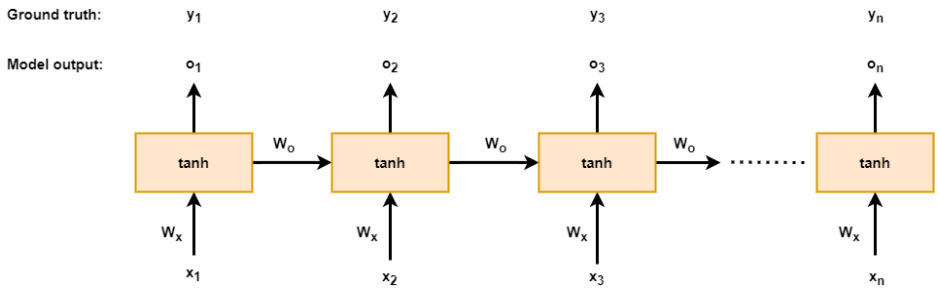


Figure S.3: Conventional RNN structure.

To optimize RNN, we use a special back propagation algorithm called back propagation through time (BPTT). We first fully unroll the model. Figure S.3 shows the model. Then, we calculate gradients of weights at each time step backwards. For a model that needs to unroll many time steps, the gradients at early steps tend to vanish. This is illustrated below.
We assume the following without loss of generality,

1. The model output o is the final prediction value.
2. Both ground truth y and model output o are one dimensional.
3. o_t is part of the input at time step $t+1$

First, the error of RNN is defined as,

$$E = \sum_{t=1}^{t=N}(e_t) = \sum_{t=1}^{t=N}(e_t(o_t, y_t)) \tag{S.4}$$

where E is the total error, $e_t(.)$ is the output of loss function at time step t.

Second, we need to calculate the gradient of weights W_o and W_x w.r.t error E. In RNN, the model weights W_x and W_o are shared at each time step. The update rule is shown in Eq. (S.5).

$$
\begin{aligned}
W_o &= W_o - \alpha_o \nabla_{W_o}(E) \\
&= W_o - \alpha_o \nabla_{W_o}(\sum_{t=1}^{t=N}(e_t(o_t, y_t))) \\
&= W_o - \alpha_o(\sum_{t=1}^{t=N} \nabla_{W_o}(e_t(o_t, y_t)))
\end{aligned} \tag{S.5}
$$

Following the same steps above, we get the updating rule for W_x.

$$
W_x = W_x - \alpha_x(\sum_{t=1}^{t=N} \nabla_{W_x}(e_t(o_t, y_t))) \tag{S.6}
$$

For both equations above, α is the learning rate. Now we only need to know $\nabla_{W_o}(e_t(o_t, y_t)$ and $\nabla_{W_x}(e_t(o_t, y_t)$.
For $\nabla_{W_o}(e_t(o_t, y_t)$, we need the chain rule,

$$
\nabla_{W_o}(e_t(o_t, y_t) = \frac{\partial e_t}{\partial o_t}\frac{\partial o_t}{\partial W_o} \tag{S.7}
$$

where, o_t is calculated as Eq. (S.8)

$$o_t = tanh(W_o o_{t-1} + W_x x_t) \tag{S.8}$$

We can see from above equation that o_t is a function of W_o and o_{t-1}, which means that o_{t-1} is also a function of W_o. Thus, in Eq. (S.7), $\frac{\partial o_t}{\partial W_o}$ is not o_{t-1} since o_{t-1} is not constant w.r.t W_o. From Eq. (S.8), we also realize that o_t is a recursive function and W_o is involved at each time step up to t. Thus,$\nabla_{W_o}(e_t(o_t, y_t))$ needs to sum up W_o's effect on o_k for $k \leq t$, where o_k can be assumed fixed. Thus, Eq. (S.7) can be rewritten as the following,

$$\begin{aligned}
\nabla_{W_o}(e_t(o_t, y_t) &= \frac{\partial e_t}{\partial o_t}\frac{\partial o_t}{\partial W_o} \\
&= \frac{\partial e_t}{\partial o_t}\sum_{k=1}^{k=t}\frac{\partial o_t}{\partial o_k}\frac{\partial o_k}{\partial W_o} \\
&= \frac{\partial e_t}{\partial o_t}\sum_{k=1}^{k=t}(\prod_{j=k}^{j=t-1}\frac{\partial o_{j+1}}{\partial o_j})\frac{\partial o_k}{\partial W_o} \qquad \text{chain rule for } \frac{\partial o_t}{\partial o_k}
\end{aligned} \tag{S.9}$$

Plug in Eq. (S.9) into Eq. (S.5), we have the final updating rule for W_o.

$$W_o = W_o - \alpha_o \sum_{t=1}^{t=N}\frac{\partial e_t}{\partial o_t}\sum_{k=1}^{k=t}(\prod_{j=k}^{j=t-1}\frac{\partial o_{j+1}}{\partial o_j})\frac{\partial o_k}{\partial W_o} \tag{S.10}$$

From Eq. (S.8) we know that $\frac{\partial o_{j+1}}{\partial o_j}$ equals to $(1 - tanh^2(W_o o_{t-1} +$

$W_x x_t))W_o$. Since $(1 - tanh^2(W_o o_{t-1} + W_x x_t))$ is between zero and one, the sequential product term in Eq. (S.10) is likely to vanish if W_o in $\frac{\partial o_{j+1}}{\partial o_j}$ is kept small. However, when W_o is kept large, the value of the sequential product term may increase exponentially, which causes exploding gradient problem. Both situations make conventional RNN difficult to train.

2.1. LSTM: How it alleviates the vanishing gradient problem

From the above illustration, we can see that the exploding/vanishing gradient problem is mainly caused by output weights W_o that is unchanged across the entire model length. The problem may be eased if the weights W_o can be changed across the model length to accommodate for each $(1 - tanh^2(W_o o_{t-1} + W_x x_t))W_o$ term. This is the core idea of how LSTM alleviates the problem.

Without loss of generality, let us assume that the LSTM model outputs one dimension and it does not have output gate. Because of the assumption, we can drop o in Eq. (8) and all subscript i. After plug in Eq. (7), Eq. (8) becomes the following,

$$h^t = tanh(f^t s^{t-1} + v^t u^t) \tag{S.11}$$

We can view u^t as the current model input, v^t is the corresponding weights. f^t and s^{t-1} are the weights and the previous cell state. Compare Eq. (S.11) with Eq. (S.8), we see that in LSTM, s^{t-1} substitute o_{t-1} to compute the final output. Also, We see that in RNN, the key problem comes

from the term $\prod_{j=k}^{j=t-1} \frac{\partial(o_{j+1}}{\partial o_j})$. Thus, in LSTM, this term becomes,

$$\prod_{j=k}^{j=t-1} \frac{\partial h^{j+1}}{\partial s^j} = \prod_{j=k}^{j=t-1} (1 - tanh^2(f^{j+1}s^j + v^{j+1}u^{j+1}))(f^{j+1} + \frac{\partial(v^{j+1}u^{j+1})}{\partial s^j}) \tag{S.12}$$

We can see that in LSTM, W_o becomes $f^{j+1} + \frac{\partial(v^{j+1}u^{j+1})}{\partial s^j}$. Because f, v, u are functions of their weights and the current model inputs, W_o now depends on current model input instead of being fixed. Since W_o can be learned to vary at each model step now, the vanishing/exploding gradient is alleviated.

3. Additional comparison for LSTM dimensions and its final prediction results

Figure S.4 below shows the comparison between the final prediction and the feature dimension values that are changing during the prediction.

References

[1] M. Awad, R. Khanna, Support Vector Regression, Apress, Berkeley, CA, 2015, pp. 67–80. doi:10.1007/978-1-4302-5990-9$_4$.
URL

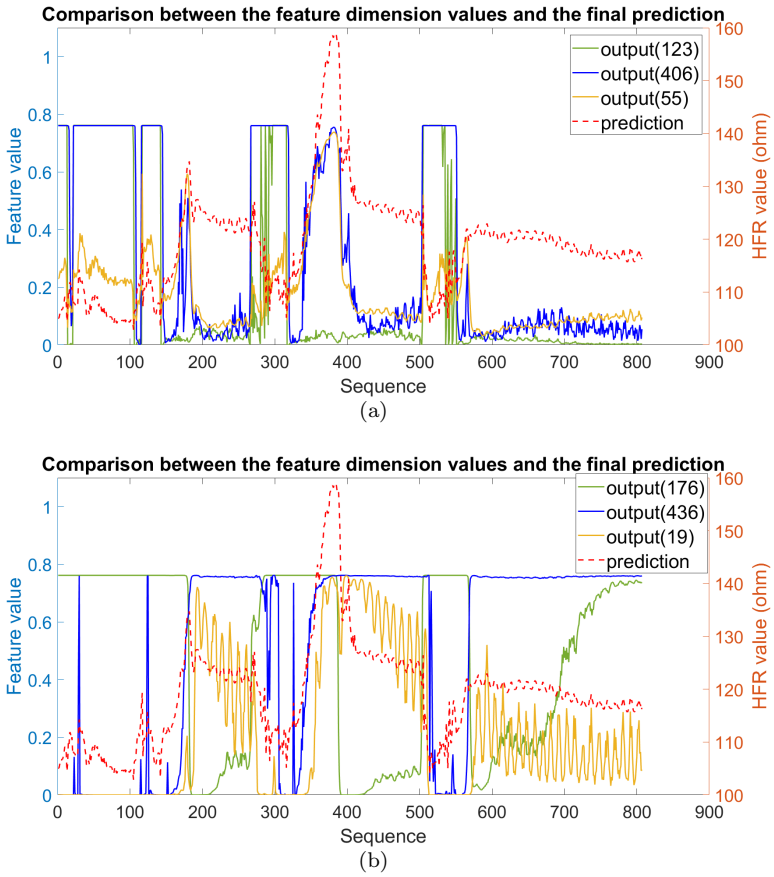


Figure S.4: Comparison between the final prediction result and the feature values at the dimension number of (a) 123^{th}, 406^{th} and 55^{th}. (b) 176^{th}, 436^{th} and 19^{th}.