

## Perspective

## Machine learning modeling for proton exchange membrane fuel cell performance

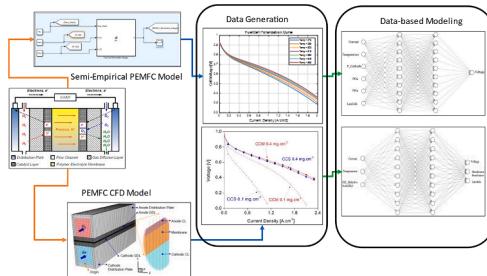
Adithya Legala, Jian Zhao, Xianguo Li <sup>\*</sup>

Department of Mechanical and Mechatronics Engineering, University of Waterloo, Waterloo, Ontario N2L 3G1 Canada

## HIGHLIGHTS

- Investigated various machine learning methods to model PEMFC performance attributes.
- Included support vector machine (SVR) and artificial neural network (ANN) with dropout technique.
- Found ANN with dropout technique advantageous for multi-variable output regression.
- Observed SVR advantageous for single-variable output regression with comparable accuracy.

## GRAPHICAL ABSTRACT



## ARTICLE INFO

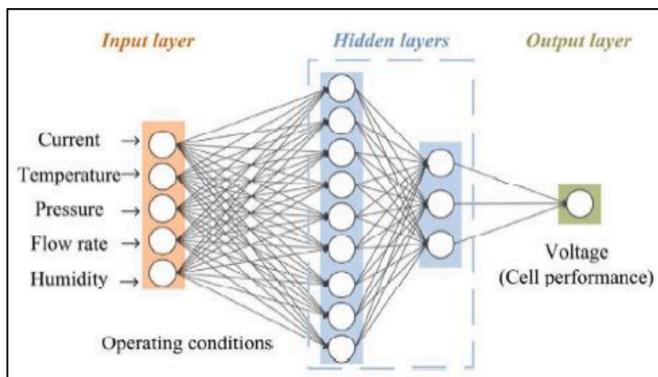
## Keywords:

Fuel cell  
Machine learning  
Artificial neural network  
Support vector machine regressor  
Data-based models

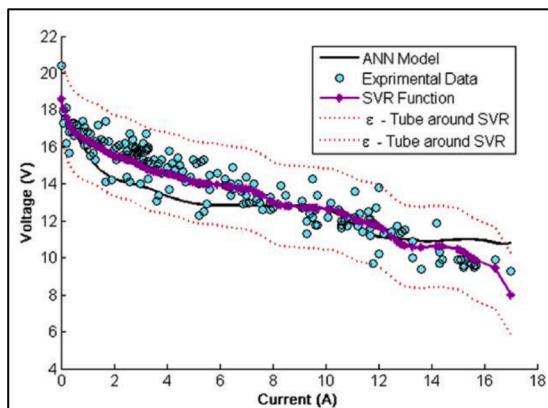
## ABSTRACT

Proton exchange membrane fuel cell (PEMFC) is considered essential for climate change mitigation, and a fast and accurate model is necessary for its control and operation in practical applications. In this study, various machine learning methods are used to develop data-based models for PEMFC performance attributes and internal states. Techniques such as Artificial Neural Network (ANN) and Support Vector Machine Regressor (SVR) are used to predict the cell voltage, membrane resistance, and membrane hydration level for various operating conditions. Varying input features such as cell current, temperature, reactant pressures, and humidity are introduced to evaluate the accuracy of the model, especially under extreme conditions. Two different sets of data are considered in this study, which are acquired from, a physics-based semiempirical model and a 1-D reduced-dimension Computational Fluid Dynamics model, respectively. The aspect of data preprocessing and hyper-parameter tuning procedures are investigated that are extensively used to calibrate the artificial neural network layers and support vector regressor to predict the fuel cell attributes. ANN clearly shows an advantage in comparison with SVR, especially on a multivariable output regression. However, the SVR is advantageous to model simple regressions as it greatly reduces the level of computation without sacrificing accuracy. Data-based models for PEMFC are successfully developed on both the data sets by adapting advanced modeling techniques and calibration procedures such as ANN incorporating the dropout technique, resulting in an  $R^2 \geq 0.99$  for all the predicted variables, demonstrating the ability to build accurate data-based models solely on data from validated physics-based models, reducing the dependency on extensive experimentation.

<sup>\*</sup> Corresponding author.E-mail address: [Xianguo.Li@uwaterloo.ca](mailto:Xianguo.Li@uwaterloo.ca) (X. Li).



**Fig. 1.** Schematic of artificial neural network architecture for fuel cell voltage prediction [22].



**Fig. 2.** Results of the support vector regression algorithm for fuel cell voltage prediction [23].

#### Abbreviations

Polymer electrolyte membrane fuel cell	PEMFC
Artificial Neural Network	ANN
Support Vector Machine	SVM
Support Vector Machine Regressor	SVR
Computational Fluid Dynamics	CFD
Group Method of Data Handling	GMDH
Recurrent Neural Network	RNN
Gas Diffusion Layer	GDL
Catalyst Layer	CL
Rectified Linear Unit	ReLU
Mean Square Error	MAE
Mean Absolute Error	MAE
Root Mean Square Error	RMSE
Learning Rate	Lr
Semi-emp	Semi-Empirical

#### 1. Introduction

A model is referred to as a virtual entity that represents the real system and emulates its behavior, this virtual entity is also referred to as a plant model and should accurately reflect all the characteristics of the system in terms of input-output relations and their sensitivities. Modeling is the manner or process of finding an appropriate mathematical representation or description of such a virtual entity. In general, physical systems modeling can be broadly categorized into empirical models (e.g., equivalent circuit models) and mathematical models which further can be divided into physics-based models and data-based models [1–3].

In most practical applications a proton exchange membrane fuel cell (PEMFC) is designed as a clean energy conversion unit and is operated in conjunction with a battery or a supercapacitor module to meet the response, power, and efficiency requirements [4,5]. This aspect needs a robust fuel cell model (plant model) to effectively evaluate and characterize the fuel cell behavior, and performance over various operating conditions to design, optimize and develop diagnostics [3,6]. A typical fuel cell model needs to incorporate properties related to porous media, polymer membrane, microscopic gas diffusion channels, electrode flooding, electron, and ion transport, and chemical kinetics across various operating conditions resulting in complex interactions affecting the accuracy of the traditional models [7,8]. As mentioned before, fuel cell models also fall into empirical, physics, and data-based categories.

Empirical models/equivalent models do not accommodate actual system components and require intensive calibration and are not suitable for performance prediction, especially when incorporating fluid, thermal management, gas dynamics, and material degradation aspects of the PEMFC. A major disadvantage of such a model is that this has no physical interpretation, cannot be scaled, and requires intensive calibration effort for field application or minor modification [9–12].

Physics-based models do not require huge sets of data to build the model but require a thorough understanding of the working mechanisms and component interactions considering time and spatial elements. However, modeling the physics behind complicated system interactions such as electrochemical reactions and membrane mechanics of fuel cells is extremely complex, experimentally tedious, requires considerable numbers of sensors to validate, and is computationally intensive. This modeling approach also needs to incorporate numerical techniques to solve the physics involved, fundamental material properties, reactant interaction mechanisms, and computationally intensive fluid dynamics calculations all of which are a limitation when it comes to real-time applications [13–17].

Data-based models are quite in contrast with the physics-based models where the entire input-output relations of the fuel cell model are established primarily based on the experimental data using statistics, probability, and network architecture avoiding the complexities of physics-based models, intensive calibration, and also providing better accuracy in most cases with fast calculation as needed for real-time control applications, especially where the complex mechanisms cannot be modelled using governing equations [15,18–21].

Data-based models tend to have a quicker turnaround in the preliminary phase as multiple iterations are computed straight away without significantly affecting the development time, and by avoiding complex physics, which can help in real-time system identification, controls, and diagnostics. Considering the PEMFC commercialization timeline it is essential to develop data-based models to accelerate the commercialization of clean energy technologies by reaping the benefits of data acquisition, storage, and analytic resources available in today's digital age.

Machine learning applications to develop data-based models are relatively in an early stage for the fuel cell area. Wang et al. [22] had summarized the recent application of artificial neural networks (ANNs) and other machine algorithms used to predict the performance, material selection, and durability estimation of fuel cells and their components. A schematic of the feed-forward artificial neural network, where current, temperature, pressure, flow rate, and humidity are used as feature vectors (inputs) to predict the voltage output of the fuel cell is shown in Fig. 1.

Similarly, Kheirandish had used Support Vector Machine Regression (SVR) to model a PEMFC and successfully predicted the voltage, power, and efficiency aspects of the system [23]. The results and boundaries of the SVR algorithm are shown in Fig. 2 which illustrates the correlation between ANN and SVR on the same dataset where both the algorithm's prediction values are in agreement with each other.

Mehrpooya had developed a hybrid neural network method to model a micro PEMFC which consists of a Group Method of Data Handling

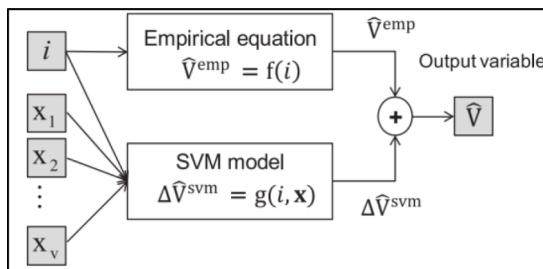


Fig. 3. Fuel cell hybrid model consisting of support vector machine and empirical equation [25].

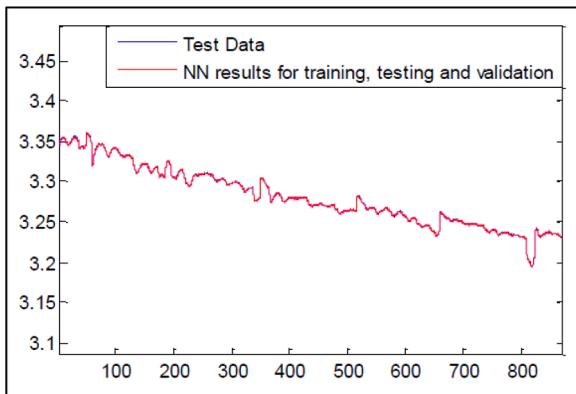


Fig. 4. Autoregressive neural network result stack voltage (V) vs time (hours) [29].

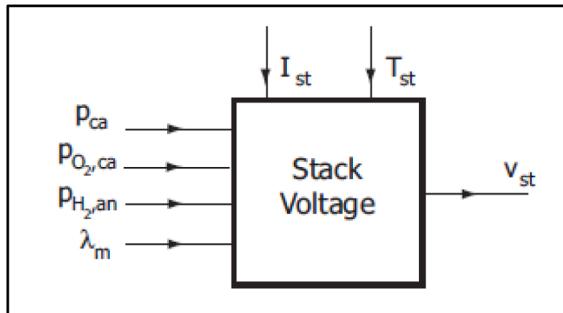


Fig. 5. Schematic of physics-based semi-empirical voltage module [30].

**Table 1**

Input and output parameters involved in the generation of Dataset 1.

Inputs (Feature vectors)	Output
Current	Fuel Cell Voltage
Temperature	
Cathode Pressure	
Oxygen (O <sub>2</sub> ) Partial Pressure	
Hydrogen (H <sub>2</sub> ) Partial Pressure	
Membrane Hydration (Lambda)	

(GMDH) neural network and empirical equations incorporated with the help of a genetic algorithm where humidity, temperature, oxygen flow rate, hydrogen flow rate, current density are considered as input feature vectors [24]. Han has used a similar hybrid approach with the implementation of SVR to account for various operating conditions in combination with a voltage empirical equation to predict the PEMFC polarization curve [25]. A schematic representation of the hybrid architecture that is a combination of SVM and empirical equation for the

fuel cell is shown in Fig. 3.

Han also has conducted a comparative study of ANN and SVR to predict the performance of PEMFC operation and estimated the effects of varying reactant pressure, and cell temperature across the range of operating conditions [26]. Deng has used a state-space system in combination with a recurrent neural network (RNN) to derive a new control-oriented data-driven model for PEMFCs and predicted the stack temperature, voltage, power, and other parameters with good accuracy [27]. Khajeh developed a hybrid model by combining the physics-based catalyst layer model with neural networks where ionomer film covering the agglomerate, agglomerate radius, platinum and carbon loading, membrane content, gas diffusion layer (GDL) penetration content, and catalyst layer (CL) thickness are used as input feature vectors to predict the activation overpotential [28]. Similarly, the durability aspects of PEMFC such as state of health and voltage over time can be predicted by an autoregressive neural network as shown by Mao and Jackson using 1200 h of PEMFC operational data [29]. The accuracy of the voltage predicted in comparison with the test data over the fuel cell life can be observed in Fig. 4.

Machine learning algorithms can be used to develop detailed data-based models for various applications but, almost all the models rely on huge sets of initial experimental data which can be time-consuming and expensive to explore the aspect of implementing new algorithms. However, this problem can be solved if large sets of data can be generated by physics-based models which are validated by a relatively small number of experiments covering a range of operating conditions. This can potentially aid in bypassing the tedious experimentation altogether or at least during the initial algorithm development phase serving as test data for performance demonstration. Most of the existing neural network models for PEMFC are developed using activation functions that are exponential in nature (logsig, tansig) predominantly in a MATLAB environment. These exponential functions are computationally intensive to train and process when compared to a non-exponential activation function like ReLU [30], especially when modeling for real-time control and diagnostic purposes. The objective of the present study is therefore to investigate such data-based modeling techniques to build machine learning models for control, diagnostic and prognostic purposes based on the data extracted through the simulation of physics-based semiempirical and CFD models, which are validated, reducing the need for tedious experimentation. As most of the data-based models for fuel cells in the literature are developed by using SVR and ANN algorithms, a similar approach is implemented by developing customized models in python environment using ReLU activation function, where advanced modeling techniques such as parameter normalization, dropout technique (used on neurons with ReLu activation function) are incorporated for improving the accuracy and computational efficiency.

## 2. Model development

### 2.1. Data acquisition

The necessary data for data-based modeling is gathered from actual PEMFC testing or through simulation of physics-based models that have been validated under a wide range of operating conditions that are also representative of real-world behaviors. Here in this case the data is acquired from physics-based models that have been validated using experimental data available in the literature. Dataset 1 is generated by simulating a semi-empirical physics-based model across different operating conditions in a MATLAB environment [31] where varying current, temperature, membrane hydration (lambda), and reactant pressures are used to predict the voltage output of the fuel cell, a schematic of the inputs and output variables of the voltage module is shown in Fig. 5.

Semi-empirical equations that are used to reconstruct the model are listed in the Appendix. As the model has been derived from the physics-based formulas the input feature vectors to the data-based model remain

**Table 2**

Input and output parameters involved in the generation of Dataset 2.

Inputs (Feature vectors)	Output
Current	Fuel Cell Voltage
Temperature	Membrane Resistance
Relative Humidity	Membrane Hydration (Lambda)

the same. Another set of data (from here on referred to as Dataset 2) is generated with the help of a 1-D physics-based CFD model [32] that had been experimentally validated. Here varying current, temperature, and humidity are used to predict the voltage output, membrane resistance, and membrane hydration level at constant reactant concentration and operating pressures. Both the feature vectors (input) and predicted variables (output) of Dataset 1 and Dataset 2 are listed in Tables 1 and 2, respectively.

The input and output attributes of the data-based models are chosen based on the governing physics of the system [33–35]. Since the objective of the present study is to develop a sufficiently accurate data-driven model for control, diagnostic and prognostic purposes, that is, to predict fuel cell performance and internal states for a given design and construction of fuel cells, the six model inputs selected for Dataset 1 to predict the voltage are the fuel cell operating conditions, which are also used as the input conditions for the physics-based models. The three model inputs selected for Dataset 2 to predict voltage, membrane resistance and membrane hydration are cell output current, the temperature of the stack and relative humidity at the cell inlet (assuming there is no change in reactant pressures), all of which have a direct correlation based on governing equations of CFD model and are necessary parameters to build a control or diagnostics model. Each dataset consists of a total of 1100 data points which are split in the ratio of 70:30 for training and validation, respectively. Both the data sets and the range of operational parameters are given in the appendix. As the datasets are generated from the validated physics-based model simulations no data points are excluded.

## 2.2. Artificial neural networks (ANNs)

Artificial Neural Network (ANN) is a combination of neurons which are the basic processing units that regulate the computational activity based on the designated activation function and associated inputs. This architecture is a clone of the biological neuron in terms of functionality and with the help of a backpropagation algorithm, it can solve and model regressions accurately.

From a mathematical perspective, the following operation can be

described as follows, internally all the inputs received by a neuron ( $x_1, x_2, x_3, \dots, x_r$ ) are multiplied by their corresponding weights ( $w_1, w_2, w_3, \dots, w_r$ ) and summed up together. Later this summation of products is assigned a threshold also referred to as bias ( $w_0$ ) where the result of this operation is passed to an activation function. A schematic representation of the neuron is shown in Fig. 6 and the mathematical representation of the neuron is shown in the following equation:

$$y(t) = \phi\left(\sum_{i=1}^r (w_i x_i - w_0)\right) \quad (1)$$

where  $y(t)$  is the output of the neuron and  $\phi$  is the activation function, the threshold  $w_0$  determines whether an aspect of a neuron generates an output or if it is inactive. This entire set of these weights and biases are considered hyperparameters which are calibrated depending on the activation function and associated error at each neuron. ANN here is used as a supervised machine learning tool where relevant inputs also referred to as feature vectors along with their corresponding outputs are provided to the model for training. During this process, relevant hyperparameters are calibrated by calculating the error at each neuron as shown below:

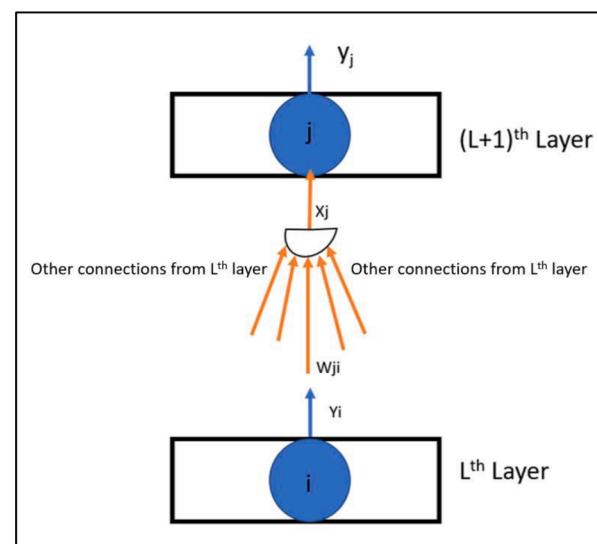


Fig. 7. Representation of backpropagation algorithm for the neural network [36].

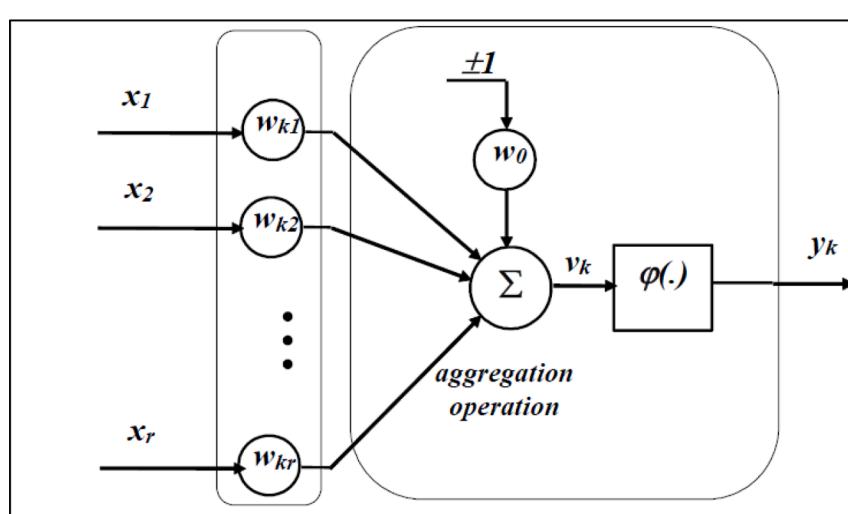
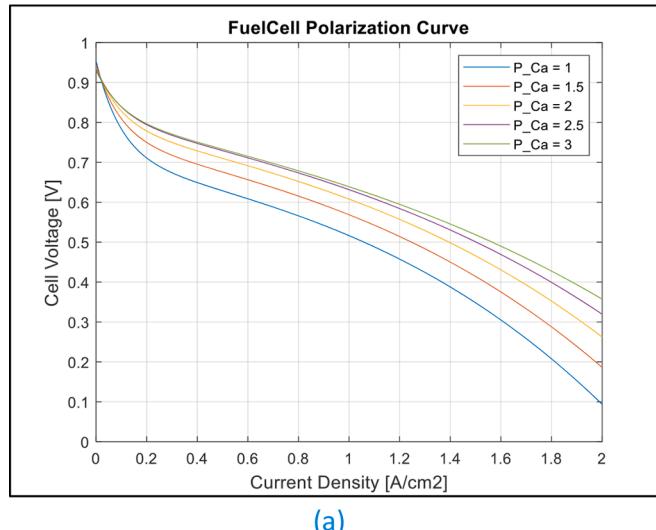
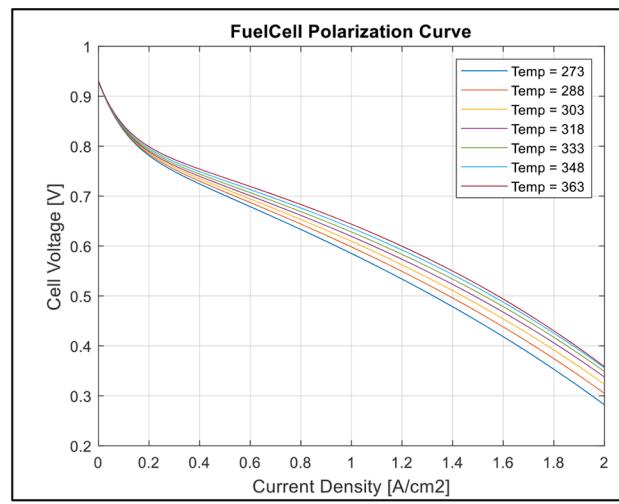


Fig. 6. Schematic of a Neuron [36].



(a)



(b)

**Fig. 8.** Fuel cell polarization curve exhibiting nonlinearity for (a) different pressures; (b) different temperatures.

$$E(n) = Q(n) - y(n) \quad (2)$$

where  $E(n)$  is the associated error at nth which is the difference between the actual (true) value of the nth element and the contemporary output of the nth neuron. The backpropagation algorithms gradually reduce this error at each training iteration referred to as epoch, by adapting the corresponding weights and biases at a defined learning rate for each neuron minimizing the total error, new weights are updated as shown below:

$$\Delta w = Lr \cdot E(n) \cdot y(n) \quad (3)$$

where  $\Delta w$  denotes the change in weight change,  $Lr$  is a positive constant known as learning rate,  $E(n)$  is the associated error,  $y(n)$  is the output of the associated neuron. The backpropagation algorithm aims to minimize the global error for a given set of data, weights for the total network are calculated as

$$\Delta w_{ji} = Lr \cdot E_{j,c} \cdot y_{i,c} \quad (4)$$

where  $E_{j,c}$  is the error signal at neuron  $j$  in layer  $L$ ; and  $y_{i,c}$  refers to the output of neuron  $i$  in layer  $L-1$  for the training case  $c$  as shown in Fig. 7.

Error signal,  $E_{j,c}$ , for each neuron is given as follows:

$$E_{j,c} = (Q_{j,c} - y_{j,c}) \cdot y_{j,c} \cdot (1 - y_{j,c}) \text{ for output neurons} \quad (5)$$

$$E_{j,c} = y_{j,c} \cdot (1 - y_{j,c}) \cdot \sum_k E_{k,c} w_{kj} \text{ for hidden layer neurons} \quad (6)$$

where  $Q$  refers to the corresponding true value,  $y$  is the corresponding neuron output,  $w$  is the associated weight;  $y_{j,c}$  the output of layer  $L$ ;  $y_{i,c}$  the output of the layer  $L-1$ ; and  $E_{k,c}$  the error of layer  $L+1$ . Detailed mathematical derivation and explanation for the backpropagation algorithm are available elsewhere [36].

### 2.2.1. Non-linearity of the data

In the case of a PEMFC, the model needs to represent both the linear and non-linear behavior of the fuel cell depending on the fuel cell's operational zone. Typically, both at low current and high current density conditions the fuel cell characteristics are nonlinear (exponential decay) as shown in Fig. 8.

### 2.2.2. Activation function

The activation functions incorporate the non-linearity to the neural networks depending on the sensitivity of the input feature vectors. Various nonlinear functions are considered to model this neural network and here is a quick mention of a few activation functions that are considered in this study.

**1 Sigmoid (Logistic function):** The Sigmoid function was introduced to ANNs in the 1990s to replace the step function and the output is centered around 0.5, a graphical representation of the sigmoid function and its derivative is shown in Fig. 9. Sigmoid function has a well-defined nonzero derivative helping stochastic gradient descent to improve at every epoch. Major issues are vanishing gradient, and computationally expensive with the exponential function involved [37].

**2 Hyperbolic Tangent (Tanh):** Tanh has characteristics like Sigmoid, and can work with Gradient Descent. Tanh function output is centered around 0, a graphical representation of the Tanh function and its derivative is shown in Fig. 9. Like sigmoid, it is also subjected to vanishing gradient and is computationally expensive, again with the same exponential function involved [37].

**3 Rectified Linear Unit (ReLU):** It is mostly implemented in deep learning and the ReLU function is composed of two linear pieces to account for non-linearities, a graphical representation of the ReLU function and its derivative is shown in Fig. 9. Unlike sigmoid and Tanh, the output of ReLU does not have a maximum value helping it to address the vanishing gradient issue and is computationally easy as there is no exponential operation [37]. After considering the initial results and literature it is decided to opt for the ReLU function to save computational expense as the activation function for the neural network considering the linear and nonlinear attributes.

### 2.2.3. Nodes & hidden layers

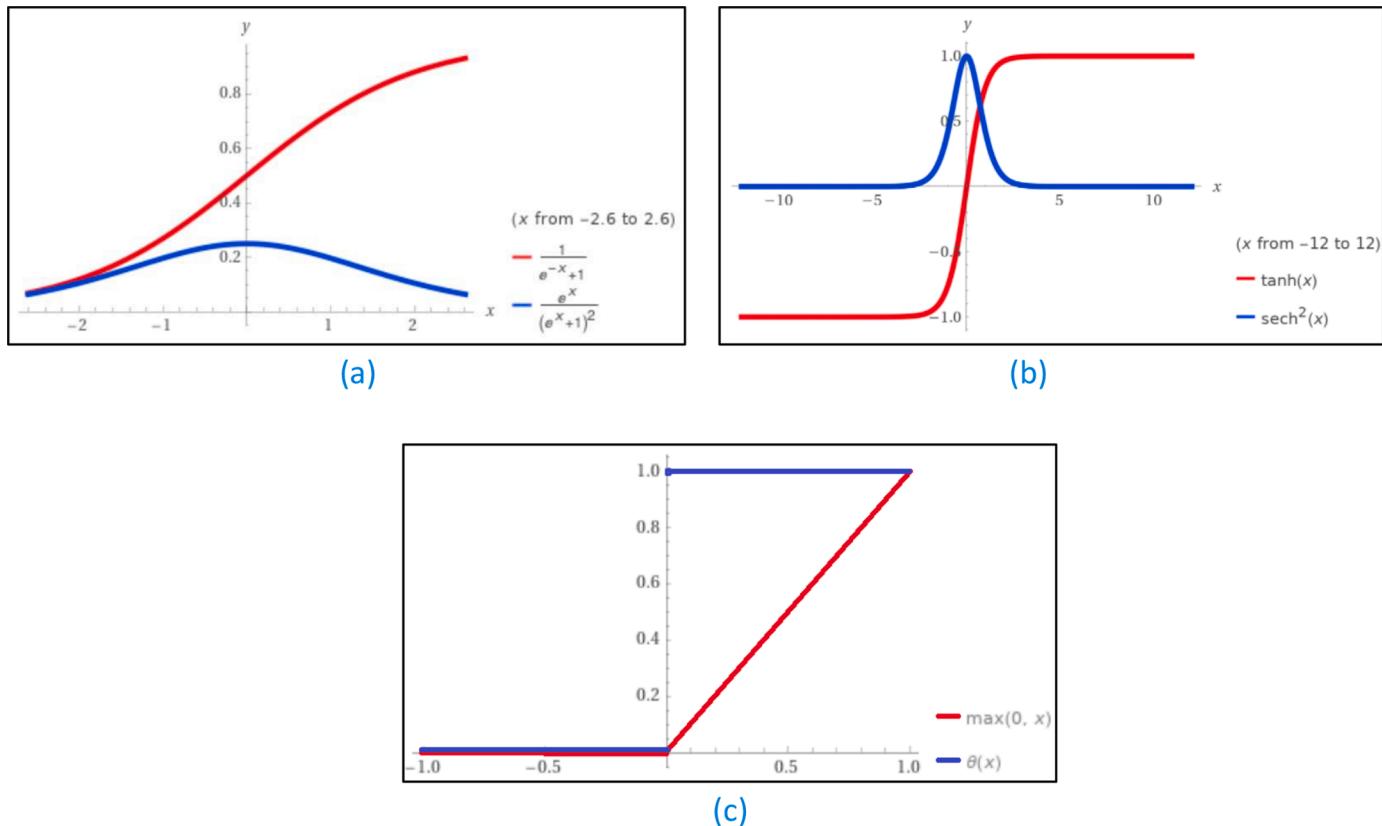
Based on the initial literature review and considering similar physical systems, it is concluded that a neural network with two hidden layers is a good starting point. Initially, 50 neurons per hidden layer are considered to model but after repeated simulations with various neuron combinations, it is reduced to 10 neurons per hidden layer.

### 2.2.4. Feature vectors

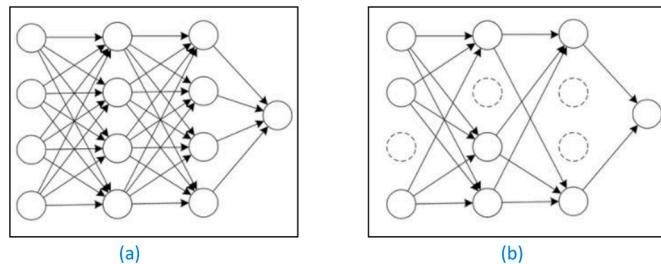
As this data is generated from the physics-based models, the inputs of these physics-based models can be used as input feature vectors for machine learning algorithms.

### 2.2.5. Loss function

Generally, Mean Square Error (MSE), Mean Absolute Error (MAE), and Huber are used to solve the regression in a neural network [38]. In



**Fig. 9.** (a) Sigmoid function (red-colored curve) and its derivative (blue-colored curve); (b) Tanh function (red-colored curve) and its derivative (blue-colored curve); (c) Rectified linear unit function (red-colored curve) and its derivative (blue-colored curve).



**Fig. 10.** Neuron connections: (a) without dropout; (b) with dropout.

this case, both MAE and MSE are considered. The mathematical representation of MAE and MSE is given below:

$$\text{MAE} = \frac{1}{n} \sum_{j=1}^n |y_j - \hat{y}_j| \quad (7)$$

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (8)$$

where  $n$  = number of elements,  $y_j$  is the actual (true value), and  $\hat{y}_j$  is the predicted (output) value.

**Training and Calibration:** Parameters like learning rates, loss function, number of epochs, and correction factors are calibrated to prevent any overtraining or undertraining of the models and are validated with the available testing data. All the iterations are ranked by quantifying the final error with the help of Coefficient of Determination ( $R^2$ ), and Root Mean Square Error (RMSE), as defined below, in addition to MAE and MSE,

$$R^2 = 1 - \frac{\text{sum squared regression (SSR)}}{\text{the total sum of squares (SST)}} \quad (9)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^n \|y(i) - \hat{y}(i)\|^2}{n}} \quad (10)$$

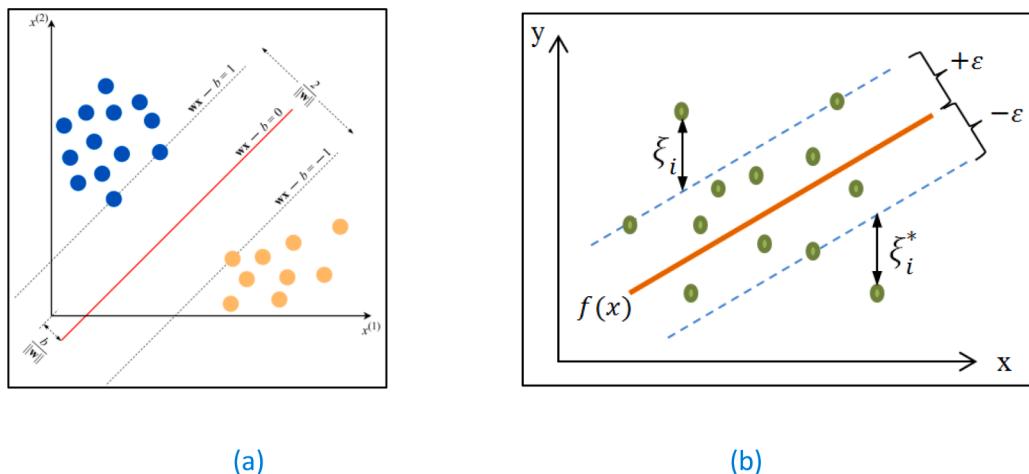
where  $n$  is the number of data points,  $y(i)$  is the  $i$  th measurement value, and  $\hat{y}(i)$  is its corresponding prediction value.

#### 2.2.6. Dropout probability

To prevent the aspect of overfitting during the training process of the neural network, certain neuron connections within the network are randomly disconnected based on a specific probability value. This specific probability number is referred to as dropout probability and needs to be calibrated depending on the data, predicted parameters, and accuracy needs. An example of a feed-forward neural network with and without dropout is shown in Fig. 10.

#### 2.3. Support vector machine regressor (SVM)

Support Vector Machine (SVM) is one of the most used techniques to resolve classification problems. The SVM helps us to find a line/hyperplane (in multidimensional space) that separates two classes. A decision boundary serves as a demarcation line to split the classes from each other. Support Vector Machine Regressor (SVR) uses a similar concept to approximate mapping from multiple input feature vectors to output based on a training sample. Here the objective is to consider the points that are within the decision boundary line and tune the SVR so that incorporates the maximum number of points, parallelly minimizing the  $\|w\| = (+\epsilon) + (-\epsilon)$ . The major tunable parameters are epsilon  $\epsilon$  and penalty factor  $C$ . Epsilon  $\epsilon$  in the SVR model refers to the epsilon tube within which no penalty is imposed in the loss function for data points



**Fig. 11.** (a) Classification of Support Vector Machine (SVM) [39]; (b) SVM regression [40].

**Table 3**

Initial ANN parameters considered for Dataset 1.

Inputs feature vectors	6
Output parameters	1
Hidden layers	2
Nodes / Neurons	10
Activation function	ReLU-1,2 layer Linear - Final
Loss function	'MAE' / 'MSE'
Learning Rate (Lr)	0.055
Iterations / Epochs	100
Momentum / correction	0.9
Dropout probability	0

predicted within a distance  $\xi$  from the actual value. A sample data set along with the corresponding SVM and SVR implementation is shown in Fig. 11.

Only data points lying outside the epsilon tube contribute to the error. The objective of the SVR is to minimize the following function

$$\text{Minimize } \frac{1}{2} \|w\|^2 + \sum_{i=1}^m (\xi_i + \xi_i^*) \quad (11)$$

Complying with the following constraints

$$y_i - (w \cdot x_i) - b \leq \varepsilon + \xi_i \quad (12)$$

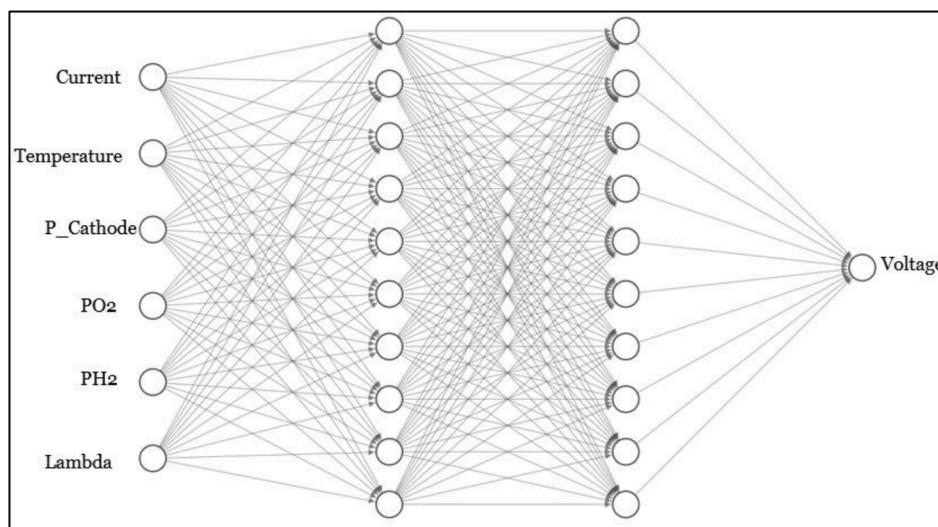
$$(w \cdot x_i) + b - y_i \leq \varepsilon + \xi_i^* \quad (13)$$

$$\xi_i, \xi_i^* \geq 0, i = 1, \dots, m \quad (14)$$

where  $x_i$  is the input feature vector,  $m$  is the total number of elements and  $w$ ,  $b$  and  $\xi$  are the pseudo hyperparameters, where  $w$  is the weight,  $b$  is the bias and  $\xi$  is the distance from the boundary determined by the SVR based on the data and epsilon ( $\varepsilon$ ) and  $C$ , which are the tunable parameters.

**Table 4**  
Initial ANN parameters considered for Dataset 2.

Inputs feature vectors	3
Output parameters	3
Hidden layers	2
Nodes	10
Activation function	ReLU-1,2 layer Linear - Final
Loss function	'MAE' / 'MSE'
Learning Rate (Lr)	0.025
Iterations/EPOCHS	100
Momentum/correction	0.9
Dropout probability	0



**Fig. 12.** Initial ANN design for Dataset 1.

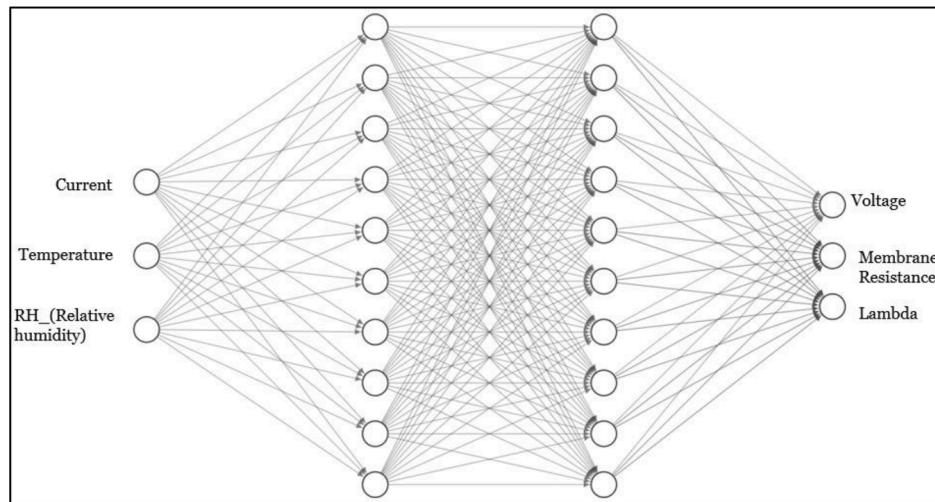


Fig. 13. Initial ANN design for Dataset 2.

**Table 5**  
Initial SVR parameters used for Dataset 1.

Inputs feature vectors	6
Output parameters	1
Epsilon	0.025
C	1.0
Training data points	770
Validation data points	330

#### 2.4. Data conditioning / preprocessing

As all the input feature vectors and output vectors differ from each other in scale, nonlinearity, and sensitivity to the model, normalization is necessary for both the input feature vectors and target output vectors before introducing them to the neural network and SVR algorithms. As all the feature vectors and output parameters are non-negative physical entities it is decided that normalization between 0 and 1 is appropriate. The normalization process is as shown below:

$$x \text{ normalized} = (x - x \text{ minimum}) / (x \text{ maximum} - x \text{ minimum}) \quad (15)$$

Both Dataset 1 and Dataset 2 are normalized between 0 and 1, and the results after normalization are shown in the appendix as well.

#### 2.5. ANN architecture and design parameters

Typically, a neural network can have any number of hidden layers, or neurons and a network with more than 2 or 3 hidden layers is considered a deep neural network whereas a network with only 1 or 2 hidden layers is considered a shallow neural network. A deep neural network requires larger computational capability and training time, which is typically used for image or complex time series regressions and object classification [35], whereas a shallow neural network relatively consumes less computational power and training time, primarily used for modeling regressions and pattern recognition [36]. Considering that the current fuel cell output parameter prediction can be regarded as a mathematical regression, a shallow feed-forward neural network is chosen. A preliminary investigation of the neural network performance with multiple hidden layers ranging from one to four each containing 50 neurons is evaluated and the network with two hidden layers provided the best value when considering the computational power required, network training time and resulting accuracy after multiple iterations. Hence it

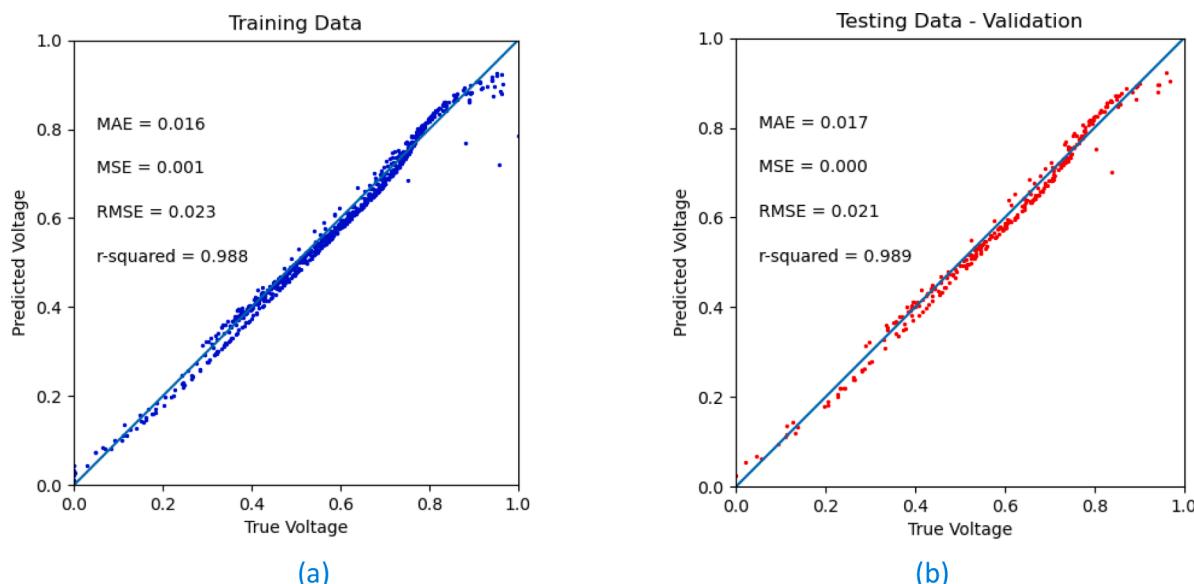
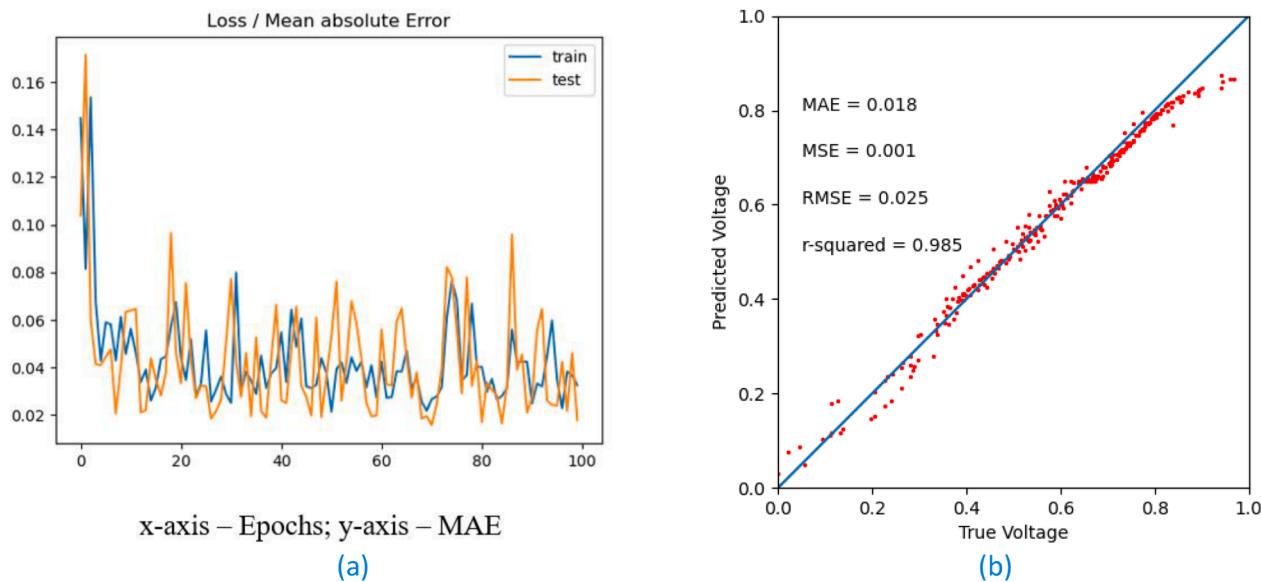
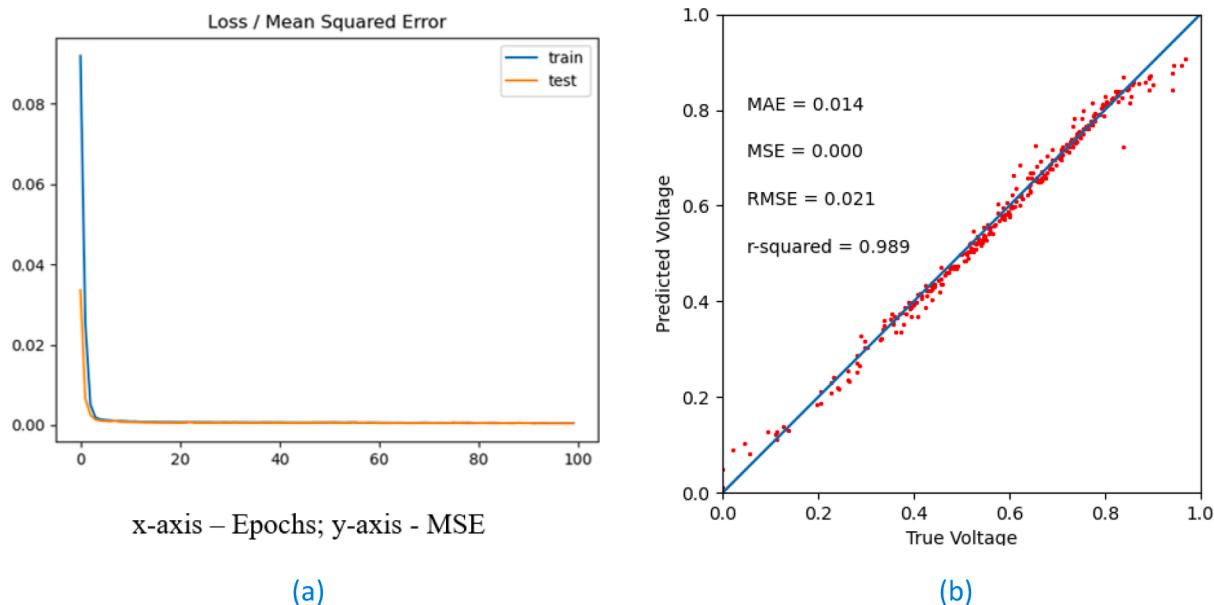


Fig. 14. SVR Results for Dataset 1: (a) Training; (b) Validation.



**Fig. 15.** Case 1 neural network result for Dataset 1 considering MAE as loss function and no dropout: (a) Loss function / MAE Vs Training Epochs; (b) Voltage prediction.



**Fig. 16.** Case 2 neural network result for Dataset 1 considering MSE as loss function and no dropout: (a) Loss function / MSE Vs Training Epochs; (b) Voltage prediction.

was decided to model the neural network with two hidden layers.

#### 2.5.1. ANN for dataset 1 (Semi-empirical)

Following are the initial structural parameters of the ANN algorithm that is applied to the semi-empirical data (Dataset 1) which consists of the following input features: PEMFC current, temperature, total cathode pressure, oxygen ( $O_2$ ), and hydrogen ( $H_2$ ) partial pressures, and membrane hydration (lambda). These are listed in Table 3 and the initial ANN design for Dataset 1 is shown in Fig. 12.

#### 2.5.2. ANN on dataset 2 (1D-CFD)

The initial structural parameters of the ANN that are applied to the 1-D CFD data (Dataset 2) consists of the following input feature vectors: PEMFC current, temperature, and relative humidity (RH). The output parameters are the predicted corresponding voltage, membrane

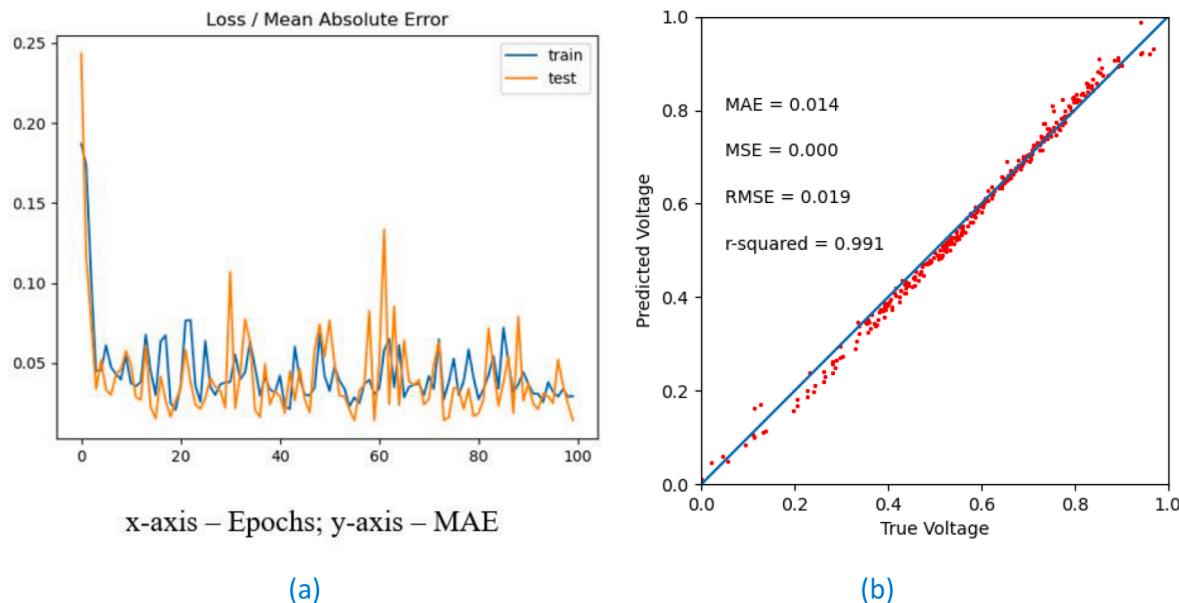
resistance, and membrane hydration (lambda). These are listed in Table 4 and the initial ANN design for Dataset 2 is shown in Fig. 13.

### 3. Results and discussion

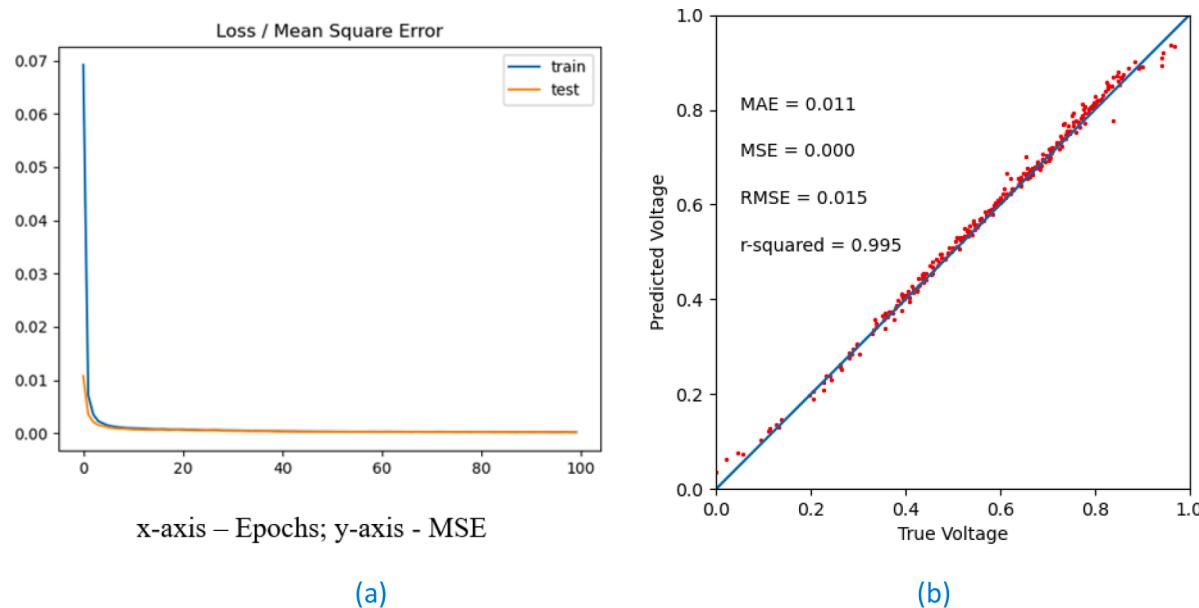
#### 3.1. SVR for dataset 1 and dataset 2

SVR technique is applied to Dataset 1 with similar feature vectors to the ANN, and is used to predict PEMFC voltage. It comprises the following design attributes and hyperparameters as listed in Table 5, the final results of the SVR for Dataset 1 during the training and validation are shown in Fig. 14.

From the above plots, it is noticed that  $R^2$  values are very close to 0.99 for both training and testing data. Thus we can conclude that the SVR technique is executed perfectly and can accurately predict the fuel



**Fig. 17.** Case 3 neural network result for Dataset 1 considering MAE as loss function with dropout: (a) Loss function / MAE Vs Training Epochs; (b) Voltage prediction.



**Fig. 18.** Case 4 neural network result for Dataset 1 considering MSE as loss function with dropout: (a) Loss function / MSE Vs Training Epochs; (b) Voltage prediction.

cell voltage baring a few extreme boundary conditions. Considering the relatively limited tunable parameters in SVR the results are considered fairly accurate in agreement with the available literature [23,26].

Whereas for Dataset 2 which is a multi-output regression problem, the SVR could not accurately converge and is not complex enough to model the associated nonlinearity. However independent SVR models can be developed and tuned to model the Dataset 2's each output vector.

### 3.2. ANN for dataset 1 (Semi-emp)

The initial results of the ANN for Dataset 1 comprising 6 inputs, 1 output, 2 hidden layers, and hyperparameters as listed in Table 3 with specific loss functions are presented below.

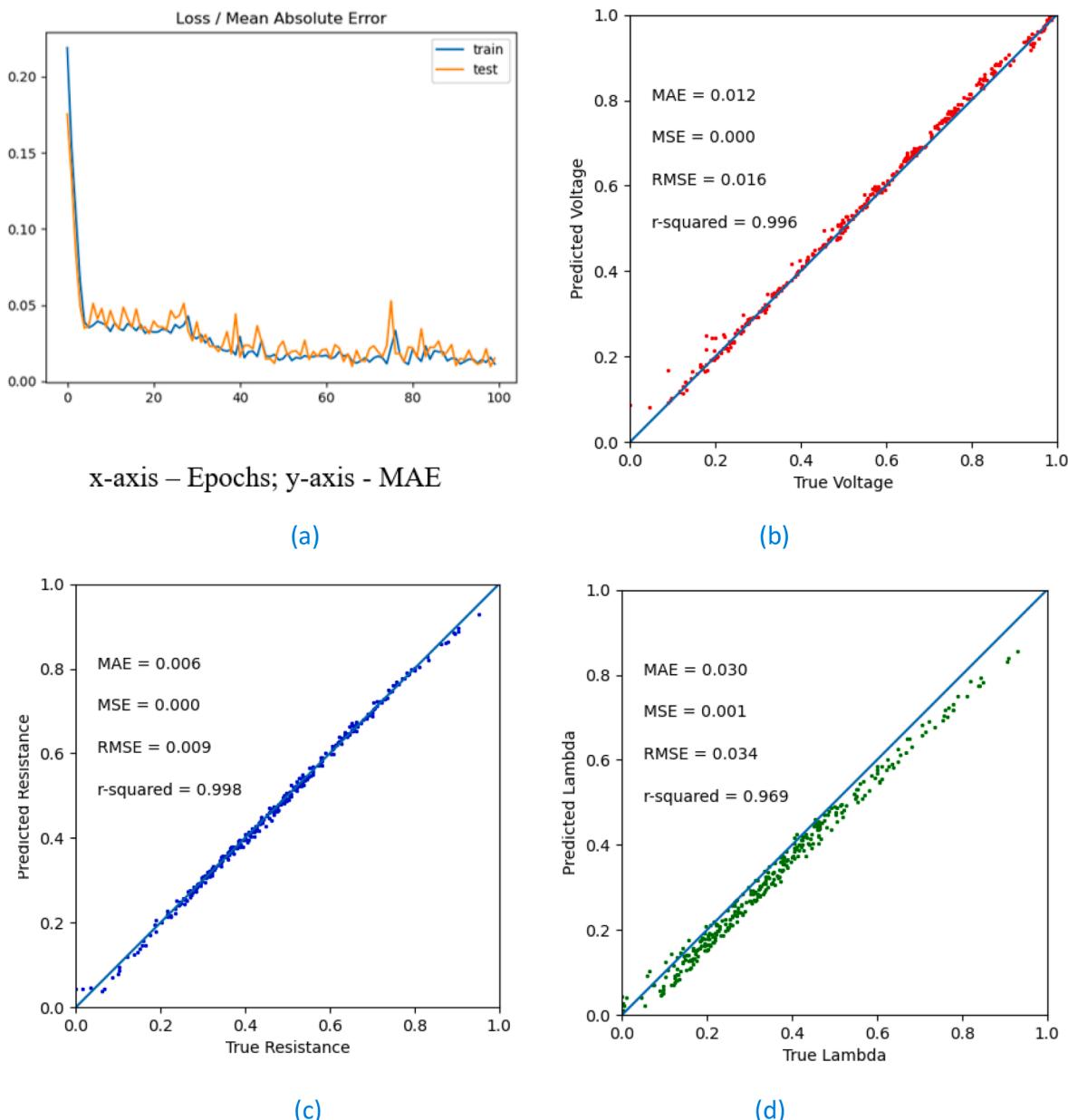
#### 3.2.1. Case 1: Mean absolute error (MAE) with no dropout

After investigation, the optimal ANN model hyperparameters are determined to be as follows: Loss function = MAE; Lr = 0.055; no dropout; and number of neurons per layer = 10. The initial results of the ANN fuel cell model developed using Dataset 1 with MAE as loss function and the above-mentioned hyperparameters are shown in Fig. 15.

As it is seen from Fig. 15, the ANN has good accuracy, and the structure of ANN (layers) is suitable for this regression but even after 100 epochs, the loss function is still unstable with MAE values fluctuating. As such, there is a need to investigate other loss functions such as MSE.

#### 3.2.2. Case 2: Mean squared error (MSE) with no dropout

After investigation, the optimal ANN model hyperparameters are



**Fig. 19.** Case 1 neural network result for Dataset 2 considering MAE as loss function with no dropout: (a) Loss function / MAE Vs Training Epochs; (b) Voltage prediction; (c) Membrane resistance prediction; (d) Membrane hydration (Lambda) prediction.

found as: Loss function = MSE,  $L_r = 0.055$ ; no dropout; number of neurons per layer = 10. The results of the ANN with MSE as the loss function and with the above-mentioned hyperparameters are shown in Fig. 16.

The MSE loss function reduces quickly with the iteration epochs, and without observable fluctuations; hence is considered stabilized and offers a good accuracy for the neural network with the  $R^2$  value of 0.989. However, it is slightly behind in terms of accuracy at extreme boundary conditions i.e., Voltage values below 0.15 and above 0.85 – these extreme cell voltage values are normally outside the range of practical operation of PEMFCs. But still, to improve the accuracy for these extreme voltage values, the number of neurons in each layer is increased and the dropout technique is incorporated for both the loss functions, and they are presented next.

### 3.2.3. Case 3: Mean absolute error (MAE) with dropout consideration

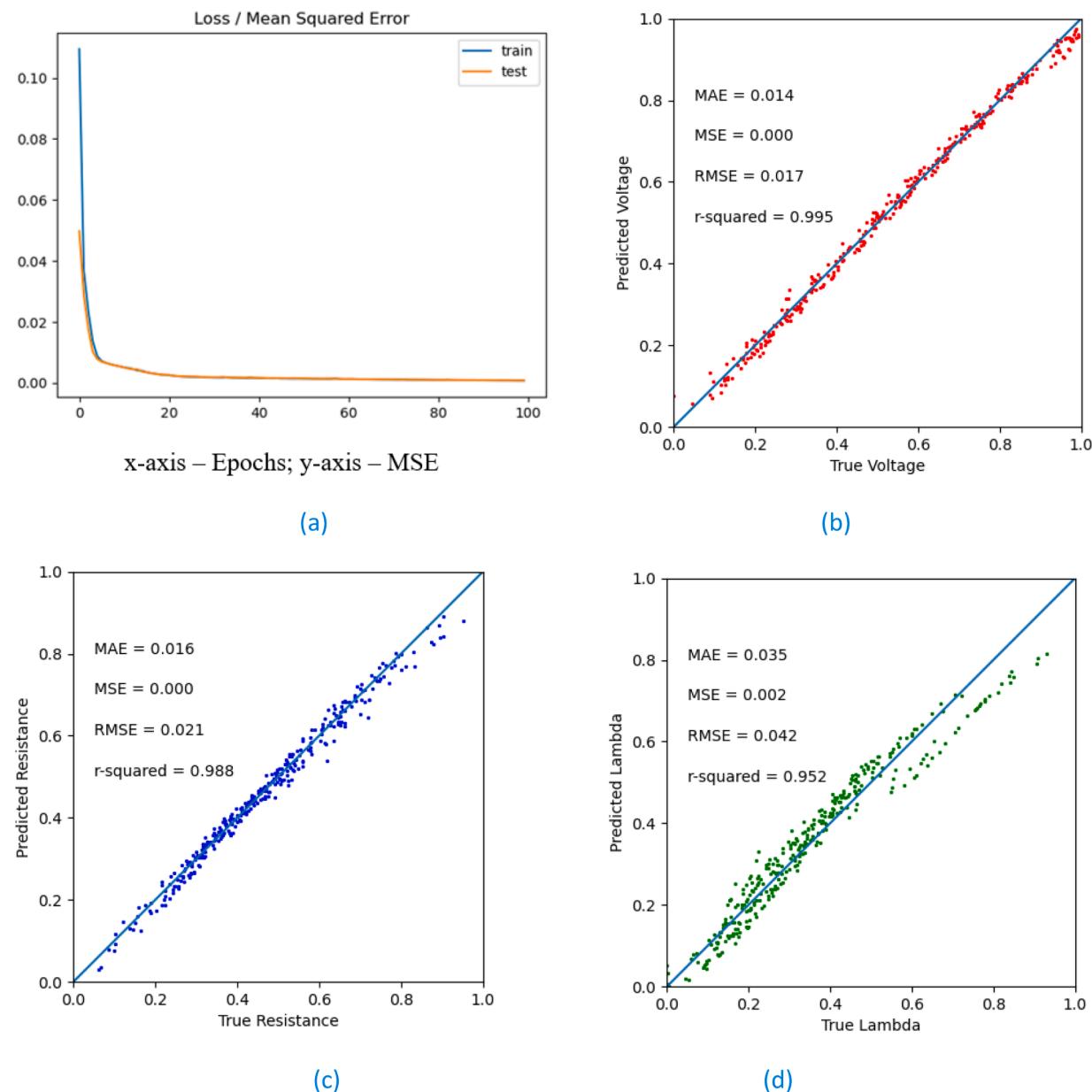
The optimal ANN model hyperparameters are determined through extensive analysis, they are: Loss function = MAE, Lr = 0.055; number of

neurons per layer = 20 in both hidden layers; dropout probability is 0.5 for both hidden layers. The corresponding results of the ANN with MAE as the loss function incorporating dropout in both hidden layers and with the above-mentioned hyperparameters are shown in Fig. 17.

Again it is observed that the MAE loss function is still unstable as shown in Fig. 15, but considering the encouraging results shown in Fig. 17 with the improved accuracy ( $R^2$  value is increased to 0.991), the loss function is replaced by MSE, and a similar dropout technique is introduced to the model by increasing the number of neurons and altering the drop probability, as shown next.

### 3.2.4. Case 4: Mean squared error (MSE) with dropout consideration

After considerable analysis, the optimal ANN model hyper-parameters are determined as: Loss function = MSE, Lr = 0.055; number of neurons per layer = 30 in both hidden layers; dropout probability = 0.5 for the 1st hidden layer and 0.25 for the 2nd hidden layer. The results of the ANN with MSE as the loss function incorporating varying dropout probability in both hidden layers and the above-mentioned



**Fig. 20.** Case 2 neural network result for Dataset 2 considering MSE as loss function with no dropout: (a) Loss function / MAE Vs Training Epochs; (b) Voltage prediction; (c) Membrane resistance prediction; (d) Membrane hydration (Lambda) prediction.

hyperparameters are shown in Fig. 18.

From Fig. 18, it is noted that the introduction of the dropout technique has resulted in a considerable improvement in the accuracy of the predicted voltage, with the  $R^2$  value of 0.995, especially considering the nonlinearity in the extreme boundary conditions (voltage less than 0.15 V and great than 0.85 V). Clearly, the use of MSE as the loss function and adoption of the dropout technique improves the accuracy of the results with fewer epoch levels (or iterations) needed to reach converged calculations.

### 3.3. ANN results for dataset 2 (1-D CFD)

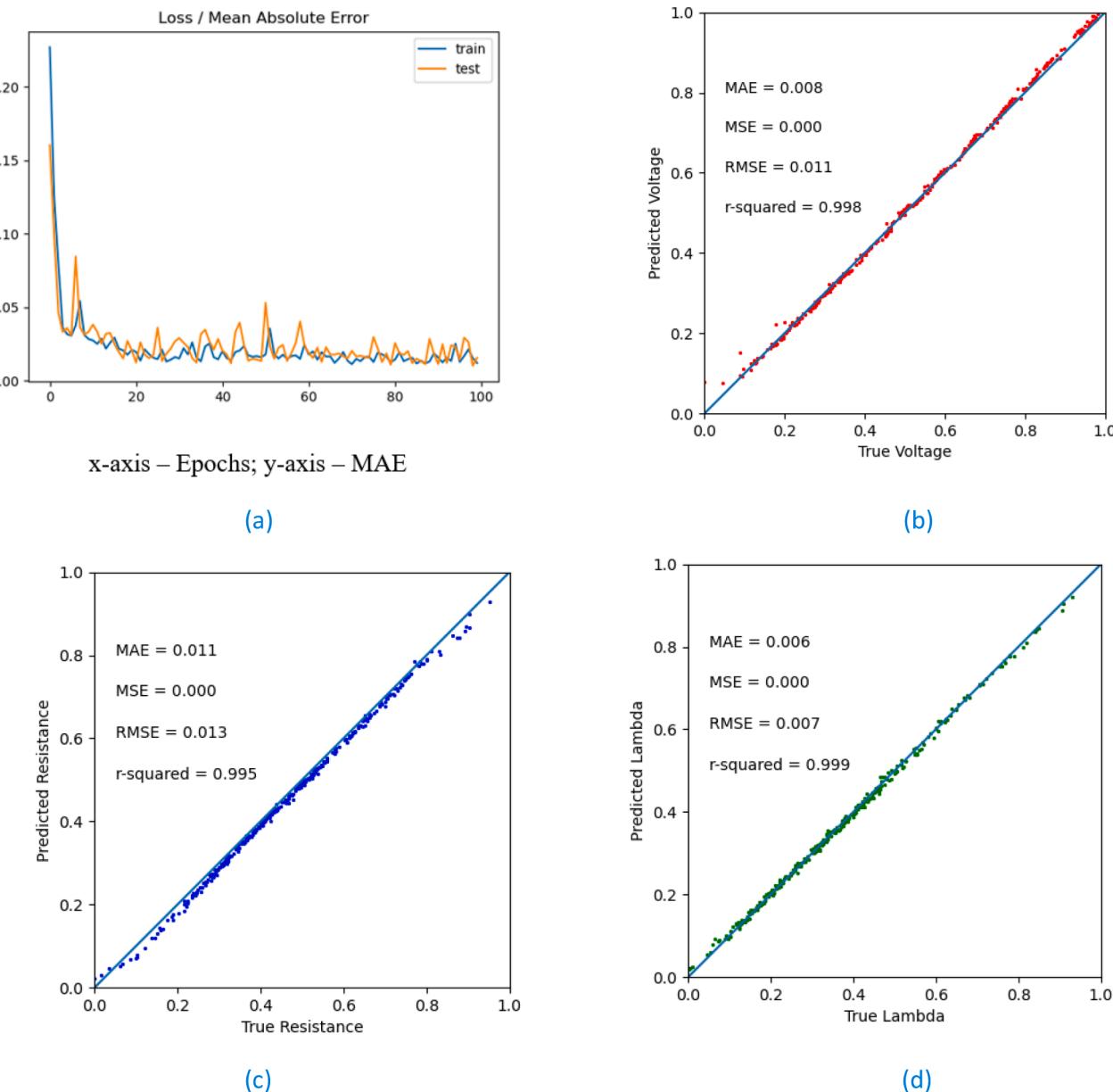
The results of the ANN for Dataset 2 comprising 3 inputs, 3 outputs, 2 hidden layers, and hyperparameters as listed in Table 4 along with specific loss functions are presented below.

#### 3.3.1. Case 1: Mean absolute error (MAE) with no dropout

The optimal ANN model hyperparameters are determined as,

following extensive investigation: Loss function = MAE,  $L_r = 0.025$ , no dropout, and number of neurons per layer = 10. The initial results of the ANN fuel cell model developed using Dataset 2 with MAE as the loss function and the above-mentioned hyperparameters are shown in Fig. 19.

Despite this being a multi-output regression problem, this neural network architecture has quite a good accuracy for all three parameters, this suggests the chosen design and hyperparameters are close to ideal. The loss function (MAE) output is also relatively stabilized, though fluctuations over the level of epochs are clearly present after 100 epochs. The accuracy in the predicted lambda ( $R^2$  value of 0.969, as compared with  $R^2$  value of 0.996 and 0.998 for the predicted cell voltage and resistance) can be improved further since a majority of the data points are persistently located beneath the correlation curve. To achieve this the loss function is replaced with MSE for further analysis, similar to the cases for Dataset 1.



**Fig. 21.** Case 3 neural network result for Dataset 2 considering MAE as loss function with dropout: (a) Loss function / MAE Vs Training Epochs; (b) Voltage prediction; (c) Membrane resistance prediction; (d) Membrane hydration (Lambda) prediction.

### 3.3.2. Case 2: Mean squared error (MSE) with no dropout

After considerable investigation, the optimal ANN model hyperparameters are determined to be: Loss function = MSE; Lr = 0.025; no dropout; and number of neurons per layer = 10. The results of the ANN with MSE as the loss function and the above-mentioned hyperparameters are shown in Fig. 20. That is, the hyperparameter values considered are exactly the same as in the previous case, except for the change in the loss function.

As it is observed from Fig. 20 the change in the loss function is kind of counterproductive and did not fully achieve the desired results even after stabilized error value. The  $R^2$  values for the predicted three parameters, 0.995, 0.988 and 0.952 for cell voltage, resistance and membrane hydration, respectively, are slightly smaller than for the previous case. After several iterations of hyperparameters resulting in no significant change, the dropout technique was again introduced by increasing the number of neurons for further investigation.

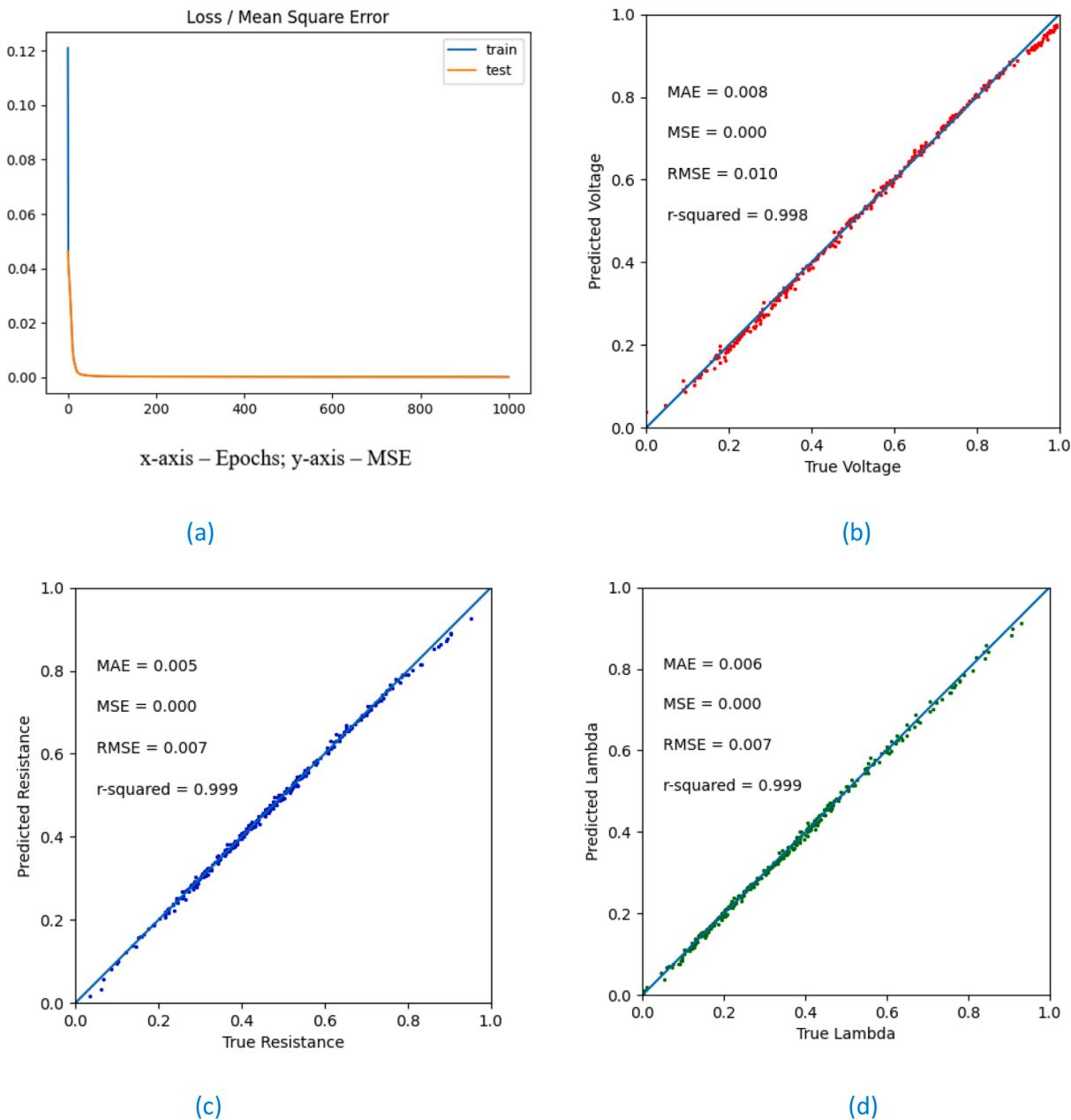
### 3.3.3. Case 3: Mean absolute error (MAE) with dropout consideration

After extensive investigation, the optimal ANN model hyperparameters are determined as follows: Loss function = MAE; Lr = 0.025; number of neurons per layer = 20 in both hidden layers; and dropout probability = 0.25 for both hidden layers. The results of the ANN with MAE as the loss function incorporating dropout probability in both hidden layers and the above-mentioned hyperparameters are shown in Fig. 21.

It can be observed from Fig. 21 that the introduction of the dropout technique on this ANN for Dataset 2 has also improved the accuracy of all the predicted parameters greatly ( $R^2$  values of 0.998, 0.995 and 0.999, respectively, for the predicted cell voltage, resistance and membrane hydration). Considering the minor fluctuations in the MAE error function even after 100 epochs the loss function is replaced with MSE for further analysis.

### 3.3.4. Case 4: Mean squared error (MSE) with dropout consideration

After extensive tuning, the epochs are increased to 1000 whereas the



**Fig. 22.** Case 4 neural network result for Dataset 2 considering MSE as loss function with dropout: (a) Loss function / MSE Vs Training Epochs; (b) Voltage prediction; (c) Membrane resistance prediction; (d) Membrane hydration (Lambda) prediction.

**Table 6**

Voltage prediction accuracy of SVR and ANN on physics-based semi-empirical data (Dataset 1).

Algorithm	Dropout	R <sup>2</sup> Value	RMSE	MAE
SVR	N/A	0.989	0.021	0.017
ANN-MAE	No	0.985	0.025	0.018
	Yes	0.991	0.019	0.014
ANN-MSE	No	0.989	0.021	0.014
	Yes	0.995	0.015	0.011

number of neurons per layer are reduced to 10 compared with the previous cases. This indicates that more time is needed for the training, but once it is trained, it is much easier to compute the various predictions from the trained model. The trained ANN model has the following optimal model hyperparameters: Loss function = MSE; Lr =

0.025; number of neurons per layer = 10 in both hidden layers; and dropout probability = 0.25 for both hidden layers. The results of the ANN model with MAE as the loss function incorporating dropout probability in both hidden layers and the above-mentioned hyperparameters are shown in Fig. 22.

From the above results, it can be concluded that changing the loss function to MSE along with the increasing the number of epochs to 1000 and implementation of the dropout technique has improved the multi-output regression performance and can accurately predict all the output parameters. It is clear that MSE is reduced quickly without fluctuations over the training epochs (iterations), and the predicted cell voltage, membrane resistance and membrane hydration have the best accuracy of all the models investigated so far, with the R<sup>2</sup> value of 0.998, 0.999 and 0.999, respectively, and the MAE less than 0.01.

It is evident that unlike SVR an ANN offers higher flexibility of multiple tunable parameters and techniques that can help to model multiple output regressions despite the non-linearity nature of the data

**Table 7**

ANN prediction accuracy (using dropout) on 1-D CFD data (Dataset 2) for all output vectors.

Algorithm	Dropout	Predicted parameter	R <sup>2</sup> Value	RMSE	MAE
ANN- MAE	No	Cell Voltage	0.996	0.016	0.012
	Yes		0.998	0.011	0.008
	No	Membrane Resistance	0.998	0.009	0.006
	Yes		0.995	0.013	0.011
	No	Membrane Hydration (Lambda)	0.969	0.034	0.030
	Yes		0.999	0.007	0.006
	No	Cell Voltage	0.995	0.017	0.014
	Yes		0.998	0.01	0.008
ANN-MSE	No	Membrane Resistance	0.988	0.021	0.016
	Yes		0.999	0.007	0.005
	No	Membrane Hydration (Lambda)	0.952	0.042	0.035
	Yes		0.999	0.007	0.006

and system attributes. The final prediction accuracy in terms of R<sup>2</sup>, root mean square error (RMSE) and mean absolute error (MAE) for ANN and SVR algorithms after tuning for both the datasets to predict cell voltage, membrane resistance and hydration (lambda) is tabulated in Tables 6 and 7 for comparison.

#### 4. Conclusions

In this study, artificial neural network (ANN) and support vector machine regressor (SVR) have been used as the machine learning methods to develop data-based models for the performance attributes and internal states of proton exchange membrane fuel cell (PEMFC). PEMFC operating conditions such as cell current, temperature, reactant pressures, and humidity are used as input feature parameters, while the output parameters include the predicted cell voltage, membrane resistance, and membrane hydration level for various operating conditions. The accuracy of the data-based models developed are evaluated, especially under extreme conditions. The data used for the present modeling study are acquired, respectively, from i) a physics-based semi-empirical model (Dataset 1), and ii) a dimensionally-reduced 1-D Computational Fluid Dynamics (CFD) model that has been validated with experimental results (Dataset 2). A total of 1100 data points for each dataset have been generated that cover the entire range of PEMFC operations. 70% of the data points are used for training, while the remaining 30% of data points are used for testing/validation. It is demonstrated that ANN clearly shows an advantage in comparison with SVR, especially on a multivariable output regression; and SVR is advantageous to model simple regressions as it greatly reduces the level of computation (no need for multiple epochs) without sacrificing accuracy. It is further shown that the machine learning techniques incorporating the dropout technique can provide very accurate predictions with R<sup>2</sup> ≥ 0.99 for all the predicted variables, demonstrating the ability to build accurate data-based models solely on data from validated physics-based models, reducing the dependency on extensive experimentation. The present study indicates that additional features vectors such as electrochemical impedance, interfacial resistances between layers, poisoning effect, aging, and catalyst leaching may be considered for better diagnostics and prognostics of PEMFC operation, including water flooding, reactant starvation, membrane mechanics, state of health and state of useful life remaining.

#### 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 work reported in this paper.

#### Acknowledgement

We would like to thank Dr. William Melek (University of Waterloo) for his guidance and support during this project. This work received financial support from Canadian Urban Transit Research and Innovation Consortium (CUTRIC) via Project Number 160028, Natural Sciences and Engineering Research Council of Canada (NSERC) via a Discovery Grant

#### Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.egyai.2022.100183.

#### References

- [1] van Amerongen J, Breedveld P. Modelling of physical systems for the design and control of mechatronic systems. *Annu Rev Control* 2003;27(1):87–117. [https://doi.org/10.1016/S1367-5788\(03\)00010-5](https://doi.org/10.1016/S1367-5788(03)00010-5).
- [2] Segura F, Andújar JM. Step by step development of a real fuel cell system. Design, implementation, control and monitoring. *Int J Hydrogen Energy* 2015;40(15):5496–508. <https://doi.org/10.1016/j.ijhydene.2015.01.178>.
- [3] Xue X, Tang J, Smirnova A, England R, Sammes N. System level lumped-parameter dynamic modeling of PEM fuel cell. *J Power Sources* 2004;133(2):188–204. <https://doi.org/10.1016/j.jpowsour.2003.12.064>.
- [4] Zhao Z, Wang T, Li M, Wang H, Wang Y. Optimization of fuzzy control energy management strategy for fuel cell vehicle power system using a multi-island genetic algorithm. *Energy Sci Eng* 2021;9(4):548–64. <https://doi.org/10.1002/ese3.835>.
- [5] Zhang H, Li X, Liu X, Yan J. Enhancing fuel cell durability for fuel cell plug-in hybrid electric vehicles through strategic power management. *Appl Energy* 2019;241:483–90. <https://doi.org/10.1016/j.apenergy.2019.02.040>. January.
- [6] Forrai A, Funato H, Yanagita Y, Kato Y. Fuel-cell parameter estimation and diagnostics. *IEEE Trans Energy Conversion* 2005;20(3):668–75. <https://doi.org/10.1109/TEC.2005.845516>.
- [7] Zhao J, Li X. A review of polymer electrolyte membrane fuel cell durability for vehicular applications: degradation modes and experimental techniques. *Energy Conversion and Manag* 2019;199:112022. <https://doi.org/10.1016/j.enconman.2019.112022>. September 2019.
- [8] Zamel N, Li X. Effective transport properties for polymer electrolyte membrane fuel cells - With a focus on the gas diffusion layer. *Prog Energy Combust Sci* 2013;39(1):111–46. <https://doi.org/10.1016/j.pecs.2012.07.002>.
- [9] Rajabzadeh M, Bathaei SMT, Aliakbar Golkar M. Dynamic modeling and nonlinear control of fuel cell vehicles with different hybrid power sources. *Int J Hydrogen Energy* 2016;41(4):3185–98. <https://doi.org/10.1016/j.ijhydene.2015.12.046>.
- [10] Belmokhtar K, Hammoudi M, Doumbia ML, Agbossou K. Modelling and fuel flow dynamic control of proton exchange membrane fuel cell. In: International Conference on Power Engineering, Energy and Electrical Drives; 2013. p. 415–20. <https://doi.org/10.1109/PowerEng.2013.6635643>. May.
- [11] Page SC, Anbukey AH, Krumdieck SP, Brouwer J. Test method and equivalent circuit modeling of a PEM fuel cell in a passive state. *IEEE Trans Energy Conversion* 2007;22(3):764–73. <https://doi.org/10.1109/TEC.2007.895857>.
- [12] Lan T, Strunz K. Modeling of multi-physics transients in PEM fuel cells using equivalent circuits for consistent representation of electric, pneumatic, and thermal quantities. *Int J Electr Power and Energy Syst* 2020;119:105803. <https://doi.org/10.1016/j.ijepes.2019.105803>. December 2019.
- [13] Khan MJ, Iqbal MT. Dynamic modelling and simulation of a fuel cell generator. *Fuel Cells* 2005;5(1):97–104. <https://doi.org/10.1002/fuce.200400054>.
- [14] Daud WRW, Rosli RE, Majlan EH, Hamid SAA, Mohamed R, Husaini T. PEM fuel cell system control: a review. *Renew Energy* 2017;113:620–38. <https://doi.org/10.1016/j.renene.2017.06.027>.
- [15] Zhao J, Li X, Shum C, McPhee J. A Review of physics-based and data-driven models for real-time control of polymer electrolyte membrane fuel cells. *Energy and AI* 2021;6:100114. <https://doi.org/10.1016/j.egyai.2021.100114>.
- [16] Bao C, Ouyang M, Yi B. Modeling and control of air stream and hydrogen flow with recirculation in a PEM fuel cell system-I. Control-oriented modeling. *Int J Hydrogen Energy* 2006;31(13):1879–96. <https://doi.org/10.1016/j.ijhydene.2006.02.031>.
- [17] Pukrushpan BJT, Stefanopoulou AG, Peng H. Avoid fuel cell oxygen starvation with air flow controllers. *IEEE Control Syst Magazine* 2004;30–46. April.
- [18] Abbaspour A, Khalilnejad A, Chen Z. Robust adaptive neural network control for PEM fuel cell. *Int J Hydrogen Energy* 2016;41(44):20385–95. <https://doi.org/10.1016/j.ijhydene.2016.09.075>.
- [19] Jia J, Liang J, Shi Y, Wen J, Pang X, Zeng J. SOH and RUL prediction of lithium-ion batteries based on Gaussian process regression with indirect health indicators. *Energies (Basel)* 2020;13(2). <https://doi.org/10.3390/en13020375>.
- [20] Wang B, Xie B, Xuan J, Jiao K. AI-based optimization of PEM fuel cell catalyst layers for maximum power density via data-driven surrogate modeling. *Energy Conversion and Manag* 2020;205. <https://doi.org/10.1016/j.enconman.2019.112460>. January.
- [21] Li S, He H, Su C, Zhao P. Data driven battery modeling and management method with aging phenomenon considered. *Appl Energy* 2020;275. <https://doi.org/10.1016/j.apenergy.2020.115340>. May.

- [22] Wang Y, Seo B, Wang B, Zamel N, Jiao K, Adroher XC. Fundamentals, materials, and machine learning of polymer electrolyte membrane fuel cell technology. *Energy and AI* 2020;1:100014. <https://doi.org/10.1016/j.egyai.2020.100014>.
- [23] Kheirandish A, Shafibady N, Dahari M, Kazemi MS, Isa D. Modeling of commercial proton exchange membrane fuel cell using support vector machine. *Int J Hydrogen Energy* 2016;41(26):11351–8. <https://doi.org/10.1016/j.ijhydene.2016.04.043>.
- [24] Mehrpooya M, Ghorbani B, Jafari B, Aghbashlo M, Pouriman M. Modeling of a single cell micro proton exchange membrane fuel cell by a new hybrid neural network method. *Therm Sci Eng Progress* 2018;7:8–19. <https://doi.org/10.1016/j.tsep.2018.04.012>. October 2017.
- [25] Han IS, Chung CB. A hybrid model combining a support vector machine with an empirical equation for predicting polarization curves of PEM fuel cells. *Int J Hydrogen Energy* 2017;42(10):7023–8. <https://doi.org/10.1016/j.ijhydene.2017.01.131>.
- [26] Han IS, Chung CB. Performance prediction and analysis of a PEM fuel cell operating on pure oxygen using data-driven models: a comparison of artificial neural network and support vector machine. *Int J Hydrogen Energy* 2016;41(24):10202–11. <https://doi.org/10.1016/j.ijhydene.2016.04.247>.
- [27] Deng Z, Chen Q, Zhang L, Zong Y, Zhou K, Fu Z. Control oriented data driven linear parameter varying model for proton exchange membrane fuel cell systems. *Appl Energy* 2020;277. <https://doi.org/10.1016/j.apenergy.2020.115540>. July.
- [28] Khajeh-Hosseini-Dalasm N, Ahadian S, Fushinobu K, Okazaki K, Kawazoe Y. Prediction and analysis of the cathode catalyst layer performance of proton exchange membrane fuel cells using artificial neural network and statistical methods. *J Power Sources* 2011;196(8):3750–6. <https://doi.org/10.1016/j.jpowsour.2010.12.061>.
- [29] Mao L, Jackson L. Comparative study on prediction of fuel cell performance using machine learning approaches. *Lecture Notes in Eng Comput Sci* 2016;1:52–7.
- [30] Chai E, Yu W, Cui T, Ren J, Ding S. An efficient asymmetric nonlinear activation function for deep neural networks. *Symmetry (Basel)* 2022;14(5):1027. <https://doi.org/10.3390/sym14051027>. May.
- [31] J.T. Pukrushpan and Jay Tawee Pukrushpan, "Modeling and control of fuel cell systems and fuel processors," 2003. [Online]. Available: [http://www-personal.umich.edu/~annastef/FuelCellPdf/pukrushpan\\_thesis.pdf](http://www-personal.umich.edu/~annastef/FuelCellPdf/pukrushpan_thesis.pdf).
- [32] J. Zhao, "Catalyst layers in polymer electrolyte membrane fuel cells: formation, characterization and performance," 2019, [Online]. Available: <https://uwaterloo.ca/handle/10012/14425>.
- [33] Tao H, Hua S, Wu K, Wu S, Liu Q, Jiao K. Two-Dimensional simulation of purge processes for dead-ended H<sub>2</sub>/O<sub>2</sub> proton exchange membrane fuel cell. *Int J Green Energy* 2022. <https://doi.org/10.1080/15435075.2022.2040508>.
- [34] Yang XG, Wang Y, Wang CY. Ultra-high fuel utilization in polymer electrolyte fuel cells part I: an experimental study. *Int J Green Energy* 2022;19(2):159–65. <https://doi.org/10.1080/15435075.2021.1941041>.
- [35] Pan M, Hu P, Gao R, Liang K. Multistep prediction of remaining useful life of proton exchange membrane fuel cell based on temporal convolutional network. *Int J Green Energy* 2022;1–15. <https://doi.org/10.1080/15435075.2022.2050377>. Mar.
- [36] Melek W. ME780: computational intelligence, lecture notes. University of Waterloo; 2018.
- [37] B. Chen, "7 popular activation functions you should know in deep learning and how to use them with keras and...," Medium, 04-Jan- 2021. [Online]. Available: <https://towardsdatascience.com/7-popular-activation-functions-you-should-know-in-deep-learning-and-how-to-use-them-with-keras-and-27b4d838dfe6>. [Accessed: 02-Mar-2022].
- [38] Mahendru K. Loss function: loss function in machine learning. Analytics Vidhya 2020. 23-Apr-[Online]. Available: <https://www.analyticsvidhya.com/blog/2019/08/detailed-guide-7-loss-functions-machine-learning-python-code/> [Accessed: 25-Feb-2022].
- [39] Burkov, Andriy. The Hundred-Page Machine Learning Book. S. l., 2019.
- [40] Chanklan R, Kaoungku N, Suksut K, Kerdprasop K, Kerdprasop Nittaya. Runoff prediction with a combined artificial neural network and support vector regression. *Int J Machine Learn Comp* 2018;8:39–43. <https://doi.org/10.18178/ijmlc.2018.8.1.660>.