



Contents lists available at ScienceDirect

# Nuclear Engineering and Technology

journal homepage: [www.elsevier.com/locate/net](http://www.elsevier.com/locate/net)

Original Article

## Modeling of zirconium alloy cladding corrosion behavior based on neural ordinary differential equation

Tao Zhang <sup>\*</sup>, Yongjun Jiao <sup>\*</sup>, Zhenhai Liu, Shuo Xing, Haoyu Wang , Kun Zhang, Yuanming Li

State Key Laboratory of Advanced Nuclear Energy Technology, Nuclear Power Institute of China, Chengdu, 610213, China



### ARTICLE INFO

#### Keywords:

Cladding corrosion model  
Differential evolution algorithm  
Neural ordinary differential equation  
Data-model fusion driven

### ABSTRACT

Current zirconium alloy cladding corrosion models are mainly semi-empirical and show significant dispersion when compared to measured data. This study introduces neural ordinary differential equation (neural ODE) to model corrosion behavior, utilizing data-model fusion approach for network training. Initially, a semi-empirical model for zirconium alloy cladding corrosion is established through differential evolution algorithm, generating a large dataset for pre-training the neural network. The network is then fine-tuned using measured data. These methods effectively address the challenges of sparse cladding corrosion data and data available only at fixed time points, resulting in a more accurate model. The results show that the differential evolution algorithm can identify a set of appropriate parameters for the semi-empirical model, achieving a standard deviation of 0.040. The neural ODE model demonstrates even higher accuracy, reducing the standard deviation to 0.031 and improving accuracy by approximately 25%. Additionally, the model demonstrates excellent generalization capacity on other time points and new power histories.

### 1. Introduction

In a typical pressurized water reactor fuel rod, the zirconium alloy cladding acts as a crucial barrier to contain radioactive fission products, which is essential for maintaining the integrity of the fuel rod [1,2]. In the reactor, the zirconium alloy cladding forms an oxide film upon contact with the coolant, leading to a reduction in the metal layer and impacting the mechanical properties of the cladding. If the zirconium alloy cladding corrodes excessively, there may be a potential risk of damage [3]. Therefore, studying and establishing a corrosion model for zirconium alloy cladding is of great significance.

The corrosion of zirconium alloys in pressurized water reactors occurs as oxygen vacancies migrate from the oxide-metal interface through the oxide layer to the oxide-coolant surface [4]. This process is highly complex and is influenced not only by the critical factor of the oxide-metal interface temperature but also by alloy composition, tubing texture, coolant chemistry, fast neutron flux, and hydrogen concentration [5]. Analyzing these factors typically can only be experimentally conducted outside the reactor [6], and even when focusing on one type of cladding, predicting in-pile corrosion remains challenging because the important variable of chemical condition is not always reported. Currently, the Arrhenius equation is used in engineering to model corrosion behavior of zirconium alloy. This semi-empirical

model relies on temperature and corrosion transition points as input parameters, while other influences, such as chemical conditions, are reflected in its coefficients. In practice, cladding corrosion models are typically developed for specific reactor types, with different cladding material models often developed separately. The chemical conditions for a given reactor type are generally consistent, as are factors such as the texture and composition of different cladding materials. Therefore, this modeling approach adequately meets most engineering requirements and has been applied in many fuel performance codes, such as FUPAC [7], FRAPCON [8] and MATPRO [4]. Similarly, this study focuses on establishing a corrosion model tailored to specific cladding materials within particular reactors. Determining coefficients typically requires measured in-pile data to refine existing models or to derive new coefficients through optimization algorithms.

However, these methods are limited by the expression ability of the equations themselves and always have a large degree of dispersion compared to measured data. Therefore, a novel method is required to achieve more precise modeling. Recent advancements in machine learning (ML) have achieved unprecedented breakthroughs in the fields of science and engineering [9,10]. Owing to its strong non-linear fitting capabilities [11], ML provides a new modeling method when the physical mechanism is not fully understood. It has already been applied in fields such as failure mechanism modeling [12], prognostics

<sup>\*</sup> Corresponding authors.

E-mail addresses: [taozhan22@gmail.com](mailto:taozhan22@gmail.com) (T. Zhang), [jiaoyongjun@npic.ac.cn](mailto:jiaoyongjun@npic.ac.cn) (Y. Jiao).

<https://doi.org/10.1016/j.net.2024.10.013>

Received 5 June 2024; Received in revised form 29 August 2024; Accepted 10 October 2024

Available online 16 October 2024

1738-5733/© 2024 Korean Nuclear Society. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

and health management [13], and materials modeling [14,15]. Currently, there have been corresponding efforts in utilizing AI to predict the corrosion behavior of materials, such as magnesium alloys [16], carbonated cementitious mortars [17], and nickel-based alloys [18]. These applications typically require a large amount of data and often ignore changes in system parameters throughout the entire corrosion process. However, the corrosion behavior of zirconium alloy cladding in the reactor is a dynamic process and its data is sparse and only exists at fixed time points, making existing methods difficult to directly apply. Among various machine learning methods, the neural ordinary differential equation (neural ODE) [19] has been identified as particularly well-suited for modeling dynamic systems. Increasing research has applied this approach to scientific problems, such as multi-step wind speed prediction [20] and virus outbreak forecasting [21]. This study employs neural ODE to predict corrosion rate, using a data-model fusion approach to effectively address the challenge of sparse data, thereby enhancing the model's accuracy and generalization capabilities.

## 2. Method

The greatest challenge in modeling the corrosion behavior of zirconium alloy cladding using neural networks lies in the measured data is sparse, and only exists at fixed time points. This is because poolside inspection can only monitor the data of fuel rods at the edges of the assembly, with the number of inspection time points being very limited and corresponding to power cycles. A general approach is to model this as a time series problem, where the input consists of the physical quantities at the current time step and the subsequent input, and the output consists of the physical quantities at the next time step. Recurrent neural networks, gated recurrent units, and long short-term memory networks can be used for this purpose. However, training recurrent neural networks generally requires a substantial number of data points over time, which is difficult to achieve due to measurement constraints. The usual practice is to use interpolation techniques to supplement the temporal data points. However, in this problem, data exists only at three time points, making it challenging to expand the data through interpolation techniques. Hence, implementing this issue with recurrent neural networks is challenging. This study mainly adopted two methods to solve this problem:

- (1) Modeling zirconium alloy cladding corrosion behavior using neural ODE. The reason for using neural networks to predict corrosion rate is that each measured oxide film thickness represents the integral of the corrosion rate over all previous time points, thereby encapsulating a large amount of information. However, solving for oxide film thickness by integrating the corrosion rate requires small time steps to control errors, leading to repeated calls to the network in the forward process. This results in a long computational graph, making the network difficult to train. Neural ODE is introduced to address this training challenge.
- (2) Using a data-model fusion driven method to train the network. This method leverages prior knowledge of the physical system to enhance the neural network's generalization to power histories and time steps outside the dataset, the process is shown in Fig. 1. First, a new semi-empirical model is established using optimization algorithms to generate a large amount of data for pre-training the neural network. This pre-training process essentially injects the physical information from the semi-empirical model into the neural network. Then, the pre-trained network is fine-tuned using measured data to better reflect the real physical laws.

The following sections will describe how to establish a semi-empirical model using optimization algorithms and how to predict corrosion rate using neural ODE.

### 2.1. Establishing a semi-empirical model using optimization algorithms

This study, like previous in-pile corrosion modeling research, considers only temperature as input, other factors are incorporated into the coefficients of a semi-empirical equation. The semi-empirical equation describing the corrosion of zirconium alloy cladding, based on the Arrhenius equation, is:

$$\begin{cases} \frac{ds^3}{dt} = A_1 e^{-\frac{Q_1}{RT}}, s \leq s_{\text{tran}} \\ \frac{ds}{dt} = A_2 e^{-\frac{Q_2}{RT}}, s > s_{\text{tran}} \end{cases} \quad (1)$$

where  $s$  is the thickness of oxidation film,  $s_{\text{tran}}$  is the transition thickness,  $t$  is the time,  $Q_1, Q_2$  are the activation energy,  $R$  is the gas constant,  $A_1, A_2$  are the coefficients, and  $T$  is the oxide-metal interface temperature. The transition point is usually determined by experiments and experience, there are time transitions and thickness transitions in FUPAC [7]. Besides, Eq. (1) describes the cubic rate before transition and linear rate after transition, and there is also a parabolic rate before transition [22], that is:

$$\frac{ds^2}{dt} = A_1 e^{-\frac{Q_1}{RT}} \quad (2)$$

There are four parameters that need to be confirmed:  $A_1, A_2, Q_1, Q_2$ . Notably, the determination of transition thickness or transition time often relies on measured data. However, available measured data is limited to fixed time, posing challenges in estimating the transition time or thickness accurately. Therefore, transition time or thickness is also considered as parameters to be optimized. The range of optimization parameters is:

- $A_1 \in (10^{-15}, 1), A_2 \in (10^{-15}, 1)$
- $Q_1/R \in (8000, 20000), Q_2/R \in (8000, 20000)$
- $s_{\text{tran}} \in (1, 10)$

The unit of  $A, Q/R, s_{\text{tran}}$  are m/s, K,  $\mu\text{m}$  respectively. The range of  $Q$  and  $s_{\text{tran}}$  refers to the models in MATPRO [4], FRAPCON [8] and FUPAC [7], and further expands the parameter range. Due to the negative exponential nature of  $Q$ , even small changes in its value can greatly impact the results, as parameter  $A$  has a wide range that enables finding appropriate values for both the upper and lower limits of  $Q$ .

The fitness function uses the mean square error (MSE) of absolute value. The reason for using absolute error instead of relative error is that the measurement uncertainty is fixed, so low measurement values have greater relative error. In addition, the design pays more attention to the accuracy of the model in predicting thick oxide films, so absolute error is used. The purpose of squared is to impose greater punishment on outliers. Solving the fitness function requires solving ordinary differential equations Eq. (1), which includes the following step:

- (1) Calculate the fluid temperature history using the power history.
- (2) Compute the oxide-metal interface temperature using the fluid temperature.
- (3) Integrate the semi-empirical equation to determine the oxide thickness.
- (4) Calculate the fitness function using MSE.

After determining the variables to be optimized and the fitness function, it is necessary to choose an appropriate optimization algorithm. Optimization algorithms can be categorized into two categories: gradient based algorithms and heuristic intelligent optimization algorithms. Gradient based algorithms use the gradient of the objective function to perform a deterministic search for a single solution, which can be prone to get trapped in local optima. The heuristic intelligent optimization algorithms are based on a bionic algorithm which has a better global optimization capability. This study evaluated various intelligent optimization algorithms, such as simulated annealing, particle swarm optimization, genetic algorithm, and differential evolution, and

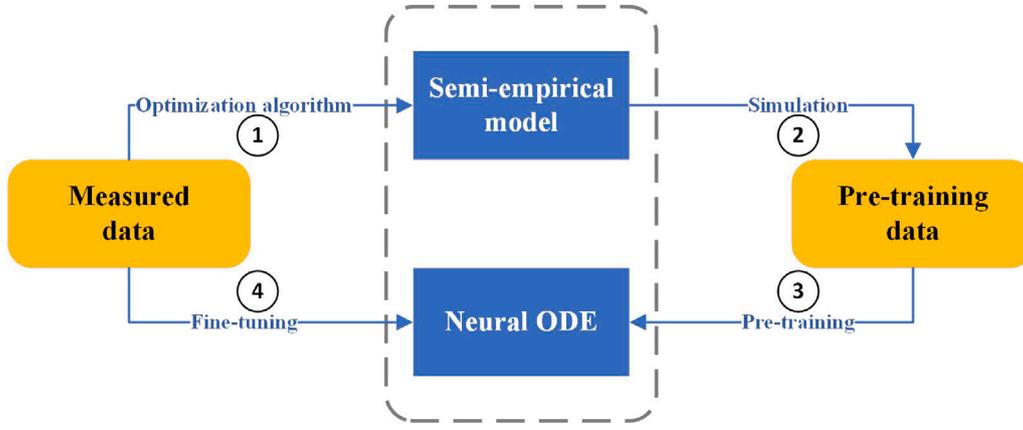


Fig. 1. Data-model fusion driven method.

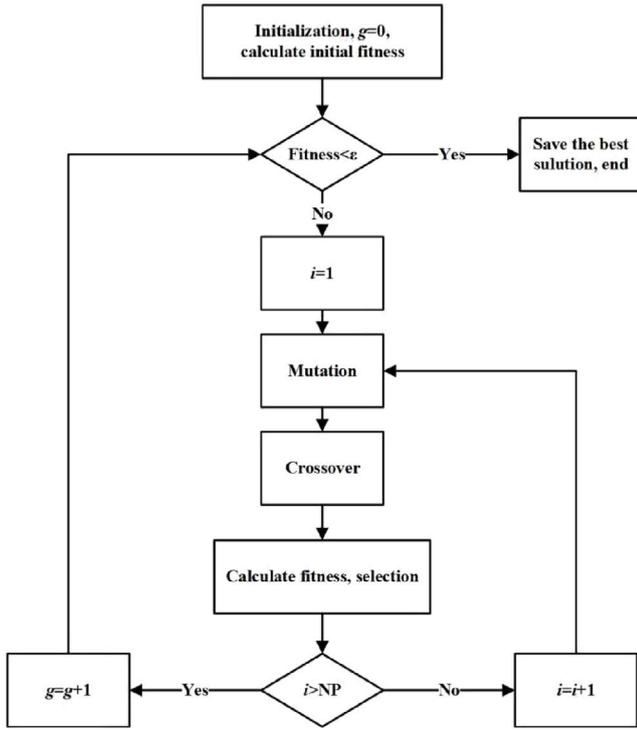


Fig. 2. Differential evolution algorithm flow.

selected the best performing differential evolution algorithm among them. Differential evolution algorithm is a heuristic random search method based on population differentiation, characterized by its simple principle, few controlled parameters, and strong robustness, making it suitable for solving global optimization problems of multi-objective continuous variables. It consists of four processes: initialization, mutation, crossover, and selection [23]. The algorithm flow is shown in Fig. 2.

## 2.2. Predicting corrosion rate using neural ODE

### 2.2.1. The principle of neural ODE

Neural ODE can be thought of as a continuous form of residual network, which can be expressed as:

$$h_{t+1} = f(h_t, \theta_t) + h_t \quad (3)$$

where  $h_t, \theta_t$  are the input and parameter of the  $t$ -th layer network respectively, and  $f$  is the output. By residual connection, the final output is  $f + h_t$ . Transform it into a continuous form:

$$\frac{dh(t)}{dt} = f(h_t, \theta) \quad (4)$$

Eq. (4) is the mathematical description of neural ODE, the output is a derivative information. By defining the derivative term as the corrosion rate, then  $h_t$  is the thickness of the oxide film at time  $t$ . Given the initial value  $h_0$ , the value  $h_t$  can be computed by a black-box differential equation solver. Numerical algorithms such as the Forward Euler or Runge–Kutta methods can be employed to calculate the thickness at each time step, with errors controlled by adjusting the step size. The main technical difficulty in training continuous-depth networks is backpropagation through the ODE solver. And the adjoint sensitivity method is used to solve this problem, it can be obtained by mathematical derivation that [19]:

$$\begin{cases} \frac{da(t)}{dt} = -a(t)^T \frac{\partial f(h(t), t, \theta)}{\partial h} \\ \frac{dL}{d\theta} = - \int a(t)^T \frac{\partial f(h(t), t, \theta)}{\partial \theta} dt \end{cases} \quad (5)$$

where  $a(t) = \partial L / \partial h(t)$ ,  $L$  is loss function. By using this method, the process of back-propagating the gradient also becomes an ODE problem with initial values, it exchanges time for space and takes up little memory.

In general, the advantages of neural ODE can be enumerated as follows: First, each layer shares a common set of network parameters, reducing the total number of parameters. Second, the utilization of the adjoint sensitivity method minimizes memory requirements. Third, the error can be controlled through the application of numerical methods.

### 2.2.2. Input features

In the semi-empirical equation, the corrosion rate is related to the oxide–metal interface temperature and the current thickness of the corrosion film. However, calculating the interface temperature is a complex process. It involves first determining the fluid temperature through thermal balance, then calculating the oxide film surface temperature using an empirical equation and linear heat generation rate, and finally deriving the interface temperature. Additionally, the calculation of oxide thickness and interface temperature is an iterative process that requires convergence of both physical quantities, which poses challenges for neural networks in handling such iterative problems. To avoid the iterative process, fluid temperature, linear heat generation rate, and oxide film thickness were selected as input parameters, with the network output being the corrosion rate. The network architecture is shown in Fig. 3.

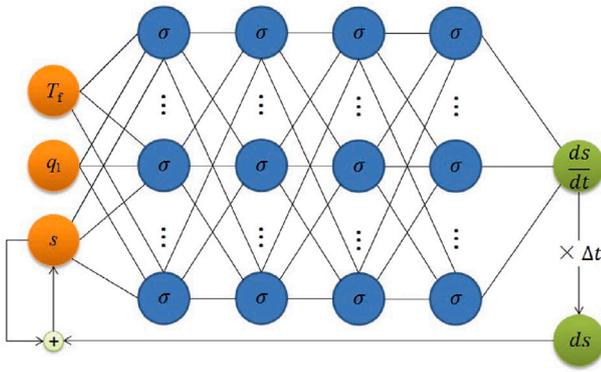


Fig. 3. Network architecture for predicting corrosion rate.

It is worth noting that the corrosion of zirconium alloys is actually influenced by many factors. Considering more factors may pose difficulties in fitting explicit empirical equations, but for neural networks, it is a very simple task that only requires increasing the number of neurons in the input layer.

### 2.2.3. Training

The pre-training dataset is derived from the randomly generated power histories of 10,000 fuel rods. Each dataset contains 79 time steps, with each time step randomly ranging from 0 to 1000 h, and power values randomly distributed between 0 and 30 kW/m. Subsequently, the semi-empirical model established through optimization algorithms is used to calculate the oxide thickness at 16 random positions on the fuel rod, the 16 positions selected for each rod are different to ensure input diversity. Finally, the pre-training dataset contains 160,000 instances.

The loss function adopted is MSE. Additionally, the growth of the oxide layer should satisfy the monotonicity constraint. To prevent the network from overfitting and generating predictions that do not conform to physical laws, the monotonicity constraint is incorporated into the loss function as a regularization term. The loss function is:

$$loss = \sum_{i=1}^{N_d} (M_i - P_i)^2 + w \sum_j^{N_h} \sum_{i=1}^{N_t} f(P^{t+1,j}, P^{t,j}) \quad (6)$$

where  $N_d$ ,  $N_h$ ,  $N_t$  represent the number of measured data, power history, and time step respectively,  $w$  is the regularization factor, and:

$$f(P^{t+1,j}, P^{t,j}) = \max(0, P^{t,j} - P^{t+1,j}) \quad (7)$$

A multilayer perceptron (MLP) is employed to predict the corrosion rate. The performance of the neural network is highly sensitive to its hyperparameters. Initially, the performance of three models with different hyperparameter configurations is evaluated on a pre-training dataset, as shown in Table 1. Among these, model 2, consisting of 4 hidden layers with 16 neurons per layer, produces the best results. Increasing the number of parameters further leads to performance degradation due to overfitting, confirming model 2 as the optimal choice. Subsequently, the impact of different fine-tuning strategies on model performance is examined, as fine-tuning typically involves reducing the learning rate and adjusting some or all of the model's parameters. Three strategies are compared: adjusting only the parameters of the last layer, the last two layers, and all layers, with the results shown in Table 2. Fine-tuning all layers yields the best performance, and further reducing the learning rate to 0.0005 improves it even more. Additionally, the forward Euler algorithm is used to integrate and solve the oxide film thickness. The ratio of the training set to the testing set is 8:2. Other hyperparameter values used for training are provided in Table 3.

Table 1  
Comparison of network hyperparameters.

Model	Neuron	Hidden layer	Validation loss
1	16	2	1.18e-2
2	16	4	0.92e-2
3	32	4	1.41e-2

Table 2  
Comparison of fine-tuning strategies.

Strategy	Fine-tuned layer	Learning rate	Validation loss
1	Last layer	0.001	4.38e-2
2	Last two layers	0.001	3.92e-2
3	All layers	0.001	3.76e-2
4	All layers	0.0005	3.73e-2

Table 3  
Hyperparameters used for training.

Hyperparameters	Pre-training	Fine-tuning
Optimizer	Adam	Adam
Size of dataset	160 000	108
Number of epoch	500	500
Batch size	10 000	32
Learning rate	0.01	0.0005
Activation function	ReLU	ReLU
Decay factor	0.95	0.90

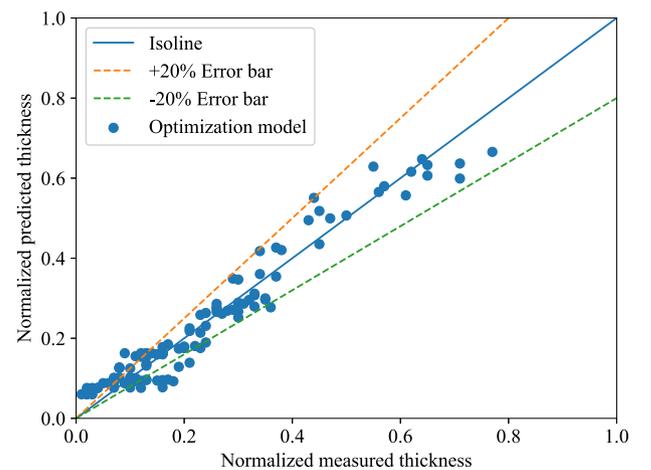


Fig. 4. Comparison of measured data and predicted data of semi-empirical model.

## 3. Result and discussion

### 3.1. Semi-empirical model

Due to the various forms of semi-empirical model, comparisons were made between the parabolic rate and cubic rate model, and the time and thickness transition. The results showed that the parabolic rate model with thickness transition performed better. Therefore, the semi-empirical model used the parabolic rate model and thickness transition, and the differential evolution algorithm was employed to optimize the parameters of the semi-empirical model. The comparison between the measured data and predicted data is shown in Fig. 4. Almost all the high-value measured data points fall within the  $\pm 20\%$  error bars, with a standard deviation (normalized, the same below) of 0.04 for the entire dataset. The whole life prediction is shown in Fig. 5. Overall, the differential evolution algorithm finds a set of optimal parameters for the semi-empirical model, which better describes the corrosion behavior of the cladding. However, due to the limited expressive capability of the semi-empirical equation, there is still some dispersion between the measured data and the predicted data.

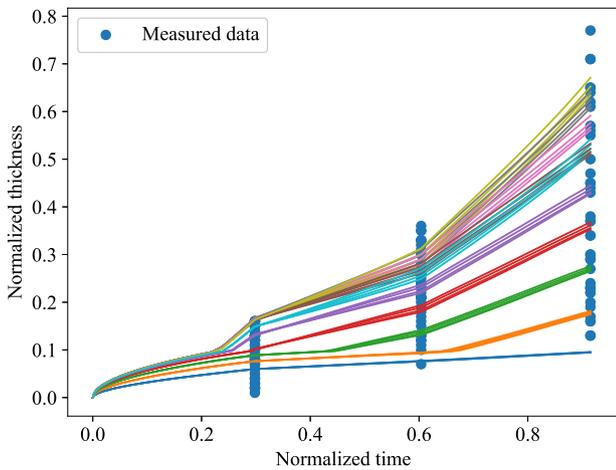


Fig. 5. Whole-life prediction of semi-empirical model.

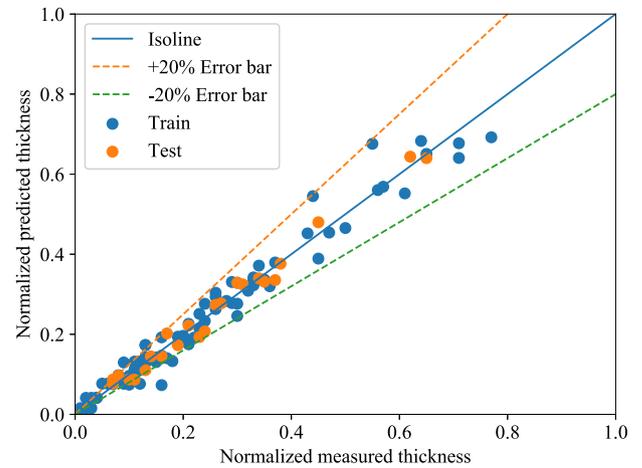


Fig. 6. Comparison of measured data and predicted data of neural network.

### 3.2. Neural network model

The comparison of the measured data and the predicted data of neural network is shown in Fig. 6. Compared to the semi-empirical model, the neural network's predicted data also fall within the  $\pm 20\%$  error bars, but its data is more tightly grouped around the isoline, indicating superior fitting. Besides, the model also shows better performance on the test dataset without overfitting. Additionally, the neural network's standard deviation decreased to 0.031, resulting in a further 25% increase in accuracy.

A more intuitive comparison is presented in Fig. 7, where Fig. 7(a) shows the error distribution histograms for the two models, and Fig. 7(b) displays boxplots and Gaussian distribution fitting curves for data points. The boxplots indicate the range covering 80% of the data. In Fig. 7(a), it is evident that the errors of the neural network model are primarily concentrated around zero, indicating better fitting performance, whereas the semi-empirical model exhibits a broader error distribution. The boxplots in Fig. 7(b) show that 80% of the data for the neural network model is concentrated within a narrower range. Additionally, the error distribution curves show that the variance of the neural network model is smaller. Overall, the neural network model exhibits smaller errors and improved fitting performance.

The whole life prediction is shown in Fig. 8. It can be found that by leveraging pre-training and monotonicity constraints, the neural network excels in accurately fitting the observed data and making reliable predictions even when measured data does not exist.

To compare the impact before and after fine-tuning, five histories were randomly selected from the pre-training dataset, the results are shown in Fig. 9. It can be found that fine-tuning not only adjusts the prediction of measurement points but also has an influence on global prediction. This indicates that the neural network predicts the corrosion rate, not limited to a fixed time.

To further validate the model's generalization capability on power histories outside the measured dataset, three power cycle histories of 256 fuel rods in an actual reactor are selected for verification, with each power cycle containing 70 time steps. Due to the lack of measured data, only the comparison between the neural network model and the semi-empirical model obtained through the optimization algorithm is performed. It should be noted that the values calculated by the semi-empirical model are not true values but rather results that can relatively reflect real physical phenomena, thereby partially validating the neural network's predictions. In addition, a neural network model is also trained using only measured data to validate the data-model fusion method's effectiveness. Both models have identical parameters,

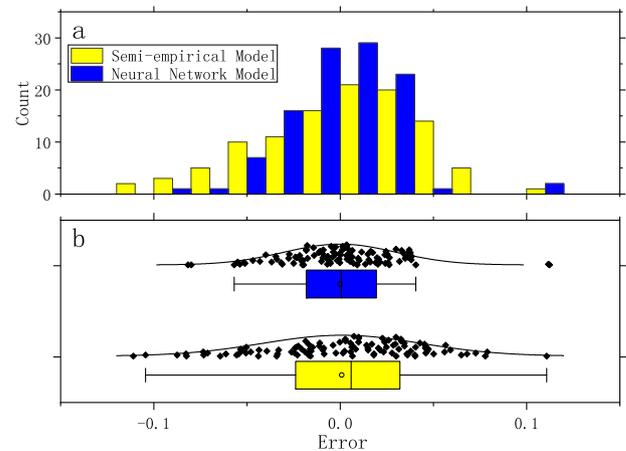


Fig. 7. Histogram and box plot of error distribution for two models.

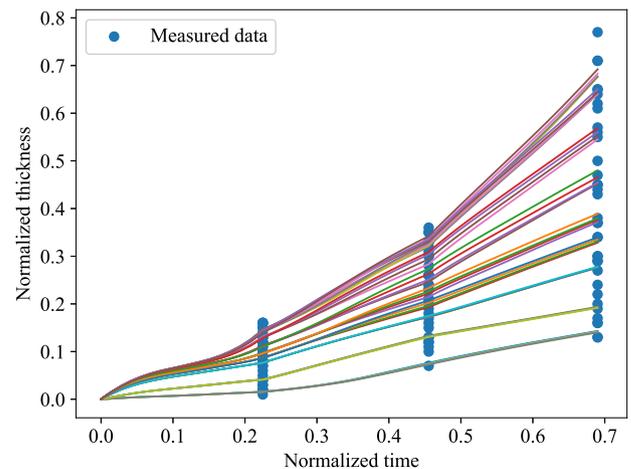


Fig. 8. Whole-life prediction of neural network.

but training with only measured data required a larger learning rate and epoch. The resulting model exhibits a standard deviation of 0.033 on the measured dataset, which is slightly higher than that of the fine-tuned model. The results are shown in Fig. 10. Training directly with measured data results in a significant difference between the

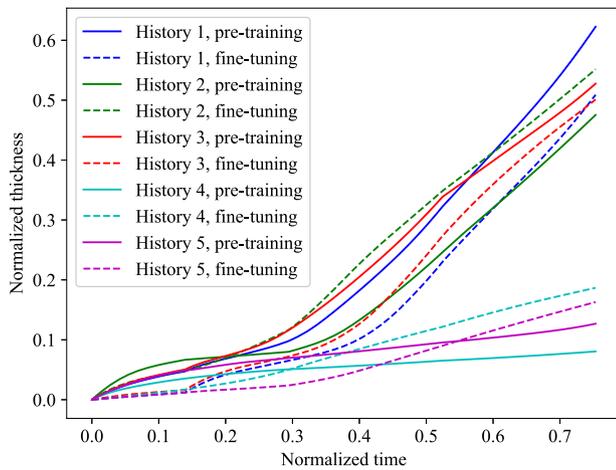
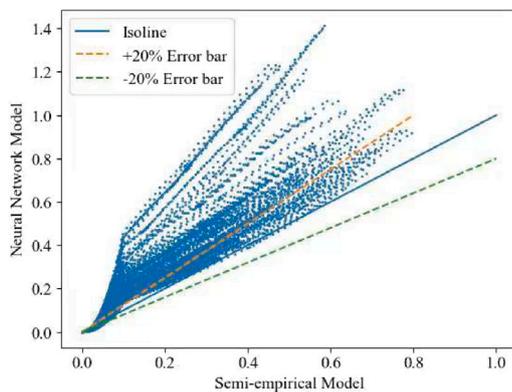
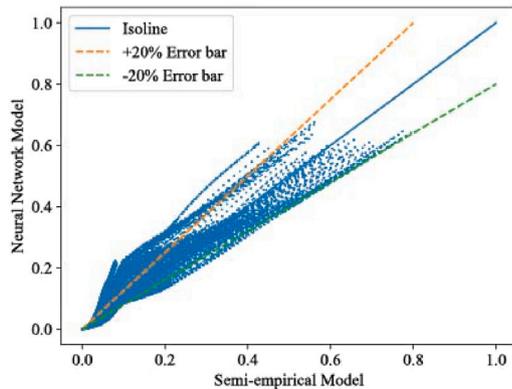


Fig. 9. Comparison before and after fine-tuning.



(a) Training using only measured data



(b) Training with data-model fusion method

Fig. 10. Verification of generalization.

obtained model and the semi-empirical model, demonstrating its poor ability to generalize power history beyond the training set. In contrast, when using the data-model fusion method, the prediction deviations of the two models for thicker oxide layers are mostly within 20%. Consequently, neural network models utilizing the data-model fusion method can improve prediction accuracy for power history outside the training dataset, enhancing the model's generalization ability to a certain extent.

## 4. Conclusion

This paper introduces neural ODE to model the corrosion behavior of zirconium alloy cladding, using a data-model fusion driven method to address the issue of sparse data. First, a semi-empirical model for zirconium cladding corrosion is established using a differential evolution algorithm. The optimized semi-empirical model fits the data well, with a standard deviation of 0.040. Subsequently, the established semi-empirical model is used to generate a large amount of pre-training data, allowing the neural network to learn the physical principles of the semi-empirical model. Finally, the neural network is fine-tuned using measured data to better reflect the actual physical laws. The resulting neural network model further reduces the standard deviation to 0.031, improving accuracy by 25% compared to the semi-empirical model, and demonstrates good generalization to new power histories outside the training set.

The data-model fusion driven method used in this study is particularly suitable for scenarios with limited data. This method first establishes an initial mathematical model based on existing data and prior knowledge such as semi-empirical model to approximate basic physical laws. Subsequently, this model is used to generate a large amount of training data for pre-training the neural network. Finally, the neural network is fine-tuned with measured data to better reflect actual physical phenomena. For dynamic problems, the neural ODE used in this study can learn the rates of change from the data. Combining these two methods can be used to construct nuclear fuel behavior models, such as irradiation growth, creep, or other dynamic processes with limited data.

## CRedit authorship contribution statement

**Tao Zhang:** Writing – original draft, Visualization, Software, Methodology, Investigation. **Yongjun Jiao:** Writing – review & editing, Supervision, Resources, Conceptualization. **Zhenhai Liu:** Writing – review & editing, Validation, Software, Conceptualization. **Shuo Xing:** Methodology. **Haoyu Wang:** Supervision. **Kun Zhang:** Project administration. **Yuanming Li:** Project administration.

## 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.

## Acknowledgments

This study did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

## Data availability

The data is not available.

## References

- [1] P. Van Uffelen, J. Hales, W. Li, G. Rossiter, R. Williamson, A review of fuel performance modelling, *J. Nucl. Mater.* 516 (2019) 373–412.
- [2] V. Kalavathi, R. Kumar Bhuyan, A detailed study on zirconium and its applications in manufacturing process with combinations of other metals, oxides and alloys – A review, *Mater. Today: Proc.* 19 (2019) 781–786.
- [3] A.T. Motta, A. Couet, R.J. Comstock, Corrosion of zirconium alloys used for nuclear fuel cladding, *Annu. Rev. Mater. Res.* 45 (2015) 311–343.
- [4] D.L. Hagerman, G.A. Reymann, MATPRO-Version 11: A Handbook of Materials Properties for Use in the Analysis of Light Water Reactor Fuel Rod Behavior, Technical Report, Idaho National Lab.(INL), Idaho Falls, ID (United States), 1979.
- [5] F. Garzarolli, D. Jorde, R. Manzel, J. Politano, P. Smerd, Waterside corrosion of Zircaloy-Clad fuel rods in a PWR environment, in: *Zirconium in the Nuclear Industry*, ASTM International, 1982.

- [6] S. Pei, F. Yang, N. Feng, J. Hu, G. Shao, G. Yuan, G. Cao, On the thermal stability and oxidation resistance of Zr/X (Cr, Ni, Si) multilayer structure, *Surf. Coat. Technol.* 440 (2022) 128500.
- [7] Y. Zhou, P. Chen, L. Zhang, W. Li, S. Xing, X. Guo, Z. Liu, X. Tu, Modeling of PWR fuel rod irradiation behaviour, *Nucl. Power Eng.* 35 (2014) 200–202.
- [8] K. Geelhood, W.G. Luscher, C. Beyer, M. Flanagan, FRAPCON-3.4: A Computer Code for the Calculation of Steady State Thermal-Mechanical Behavior of Oxide Fuel Rods for High Burnup, US Nuclear Regulatory Commission, Office of Nuclear Regulatory Research, 2011.
- [9] H. Wang, T. Fu, Y. Du, W. Gao, K. Huang, Z. Liu, P. Chandak, S. Liu, P. Van Katwyk, A. Deac, A. Anandkumar, K. Bergen, C.P. Gomes, S. Ho, P. Kohli, J. Lasenby, J. Leskovec, T.Y. Liu, A. Manrai, D. Marks, B. Ramsundar, L. Song, J. Sun, J. Tang, P. Velickovic, M. Welling, L. Zhang, C.W. Coley, Y. Bengio, M. Zitnik, Scientific discovery in the age of artificial intelligence, *Nature* 620 (7972) (2023) 47–60.
- [10] X. Zhang, L. Wang, J. Helwig, Y. Luo, C. Fu, Y. Xie, M. Liu, Y. Lin, Z. Xu, K. Yan, Artificial intelligence for science in quantum, atomistic, and continuum systems, 2023, arXiv preprint arXiv:2307.08423.
- [11] T. Chen, H. Chen, Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems, *IEEE Trans. Neural Netw. Learn. Syst.* 6 (4) (1995) 911–917.
- [12] S.-P. Zhu, X. Niu, B. Keshtegar, C. Luo, M. Bagheri, Machine learning-based probabilistic fatigue assessment of turbine bladed disks under multisource uncertainties, *Int. J. Struct. Integr.* 14 (6) (2023) 1000–1024.
- [13] S.-P. Zhu, L. Wang, C. Luo, J.A. Correia, A.M. De Jesus, F. Berto, Q. Wang, Physics-informed machine learning and its structural integrity applications: state of the art, *Phil. Trans. R. Soc. A* 381 (2260) (2023) 20220406.
- [14] W. Zhang, X. Wang, Y. Ai, W. Zhang, Selection of mechanical properties of uranium and uranium alloys after corrosion based on machine learning, *Mater. Today Commun.* 38 (2024) 107606.
- [15] E.J. Kautz, A.R. Hagen, J.M. Johns, D.E. Burkes, A machine learning approach to thermal conductivity modeling: A case study on irradiated uranium-molybdenum nuclear fuels, *Comput. Mater. Sci.* 161 (2019) 107–118.
- [16] A. Moses, D. Chen, P. Wan, S. Wang, Prediction of electrochemical corrosion behavior of magnesium alloy using machine learning methods, *Mater. Today Commun.* 37 (2023) 107285.
- [17] H. Ji, H. Ye, Machine learning prediction of corrosion rate of steel in carbonated cementitious mortars, *Cem. Concr. Compos.* 143 (2023) 105256.
- [18] T. Dudziak, P. Gajewski, B. Śnieżyński, V. Deodshmukh, M. Witkowska, W. Ratuszek, K. Chruściel, Neural network modelling studies of steam oxidised kinetic behaviour of advanced steels and Ni-based alloys at 800 °C for 3000 h, *Corros. Sci.* 133 (2018) 94–111.
- [19] R.T.Q. Chen, Y. Rubanova, J. Bettencourt, D.K. Duvenaud, Neural ordinary differential equations, in: *Advances in Neural Information Processing Systems*, Vol. 31, Curran Associates, Inc., 2018.
- [20] R. Ye, X. Li, Y. Ye, B. Zhang, DynamicNet: A time-variant ODE network for multi-step wind speed prediction, *Neural Netw.* 152 (2022) 118–139.
- [21] M. Núñez, N.L. Barreiro, R.A. Barrio, C. Rackauckas, Forecasting virus outbreaks with social media data via neural ordinary differential equations, *Sci. Rep.* 13 (1) (2023) 10870.
- [22] F. Lefebvre, C. Lemaignan, Irradiation effects on corrosion of zirconium alloy claddings, *J. Nucl. Mater.* 248 (1997) 268–274.
- [23] R. Storn, K. Price, Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces, *J. Glob. Optim.* 11 (1997) 341–359.