2025 10th International Conference on Machine Learning Technologies (ICMLT 2025), Helsinki, Finland, May 23-25, 2025. Final accepted manuscript.

SAFE-IML: Sparsity-Aware Feature Extraction for Interpretable Machine Learning with Two-Stage Neural Network Modelling

Hua-Liang Wei [1,2]
[1] *Department of Automatic Control and Systems Engineering*
School of Electrical and Electornic Engineering
[2] *Centre of Machine Intelligence*
The University of Sheffield
Sheffield, S1 3JD, UK
w.hualiang@sheffield.ac.uk; ORCID: 0000-0002-4704-7346

***Abstract*—In recent years, model interpretability has attracted significantly increasing attention and research interests from different backgrounds and perspectives. This paper focuses on interpretation of machine learning models, aiming to propose a new sparsity-aware feature extraction (SAFE) approach to significantly improve the interpretability of neural network models. The SAFE method includes the following two steps: 1) the first step starts with a set of features used for training machine learning models, to generate a significantly large number of new features; 2) with the awareness that augmented feature space is usually redundant, the second step is focused on dimensionality reduction to identify the most important features. These important features will then be used to train neural network models, enabling much better interpretability of learning results, as well as models themselves. The proposed method is referred to as Sparsity-Aware Feature Extraction for Interpretable Machine Learning (SAFE-IML). Two illustrative examples are provided to demonstrate the applicability and efficacy of SAFE-IML.**

Keywords—machine learning, model interpretability, feature engineering, feature selection, neural network, sparse modelling

I. INTRODUCTION

A. Why Is Model Interpretabilty Important?

Data driven modelling techniques based on sequentially or non-sequentially observed data, are ubiquitously used in all fields of science and technology. In many practical applications, model interpretability is highly important and useful for obtaining insights into physical or mechanistic understanding of the mechanism or dynamics that govern the system or process of interest. For example, in medicine and healthcare, machine learning (ML) techniques have been widely used to solve various data-driven modelling problems. Arguably, model interpretability will be a key factor determining whether ML technologies can fully achieve their promise of efficiency and safety in solving challenging problems in medicine and healthcare [1]-[3]. In weather and climate studies, the application of ML techniques has quickly increased in recent years (see e.g. [4],[5]). In these studies, the model prediction accuracy is important, but the identification of important drivers (variables) is equally important or even more desirable [6]-[10]. The past few decades have witnessed the fast growth of ML and its applications everywhere, but meanwhile there has been an increasing interest and demand for exploring transparent and interpretable ML models.

B. Model Sparsity

A wide class of ML modelling problems can be considered as a multi-input single-output (MISO) or multi-input multi-output (MIMO) modelling problem. For simplicity of notation, take the MISO case as an example, where we have a system whose response (output), y, is potentially determined by n input variables: x_1, x_2, …, x_n. Assume that there exists a linear or nonlinear functional relationship between the input $\mathrm{x}=[x_1, x_2, \ldots, x_n]$ and the output y, such that $y = f_{\text{true}}(\mathrm{x})$, where the true function f_{true} is in general unknown due to the lack of knowledge of the system. Given an input-output dataset of a system, the central task or objective of data-driven modelling is to induce a model f from the data such that $f(\mathrm{x})$ can approximate or represent the true function $f_{\text{true}}(\mathrm{x})$ as close as possible.

Various modelling methods are available in the literature. Many traditional models used for multiple linear regression, linear and nonlinear system identification [11], [12] are transparent and easy to explain. Models produced by classical ML methods e.g. decision tree, logistic regression, support vector machine and fuzzy logic are usually interpretable. On the contrary, many other ML models including deep learning and deep neural networks are unexplainable or difficult to interpret.

In many practical applications, models used may be either oversimplified or overcomplicated. For example, assume a system output y is determined by three input variables: x_1, x_2, x_3. If the true model structure of the system is $y = \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2 + \theta_4 x_3^2$, then a linear model $y = ax_1 + bx_2 + cx_3$ would be too simplified to represent the system. On the contrary, without any a priori knowledge of the system, the following model structure may be used represent the system:

$$y = \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_1^2 + \beta_5 x_1 x_2 + \beta_6 x_1 x_3 + \beta_7 x_2^2 + \beta_8 x_2 x_3 + \beta_9 x_3^2 \quad (1)$$

Clearly, model (1) is too complicated to well represent the original system behaviour. Good ML algorithms should be able

to either correctly select the four true model terms, x_1, x_2, x_1x_2, x_3^2, or effectively determine that the coefficients of the five spurious model terms, x_3, x_1^2, x_2^2, x_1x_3, x_2x_3, to be zero, that is, $\beta_3=\beta_4=\beta_6=\beta_7=\beta_8=0$, leading to a sparse model: $y = \beta_1x_1 + \beta_2x_2 + \beta_5x_1x_2 + \beta_9x_3^2$. The procedure of reducing and refining an overcomplicated model to a reasonably simpler one is referred to as sparse model identification [13]-[15].

Consider another example of model sparsity. Table I presents eight samples of a multi-input system. Two algorithms were applied to the data to build two models as follows:

$$y = -x_2 + x_3 - 1.5x_1x_2 + 2x_1x_3 - x_2x_3 \quad (2)$$

$$y = 2x_1 + x_2 - 0.5x_1x_2 \quad (3)$$

While both models perfectly characterise the input-output relationship between the input-output data, model (3) is clearly more compact and parsimonious than model (2). Therefore, model (3) is sparser than and more preferable to model (2).

TABLE I. A SMALL DATASET FOR A 3-INPUT 1-OUTPUT SYSTEM

Inputs & output	1	2	3	4	5	6	7	8
x_1	0	0	0	1	1	1	2	2
x_2	0	1	2	0	1	2	0	1
x_3	0	1	4	1	2	5	4	5
y	0	1	2	2	3.5	5	4	6

A variety of methods and algorithms for sparse model identification have been developed over the past years, including orthogonal least squares (OLS) [16],[17] adaptive orthogonal search (AOS) [18], orthogonal matching pursuit (OMP) [19],[20], least absolute shrinkage and selection operator (LASSO) [21],[22], and sparse Bayesian learning (SBL) [23]-[28], among others. In comparison with other methods, OLS and AOS have an attractive advantage in that they use a simple but effective index [12],[16], called the error reduction ratio (ERR), to measure the significance of model terms; this index gives a clear assessment of the contribution made by each term included in the model to explaining the variation in the response [14],[16],[29].

C. Interpreting Machine Learning Models

Model interpretability is an important requirement for many applications or even part of the essential requirements for the development of trustworthy ML models used for real-life problem solving. Most ML models are complicated and difficult to understand due to their black-box architecture. Massive efforts have been made to try and understand results produced by complicated ML models, and a vast number of publications on model interpretability have been added to the literature from diverse communities in recent years.

A natural way to make machine learning interpretable is to employ transparent models (see e.g. [12]-[15]). For complicated opaque models, many methods have been developed for odel interpretation, including the two most popular and widely commonly used methods, Local Interpretable Model-agnostic Explanations (LIME) [30] and SHapley Additive exPlanations (SHAP) [31]-[34]. LIME has a major deficiency in that it uses a kernel, whose parameters determine how accurate the ML model interpretation is. The explanations given by LIME can be instable and inconsistent, e.g., for two samples that are very close to each other, the method may give very different explanations [35]. One of the main disadvantages of SHAP is its highly heavy computational workload. Another major drawback is related to its theoretical limitations. For example, a concern about the existing definitions of SHAP scores was raised recently [36]-[38]. It was proven and shown that "the existing definition will necessarily yield misleading information about the relative importance of features for predictions."

D. Contributions of This Work

This paper aims to propose a new feature engineering enhanced interpretable machine learning (IML) framework that shows better performance in terms of both model interpretability and prediction capability, by taking advantage of sparsity-aware feature extraction (SAFE) and neural network (NN) approaches. The work makes a novel contribution through designing, implementing and testing a two-stage NN modelling framework as follows: (1) the first NN is used for feature generation, selection and extraction; and (2) the second NN is implemented using a regression neural network. The proposed method is referred to as Sparsity-Aware Feature Extraction for Interpretable Machine Learning (SAFE-IML).

II. PROBLEM SPECIFICATION

A wide range of practical data-driven modelling problems can be represented as an input-output data-based model identification task as follows. There is an output y that depends on an input vector of n variables denoted by $\mathbf{x}=[x_1, \ldots, x_n]$. Assume a set of observation pairs are available, denoted by $\{y(t), \mathbf{x}(t)\}$ with $\mathbf{x}(t) = [x_1(t),..., x_n(t)]$ and (t =1,…,N). The true quantitative representation of the relationship between the output y and the input $\mathbf{x}$ is in general unknown or may never be known. The central task of data modelling is to build a mathematical model, $y(t) = f(\mathbf{x}(t)) + e(t)$ (here $e(t)$ is noise), that can approximate or represent the true input-output relationship, $y(t) = f_{\text{true}}(\mathrm{x}(t))$, as accurate as possible.

A variety of model structures and building blocks have been proposed to construct the function f, including polynomials, radial basis functions, wavelets, fuzzy sets, neural networks, decision trees and random forests, support vector machine, deep learning and deep neural networks (see e.g. [11], [39]-[41]).

A multivariate nonlinear function can often be decomposed into a number of polynomial functional components as:

$$f(x_1, x_2, \cdots, x_n) = a_0 + \sum_{1\le i\le n} b_i x_i + \sum_{1\le i<j\le n} c_{ij} x_i x_j + \sum_{1\le i<j<k\le n} d_{ijk} x_i x_j x_k + \ldots + error \quad (4)$$

where a_0 is a constant, b_i, c_{ij}, d_{ijk}, …, are coefficients of the linear, quadratic and cubic terms. Previous experiences show that a decomposition of up to quadratic or cubic terms can usually provide sufficiently satisfactory approximation.

For a dynamical system, the input-output relationship can also be represented using a similar decomposition. Taking the following 2-input 1-output dynamical system as an example:

$$y(t) = f(y(t-1), y(t-2), u_1(t-1), ..., u_1(t-3), u_2(t-1), ..., u_2(t-3)) + e(t) \quad (5)$$

Define

$$x_1(t) = y(t-1), \quad x_2(t) = y(t-2),$$
$$x_3(t) = u_1(t-1), \quad x_4(t) = u_1(t-2), \quad x_5(t) = u_1(t-3),$$
$$x_6(t) = u_2(t-1), \quad x_7(t) = u_2(t-2), \quad x_8(t) = u_2(t-3),$$

Then model (5) can be written as:

$$y(t) = f(x_1(t), x_2(t), ..., x_8(t)) + e(t) \quad (6)$$

which can be decomposed into a set of polynomial elements.

Polynomial decomposition has several attractive properties, e.g., (1) it is transparent and interpretable; (2) it can be arranged to a linear-in-the-parameters form which is easy to compute and manage; (3) for dynamical systems, it enables to perform analysis not only in the time domain, but also in the frequency domain where important insightful information hidden in the time-domain signals can be revealed, allowing better understanding of original physical systems [12],[42].

Like any other approach, polynomial decomposition has its own shortcomings. For example, in comparison with complicated neural networks, polynomial decomposition may not be able to represent highly nonlinear complex behaviour. However, the computational results produced by complicated neural networks may be extremely hard to explain or understand, especially when the modelling task potentially involves a very large number of input variables.

The above observations motivate us to explore the advantages and disadvantages of polynomial decomposition and neural networks, to develop a SAFE-IML framework.

III. METHOD

This section introduces the proposed two-stage neural network modelling framework, SAFE-IML.

A. The Structure of SAFE-IML

The diagram of SAFE-IML is shown in Fig. 1, where NN1is a 3-layer neural network which is used for feature generation and dimensionality reduction (feature selection and extraction), and NN2 represents 4-layer regression neural network. Details of these networks are given in the following section.

B. The First-Stage Network: NN1

1) Feature generation and augmentation

The input to the first-stage network is $\mathbf{x}=[x_1, x_2, \ldots, x_n]$. The n variables are augmented to a higher feature space. The new feature space is determined by D_0, D_1, D_2 and D_3, which are the complete collection of constant, linear, quadratic and cubic features (model terms), respectively, as follows:

$$D_0 = \{x_0\} \ (x_0 \equiv 1);$$
$$D_1 = \{x_1, x_2, ..., x_n\};$$
$$D_2 = \{x_1^2, x_1x_2, ..., x_1x_n, x_2^2, x_2x_3, ..., x_2x_n, ..., x_{n-1}x_n, x_n^2\};$$
$$D_3 = \{x_3^2, x_1x_1x_2, ..., x_1x_1x_n, ..., x_{n-1}x_nx_n, x_n^3\}.$$

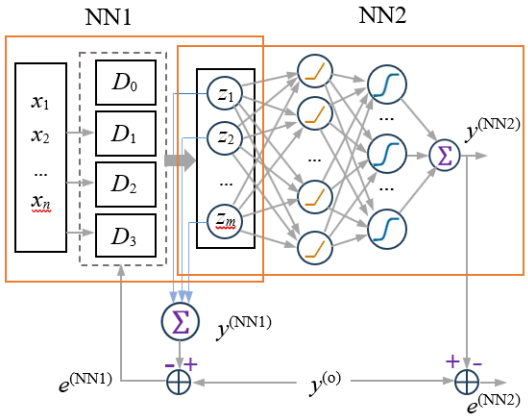

Fig. 1. The diagram of the proposed SAFE-IML framework.

In this way, the original feature space is significantly augmented. The error signal $e^{(NN1)}$ as feedback is sent to the augmented space layer to refine the parameter estimation of the identified model in the feature reduced layer; this is very useful in dynamical system modelling (e.g. when autoregressive terms are included in the model).

2) Feature subset selecton and dimentionality reduction

Let $D = D_0 \cup D_1 \cup D_2 \cup D_3$. It can be known that the set D includes a total of $M = C_n^3 = n(n-1)(n-2)/6$ different elements. The feature library D can also be defined as $D_0 \cup D_1 \cup D_2$ which can often work well for many practical applications. Given a set of samples, $\{y(t), \mathbf{x}(t)\}$ (t =1, 2,…, N), sparse learning algorithms, e.g. OLS [16], AOS [17], LASSO [21],[22], and sparse Bayesian learning [23]-[27], can be used to determine a small subset consisting of m (<< M) important features selected from D. Denote by $\mathbf{z} = [z_1, z_2, \ldots, z_m]$ the m selected features. The principle is that the subset $\mathbf{z}$ should sufficiently well represents the system output $\mathbf{y}$ in the sense that:

a) $\hat{y}(t) = \sum_{k=1}^{m} \beta_k z_k(t)$, $\hat{y}(t)$ is the model prediction value.

b) The overall error $\left\|\mathbf{y}^{(o)} - \hat{\mathbf{y}}\right\|^2$ is satisfactory small, where $\mathbf{y}^{(o)}$ is the observation vector.

C. The Second Neural Network: NN2

The output of NN1, $\mathbf{z} = [z_1, z_2, \ldots, z_m]$, is the input of NN2. The main objective of the second-stage modelling is twofold: (1) to explore the significance of m features and the individual original variables involved in $\mathbf{z}$; and (2) to explore the potential of improving the prediction performance obtained in NN1 through NN2. For (1), an explanation of model prediction performance will be explored using SHAP values.

Note that Fig. 1 only shows a simple case with two fully connected hidden layers and an output layer in NN2. In practical applications, more hidden layers may be added where necessary to achieve potentially strong and better prediction ability, but at the price of weakening model interpretability.

The Shapley value was originally used in cooperative game theory [43], as a method for players to assess a priori how much they each would expect to befit from playing a game. The SHAP method has gradually become the most dominant and most commonly used approach for ML model interpretation since the breakthrough paper by Lundberg and Lee [32]. If a modelling task involves n different elements (variables, regressiors or terms), then each variable can be considered as a player in building a target model, and their contribution and importance can be assessed using the SHAP values. However, the determination of the contribution and importance does depend on the specific model type and model structure chosen and how the available model building elements are used to implement the target model [44].

IV. Examples and Applications

This section provides two examples to illustrate the applicability of the proposed SAFE-IML framework. All the numerical experiments were conducted using MATLAB R2024b. For each example, we report the following:

1) *Output results of NN1*
2) *Output results of NN2 driven by the output of NN1*
3) *Output results of NN2 driven by the origianl inputs, without using any output results of NN1*

A. Example 1: Hidimensional Linear Regression Model

Consider the following model:

$$y = f(x_1, x_2, ..., x_{1000}) = \sum_{k=1}^{1000} \beta_k x_k + \xi \tag{7}$$

where p =1000, x_k (k=1, 2, …, p) are independent variables; β_{10} = 1, β_{20} = 2, β_{30}= –3, and β_k = 0 if $k \neq$10, 20 or 30; ξ is noise. The model was simulated with the following settings: each of the p variables was independently set to be a zero-mean Gaussian process with standard deviation σ_x = 1. The random noise ξ was set to be a zero-mean Gaussian process with standard deviation σ_ξ = 0.5. A total of n=100 simulation samples, that is, output/input pairs $\{y(k), \mathbf{x}(k)\}$ (k=1,2,…,100), were recorded; each input sample has 1000 element values. The first 50 samples were used for model training and the remaining data were used for model testing.

This is a very typical small sample size and "large p, small n" problem. Under the assumption that no a priori knowledge is available about the importance of the 1000 input variables, we applied SAFE-IML to the available 100 samples.

1) Output results of NN1

Both AOS and SBL algorithms were performed in NN1, and the output results are reported in Table II. Clearly, all the three true variables were correctly determined and their importance was well explained with the ERR values [14],[16],[29]. The output of NN1 on the test dataset, $y^{(NN1)}$, is shown in Fig. 2.

2) Output results of NN2 driven by the output of NN1

Driven by the output of NN1, i.e., the three variables, z_1 = x_{30}, z_2 = x_{20} and z_3 = x_{10}, the output of NN2, $y^{(NN2)}$, is shown in Fig. 3. To save space and give a better visualisation, only the predictions on the test data are shown here, but we also calculated the MSE and R^2 values of $y^{(NN2)}$, over the training dataset, which are 9.7263e-07 and 1, respectively. These results show that NN2 performed perfect on the training data, but its performance on the test data is not as good as NN1, meaning that the overall performance of NN1 was not enhanced or improved through NN2.

TABLE II. Output Results of NN1 for Example 1

Variable	Coefficient	Importance (ERR)
$z_1 = x_{30}$	-2.9234	64.3447%
$z_2 = x_{20}$	2.0100	26.3311%
$z_3 = x_{10}$	1.0064	7.4440%
		$\sum$ = 98.1198%

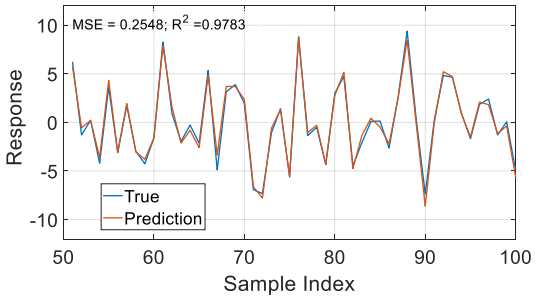

Fig. 2. A comparison between the NN1 prediction and the actual observations on the test dataset (Example 1).

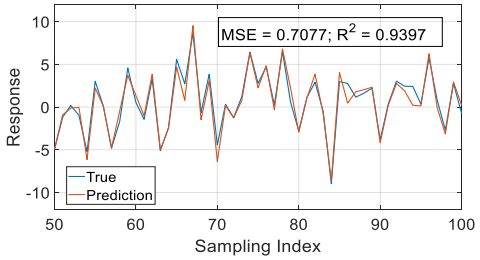

Fig. 3. A comparison between the NN1+NN2 prediction and the actual observations on the test dataset (Example 1).

The SHAP values of the three variables z_1, z_2 and z_3 are shown in Fig. 4. Clearly, the importance of three variables assessed by Shapley values is perfectly consistent with that measured by the ERR index.

3) Output results of NN2 driven by the original inputs, without using any output results of NN1

In this case, NN2 was trained using the original 1000 variables as input. The prediction results on the training and test datasets, together with the distribution of the Shapley values of the 1000 input variables, are shown in Figs. 5 and 6, respectively. It can be observed that while the network performed good on the training data, it showed very pool generalisation ability on the tests dataset. From the Shapley distribution, it failed to identify the three important variables, x_{10}, x_{20} and x_{30}.

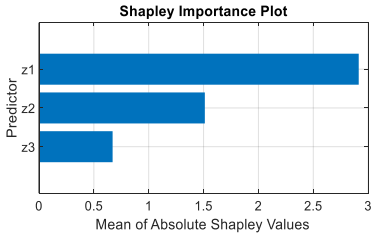

Fig. 4. Shapley values of the three variables z_1=x_{30}, z_2=x_{20} and z_3=x_{10}.

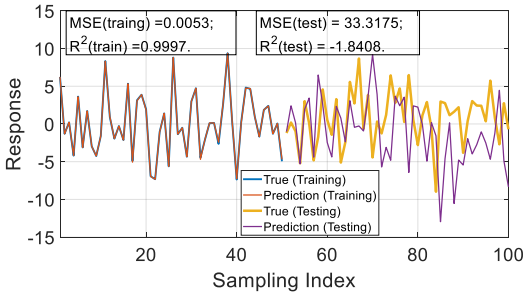

Fig. 5. A comparison between the NN2 predictions and the actual observations without using the output of NN1 (Example 1).

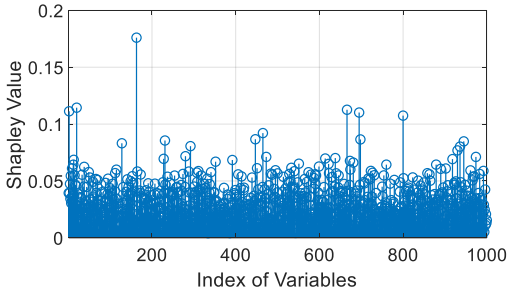

Fig. 6. Shapley values of the 1000 original variables calculated based on NN2. These variables were used as input to NN2; the output of NN1 was not used.

B. *Example 2: Mutiple Nonlinear Regression Model*

In Example 1, all the true model terms are included in the augmented feature library. In Example 2, only one of the true model terms, i.e., x_8, is included the augmented feature space and all the other true model terms are not included in the library, making the feature reduction procedure, i.e., the determination of important model terms, more challenging. The model used to generate data is as follows:

$$y = f(x_1,...,x_{21}) = x_2 \mid x_5 \mid + x_8 + x_{13}\sin(x_{13}) + \frac{4x_{20}}{1+e^{x_{21}}} + \xi \quad (8)$$

where $x_k\,(k=1, 2, \ldots, 21)$ are independent variables, each follows a continuous uniform distribution on (-1,1); ξ is noise, following a Gaussian distribution with standard deviation $\sigma_\xi = 0.1$.

Model (8) was simulated and 200 output/input pairs $\{y(k), \mathbf{x}(k)\}$ (k=1,2,…,21) were recorded. The first 100 samples were used for model training and the remaining samples were used for model testing. The size of the augmented feature space in NN1 is 2024. The main results are reported below.

1) *Output results of NN1*

The output of NN1, i.e., the selected important variables, z_k (k=1,2, …,6) are listed in Table III, together with the values of the ERR index. The prediction, $y^{(NN1)}$, from the model reported in Table III is shown in Fig. 7.

TABLE III. OUTPUT RESULTS OF NN1 FOR EXAMPLE 2

Variable	Coefficient	Importance (ERR)
$z_1 = x_{20}$	2.0243	72.4367%
$z_2 = x_8$	0.9977	11.2821%
$z_3 = x_{13}^2$	0.9230	6.6351%
$z_4 = x_{20}x_{21}$	-0.9208	4.2885%
$z_5 = x_2x_5^2$	0.9158	4.4853%
$z_6 = x_2$	0.1843	0.2171%
		$\sum$ = 99.3447%

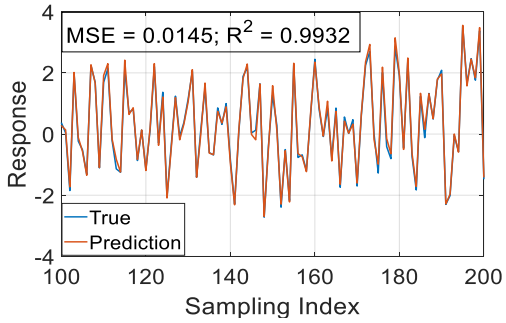

Fig. 7. A comparison between the NN1 prediction and the actual observations on the test dataset (Example 2).

2) *Output results of NN2 driven by the output of NN1*

It followed that the output of NN2, $y^{(NN2)}$, driven by z_k (k= 1, 2, ..., 6), is almost identical to $y^{(NN1)}$. The MSE and R^2 values of $y^{(NN2)}$ on the test data are 0.0147 and 0.9931, respectively. The SHAP values of the six variables, z_1, z_2, …, z_6, are shown in Fig. 8, where it can be seen that the importance of six features assessed by Shapley values is perfectly consistent with that measured by the ERR index.

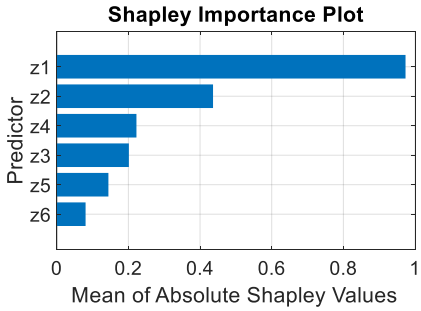

Fig. 8. Shapley values of the six features, z_1, z_2, …, z_6, listed in Table III.

It is interesting to know the importance of each of the model terms for determining the prediction value of a specific point in the response. For example, it is interesting to know the importance of the six model terms, z_1, z_2, …, z_6, for predicting the following three points: lowest peak, highest peak and the end point of the test data, which are corresponding the 148th sample with the prediction value -2.625, the 195th sample with the prediction value 3.6431 and the 200th sample with the prediction value -1.4171. Through NN2, we calculated the importance of the six model terms for each of the three specific points, which are shown in Fig. 9.

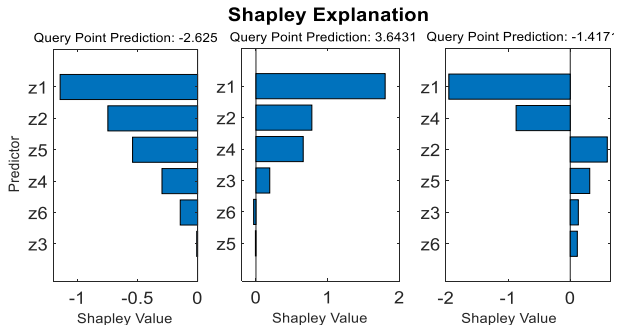

Fig. 9. Shapley values of the six features (model terms), z_1, z_2, …, z_6, for the three specific points: the 145th sample with the prediction value -2.625, the 195th sample with the prediction value 3.6431 and the 200th sample with the prediction value -1.4171.

3) Output results of NN2 driven by the origianl inputs, without using any output results of NN1

In this case, NN2 was trained using the original 21 variables as input. The prediction results on the training and test datasets are shown in Fig. 10, and the distribution of the Shapley values of the 21 input variables is shown in Fig. 11. Clearly, while the network performed good on the training data, it showed slightly weak generalisation ability on the test dataset. From the Shapley distribution, two variables, x_{20} and x_8, which are important for model (8), were correctly identified by the Shapley value, but they failed to identify all the other important variables, such as x_2, x_5, x_{13}, and x_{21}.

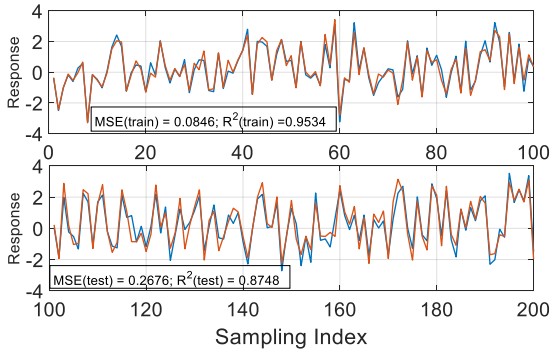

Fig. 10. A comparison between the NN2 prediction and the actual observations without using the output of NN1 (Example 2). Blue curve: observations; red curve: model predictions.

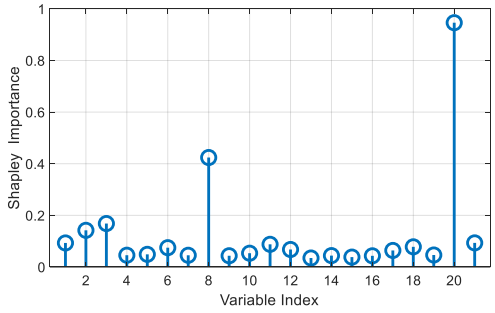

Fig. 11. Shapley values of the 21orginal input variables calculated from NN2. These variables were used as input to NN2; the output of NN1 was not used.

V. Conclusion

The paper investigated state-of-the-art quantitative methods for model interpretability evaluation. It focused on exploring the strengths and weaknesses of traditional feature engineering and the popular SHAP methods. It showed, through two illustrative examples, how the proposed SAFE-IML worked by using a small number of explainable features obtained in the first NN to train the second NN to acquire further explanation using SHAP scores and meanwhile maintain the predictive ability of the second NN. The strong interpretability of the second NN cannot be achieved without using the information produced by the first NN. While the results and findings are interesting and promising, the proposed method still has a few limitations, e.g., for a problem involves many variables, the number of features generated by the polynomial-based approach may very large, this potentially leads to heavy computational load in the dimensionality reduction (subset selection) procedure. Another limitation is to choose the second NN model. A simple model structure is easy to interpret and takes less computation time but it may not have enough learning ability, whereas a complex network model may have strong learning ability but it requires more computational time and lacks interpretability. Our future work will be focusing on finding solutions to these issues.

Acknowledgment

The authors gratefully acknowledge that this work was supported in part by STFC (Ref. ST-Y001524-1), NERC (Ref. NE/W005875/1, Ref. NE/V001787/1, Ref. NE/Y503290/1 and Ref. NE/V002511/1).

References

[1] G. Stiglic et al., ‘‘Interpretability of machine learning-based prediction models in healthcare,’’ WIREs Data Mining Knowl. Discovery, vol. 10, no. 5, p. e1379, Sep. 2020.

[2] A. Vellido, “The importance of interpretability and visualization in machine learning for applications in medicine and health care,” Neural Comput. Appl., vol. 32, no. 24, pp. 18069-18083, Dec. 2020.

[3] C. Priya and P. M. Durai Raj Vincent, “An efficient CSPK-FCM explainable artificial intelligence model on COVID-19 data to predict the emotion using topic modeling,” J. Adv. Inf. Technol., vol. 14, no. 6, pp. 1390-1402, Dec. 2023.

[4] M. Chantry et al. (Editors), "Machine learning for weather and climate modelling," Phil. Trans. R. Soc., Theme Issue, Feb. 2021.

[5] B. Dong et al., "Key drivers of large scale changes in North Atlantic atmospheric and oceanic circulations and their predictability," Clim. Dyn., vol. 63, no. 2, art. no. 113, Feb. 2025.

[6] R. Yang et al., "Interpretable machine learning for weather and climate prediction: A review," Atmos. Environ., 338, 120797, Dec. 2024.

[7] T. Hu et al., "Crop yield prediction via explainable AI and interpretable machine learning: Dangers of black box models for evaluating climate change impacts on crop yield," Agric. For. Meteorol., vol. 336, 109458, Jan. 2023.

[8] R.J. Hall et al., "Complex systems modelling of UK winter wheat yield," Comput. Electron. Agr., vol. 209, art. no. 107855, June 2023.

[9] R.J. Hall, H.-L. Wei, and E. Hanna, "Complex systems modelling for statistical forecasting of winter North Atlantic atmospheric variability: A new approach to North Atlantic seasonal forecasting," Q. J. R. Meteorol Soc., vol. 145, pp. 2568-2585, June 2019.

[10] Y. Sun et al., "Probabilistic seasonal forecasts of North Atlantic atmospheric circulation using complex systems modelling and comparison with dynamical models," Meteorol. Appl., vol. 31, no. 1, e2178, Feb. 2024.

[11] L. Ljung, System Identification: Theory for the User. Englewood Cliffs, NJ, USA: Prentice-Hall, 1987.

[12] S. A. Billings, Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains. Hoboken, NJ, USA: John Wiley & Sons, 2013.

[13] S. A. Billings and H.-L. Wei, ''Sparse model identification using a forward orthogonal regression algorithm aided by mutual information,'' IEEE Trans. Neural Netw., vol. 18, no. 1, pp. 306–310, Jan. 2007.

[14] H.-L. Wei and S. A. Billinsg, "Modelling COVID-19 pandemic dynamics using transparent, interpretable, parsimonious and simulatable (TIPS) machine learning models: A case study from systems thinking and system identification perspectives," in Recent Advances in AI-enabled Automated Medical Diagnosis, CRC Press, 2022, pp. 13-28.

[15] H.-L. Wei, "System identification-informed transparent and explainable machine learning with application to power consumption forecasting," in Proc. 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), Tenerife, Spain, July 2023, pp. 1-6.

[16] S. Chen, S. A. Billings, and W. Luo, "Orthogonal least squares methods and their application to non-linear system identification," *Int. J. Control*, vol. 50, no. 5, pp. 1873–1896, Nov. 1989.

[17] Q. M. Zhu and S. A. Billings, "Fast orthogonal identification of nonlinear stochastic models and radial basis function neural networks", Int. J. Control, vol. 64, pp. 871-886, 1996.

[18] S. A. Billings and H.-L. Wei, "An adaptive orthogonal search algorithm for model subset selection and non-linear system identification," *Int. J. Control*, vol. 81, no. 5, pp. 714-724, April 2008.

[19] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proc. 27th Asilomar Conf. Signals, Syst. Comput.*, vol. 1. Pacific Grove, CA, USA, 1993, pp. 40–44.

[20] J. A. Tropp and A. C. Gilbert, ''Signal recovery from random measurements via orthogonal matching pursuit,'' *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.

[21] R. Tibshirani, "Regression shrinkage and selection via the LASSO," *J. Roy. Stat. Soc., Ser. B (Methodol.)*, vol. 58, no. 1, pp. 267–288, 1996.

[22] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning, 2nd ed. New York: Springer, 2009.

[23] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, pp. 211–244, Jun. 2001

[24] M. E. Tipping, "Bayesian inference: an introduction to principles and practice in machine learning," in *Proc. 9th International Workshop on Artificial Intelligence and Statistics*, C. Bishop et al., eds., Key West, FL, USA, 3–6 Jan. 2003, pp. 276–283.

[25] M. E. Tipping, "Bayesian inference: an introduction to principles and practice in machine learning," in *Advanced Lectures on Machine Learning*, O. Bousquet et al., eds., Berlin, Heidelberg, Germany: Springer, 2004, pp. 41–62.

[26] W. R. Jacobs, T. Baldacchino, and S. R. Anderson, "Sparse Bayesian identification of polynomial NARX models," *IFAC-PapersOnLine*, vol. 48, no. 28, pp. 172–177, Dec. 2015.

[27] N. Zheng, Y. Li, W. Shi, and Q. Xie, "Sparse Bayesian based NARX modeling of cortical response: Introducing information entropy for enhancing the stability," *Neurocomputing*, vol. 626, 129569, April 2025.

[28] H.-L. Wei, "System identification and interpretable modelling of dynamical systems with small data using sparse Bayesian learning," under review, 2025.

[29] H.-L. Wei, S. A. Billings, and J. Liu, "Term and variable selection for nonlinear system identification," Int. J. Control, vol. 77, pp. 86–110, Jan. 2004.

[30] M.T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you? Explaining the predictions of any classifier," in Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 1135–1144.

[31] D. Slack, S. Hilgard, E. Jia, S. Singh, and H. Lakkaraju, "Fooling lime and shap: Adversarial attacks on post hoc explanation methods," in Proc. AAAI/ACM Conf. AI Ethics Soc., Feb. 2020, pp. 180–186.

[32] S. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in Proc. NeurIPS, Dec. 2017, pp. 1–10.

[33] S. Lundberg et al., "From local explanations to global understanding with explainable AI for trees," Nature Mach. Intell., vol. 2, pp. 56-67, Jan. 2020.

[34] S. Mangalathu, S.H. Hwang, and J. S. Jeon, "Failure mode and effects analysis of RC members based on machine-learning-based SHapley Additive exPlanations (SHAP) approach," Eng. Struct., vol. 219, 110927, Sep. 2020.

[35] C. Molnar, Interpretable Machine Learning: A Guide for Making Black Box Models Explainable. Online Book, 2021.

[36] X. Huang and J. Marques-Silva, ''On the failings of Shapley values for explainability,'' Int. J. Approx. Reasoning, vol. 171, art. no. 109112, Aug. 2024.

[37] J. Marques-Silva and X. Huang, "Explainability is NOT a game," CoRR, vol. abs/2307.07514, June 2023.

[38] O. Letoffe, X. Huang, and J. Marques-Silva, "Towards trustable SHAP scores," in AAAI, 2025 (accepted).

[39] C. M. Bishop, Neural Networks for Pattern Recognition. NY, USA: Oxford Univ. Press, 1996.

[40] S. Haykin, Neural networks: A Comprehensive Foundation (2nd Edition). Hoboken, N.J. USA: Prentice Hall, 1999.

[41] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: MIT Press, 2016.

[42] S. A. Billings and J. C. Peyton Jones, "Mapping non-linear integro-differential equations into the frequency domain," Int. J. Control, vol. 52,no. 4, pp. 863–879, Oct. 1990.

[43] L. S. Shapley, "A value for n-person games," Contributions to Theory Games, vol. 2, no. 28, pp. 307–317, 1953.

[44] M. Sundararajan and A. Najmi, ''The many Shapley values for model explanation,'' in Proc. Int. Conf. Mach. Learn., pp. 9278–9629, Oct. 2020.