Journal of Big Data

RESEARCH Open Access

A novel sensitivity-based method for feature selection

Dayakar L. Naik and Ravi kiran*

*Correspondence:
ravi.kiran@ndsu.edu
Department of Civil & Environmental Engineering, North Dakota State University, Fargo, ND 58105, USA

Abstract

Sensitivity analysis is a popular feature selection approach employed to identify the important features in a dataset. In sensitivity analysis, each input feature is perturbed one-at-a-time and the response of the machine learning model is examined to determine the feature's rank. Note that the existing perturbation techniques may lead to inaccurate feature ranking due to their sensitivity to perturbation parameters. This study proposes a novel approach that involves the perturbation of input features using a complex-step. The implementation of complex-step perturbation in the framework of deep neural networks as a feature selection method is provided in this paper, and its efficacy in determining important features for real-world datasets is demonstrated. Furthermore, the filter-based feature selection methods are employed, and the results obtained from the proposed method are compared. While the results obtained for the classification task indicated that the proposed method outperformed other feature ranking methods, in the case of the regression task, it was found to perform more or less similar to that of other feature ranking methods.

Keywords: Complex-step perturbation approach (CSPA), Feature Ranking, Regression, Classification, Feature relevance, And neural networks

Introduction

Feature selection is a process of identifying a subset of features that dictate the prediction accuracy of the target variables/class labels in a given machine learning task [1–3]. Identification of relevant features improves the machine learning (ML) models' generalized performance and facilitates a better understanding of the data in relation to the ML model [4]. For performing the task of feature selection, various methods have been proposed by researchers in the past. These methods could be broadly grouped into six categories, namely, filter methods, wrapper methods, embedded methods, hybrid methods, ensemble methods, and integrative methods [5–7]. While filter methods select features based on a performance metric regardless of the supervised learning algorithm [8–12], the wrapper methods choose feature subset by iteratively examining a certain or an ensemble of the ML algorithm's performance for selected features [13]. Examples of filter methods include Pearson correlation coefficient, information gain, gain ratio, Chi-square, Fisher score, ReliefF, etc., and examples of wrapper method include sequential feature selection, genetic algorithms, etc. On the other hand, in embedded methods, the feature selection algorithm is integrated into the learning algorithm [5, 9, 13]. Examples of the embedded method include decision tree, random forest, support vector machine recursive feature elimination (SVM-RFE). When compared to filter-based approaches, the embedded approach yields higher accuracy because of its interaction with a specific classification model. A comprehensive review of these three methods' description and comparison is discussed by various researchers in the literature [4, 5, 14–19].

In hybrid methods, multiple conjunct primary feature selection methods are applied consecutively [6]. For instance, Liu et al. [20] proposed a hybrid feature selection method in which mutual information was first applied to identify the relevant features from the feature set, and then the wrapper method was applied subsequently to choose the subset of best features from the relevant features. Ensemble feature selection methods use an aggregate of feature subsets of diverse base classifiers [6]. For instance, Hoque et al. [21] proposed an Ensemble Feature Selection—Feature Selection (EMI-FS) in which information gain, gain ratio, ReliefF, symmetric uncertainty, and Chi-square were employed as base filter methods to obtain the relevant subset of features which were subsequently combined to extract the optimal subset. In the integrative feature selection method, the external knowledge of feature selection is integrated [6]. For example, Cindy et al. [7], proposed an integrative gene selection approach in which gene rankings are determined by considering both the statistical significance of a gene in the dataset and the biological background information acquired through research. In this paper, we restrict our scope to the embedded feature selection methods that incorporate feed-forward neural networks/multi-layer perceptron as the learning models.

Multi-layer Perceptron (MLP) is a basic type of neural network that learns a function $g : \mathbb{R}^q \rightarrow \mathbb{R}^m$ by training on a dataset, where q is the number of inputs and m is the number of outputs. MLP's were employed for performing feature selection by various researchers in the past. For instance, Setiono and Liu [22] developed a neural network feature selector method based on backward elimination wherein weights of low magnitude in the network were converged to zero by adding a penalty term to the error function. Sindhwani et al. [23] presented a maximum output information algorithm for feature selection. Liefeng Bo [24] proposed MLP Embedded Feature Selection (MLP-EFS), in which each feature is multiplied by the corresponding scaling factor. By applying truncated Laplace prior to the scaling factors, feature selection is integrated into MLP-EFS.

Notwithstanding to methods mentioned above, sensitivity analysis of MLP and support vector machines (SVM) was also carried out to perform feature selection. For instance, Ruck et al. [25] developed a technique that analyzes the weights in MLP to determine essential features. Gasca et al. [26] proposed a saliency measure that estimates the input features' relative contribution to the output neurons. Utans et al. [27] proposed a 'sensitivity-based-pruning (SBP)' to remove irrelevant input features from a nonlinear regression model. Acir et al. [28] implemented the perturbation method in the framework of SVM to perform feature selection for classification of Electrocardiogram (ECG) beats. Sensitivity analysis examines the change in the target output when one of the input features is perturbed, i.e., first-order derivatives of the target variable with respect to the input feature are evaluated. Herein we refer the first-order derivative term as the feature sensitivity metric. The higher the magnitude of change in feature sensitivity metric, the higher is the importance of input feature. At this juncture, it is important to note that sensitivity analysis methods involve computation of the feature sensitivity metric or first-order derivative for identifying important features. In general backpropagation algorithm (for MLP), is employed or finite difference schemes [29–32] is used for computing feature sensitivity metric. Employing numerical differentiation techniques such as finite difference approximation (FDA) (see Eq. 1) and central finite difference approximation (CFDA) (see Eq. 2) results in inaccurate computation of derivatives [33, 34] because of inappropriate choice of step size. For instance, Juana et al. [35] introduced the iterative perturbation method for auto-tuning the step size for SVM. Such errors arising due to the choice of smaller step sizes are referred to as subtractive cancellation errors.

Finite difference approximation (FDA)

$$g'\left(x_1, x_2, \ldots x_k, \ldots x_q\right) \approx \frac{\left(f\left(x_1, x_2, \ldots x_k + h, \ldots x_q\right) - f\left(x_1, x_2, \ldots x_k, \ldots x_q\right)\right)}{h} \tag{1}$$

Central finite difference (CFDA)

$$g'\left(x_1, x_2, \ldots x_k, \ldots x_q\right) \approx \frac{\left(f\left(x_1, x_2, \ldots x_k + h, \ldots x_q\right) - f\left(x_1, x_2, \ldots x_k - h, \ldots x_q\right)\right)}{2h} \tag{2}$$

where $\boldsymbol{x} = \left(x_1, x_2, \ldots x_k, \ldots x_q\right)' \in \mathbb{R}^{q \times 1}$ are the inputs, q is the number of inputs, $g(.)$ is the function mapping the inputs to the output variable and, $g'(.)$ is the first partial derivative approximation of $f(.)$ with respect to the input x_k. The feature x_k is perturbed in both the cases to get the first derivative as seen in Eq. (1) and (2).

In this paper, a novel Complex-step sensitivity analysis-based feature selection method referred to as CS-FS is proposed, which incorporates a complex-step perturbation of the input feature to compute the feature sensitivity metric and identify the important features. It evaluates the analytical quality first-order derivatives without the need for extra computations in neural networks or SVM machine learning models. A brief overview of the complex step perturbation approach is provided in section "Overview of complex-step perturbation approach (CSPA)", and its implementation in the framework of FFNN to perform feature selection is described in section "Complex-step feature selection method". The details of the dataset are provided in section "Numerical experiments" and the efficacy of the proposed method is then demonstrated on real-world datasets in section "Results", and the summary and future work are provided in Section "Summary and future work".

Overview of complex-step perturbation approach (CSPA)

CSPA, originally referred to as complex-step derivative approximation (CSDA), was proposed by Lyness and Moler [36] to evaluate the first-order derivative of analytic functions. A simplified version of mathematical derivation for computing the first-order derivative of a scalar function using complex-step perturbation was then provided by Squire and Trapp [37] which is as follows.

Consider a holomorphic function $f(.)$ which is infinitely differentiable. The Taylor series expansion of the function $f(.)$ evaluated at the complex perturbed point $x_0 + ih$ is expressed as

$$f(x_0 + ih) = f(x_0) + ihf'(x_0) - \frac{h^2}{2!}f''(x_0) - \frac{ih^3}{3!}f'''(x_0) + \dots \tag{3}$$

where, h is the step size and $i^2 = -1$.

By taking the imaginary component of $f(x_0 + ih)$, and truncating the higher-order terms in the Taylor series, the first-order derivative can be expressed as

$$f'(x_0) = \frac{\text{Imag}\left(f(x_0 + ih)\right)}{h} + \mathcal{O}\left(h^2\right) \tag{4}$$

where, Imag (*) denotes the imaginary component and $\mathcal{O}\left(h^2\right)$ is the second-order truncation error. It is evident from Eq. 4 that the first-order derivative evaluated using the CSPA technique is not prone to subtractive cancellation errors (see Eq. 1 and Eq. 2) due to the absence of subtractive operations. Furthermore, a choice of the small magnitude of h could possibly eliminate the truncation error $\mathcal{O}\left(h^2\right)$ too. A simple example illustrating the accuracy of CSPA over finite difference schemes can be found elsewhere [38, 39]. Some examples of the fields where CSPA is currently gaining a lot of attention for performing sensitivity analysis includes aerospace [40–43], computational mechanics [38, 39, 44], estimation theory (e.g., second-order Kalman filter) [45].

Complex-step feature selection method

In the proposed method, we implement a complex-step perturbation in the framework of feed-forward neural networks to illustrate the task of feature selection. Note that this could be extended to other ML models such as SVM whose decision function is holomorphic. Higher the change in the magnitude of the output variable $y \in \mathbb{R}$ of the FFNN with respect to the input feature $x_k \in \mathbb{R}$, higher is the importance of the feature x_k. For a multivariate function, the extended form of CSPA can be expressed as

$$g'\left(x_1, x_2, \dots x_k, \dots x_q\right) = \frac{\text{Imag}\left(g\left(x_1, x_2, \dots x_k + ih, \dots x_q\right)\right)}{h} + \mathcal{O}\left(h^2\right) \tag{5}$$

where $\boldsymbol{x} = \left(x_1, x_2, \dots x_k, \dots x_q\right)' \in \mathbb{R}^{q \times 1}$ is a vector of input features, q is the number of input features, $g(.)$ is the function mapping the input features to the output target variable and, $g'(.)$ is the first-order derivative approximation of $g(.)$ with respect to the k^{th} input feature x_k.

Feature selection for regression using complex-step sensitivity

The proposed feature selection method for the regression task involves four steps (see Fig. 1). In the first step, an FFNN is configured and trained for a given dataset. Configuring the FFNN is a trial-and-error process that involves finding the appropriate number of neurons and hidden layers in a network. A neural network is said to be configured when it is capable of learning a mathematical mapping between the input features and the associated target variable such that it could be generalized to the unseen data instances. In the second step, one of the input features, x_k is chosen at a time and is perturbed with

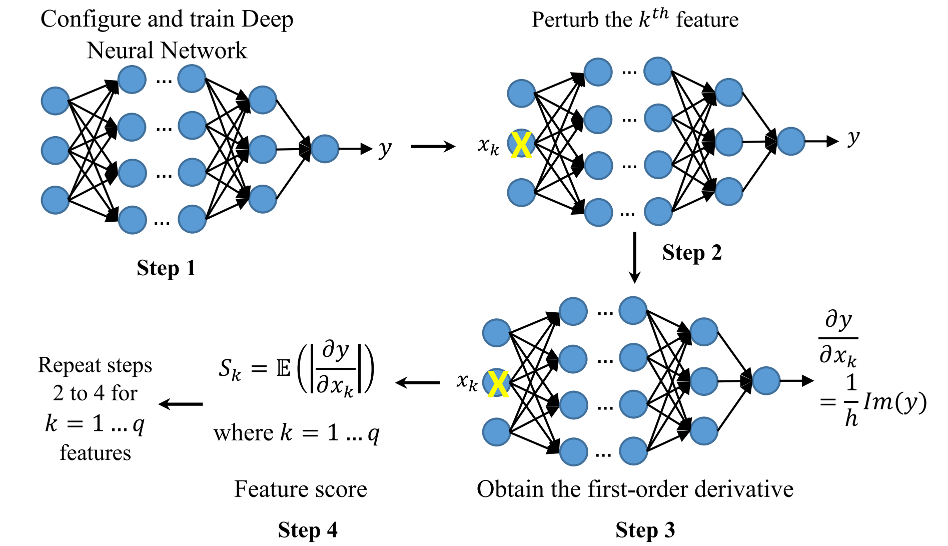
Configure and train Deep Neural Network
y
Step 1
Perturb the k^{th} feature
x_k
y
Step 2
x_k
$\frac{\partial y}{\partial x_k} = \frac{1}{h} Im(y)$
Obtain the first-order derivative
Step 3
$S_k = \mathbb{E}\left(\left|\frac{\partial y}{\partial x_k}\right|\right)$
where $k = 1 \dots q$
Feature score
Step 4
Repeat steps 2 to 4 for $k = 1 \dots q$ features

Fig. 1 Steps involved in the complex-step sensitivity for regression task

an imaginary step size of ih (where$h \ll 10^{-8}$). Feedforward operation is then performed with the perturbed feature on the trained FFNN, and the results in the output layer are obtained. In the third step, the imaginary components of the output neurons' results are extracted for each perturbed feature and are divided with the step size (h) (see Eq. 5), i.e., the first-order derivative of the target output with respect to the input feature is evaluated. Note that step 2 and step 3 are repeated for all instances in the dataset, and the average absolute magnitude of the first-order derivative of the target output with respect to the input feature is evaluated. For example, if y is the target output variable and x_{jk} is the kth feature in the jth observation that is complex-step perturbed (ih), then the first order derivative of the target output with respect to the input feature averaged over all instances of datasets is expressed as (see Eq. 6)

$$\frac{\partial y}{\partial x_k} = \frac{1}{N} \sum_{j=1}^{N} \left| \frac{\partial y}{\partial x_{jk}} \right| \tag{6}$$

where, N denotes the number of instances in the dataset, $k = 1 \dots q$ indicates the input feature, and j represents the observation number in the dataset. In the fourth and final step, the rank of each input feature is determined based on the magnitude of the first-order derivatives evaluated, as shown in Eq. 5. The feature with a higher magnitude of the first-order derivative is assigned a higher rank and vice versa. Note that for training the feedforward neural network, a backpropagation algorithm, in conjunction with the Levenberg–Marquardt optimization technique, is employed in this study [46].

Feature selection for classification using complex-step sensitivity

Unlike regression, a modification to step 3 is needed in the proposed method when feature selection is performed on the classification task, i.e., evaluating the first-order derivative of target output with respect to perturbed input feature. The need for modification could be attributed to two reasons: (1) discrete output in the output layer and (2) multiple first-order derivatives yielded by the feed-forward neural network output layer (SoftMax layer) (see Fig. 2). Considering the fact that the inputs fed to the SoftMax activation neurons in the output layer are not discrete, the first-order derivatives of such inputs could still be evaluated. These first-order derivatives will aid in providing information about the importance of the input features. If Σ_r represents the net function of *r*th neuron in the SoftMax layer, then the first-order derivative of the net function Σ_r with respect to the *k*th feature x_k is expressed as (see Eq. 7)

$$\left(\frac{\partial \Sigma_\mathrm{r}}{\partial x_k}\right) = \frac{1}{h}\mathrm{Imag}(\Sigma_r(x_k + ih)) \tag{7}$$

where, $r = 1 \ldots .m$ and m indicates the number of class labels. To quantify the change in the target output with respect to the *k*th input feature x_k, the average of the first-order derivatives obtained for all neurons in the output layer is determined. This average magnitude is referred to as saliency (S_k) of *k*th input feature [25] and is expressed as (see Eq. 8):

$$S_k = \frac{1}{N}\sum_{j=1}^{N}\left(\sum_{r=1}^{m}\left|\left(\frac{\partial \Sigma_\mathrm{r}}{\partial x_{jk}}\right)\right|\right) \tag{8}$$

where r denotes the neuron in the SoftMax output layer, m represents the number of class labels, Σ_r represents the net function of *r*th neuron in the SoftMax layer. The rank of each input feature is then determined based on the magnitude of the first-order derivatives for each perturbed feature x_k determined as shown in Eq. 8.

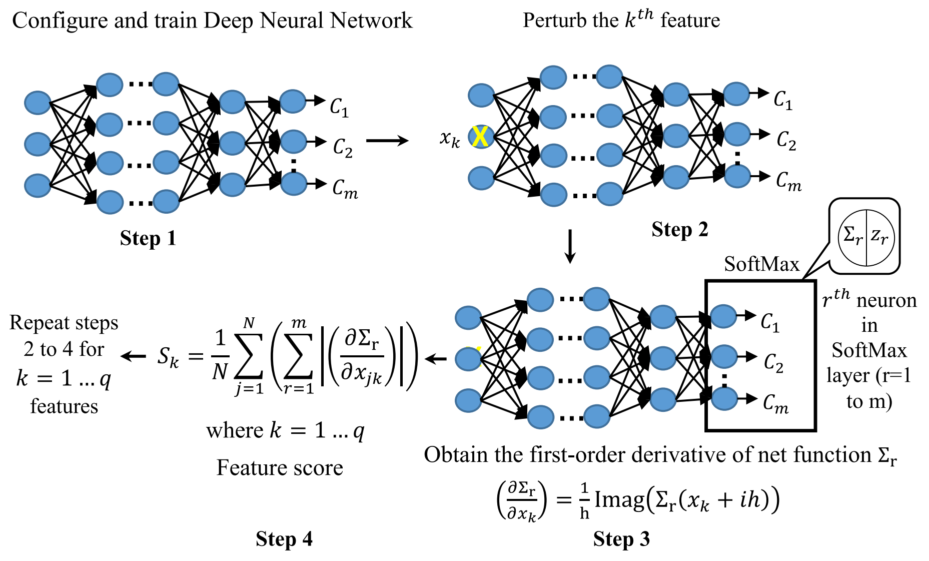
Configure and train Deep Neural Network
Perturb the k^{th} feature
C_1
C_2
C_m
Step 1
x_k
C_1
C_2
C_m
Step 2
SoftMax
Σ_r z_r
r^{th} neuron in SoftMax layer (r=1 to m)
Repeat steps 2 to 4 for $k = 1 \ldots q$ features
$S_k = \frac{1}{N}\sum_{j=1}^{N}\left(\sum_{r=1}^{m}\left|\left(\frac{\partial \Sigma_\mathrm{r}}{\partial x_{jk}}\right)\right|\right)$
where $k = 1 \ldots q$
Feature score
C_1
C_2
C_m
Obtain the first-order derivative of net function Σ_r
$\left(\frac{\partial \Sigma_\mathrm{r}}{\partial x_k}\right) = \frac{1}{\mathrm{h}}\mathrm{Imag}\left(\Sigma_\mathrm{r}(x_k + ih)\right)$
Step 4
Step 3

Fig. 2 Steps involved in the complex-step sensitivity for the classification task

Numerical experiments

In this section, numerical experiments are performed to demonstrate the effectiveness of the proposed method.

Datasets

Three real-world datasets, each for regression and classification problems, are employed to demonstrate the proposed method's efficacy. The datasets are obtained from the UCI open-source data repository [47]. For regression problems, the body fat percentage dataset, abalone dataset, and wine quality dataset are chosen, and, for the classification task, a vehicle dataset, segmentation dataset, and breast cancer dataset are chosen. One of the main reasons for choosing these datasets is that they are commonly adopted in the literature of feature selection. On the other hand, the results obtained from some of the chosen datasets such as body fat percentage, wine quality, segmentation are easily interpretable and aids in ensuring the verification of the proposed method. While most of the chosen datasets have descriptive features that are continuous in nature, the proposed method can be extended to the datasets consisting of discrete input features. The descriptive features and target variables for each dataset are mentioned as follows.

Regression

Body fat percentage dataset [48]: Features—(1) Age (years), (2) Weight (kg), (3) Height (cm), (4) Neck (cm), (5) Chest (cm), (6) Abdomen (cm), (7) Hip (cm), (8) Thigh (cm), (9) Knee (cm), (10) Ankle (cm), (11) Biceps (cm), (12) Forearm (cm), (13) Wrist (cm); Target variable—percentage of body fat.

Abalone dataset [49]: Features—(1) Female, (2) Infant, (3) Male, (4) Length (gms.), (5) Diameter (gms.), (6) Height (gms.), (7) Whole weight (gms.), (8) Shucked weight (gms.), (9) Viscera weight (gms.), (10) Shell weight (gms.); Target variable—Number of rings.

Wine quality dataset [50]: Features—(1) fixed acidity, (2) volatile acidity, (3) citric acid, (4) residual sugar, (5) chlorides, (6) free sulfur dioxide, (7) total sulfur dioxide, (8) density, (9) pH, (10) sulfates, (11) alcohol; Target variable – quality score (1 to 10).

Classification

Vehicle dataset [51]: Features—(1) Compactness, (2) circularity, (3) radius circularity, (4) radius ratio, (5) axis aspect ratio, (6) maximum length aspect ratio, (7) scatter ratio, (8) elongatedness, (9) axis rectangularity, (10) maximum length rectangularity, (11) scaled variance major, (12) scaled variance minor, (13) scaled radius of gyration, (14) skewness major, (15) skewness minor, (16) kurtosis major, (17) kurtosis minor, (18) hollow ratio; Target variable—Class label 1 (van), Class label 2 (Saab), Class label 3 (bus), Class label 4 (Opel).

Segmentation dataset [47]: Features—(1) region-centroid-col (2) region-centroid-row (3) short-line-density (4) the results of a line extraction algorithm that counts how many lines of length (5) vedge-mean (6) vedge-sd (7) hedge-mean (8) hedge-sd (9) intensity-mean (10) rawred-mean (11) rawblue-mean (12) rawgreen-mean (13)

Table 1 Description of the datasets used for regression task

Dataset name	Instances	No. of features	No. of target variables
Bodyfat	252	13	1
Abalone	4177	10	1
Wine quality	4898	11	1

Table 2 Description of the datasets used for the classification task

Dataset name	Instances	No. of features	No. of class labels
Vehicle	846	30	4
Segmentation	210	18	7
Breast cancer	569	18	2

exred-mean (14) exblue-mean (15) exgreen-mean (16) value-mean (17) saturatoin-mean (18) hue-mean; Target variable—Class label 1 (Window), Class label 2 (foilage), Class label 3 (brickface), Class label 4 (path), Class label 5 (cement), Class label 6 (grass), Class label 7 (sky).

Breast cancer dataset [52]: Features—(1) radius1, (2) texture1, (3) perimeter1, (4) area1, (5) smoothness1, (6) compactness1, (7) concavity1, (8) concave points1, (9) symmetry1, (10) fractal dimension1, (11) radius2, (12) texture2, (13) perimeter2, (14) area2, (15) smoothness2, (16) compactness2, (17) concavity2, (18) concave points2, (19) symmetry2, (20) fractal dimension2, (21) radius3, (22) texture3, (23) perimeter3, (24) area3, (25) smoothness3, (26) compactness3, (27) concavity3, (28) concave points3, (29) symmetry3, (30) fractal dimension3; Target variable—Class label 1 (Benign), Class label 2 (Malignant).

Other details about regression and classification datasets are provided in Table 1 and Table 2, respectively.

Configuring feed-forward neural networks

Feed-forward neural networks (FFNN) with three hidden layers (HL) are configured to train on the regression and classification datasets. While a configuration of 1st HL—20 neurons, 2nd HL—10 neurons, and 3rd HL—5 neurons is employed to train on regression datasets, a configuration of 1st HL—60 neurons, 2nd HL—40 neurons, and 3rd HL—20 neurons is employed to train on classification datasets. A Rectified Linear Unit (ReLU) nonlinear function is used as an activation function for all the configurations [53]. Note that different architectures and model parameters yield different results if a suitable configuration is not adopted. In this study, various trail configurations of increased complexity (i.e., more hidden neurons and hidden layers) were examined before choosing a suitable configuration. Herein, the suitable configuration refers to the model architecture for which further improvement in performance was not observed with an increase in complexity of architecture. For training, validating, and testing the chosen configurations, the datasets are randomly partitioned into 70:15:15 ratio,

Table 3 Important features identified by various feature selection methods for regression task (ranked in the descending order of their importance)

Bodyfat dataset				**Abalone dataset**				**Wine quality dataset**			
Corr	**ReliefF**	**MI**	**CS-FS**	**Corr**	**ReliefF**	**MI**	**CS-FS**	**Corr**	**ReliefF**	**MI**	**CS-FS**
6	6	6	6	10	10	10	7	11	2	8	11
5	5	5	3	5	7	5	8	9	11	11	4
7	7	7	13	6	8	7	6	10	6	4	6
2	2	2	4	4	9	6	10	6	9	7	2
8	8	8	8	7	5	9	9	3	7	5	7
9	9	9	2	9	6	4	4	4	1	6	5
1	1	11	1	8	4	8	5	1	10	3	1
11	11	4	7	1	2	2	2	7	8	2	9
3	3	1	5	3	3	1	3	2	3	9	3
4	4	13	12	2	1	3	1	5	4	1	8
10	10	12	11					8	5	10	10
13	13	10	10								
12	12	3	9								

respectively. Note that in the case of the classification task, the partition ratio is maintained consistently for each class label, i.e., 70:15:15 of training, validation, and testing data from each class label is chosen. To ensure that the chosen configurations yield repeatable results, the training operation is performed 100 times with the same partition ratio but with the replacement of instances randomly selected in every iteration. The performance metric, namely mean squared error (MSE) and accuracy, are evaluated for regression and classification datasets, respectively, for chosen configurations. The average MSE error for body fat percentage, abalone, and wine quality datasets is determined to be 20.41, 4.6, and 0.53, respectively. The average accuracy for the vehicle, segmentation, and breast cancer dataset is determined to be 75%, 80% and, 90%, respectively. The addition of more hidden layers or neurons in each hidden layer to the chosen configuration was found to yield similar MSE errors or accuracies and hence are not considered in this study.

Results

Followed by the determination of FFNN configuration, the rank of the features in each dataset is evaluated using the proposed method. Furthermore, other feature ranking methods are also considered in this study for the sake of comparison. An open-source software WEKA is employed for this purpose. While feature ranking methods such as Pearson correlation coefficient, ReliefF and, mutual information are used for regression task, symmetric uncertainty, information gain, gain ratio, reliefF and, chi-square is employed for the classification task. The efficacy of all feature ranking methods is then assessed by evaluating the performance of FFNN, wherein the size of the input layer is increased by one feature in each succession. In other words, the performance of FFNN for the only top-most feature is first assessed, and then the process is repeated by including the second most important feature and so on.

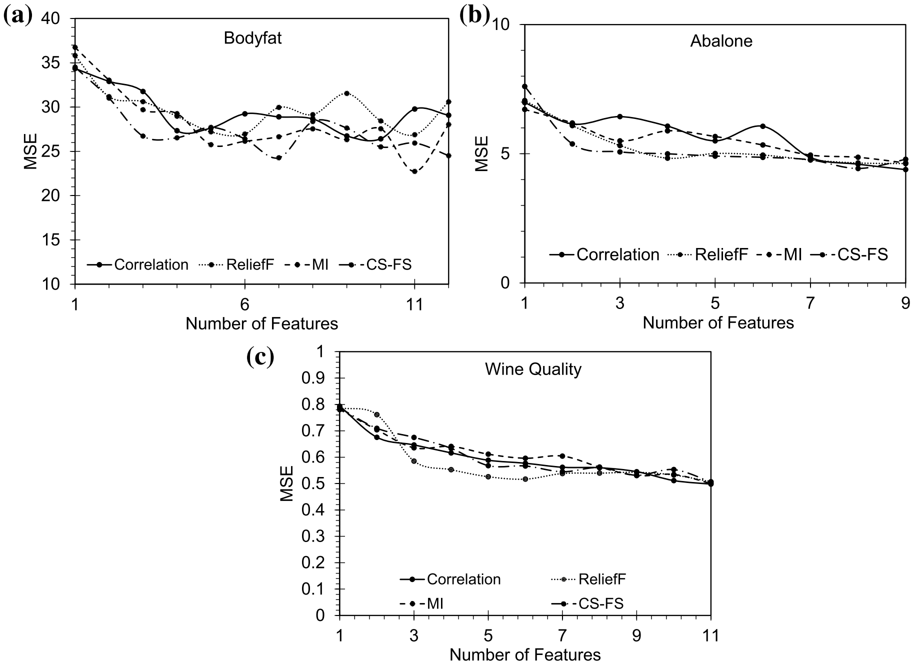

Fig. 3 Comparison of the complex-step sensitivity method with other feature selection methods for regression task

Regression

From Table 3, it can be inferred that all four feature ranking methods yielded feature 6 (Abdomen) as the most important feature and feature 10 (Ankle) as the least relevant feature for determining the percentage of body fat. While the top six features determined using Pearson correlation coefficient, ReliefF and, mutual information method are noticed to be similar; the proposed method yielded different feature ranks. Furthermore, the MSE for body fat dataset with each feature's inclusion is evaluated for all four feature ranking methods and is shown in Fig. 3a. From Fig. 3a, it is evident that the overall trend of MSE for FFNN decreases with the inclusion of each feature. While the proposed method was found to yield lower MSE with only seven top-most features, the mutual information method yielded lower MSE for eleven features for the bodyfat dataset. In other words, the filter based approach was found to be ineffective at determining a subset of important features that could reduce the MSE. According to the proposed method, following features are found to be least important as they do not contribute further for reduction of MSE: (5) Chest (cm), (7) Hip (cm), (9) Knee (cm), (10) Ankle (cm), (11) Biceps (cm), (12) Forearm (cm).

In the case of the abalone dataset, the least relevant features are determined to be the same by all four feature ranking methods, i.e., feature 1 (female), feature 2 (infant), and feature 3 (male) are identified to be the least relevant (see Table 3). While the remaining seven features' rank was found to vary, feature 10 (shell weight) and feature 7 (whole weight) were common in the top four features for all feature

Table 4 Important features identified by various feature selection methods for classification task (ranked in the descending order of their importance)

Method	Feature ranking
Vehicle dataset	
ReliefF	8, 7, 12, 9, 3, 11, 18, 4, 2, 1, 13, 10, 16, 14, 17, 6, 15, 5
Symmetric uncertainty	12, 7, 8, 11, 9, 6, 3, 4, 1, 13, 2, 14, 10, 17, 18, 5, 16, 15
Info gain	12, 7, 8, 11, 9, 3, 6, 2, 1, 4, 13, 10, 14, 17, 18, 5, 16, 15
Gain ratio	11, 9, 12, 7, 4, 8, 6, 3, 5, 18, 13, 14, 1, 2, 16, 10, 15, 17
Chi-squared	12, 7, 8, 9, 11, 3, 6, 1, 2, 10, 14, 13, 4, 17, 18, 5, 16, 15
CS-FS	10, 8, 5, 17, 14, 18, 11, 3, 6, 12, 7, 1, 9, 4, 13, 2, 15, 16
Segmentation dataset	
ReliefF	11, 16, 18, 9, 12, 10, 2, 15, 14, 13, 17, 1, 5, 7, 3, 4, 6, 8
Symmetric uncertainty	18, 10, 9, 16, 12, 11, 15, 17, 2, 14, 13, 7, 8, 5, 6, 3, 4, 1
Info gain	18, 9, 12, 16, 10, 11, 15, 17, 13, 14, 2, 7, 8, 5, 6, 3, 4, 1
Gain ratio	10, 11, 9, 16, 18, 2, 12, 14, 15, 17, 13, 8, 7, 5, 6, 3, 4, 1
Chi-squared	18, 12, 9, 16, 10, 11, 13, 15, 17, 14, 2, 7, 8, 5, 6, 3, 4, 1
CS-FS	2, 18, 15, 13, 10, 16, 11, 12, 17, 9, 14, 6, 8, 7, 5, 4, 3, 1
Breast cancer dataset	
ReliefF	28, 8, 21, 23, 3, 1, 7, 24, 4, 27, 26, 6, 22, 25, 11, 2, 14, 13, 29, 30, 10, 18, 5, 16, 9, 17, 19, 15, 12, 20
Symmetric uncertainty	23, 21, 24, 28, 8, 3, 7, 4, 1, 27, 14, 11, 13, 6, 26, 17, 2, 18, 22, 25, 29, 16, 5, 30, 9, 19, 20, 10, 12, 15
Info gain	23, 24, 21, 28, 8, 3, 4, 1, 7, 14, 27, 11, 13, 26, 6, 17, 18, 22, 2, 29, 16, 25, 9, 5, 30, 20, 19, 10, 12, 15
Gain ratio	23, 21, 24, 28, 8, 7, 27, 3, 4, 1, 14, 6, 11, 13, 26, 17, 2, 19, 18, 25, 22, 29, 5, 16, 30, 9, 20, 12, 10, 15
Chi-squared	23, 21, 24, 28, 8, 3, 4, 1, 7, 14, 27, 11, 13, 26, 6, 17, 18, 22, 2, 29, 25, 16, 9, 5, 30, 20, 19, 10, 12, 15
CS-FS	21, 23, 28, 20, 8, 4, 7, 11, 24, 17, 15, 2, 22, 30, 12, 26, 13, 16, 1, 14, 10, 9, 29, 25, 18, 19, 6, 3, 27, 5

ranking methods, including the proposed method. Similar to the body fat dataset, the MSE of FFNN with the inclusion of each feature is determined for all feature ranking methods and is shown in Fig. 3b. From Fig. 3b, it can be inferred that the trend of ReliefF and the proposed method are similar. Both ReliefF and the proposed method identified feature 5 (diameter), feature 6 (height), feature 7 (whole weight), and feature 10 (shell weight) as the top 4 features that yield the lowest MSE. In other words, ReliefF was found to be effective among all the filter-based methods. According to the proposed method, following features are found to be least important as they do not contribute further for reduction of MSE: (1) Female, (2) Infant, (3) Male, (4) Length (gms.), (8) Shucked weight (gms.), (9) Viscera weight (gms.).

Interestingly, in the wine quality dataset, all four feature ranking methods yielded different ranks for the features (see Table 3). However, feature 11 (alcohol) is determined to be one of the top two features by all four feature ranking methods. Furthermore, feature 6 (free sulfurdioxide) is determined to be common among first four features determined by all feature ranking methods except mutual information. The MSE of FFNN with each feature's inclusion is determined for all feature ranking methods and is shown in Fig. 3c. The trend obtained in Fig. 3c, reveals that all feature ranking methods performed more or less similar.

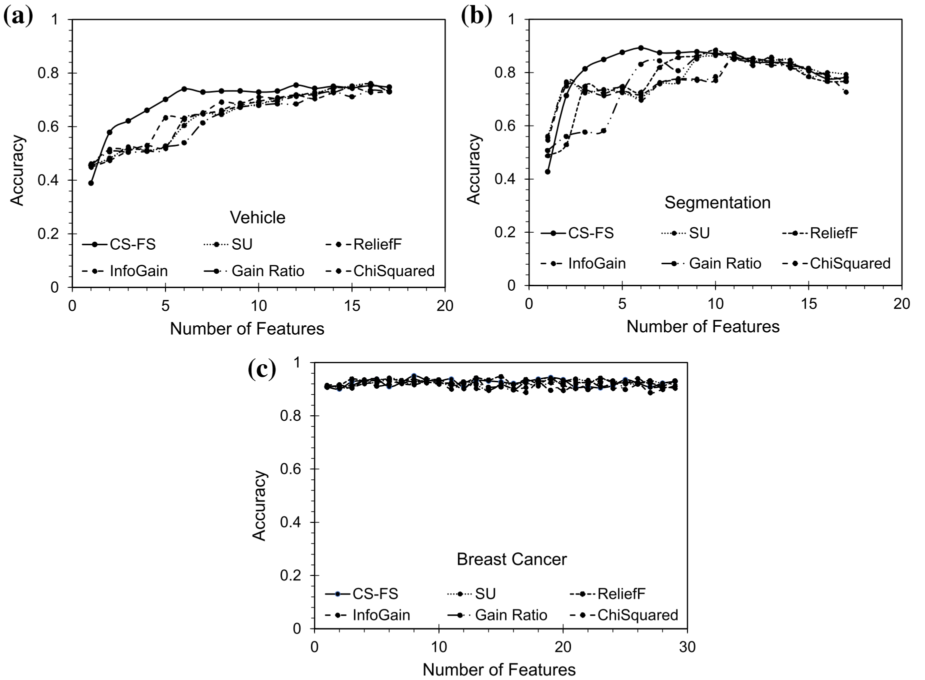

Fig. 4 Comparison of the complex-step sensitivity method with other feature selection methods for the classification task

Classification

From Table 4, it can be inferred that all feature ranking methods employed for the classification task identified similar least relevant features for the vehicle dataset (feature 15 (skewness minor), feature 16 (kurtosis major)). However, the rank of the remaining features was found to vary. While feature 12 (scaled variance minor), feature 7 (scatter ratio) and feature 8 (elongatedness) was found to be the top three features for symmetric uncertainty, information gain, gain ratio, reliefF and, chi-square, feature 10 (maximum length rectangularity), feature 8 (elongatedness) and feature 5 (axis aspect ratio) was found to be the top 3 features for the proposed method, i.e., feature 8 (elongatedness) was found to be common among top 3 features predicted by all feature ranking methods. Furthermore, the trend of the accuracy is determined for vehicle dataset for all feature ranking methods with the inclusion of each feature in succession and is shown in Fig. 4a. From Fig. 4a, it is evident that the accuracy of the FFNN increases with the addition of each feature for the vehicle dataset. The proposed method yielded an accuracy of 75% by selecting only the top 6 features and was found to outperform the other feature ranking methods. The top 6 features are identified as follows: (5) axis aspect ratio, (8) elongatedness, (10) maximum length rectangularity, (14) skewness major, (17) kurtosis minor and (18) hollow ratio.

Similar to the vehicle dataset, all feature ranking methods employed in the case of the segmentation dataset obtained the same least relevant features (feature 1 (region-centroid-col), feature 3 (short-line density), feature 4 (lines of length), feature 6 (vedge-sd), and feature 8 (hedge-sd)). While the rank of the top features was found to vary for all feature ranking methods, feature 10 (rawred-mean), feature 16 (value-mean), and feature 18 (hue-mean) were found to be common among the top-most 6 features. The trend of the accuracy for the segmentation dataset is determined for all feature ranking methods with the inclusion of each feature in succession and is shown in Fig. 4b. From Fig. 4b, it is evident that the accuracy of the FFNN increases with the addition of each feature for the segmentation dataset. Among all the feature ranking methods, the proposed method was found to outperform yielding the highest accuracy of 90% with only the top 6 features. In other words, the filter based methods suggested top 10 features are important for achieving an accuracy of 85%.

Interestingly, in the breast cancer dataset, all feature ranking methods resulted in similar top-most features, i.e., feature 21 (radius3) and feature 23 (perimeter3). While symmetric uncertainty, information gain, gain ratio, reliefF and, chi-square identified feature 10 (fractal dimension1), feature 12 (texture2), and feature 15 (smoothness2) as least relevant, the proposed method identified the feature 3 (perimeter1), feature 5 (smoothness1) and feature 27 (concavity3) are least relevant. Similar to the vehicle and segmentation dataset, the trend of accuracy is obtained for the breast cancer dataset with the inclusion of each feature in each succession and is shown in Fig. 4c In the case of the breast cancer dataset, the trend of all feature ranking methods was found to be more or less similar. An accuracy of 93% is achieved by the inclusion of the top two features, i.e., feature 21 (radius3) and feature 23 (perimeter3).

Summary and future work

A novel complex-step sensitivity analysis-based feature selection method is proposed in this study for regression and classification tasks. A step-by-step process involved in implementing the proposed method in the framework of FFNN is described, and its efficacy on real-world datasets is demonstrated. Three real-world datasets, namely, body fat percentage dataset, abalone dataset, and wine quality dataset, are chosen for the regression task and, three datasets, namely vehicle dataset, segmentation dataset, and breast cancer dataset, are chosen for the classification task. While the proposed method was found to outperform other popular feature ranking methods for classification datasets (vehicle, segmentation, and breast cancer), it was found to perform more or less similar with other methods in the case of regression datasets (body fat, abalone, and wine quality). An average MSE of 20.41, 4.6, and 0.53 were observed for body fat, abalone, and wine quality datasets, respectively, and an average accuracy of 75%, 80%, and 90% was observed for the vehicle segmentation and breast cancer datasets, respectively. Furthermore, the top-most relevant features and irrelevant features are identified for all the employed datasets. At this juncture, it is also important to note that the proposed method possesses the advantage of performing sensitivity analysis through the forward propagation of FFNN, i.e., no backpropagation is required for evaluating the derivatives.

In future work, the authors intend to extend the proposed method to the multiple output regression problems. In addition to this, the authors would also like to investigate the influence of different activation functions (e.g., Sigmoid, tanh, Softplus, Leaky ReLU, etc.). Other supervised ML classification algorithms will be employed, and the efficacy of the proposed method will be examined. Note that often complete dataset may not be required for training the FFNN when the size of the dataset is large. Hence the influence of a number of instances on the determination of the important features would also be studied. Furthermore, the proposed method would also be extended to the datasets that consists of discrete and continuous features and also include redundant features.

Acknowledgements
Research presented in this paper was supported by the National Science Foundation under NSF EPSCoR Track-1 Cooperative Agreement OIA #1946202. Any opinions, findings, and conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Authors' contributions
RK: Conception, design of work, interpretation of results, revising the manuscript, and acquiring funding. DLN: execution, data generation, coding, first draft preparation, interpretation of results, and revision of manuscript. Both authors have read and approved the final manuscript.

Funding
National Science Foundation under NSF EPSCoR Track-1 Cooperative Agreement OIA #1946202.

Availability of data and materials
All the datasets employed in this study are obtained from UCI open-source data repositories [48–52].

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Received: 16 May 2021 Accepted: 12 September 2021
Published online: 09 October 2021

References

1. Dash M, Liu H. Feature Selection for Classification. Intell Data Anal. 1997. https://doi.org/10.3233/IDA-1997-1302.
2. Blum AL, Langley P. Selection of relevant features and examples in machine learning. Artif Intell. 1997;97(1–2):245–71.
3. Sánchez-Maroño N, Alonso-Betanzos A, Tombilla-Sanromán M. Filter methods for feature selection—a comparative study. Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics). Vol 4881, LNCS, Springer: Berlin; 2007. pp. 178–87. https://doi.org/10.1007/978-3-540-77226-2_19.
4. Chandrashekar G, Sahin F. A survey on feature selection methods q. Comput Electr Eng. 2014;40:16–28. https://doi.org/10.1016/j.compeleceng.2013.11.024.
5. Bolón-Canedo V, Sánchez-Maroño N, Alonso-Betanzos A. A review of feature selection methods on synthetic data. Knowl Inf Syst. 2013;34:483–519. https://doi.org/10.1007/s10115-012-0487-8.
6. Tadist K, Nikolov NS, Mrabti F, Zahi A. Feature selection methods and genomic big data: a systematic review. J Big Data. 2019. https://doi.org/10.1186/s40537-019-0241-0.
7. Perscheid C, Grasnick B, Uflacker M. Integrative gene selection on gene expression data: providing biological context to traditional approaches. J Integr Bioinform. 2018. https://doi.org/10.1515/jib-2018-0064.
8. Jović A, Brkić K, Bogunović N. A review of feature selection methods with applications. In: 2015 38th Int Conv Inf Commun Technol Electron Microelectron MIPRO 2015—Proc, Institute of Electrical and Electronics Engineers Inc. 2015; pp. 1200–5. https://doi.org/10.1109/MIPRO.2015.7160458.
9. Asir D, Appavu S, Jebamalar E. Literature review on feature selection methods for high-dimensional data. Int J Comput Appl. 2016. https://doi.org/10.5120/ijca2016908317.
10. Naik DL, Sajid HU, Kiran R. Texture-based metallurgical phase identification in structural steels: a supervised machine learning approach. Metals. 2019;9:546. https://doi.org/10.3390/met9050546.
11. Naik DL, Kiran R. Naïve Bayes classifier, multivariate linear regression and experimental testing for classification and characterization of wheat straw based on mechanical properties. Ind Crops Prod. 2018. https://doi.org/10.1016/j.indcrop.2017.12.034.
12. Naik DL, Kiran R. Identification and characterization of fracture in metals using machine learning based texture recognition algorithms. Eng Fract Mech. 2019. https://doi.org/10.1016/j.engfracmech.2019.106618.
13. Dong NT, Winkler L, Khosla M. Revisiting feature selection with data complexity for biomedicine. bioRxiv. 2019. https://doi.org/10.1101/754630.
14. Hua J, Tembe WD, Dougherty ER. Performance of feature-selection methods in the classification of high-dimension data. Pattern Recognit. 2009;42:409–24. https://doi.org/10.1016/j.patcog.2008.08.001.

15. Cilia N, De Stefano C, Fontanella F, Raimondo S, di Freca AS. An experimental comparison of feature-selection and classification methods for microarray datasets. Information. 2019;10:109. https://doi.org/10.3390/info10030109.
16. Hira ZM, Gillies DF. A review of feature selection and feature extraction methods applied on microarray data. Adv Bioinformatics. 2015. https://doi.org/10.1155/2015/198363.
17. Remeseiro B, Bolon-Canedo V. A review of feature selection methods in medical applications. Comput Biol Med. 2019;112: 103375. https://doi.org/10.1016/j.compbiomed.2019.103375.
18. Garrett D, Peterson DA, Anderson CW, Thaut MH. Comparison of linear, nonlinear, and feature selection methods for EEG signal classification. IEEE Trans Neural Syst Rehabil Eng. 2003;11:141–4. https://doi.org/10.1109/TNSRE.2003.814441.
19. Refaeilzadeh P, Tang L, Liu H. On comparison of feature selection algorithms. In: Proceedings of AAAI workshop on evaluation methods for machine learning II, vol. 3, no. 4; 2007.
20. Liu J, Wang G. A hybrid feature selection method for data sets of thousands of variables. In Proc—2nd IEEE Int Conf Adv Comput Control. ICACC 2010. 2010;2:288–91. https://doi.org/10.1109/ICACC.2010.5486671.
21. Hoque N, Singh M, Bhattacharyya DK. EFS-MI: an ensemble feature selection method for classification. Complex Intell Syst. 2018;4:105–18. https://doi.org/10.1007/s40747-017-0060-x.
22. Setiono R, Liu H. Neural-network feature selector. IEEE Trans Neural Networks. 1997;8:654–62. https://doi.org/10.1109/72.572104.
23. Sindhwani V, Rakshit S, Deodhare D, Erdogmus D, Principe JC, Niyogi P. Feature selection in MLPs and SVMs based on maximum output information. IEEE Trans Neural Networks. 2004;15:937–48. https://doi.org/10.1109/TNN.2004.828772.
24. Bo L, Wang L, Jiao L. Multi-layer perceptrons with embedded feature selection with application in cancer classification. Chinese J Electron. 2006;15:832–5.
25. Ruck DW, Rogers SK, Kabrisky M. Feature selection using a multilayer perceptron. J Neural Network Comput. 1990;2(2):40–8.
26. Gasca E, Sánchez JS, Alonso R. Eliminating redundancy and irrelevance using a new MLP-based feature selection method. Pattern Recognit. 2006;39:313–5. https://doi.org/10.1016/j.patcog.2005.09.002.
27. Utans J, Moody J, Rehfuss S, Siegelmann H. Input variable selection for neural networks: application to predicting the U.S. business cycle. IEEE/IAFE Conf Comput Intell Financ Eng Proc, IEEE. 1995; pp. 118–22. https://doi.org/10.1109/cifer.1995.495263.
28. Acir N. A support vector machine classifier algorithm based on a perturbation method and its application to ECG beat recognition systems. Expert Syst Appl. 2006;31:150–8. https://doi.org/10.1016/J.ESWA.2005.09.013.
29. Montaño JJ, Palmer A. Numeric sensitivity analysis applied to feedforward neural networks. Neural Comput Appl. 2003;12:119–25. https://doi.org/10.1007/s00521-003-0377-9.
30. Güne¸ A, Baydin G, Pearlmutter BA, Siskind JM. Automatic differentiation in machine learning: a survey. 2018.
31. Jerrell ME. Automatic differentiation and interval arithmetic for estimation of disequilibrium models. Comput Econ. 1997;10:295–316. https://doi.org/10.1023/A:1008633613243.
32. Hashem S. Sensitivity analysis for feedforward artificial neural networks with differentiable activation functions, Institute of Electrical and Electronics Engineers (IEEE); 2003; pp. 419–24. https://doi.org/10.1109/ijcnn.1992.287175.
33. Driscoll TA, Braun RJ. Fundamentals of numerical computation, vol. 154. SIAM; 2017.
34. Boudjemaa R, Cox MG, Forbes AB, Harris PM. Report to the National Measurement Directorate, Department of Trade and Industry From the Software Support for Metrology Programme Automatic Differentiation Techniques and their Application in Metrology. 2003.
35. Canul-Reich J, Hall LO, Goldgof DB, Korecki JN, Eschrich S. Iterative feature perturbation as a gene selector for microarray data. Int J Patt Recogn Artif Intell. 2012. https://doi.org/10.1142/S0218001412600038.
36. Lyness JN, Moler CB. Numerical differentiation of analytic functions. SIAM J Numer Anal. 1967;4:202–10. https://doi.org/10.1137/0704019.
37. Squire W, Trapp G. Using complex variables to estimate derivatives of real functions. SIAM Rev. 1998;40:110–2. https://doi.org/10.1137/S003614459631241X.
38. Kiran R, Khandelwal K. Complex step derivative approximation for numerical evaluation of tangent moduli. Comput Struct. 2014;140:1–13. https://doi.org/10.1016/j.compstruc.2014.04.009.
39. Kiran R, Khandelwal K. Automatic implementation of finite strain anisotropic hyperelastic models using hyper-dual numbers. Comput Mech. 2015;55:229–48. https://doi.org/10.1007/s00466-014-1094-1.
40. Martins J, Sturdza P, Alonso J, R A Martins JR, Alonso JJ. . The complex-step derivative approximation. ACM Trans Math Softw. 2003;29:245–62. https://doi.org/10.1145/838250.838251ï.
41. Conolly J, Lake M. Geographical information systems in archaeology. Cambrigde: Cambridge University Press; 2006.
42. Zhu J-J, et al. Quantum dots for DNA biosensing. Berlin, Heidelberg: Springer; 2013.
43. Campbell AR. Numerical analysis of complex-step differentiation in spacecraft trajectory optimization problems. Doctoral Dissertation; 2011.
44. Kiran R, Li L, Khandelwal K. Complex perturbation method for sensitivity analysis of nonlinear trusses. J Struct Eng. 2017;143:04016154. https://doi.org/10.1061/(asce)st.1943-541x.0001619.
45. Lai KL, Crassidis JL, Cheng Y, Kim J. New complex-step derivative approximations with application to second-order Kalman filtering. Collect Tech Pap—AIAA Guid Navig Control Conf. 2005;2:982–98. https://doi.org/10.2514/6.2005-5944.
46. Christopher MB. Neural networks for pattern recognition. New york: Oxford University Press; 1995.
47. UCI Machine Learning Repository 2021. https://archive.ics.uci.edu/ml/index.php. Accessed 7 Apr s2021.
48. Johnson RW. Fitting percentage of body fat to simple body measurements. J Stat Educ. 1996. https://doi.org/10.1080/10691898.1996.11910505.
49. Nash WJ. 7he population biology of abalone (_Haliotis_ species) in Tasmania. I. Blacklip abalone (_H. rubra_) from the North Coast and Islands of Bass Strait. 1994.
50. Cortez P, Cerdeira A, Almeida F, Matos T, Reis J. Modeling wine preferences by data mining from physicochemical properties. Decis Support Syst. 2009;47:547–53. https://doi.org/10.1016/j.dss.2009.05.016.

51. Siebert JP. Vehicle recognition using rule based methods. Turing Institute Research Memorandum TIRM-87-0.18; 1987.
52. Mangasarian OL, Street WN, Wolberg WH. Breast cancer diagnosis and prognosis via linear programming. Oper Res. 1995;43:570–7. https://doi.org/10.1287/opre.43.4.570.
53. Ding B, Qian H, Zhou J. Activation functions and their characteristics in deep neural networks. In Proc 30th Chinese Control Decis Conf CCDC 2018, Institute of Electrical and Electronics Engineers Inc. 2018; pp. 1836–41. https://doi.org/10.1109/CCDC.2018.8407425.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.