HHS Public Access

Author manuscript

IEEE Access. Author manuscript; available in PMC 2022 April 22.

Published in final edited form as:

IEEE Access. 2022 ; 10: 36166–36176. doi:10.1109/access.2022.3163257.

LINA: A Linearizing Neural Network Architecture for Accurate First-Order and Second-Order Interpretations

ADRIEN BADRÉ, **CHONGLE PAN**

School of Computer Science, The University of Oklahoma, Norman, OK 73019, USA

Abstract

While neural networks can provide high predictive performance, it was a challenge to identify the salient features and important feature interactions used for their predictions. This represented a key hurdle for deploying neural networks in many biomedical applications that require interpretability, including predictive genomics. In this paper, linearizing neural network architecture (LINA) was developed here to provide both the first-order and the second-order interpretations on both the instance-wise and the model-wise levels. LINA combines the representational capacity of a deep inner attention neural network with a linearized intermediate representation for model interpretation. In comparison with DeepLIFT, LIME, Grad*Input and L2X, the first-order interpretation of LINA had better Spearman correlation with the ground-truth importance rankings of features in synthetic datasets. In comparison with NID and GEH, the second-order interpretation results from LINA achieved better precision for identification of the ground-truth feature interactions in synthetic datasets. These algorithms were further benchmarked using predictive genomics as a real-world application. LINA identified larger numbers of important single nucleotide polymorphisms (SNPs) and salient SNP interactions than the other algorithms at given false discovery rates. The results showed accurate and versatile model interpretation using LINA.

INDEX TERMS

Interpretable machine learning; predictive genomics; bioinformatics; deep neural networks

I. INTRODUCTION

An interpretable machine learning algorithm should have a high representational capacity to provide strong predictive performance, and its learned representations should be amenable to model interpretation and understandable to humans. The two desiderata are generally difficult to balance. Linear models and decision trees generate simple representations for model interpretation, but have low representational capacities for only simple prediction

Corresponding author: Chongle Pan (cpan@ou.edu).

SUPPLEMENTARY MATERIAL

Supplementary Material available under the tab “Media.” The source code is freely available under the MIT license at https://github.com/thepanlab/LINA.

tasks. Neural networks and support vector machines have high representational capacities to handle complex prediction tasks, but their learned representations are often considered to be "black-boxes" for model interpretation [1].

Predictive genomics is an exemplar application that requires both a strong predictive performance and high interpretability. In this application, the genotype information for a large number of SNPs in a subject's genome is used to predict the phenotype of this subject. While neural networks have been shown to provide better predictive performance than statistical models [2], [3], statistical models are still the dominant methods for predictive genomics, because geneticists and genetic counselors can understand which SNPs are used and how they are used as the basis for certain phenotype predictions. Neural network model have also been used in many other important bioinformatics applications [4]–[6] that can benefit from model interpretation. To make neural networks more useful for predictive genomics and other applications, we developed a new neural network architecture, referred to as linearizing neural network architecture (LINA), to provide both first-order and second-order interpretations and both instance-wise and model-wise interpretations.

Model interpretation reveals the input-to-output relationships that a machine learning model has learned from the training data to make predictions [7]. The first-order model interpretation aims to identify individual features that are important for a model to make predictions. For predictive genomics, this can reveal which individual SNPs are important for phenotype prediction. The second-order model interpretation aims to identify important interactions among features that have a large impact on model prediction. The second-order interpretation may reveal the XOR interaction between the two features that jointly determine the output. For predictive genomics, this may uncover epistatic interactions between pairs of SNPs [8], [9].

A general strategy for the first-order interpretation of neural networks, first introduced by Saliency [10], is based on the gradient of the output with respect to (w.r.t.) the input feature vector. A feature with a larger partial derivative of the output is considered more important. The gradient of a neural network model w.r.t. the input feature vector of a specific instance can be computed using backpropagation, which generates an instance-wise first-order interpretation. The Grad*Input algorithm [11] multiplies the obtained gradient element-wise with the input feature vector to generate better scaled importance scores. As an alternative to using the gradient information, the Deep Learning Important FeaTures (DeepLIFT) algorithm explains the predictions of a neural network by backpropagating the activations of the neurons to the input features [11]. The feature importance scores are calculated by comparing the activations of the neurons with their references, which allows the importance information to pass through a zero gradient during backpropagation. The Class Model Visualization (CMV) algorithm [10] computes the visual importance of pixels in convolution neural network (CNN).It performs backpropagation on an initially dark image to find the pixels that maximize the classification score of a given class.

While the algorithms described above were developed specifically for neural networks, model-agnostic interpretation algorithms can be used for all types of machine learning models. Local Interpretable Model-agnostic Explanations (LIME) [12] fits a linear model

to synthetic instances that have randomly perturbed features in the vicinity of an instance. The obtained linear model is analyzed as a local surrogate of the original model to identify the important features for the prediction on this instance. Because this approach does not rely on gradient computation, LIME can be applied to any machine learning model, including non-differentiable models. Previously, we combined LIME and DeepLIFT to interpret a feedforward neural network model for predictive genomics [2]. Kernel SHapley Additive exPlanations (SHAP) [13] uses a sampling method to find the Shapley value for each feature of a given input. The Multi-Objective Counterfactuals (MOC) method [14] searches for the counterfactual explanations for an instance by solving a multi-objective optimization problem. The importance scores calculated by the L2X algorithm [15] are based on the mutual information between the features and the output from a machine learning model. L2X is efficient because it approximates the mutual information using a variational approach.

The second-order interpretation is more challenging than the first-order interpretation because d features would have $(d^2 - d)/2$ possible interactions to be evaluated. Computing the Hessian matrix of a model for the second-order interpretation is conceptually equivalent to, but much more computationally expensive than, computing the gradient for the first-order interpretation. Group Expected Hessian (GEH) [16] computes the Hessian of a Bayesian neural network for many regions in the input feature space and aggregates them to estimate an interaction score for every pair of features. The additive grooves algorithm [17] estimates the feature interaction scores by comparing the predictive performance of the decision tree containing all features with that of the decision trees with pairs of features removed. Neural Interaction Detection (NID) [18] avoids the high computational cost of evaluating every feature pair by directly analyzing the weights in a feedforward neural network. If some features are strongly connected to a neuron in the first hidden layer and the paths from that neuron to the output have high aggregated weights, then NID considers these features to have strong interactions.

Model interpretations can be further classified as instance-wise interpretations or model-wise interpretations. Instance-wise interpretation algorithms, including Saliency [10], LIME [12] and L2X [15], provide an explanation for a model's prediction for a specific instance. For example, an instance-wise interpretation of a neural network model for predictive genomics may highlight the important SNPs in a specific subject which are the basis for the phenotype prediction of this subject. This is useful for intuitively assessing how well grounded the prediction of a model is for a specific subject. Model-wise interpretation provides insights into how a model makes predictions in general. CMV [10] was developed to interpret CNN models. Instance-wise interpretation methods can also be used to explain a model by averaging the explanations of all the instances in a test set. A model-wise interpretation of a predictive genomics model can reveal the important SNPs for a phenotype prediction in a large cohort of subjects. Model-wise interpretations shed light on the internal mechanisms of a machine learning model.

In this study, we designed the LINA architecture and developed the first-order and second-order interpretation algorithms for LINA. The interpretation performance of the new methods was benchmarked using synthetic datasets and a predictive genomics application

in comparison with state-of-the-art (SOTA) interpretation methods. The interpretations from LINA were more versatile and more accurate than those from the SOTA methods.

II. METHODS

A. LINA ARCHITECTURE

The key feature of the LINA architecture (Supplementary Figure 1) is the linearization layer, which computes the output as an element-wise multiplication product of the input features, attention weights, and coefficients:

$$y = S\left[K^T(A \circ X) + b\right] = S\left(\sum_{i=1}^{d} k_i a_i x_i + b\right) \quad (1)$$

where y is the output, X is the input feature vector, $S()$ is the activation function of the output layer, ○ represents the element-wise multiplication operation, K and b are respectively the coefficient vector and bias that are constant for all instances, and A is the attention vector that adaptively scales the feature vector of an instance. X, A and K are three vectors of dimension d, which is the number of input features. The computation by the linearization layer and the output layer is also expressed in a scalar format in Equation (1). This formulation allows the LINA model to learn a linear function of the input feature vector, coefficient vector, and attention vector.

The attention vector is computed from the input feature vector using a multi-layer neural network, referred to as the inner attention neural network in LINA. The inner attention neural network must be sufficiently deep for a prediction task owing to the designed low representational capacity of the remaining linearization layer in a LINA model. In the inner attention neural network, all hidden layers use a non-linear activation function, such as ReLU, but the attention layer uses a linear activation function to avoid any restriction in the range of the attention weights. This is different from the typical attention mechanism in existing attentional architectures which generally use the softmax activation function.

B. THE LOSS FUNCTION

The loss function for LINA is composed of the training error loss, regularization penalty on the coefficient vector, and regularization penalty on the attention vector:

$$loss = E(Y, Y_{true}) + \beta \parallel K \parallel_2 + \gamma \parallel A - 1 \parallel_1 \quad (2)$$

where E is a differentiable convex training error function, $\|K\|_2$ is the L2 norm of the coefficient vector, $\|A - 1\|_1$ is the L1 norm of the attention vector minus 1, and β and γ are the regularization parameters. The coefficient regularization sets 0 to be the expected value of the prior distribution for K, which reflects the expectation of un-informative features. The attention regularization sets 1 to be the expected value of the prior distribution for A, which reflects the expectation of a neutral attention weight that does not scale the input feature. The values of β and γ and the choices of L2, L1, and L0 regularization for the coefficient and attention vectors are all hyperparameters that can be optimized for predictive performance on the validation set.

C. FIRST-ORDER INTERPRETATION

LINA derives the instance-wise first-order interpretation from the gradient of the output, y, w.r.t the input feature vector, X. The output gradient can be decomposed as follows:

$$\frac{\partial y}{\partial x_i} = k_i a_i + \sum_{j=1}^{d} k_j \frac{\partial a_j}{\partial x_i} x_j \tag{3}$$

Proof: Let us derive $\frac{\partial y}{\partial x_i}$ for a regression task:

$$\begin{aligned}
\frac{\partial y}{\partial x_i} &= \frac{\partial k_i a_i x_i}{\partial x_i} + \sum_{\substack{j=1 \\ j \neq i}}^{d} \frac{\partial k_j a_j x_j}{\partial x_i} + \frac{\partial b}{\partial x_i} \\
&= k_i \frac{\partial (a_i x_i)}{\partial x_i} + \sum_{\substack{j=1 \\ j \neq i}}^{d} k_j \frac{\partial (a_j x_j)}{\partial x_i} \\
&= k_i \left(\frac{\partial a_i}{\partial x_i} x_i + a_i \right) + \sum_{\substack{j=1 \\ j \neq i}}^{d} k_j \frac{\partial a_j}{\partial x_i} x_j \\
&= k_i a_i + \sum_{j=1}^{d} k_j \frac{\partial a_j}{\partial x_i} x_j
\end{aligned}$$

End-of-proof.

The decomposition of the output gradient in LINA shows that the contribution of a feature in an attentional architecture comprises (i) a direct contribution to the output weighted by its attention weight and (ii) an indirect contribution to the output during attention computation. This indicates that using attention weights directly as a measure of feature importance omits the indirect contribution of a feature in the attention mechanism.

For the instance-wise first-order interpretation, we defined $\mathrm{FQ_i} = \frac{\partial y}{\partial x_i}$ as the full importance score for feature i, $\mathrm{DQ_i} = k_i a_i$ as the direct importance score for feature i, and $\mathrm{IQ_i} = \sum_{j=1}^{d} k_j \frac{\partial a_j}{\partial x_i} x_j$ as the indirect importance score for feature i.

For the model-wise first-order interpretation, we defined the model-wise full importance score ($\mathrm{FP_i}$), direct importance score ($\mathrm{DP_i}$), and indirect importance score ($\mathrm{IP_i}$) for feature i as the averages of the absolute values of the corresponding instance-wise importance scores of this feature across all instances in the test set:

$$\mathrm{FP_i} = \overline{|\mathrm{FQ_i}|} \tag{4}$$

$$\mathrm{DP_i} = \overline{|\mathrm{DQ_i}|} \tag{5}$$

$$\mathrm{IP_i} = \overline{|\mathrm{IQ_i}|} \tag{6}$$

Because absolute values are used, the model-wise $\mathrm{FP_i}$ of feature i is no longer a sum of its $\mathrm{IP_i}$ and $\mathrm{DP_i}$.

D. SECOND-ORDER INTERPRETATION

It is computationally expensive and unscalable to compute the Hessian matrix for a large LINA model. Here, the Hessian matrix of the output w.r.t. the input feature vector is approximated using the Jacobian matrix of the attention vector w.r.t. the input feature vector in a LINA model, which is computationally feasible to calculate. An approximation is derived as follows.

$$\frac{\partial^2 y}{\partial x_i \partial x_j} = K^T \frac{\partial}{\partial x_i} \begin{bmatrix} x_1 \frac{\partial a_1}{\partial x_j} \\ \vdots \\ x_{j-1} \frac{\partial a_{j-1}}{\partial x_j} \\ \boldsymbol{x_j \frac{\partial a_j}{\partial x_j} + a_j} \\ x_{j+1} \frac{\partial a_{j+1}}{\partial x_j} \\ \vdots \\ x_n \frac{\partial a_n}{\partial x_j} \end{bmatrix} = K^T \begin{bmatrix} x_1 \frac{\partial^2 a_1}{\partial x_i \partial x_j} \\ \vdots \\ x_{i-1} \frac{\partial^2 a_i - 1}{\partial x_i \partial x_j} \\ \boldsymbol{x_i \frac{\partial^2 a_i}{\partial x_i \partial x_j} + \frac{\partial a_i}{\partial x_j}} \\ x_{i+1} \frac{\partial^2 a_{i+1}}{\partial x_i \partial x_j} \\ \vdots \\ x_{j-1} \frac{\partial^2 a_{j-1}}{\partial x_i \partial x_j} \\ \boldsymbol{x_j \frac{\partial^2 a_j}{\partial x_i \partial x_j} + \frac{\partial a_j}{\partial x_i}} \\ x_{j+1} \frac{\partial^2 a_{j+1}}{\partial x_i \partial x_j} \\ \vdots \\ x_n \frac{\partial^2 a_n}{\partial x_i \partial x_j} \end{bmatrix} \tag{7}$$

By omitting the second-order derivatives of the attention weights, Equation (7) can be simplified as

$$\frac{\partial^2 y}{\partial x_i \partial x_j} \approx k_j \frac{\partial a_j}{\partial x_i} + k_i \frac{\partial a_i}{\partial x_j} \tag{8}$$

Equation (8) shows an approximation of the Hessian of the output using the Jacobian of the attention vector. The K-weighted sum of the omitted second-order derivatives of the attention weights constitutes the approximation error. The performance of the second-order interpretation based on this approximation is benchmarked using synthetic and real-world datasets.

For instance-wise second-order interpretation, we define a directed importance score of feature r to feature c:

$$\mathrm{SQ}_{\mathrm{r}}^{\mathrm{c}} = k_c \frac{\partial a_c}{\partial x_r} \tag{9}$$

This measures the importance of feature r in the calculation of the attention weight of feature c. In other words, this second-order importance score measures the importance of feature r to the direct importance score of feature c for the output.

For the model-wise second-order interpretation, we defined an undirected importance score between feature r and feature c based on their average instance-wise second-order importance score in the test set:

$$\mathrm{SP}_{\mathrm{c,r}} = \overline{\left| SQ_r^c + SQ_c^r \right|} \tag{10}$$

E. RECAP OF THE LINA IMPORTANCE SCORES

The notations and definitions of all the importance scores for a LINA model are recapitulated below. FQ and SQ are selected as the first-order and second-order importance score, respectively, for instance-wise interpretation. FP and SP are used as the first-order and second-order importance scores, respectively, for model-wise interpretation.

Order	Target	Acronym	Definition
First-order	Instance-wise	FQ	$\mathrm{FQ_i} = \mathrm{DQ_i} + \mathrm{IQ_i}$
		DQ	$\mathrm{DQ_i} = k_i a_i$
		IQ	$\mathrm{IQ_i} = \sum_{c=1}^{d} \mathrm{SQ}_i^c x_c$
	Model-wise	FP	$\mathrm{FP_i} = \overline{\left|\mathrm{FQ_i}\right|}$
		DP	$\mathrm{DP_i} = \overline{\left|\mathrm{DQ_i}\right|}$
		IP	$\mathrm{IP_i} = \overline{\left|\mathrm{IQ_i}\right|}$
Second-order	Instance-wise	SQ	$\mathrm{SQ}_{\mathrm{r}}^{\mathrm{c}} = k_c \frac{\partial a_c}{\partial x_r}$
	Model-wise	SP	$\mathrm{SP}_{\mathrm{c,r}} = \overline{\left\| SQ_r^c + SQ_c^r \right\|}$

III. DATA AND EXPERIMENTAL SETUP

A. CALIFORNIA HOUSING DATASET

The California housing dataset [19] was used to formulate a simple regression task, which is the prediction of the median sale price of houses in a district based on eight input features (Supplementary Table 1). The dataset contained 20640 instances (districts) for model training and testing.

B. FIRST-ORDER BENCHMARKING DATASETS

Five synthetic datasets, each containing 20,000 instances, were created using the sigmoid functions to simulate binary classification tasks. These functions were created following the examples in [15] for the first-order interpretation benchmarking. All five datasets included ten input features. The values of the input features were independently sampled from a standard Gaussian distribution: $x_i \sim N(0, 1)$, $i \in \{1, 2, \ldots, 10\}$. The target value was set to 0, if the sigmoid function output is (0, 0.5). The target value was set to 1, if the sigmoid function output is [0.5, 1). We used the following five sigmoid functions of different subsets of the input features:

(F1): $Sig\left(4 * X_1^2 - 3 * X_2^2 - 2 * X_3^2 + X_4^2\right)$. This function contains four important features with independent squared relationships with the target. The ground-truth rankings of the features by first-order importance are X_1, X_2, X_3, and X_4. The remaining six uninformative features are tied in the last rank.

(F2): $Sig(-10 * \sin(X_1) + 2 * abs(X_2) + X_3 - \exp(-X_4))$. This function contains four important features with various non-linear additive relationships with the target. The ground-truth ranking of the features is X_1, X_4, X_2, and X_3. The remaining six uninformative features are tied in the last rank.

(F3): $Sig(4 * X_1 * X_2 * X_3 + X_4 * X_5 * X_6)$. This function contains six important features with multiplicative interactions among one another. The ground-truth ranking of the features is X_1, X_2 and X_3 tied in the first rank, X_4, X_5 and X_6 tied in the second rank, and the remaining uninformative features tied in the third rank.

(F4): $Sig(-10 * \sin(X_1 * X_2 * X_3) + abs(X_4 * X_5 * X6))$. This function contains six important features with multiplicative interactions among one another and non-linear relationships with the target. The ground-truth ranking of the features is X_1, X_2 and X_3 tied in the first rank, X_4, X_5 and X_6 tied in the second rank, and the other four uninformative features tied in the third rank.

(F5): $Sig(-20 * \sin(X_1 * X_2) + 2 * abs(X_3) + X_4 * X_5 - 4 * \exp(-X_6))$. This function contains six important features with a variety of non-linear relationships with the target. The ground-truth ranking of the features is X_1 and X_2 tied in the first rank, X_6 in the second, X_3 in the third, X_4 and X_5 tied in the fourth, and the remaining uninformative features tied in the fifth.

C. SECOND-ORDER BENCHMARKING DATASET

Ten regression synthetic datasets, referred to as F6-A, F7-A, F8-A, F9-A, and F10-A (−A datasets) and F6-B, F7-B, F8-B, F9-B, and F10-B (−B datasets) were created. The −A datasets followed the examples in [18] for the second-order interpretation benchmarking. The −B datasets used the same functions below to compute the target as the −A datasets, but included more uninformative features to benchmark the interpretation performance on high-dimensional data. Each −A dataset contained 5,000 instances. Each −B dataset contained 10,000 instances. The five −A datasets included 13 input features. The five −B datasets included 100 input features, some of which were used to compute the target. In F7-A/B, F8-A/B, F9-A/B, and F10-A/B, the values of the input features of an instance were

independently sampled from a standard uniform distribution: $X_i \sim U(-1, 1)$, $i \in \{1, 2, \ldots, 13\}$ in the −A datasets or $i \in \{1, 2, \ldots, 100\}$ in the −B datasets. In the F6 dataset, the values of the input features of an instance were independently sampled from two uniform distributions: $X_i \sim U(0, 1)$, $i \in \{1, 2, 3, 6, 7, 9, 11, 12, 13\}$ in the −A datasets and $i \in \{1, 2, 3, 6, 7, 9, 11, \ldots, 100\}$ in the −B datasets; and $X_i \sim U(0.6, 1)$, $i \in \{4, 5, 8, 10\}$ in both. The value of the target for an instance was computed using the following five functions:

(F6-A) and (F6-B): $\pi^{X_1 * X_2} * \sqrt{X_3} + sin^{-1}(X_4) + \log(X_3 + X_5) + \frac{X_9}{X_{10}} * \sqrt{\frac{X_7}{X_8}} - X_2 * X_7$.

This function contains eleven pairwise feature interactions: $\{(X_1, X_2), (X_1, X_3), (X_2, X_3), (X_3, X_5), (X_7, X_8), (X_7, X_9), (X_7, X_{10}), (X_8, X_9), (X_8, X_{10}), (X_9, X_{10}), (X_2, X_7)\}$.

(F7-A) and (F7-B):

$\exp(|X_1 - X_2|) + |X_2 * X_3| - X_3^{2|X_4|} + \log\left(X_4^2 + X_5^2 + X_7^2 + X_8^2\right) + X_9 + X_{10}^2 \frac{1}{1 + X_{10}^2}$. This function contains nine pairwise interactions: $\{(X_1, X_2), (X_2, X_3), (X_3, X_4), (X_4, X_5), (X_4, X_7), (X_4, X_8), (X_5, X_7), (X_5, X_8), (X_7, X_8)\}$.

(F8-A) and (F8-B):

$\sin(|X_1 * X_2| + 1) - \log(|X_3 * X_4| + 1) + \cos(X_5 + X_6 - X_8) + \sqrt{X_8^2 + X_9^2 + X_{10}^2}$. This function contains ten pairwise interactions: $\{(X_1, X_2), (X_3, X_4), (X_5, X_6), (X_4, X_7), (X5, X_6), (X_5, X_8), (X_6, X_8), (X_8, X_9), (X_8, X_{10}), (X_9, X_{10})\}$.

(F9-A) and (F9-B):

$\tanh(X_1 * X_2 + X_3 * X_4) * \sqrt{|X_5|} + \log\left[(X_6 * X_7 * X_8)^2 + 1\right] + X_9 * X_{10} + \frac{1}{1 + |X_{10}|}$. This function contains thirteen pairwise interactions: $\{X_1, X_2), (X_1, X_3), (X_2, X_3), (X_2, X_4), (X_3, X_4), (X_1, X_5), (X_2, X_5), (X_3, X_5), (X_4, X_5), (X_6, X_7), (X_6, X_8), (X_7, X_8), (X_9, X_{10})\}$.

(F10-A) and (F10-B): $\cos(X_1 * X_2 * X_3) + \sin(X_4 * X_5 * X_6)$. This function contains six pairwise interactions: $\{(X_1, X_2), (X_1, X_3), (X_2, X_3), (X_4, X_5), (X_4, X_6), (X_5, X_6)\}$.

D. BREAST CANCER DATASET

The Discovery, Biology, and Risk of Inherited Variants in Breast Cancer (DRIVE) project [20] generated a breast cancer dataset (NIH dbGaP accession number: phs001265.v1.p1) for genome-wide association study (GWAS) and predictive genomics. This cohort contained 26,053 case subjects with malignant tumor or *in situ* tumor and 23,058 control subjects with no tumor. The task for predictive genomics is a binary classification of subjects between cases and controls. The breast cancer dataset was processed using PLINK [21] as described previously [2] to compute the statistical significance of the SNPs. Out of a total of 528,620 SNPs, 1541 SNPs had a p-value lower than 10^{-6} and were used as the input features for predictive genomics. To benchmark the performance of the model interpretation, 1541 decoy SNPs were added as input features. The frequencies of homozygous minor alleles, heterozygous alleles, and homozygous dominant alleles were the same between decoy SNPs and real SNPs. Because decoy SNPs have random relationships with the case/

control phenotype, they should not be selected as important features or be included in salient interactions by model interpretation.

E. IMPLEMENTATIONS AND EVALUATION STRATEGIES

The California Housing Dataset was partitioned into a training set (70%), a validation set (20%), and a test set (10%). The eight input features were longitude, latitude, median age, total rooms, total bedrooms, population, households, and median income. The median house value was the target of the regression. All the input features were standardized to zero mean and unit standard deviation based on the training set. Feature standardization is critical for model interpretation in this case because the scale for the importance scores of a feature is determined by the scale for the values of this feature and comparison of the importance scores between features requires the values of the features to be in the same scale. The LINA model comprised an input layer (8 neurons), five fully connected hidden layers (7, 6, 5, 4 and 3 neurons), and an attention layer (8 neurons) for the inner attention neural network, followed by a second input layer (8 neurons), a linearization layer (8 neurons), and an output layer (1 neuron). The hidden layers used ReLU as the activation function. No regularization was applied to the coefficient vector and L1 regularization was applied to the attention vector ($\gamma = 10^{-6}$). The LINA model was trained using the Adam optimizer with a learning rate of 10^{-2}. The predictive performance of the obtained LINA model was benchmarked to have an RMSE of 71055 in the test set. As a baseline model for comparison, a gradient boosting model achieved an RMSE of 77852 in the test set using 300 decision trees with a maximum depth of 5.

For the first-order interpretation, each synthetic dataset was split into a cross-validation set (80%) for model training and hyperparameter optimization and a test set (20%) for performance benchmarking and model interpretation. A LINA model and a feedforward neural network (FNN) model were constructed using 10-fold cross-validation. For the first four synthetic datasets, the inner attention neural network in the LINA model had 3 layers containing 9 neurons in the first layer, 5 neurons in the second layer, and 10 neurons in the attention layer. The FNN had 3 hidden layers with the same number of neurons in each layer as the inner attention neural network in the LINA model. For the fifth function with more complex relationships, the first and second layers were widened to 100 and 25 neurons, respectively, in both the FNN and LINA models to achieve a predictive performance similar to the other datasets in their respective validation sets. Both the FNN and LINA models were trained using the Adam optimizer. The learning rate was set to 10^{-2}. The mini-batch size was set to 32. No hyperparameter tuning was performed. The LINA model was trained with the L2 regularization on the coefficient vector ($\beta = 10^{-4}$) and the L1 regularization on the attention vector ($\gamma = 10^{-6}$). The values of β and γ were selected from 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}, and 0 based on the predictive performance of the LINA model on the validation set. Batch normalization was used for both architectures. Both the FNN and LINA models achieved predictive performance at approximately 99% AUC on the test set in the five first-order synthetic datasets, which was comparable to [15]. Deep Lift [11], LIME [12], Grad*Input [11], L2X [15] and Saliency [10] were used to interpret the FNN model and calculate the feature importance scores using their default configurations. FP, DP, and IP scores were used as the first-order importance scores for the LINA model. We compared the

performances of the first-order interpretation of LINA with DeepLIFT, LIME, Grad*Input and L2X. The interpretation accuracy was measured using the Spearman rank correlation coefficient between the predicted ranking of features by their first-order importance and the ground-truth ranking. This metric was chosen because it encompasses both the selection and ranking of the important features.

For the second-order interpretation benchmarking, each synthetic dataset was also split into a cross-validation set (80%) and a test set (20%). A LINA model, an FNN model for NID, and a Bayesian neural network (BNN) for GEH as shown in [16], were constructed based on the neural network architecture used in [18] using 10-fold cross-validation. The inner attention neural network in the LINA model uses 140 neurons in the first hidden layer, 100 neurons in the second hidden layer, 60 neurons in the third hidden layer, 20 neurons in the fourth hidden layer, and 13 neurons in the attention layer. The FNN model was composed of 4 hidden layers with the same number of neurons in each layer as LINA's inner attention neural network. The BNN model uses the same architecture as that of the FNN model. The FNN, BNN and LINA models were trained using the Adam optimizer with a learning rate of 10^{-3} and a mini-batch size of 32 for the −A datasets and 128 for the −B datasets. The LINA model was trained using L2 regularization on the coefficient vector ($\beta = 10^{-4}$) and the L1 regularization on the attention vector ($\gamma = 10^{-6}$) with batch normalization. Hyperparameter tuning was performed as described above to optimize the predictive performance. The FNN and BNN models were trained using the default regularization parameters, as shown in [16], [18]. Batch normalization was used for LINA. The FNN, BNN and LINA models all achieved R^2 scores of more than 0.99 on the test sets of the five −A datasets, as in the examples in [18], while their R^2 scores ranged from 0.91 to 0.93 on the test set of the five high-dimensional −B datasets. Pairwise interactions in each dataset were identified from the BNN model using GEH [16], the FNN model using NID [18], and the LINA model using the SP scores. For GEH, the number of clusters was set to the number of features and the number of iterations was set to 20. NID was run using its default configuration. For a dataset with m pairs of ground-truth interactions, the top-m pairs with the highest interaction scores were selected from each algorithm's interpretation output. The percentage of ground-truth interactions in the top-m predicted interactions (i.e., the precision) was used to benchmark the second-order interpretation performance of the algorithms.

For the breast cancer dataset, 49111 subjects in the breast cancer dataset were randomly divided into the training set (80%), validation set (10%), and test set (10%). The FNN model and the BNN model had 3 hidden layers with 1000, 250 and 50 neurons as described previously [2]. The same hyperparameters were used in our previous study [2]. The inner attention neural network in the LINA model also used 1000, 250 and 50 neurons before the attention layer. All of these models had 3082 input neurons for 1541 real SNPs and 1541 decoy SNPs. β was set to 0.01 and γ to 0, which were selected from 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}, and 0 based on the predictive performance of the LINA model on the validation set. Early stopping based on the validation AUC score was used during training. The FNN, BNN and LINA models achieved a test AUC of 64.8%, 64.8% and 64.7% on the test set, respectively, using both the 1541 real SNPs with p-values less than 10^{-6} and the 1541 decoy SNPs. The test AUCs of these models were lower than that of the FNN model in our previous study [2] at 67.4% using real 5,273 SNPs with p-values less than 10^{-3} as

input. As the same FNN architecture design was used in the two studies, the reduction in the predictive performance in this study can be attributed to the use of more stringent p-value filtering to retain only real SNPs with a high likelihood of having a true association with the disease and the addition of decoy SNPs for benchmarking the interpretation performance.

Deep Lift [11], LIME [12], Grad*Input [11], L2X [15] and Saliency [10] were used to interpret the FNN model and calculate the feature importance scores using their default configurations. The FP score was used as the first-order importance score for the LINA model. After the SNPs were filtered at a given importance score threshold, the false discovery rate (FDR) was computed from the retained real and decoy SNPs above the threshold. The number of retained real SNPs was the total positive count for the FDR. The number of false positive hits (i.e., the number of unimportant real SNPs) within the retained real SNPs was estimated as the number of retained decoy SNPs. Thus, FDR was estimated by dividing the number of retained decoy SNPs by the number of retained real SNPs. An importance-score-sorted list of SNPs from each algorithm was filtered at an increasingly stringent score threshold until reaching the desired FDR level. The interpretation performance of an algorithm was measured by the number of top-ranked features filtered at 0.1%, 1% and 5% FDR and the FDRs for the top-100 and top-200 SNPs ranked by an algorithm. For the second-order interpretation, pairwise interactions were identified from the BNN model using GEH [16], from the FNN model using NID [18], and from the LINA model using the SP scores. For GEH, the number of clusters was set to 20 and the number of iterations was set to 20. While LINA and NID used all 4,911 subjects in the test set and completed their computation within an hour, the GEH results were computed for only 1000 random subjects in the test set over >2 days because GEH would have taken approximately two months to complete the entire test set with its n^2 computing cost where n is the number of subjects. NID was run using its default configuration in the FNN model. The interpretation accuracy was also measured by the numbers of top-ranked pairwise interactions detected at 0.1%, 1% and 5% FDR and the FDRs for the top-1000 and top-2000 interaction pairs ranked by an algorithm. A SNP pair was considered to be false positive if one or both of the SNPs in a pair was a decoy.

IV. RESULTS AND DISCUSSION

A. DEMONSTRATION OF LINA ON A REAL-WORLD APPLICATION

In this section, we demonstrate LINA using the California housing dataset, which has been used in previous model interpretation studies for algorithm demonstration [16], [18]. Four types of interpretations from LINA were presented, including the instance-wise first-order interpretation, the instance-wise second-order interpretation, the model-wise first-order interpretation, and the model-wise second-order interpretation.

1) INSTANCE-WISE INTERPRETATION—Supplementary Table 1 shows the prediction and interpretation results of the LINA model for an instance (district # 20444) that had a true median price of $208600. The predicted price of $285183 was simply the sum of the eight element-wise products of the attention, coefficient, and feature columns plus the bias. This provided an easily understandable representation of the intermediate

computation behind the prediction for this instance. For example, the median age feature had a coefficient of 213 in the model. For this instance, the median age feature had an attention weight of −275, which switched the median age to a negative feature and amplified its direct effect on the predicted price in this district.

The product of the attention weight and coefficient yielded the direct importance score of the median age feature (i.e., DQ = −58,524), which represented the strength of the local linear association between the median age feature and the predicted price for this instance. By assuming that the attention weights of this instance are fixed, one can expect a decrease of $58,524 in the predicted price for an increase in the median age by one standard deviation (12.28 years) for this district. But this did not consider the effects of the median age increase on the attention weights, which was accounted for by its indirect importance score (i.e., IQ = 91,930). The positive IQ indicated that a higher median age would increase the attention weights of other positive features and increase the predicted price indirectly. Combining the DQ and IQ, the positive FQ of 33,407 marked the median age to be a significant positive feature for the predicted price, perhaps through the correlation with some desirable variables for this district. This example suggested a limitation of using the attention weights themselves to evaluate the importance of features in the attentional architectures. The full importance scores represented the total effect of a feature's change on the predicted price. For this instance, the latitude feature had the largest impact on the predicted price.

Supplementary Table 2 presents a second-order interpretation of the prediction for this instance. The median age row in Supplementary Table 2 shows how the median age feature impacted the attention weights of the other features. The two large positive SQ values of median age to the latitude and longitude features indicated significant increases of the two location features' attention weights with the increase of the median age. In other words, the location become a more important determinant of the predicted price for districts with older houses. The total bedroom feature received a large positive attention weight for this instance. The total bedroom column in Supplementary Table 2 shows that the longitude and latitude features are the two most important determinants for the attention weights of the total bedroom feature. This suggested how a location change may alter the direct importance of the total bedroom feature for the price prediction of this district.

2) MODEL-WISE INTERPRETATION—Figure 1 shows the first-order model-wise interpretation results across districts in the California Housing dataset. The longitude, latitude and population were the three most important features. The longitude and latitude had both high direct importance scores and high indirect importance scores. However, the population feature derived its importance mostly from its heavy influence on the attention weights as measured by its indirect importance score.

Figure 2 shows the second-order model-wise interpretation results for pairs of different features. Among all the feature pairs, the latitude and longitude features had the most prominent interactions, which was reasonable because the location was jointly determined by these two features.

Some significant differences existed between the instance-wise interpretation and model-wise interpretation (e.g., Supplementary Table 1 vs. Figure 1 and Supplementary Table 2 vs. Figure 2). This illustrates the need for both instance-wise and model-wise interpretation methods for different purposes.

B. BENCHMARKING OF THE FIRST-ORDER AND SECOND-ORDER INTERPRETATIONS USING SYNTHETIC DATASETS

In real-world applications, the true importance of features for prediction cannot be determined with certainty and may vary among different models. Therefore, previous studies on model interpretation [12], [16] benchmarked their interpretation performance using synthetic datasets with known ground-truth of feature importance. In this study, we also compared the interpretation performance of LINA with the SOTA methods using synthetic datasets created as in previous studies [15], [18].

The performance of the first-order interpretation of LINA was compared with DeepLIFT, LIME, Grad*Input and L2X (Table 1). The three first-order importance scores from LINA, including FP, DP and IP, were tested. The DP score performed the worst among the three, especially in the F3 and F4 datasets which contained interactions among three features. This suggested the limitation of using attention weights as a measure of feature importance. The FP score provided the most accurate ranking among the three LINA scores because it accounted for the direct contribution of a feature and its indirect contribution through attention weights. The first-order importance scores were then compared among different algorithms. L2X and LIME distinguished many important features correctly from un-informative features, but their rankings of the important features were often inaccurate. The gradient-based methods produced mostly accurate rankings of the features based on their first-order importance. Their interpretation accuracy generally decreased in datasets containing interactions among more features. Among all the methods, the LINA FP scores provided the most accurate ranking of the features on average.

The performance of the second-order interpretation of LINA was compared with those of GEH and NID (Table 2). There were a total of 78 possible pairs of interactions among 13 features in each −A synthetic dataset and there were 4950 possible pairs of interactions among 100 features in each −B synthetic dataset. The precision from random guesses was only ~12.8% on average in the −A datasets and less than 1% in the −B datasets. The three second-order algorithms all performed significantly better than the random guess. In the −A datasets, the average precision of LINA SP was ~80%, which was ~12% higher than that of NID and ~29% higher than that of GEH. The addition of 87 un-informative features in the −B datasets reduced the average precision of LINA by ~15%, that of NID by ~13%, and that of GEH by ~22%. In the −B datasets, the average precision of LINA SP was ~65%, which was ~9% higher than that of NID and ~35% higher than that of GEH. This indicates that more accurate second-order interpretations can be obtained from the LINA models.

C. BENCHMARKING OF THE FIRST-ORDER AND SECOND-ORDER INTERPRETATION USING A PREDICTIVE GENOMICS APPLICATION

As the performance benchmarks in synthetic datasets may not reflect those in real-world applications, we engineered a real-world benchmark based on a breast cancer dataset for predictive genomics. While it was unknown which SNPs and which SNP interactions were truly important for phenotype prediction, the decoy SNPs added by us were truly unimportant. Moreover, a decoy SNP cannot have a true interaction, such as XOR or multiplication, with a real SNP to have a joint impact on the disease outcome. Thus, if a decoy SNP or an interaction with a decoy SNP is ranked by an algorithm as important, it should be considered a false positive detection. As the number of decoy SNPs was the same as the number of real SNPs, the false discovery rate can be estimated by assuming that an algorithm makes as many false positive detections from the decoy SNPs as from the real SNPs. This allowed us to compare the number of positive detections by an algorithm at certain FDR levels.

The first-order interpretation performance of LINA was compared with those of DeepLIFT, LIME, Grad*Input and L2X (Table 3). At 0.1%, 1%, and 5% FDR, LINA identified more important SNPs than other algorithms. LINA also had the lowest FDRs for the top-100 and top-200 SNPs. The second-order interpretation performance of LINA was compared with those of NID and GEH (Table 4). At 0.1%, 1%, and 5% FDR, LINA identified more pairs of important SNP interactions than NID and GEH did. LINA had lower FDRs than the other algorithms for the top-1000 and top-2000 SNP pairs. Both L2X and GEH failed to output meaningful importance scores in this predictive genomics dataset. Because GEH needed to compute the full Hessian, it was also much more computationally expensive than the other algorithms.

The existing model interpretation algorithms and LINA can provide rankings of the features or feature interactions based on their importance scores at arbitrary scales. We demonstrated that decoy features can be used in real-world applications to set thresholds for first-order and second-order importance scores based on the FDRs of retained features and feature pairs. This provided an uncertainty quantification of the model interpretation results without knowing the ground-truth in real-world applications.

The predictive genomics application provided a real-world test of the interpretation performance of these algorithms. In comparison with the synthetic datasets, the predictive genomics dataset was more challenging for model interpretation, because of the low predictive performance of the models and the large number of input features. For this real-world application, LINA was shown to provide better first-order and second-order interpretation performance than existing algorithms on a model-wise level. Furthermore, LINA can provide instance-wise interpretation to identify important SNP and SNP interactions for the prediction of individual subjects. Model interpretation is important for making biological discoveries from predictive models, because first-order interpretation can identify individual genes involved in a disease [22], [23] and second-order interpretation can uncover epistatic interactions among genes for a disease [24], [25]. These discoveries may provide new drug targets [26]–[28] and enable personalized formulation of treatment plans [29]–[31] for breast cancer.

V. CONCLUSION

In this study, we designed a new neural network architecture, referred to as LINA, for model interpretation. LINA uses a linearization layer on top of a deep inner attention neural network to generate a linear representation of model prediction. LINA provides the unique capability of offering both first-order and second-order interpretations and both instance-wise and model-wise interpretations. The interpretation performance of LINA was benchmarked to be higher than the existing algorithms on synthetic datasets and a predictive genomics dataset.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgments

This work was supported in part by the National Center for Complementary & Integrative Health, in part by the National Institute of General Medical Sciences of the National Institutes of Health (NIH) under Award R01AT011618, in part by the Startup Fund to Chongle Pan in The University of Oklahoma.

Biographies

ADRIEN BADRÉ received the M.S. degree in computer science from the Institut Supérieur Informatique, de Modélisation et de leurs Applications, France, in 2018, and the M.S. degree in data science and analytics from The University of Oklahoma, USA, in 2018, where he is currently pursuing the Ph.D. degree. His current research interests include machine learning, deep learning, interpretable AI, and predictive genomics.

CHONGLE PAN is currently an Associate Professor in computer science and microbiology with The University of Oklahoma (OU), Norman. Prior to joining OU, in 2018, he was a Senior Research Scientist with the Computer Science and Mathematics Division, Oak Ridge National Laboratory. His research interests include bioinformatics, omics data analysis, machine learning, natural language models, and parallel computing.

REFERENCES

[1]. Bermeitinger B, Hrycej T, and Handschuh S, "Representational capacity of deep neural networks: A computing study," in Proc. 11th Int. Joint Conf. Knowl. Discovery, Knowl. Eng. Knowl. Manage, 2019, pp. 532–538, doi: 10.5220/0008364305320538.

[2]. Badré A, Zhang L, Muchero W, Reynolds JC, and Pan C, "Deep neural network improves the estimation of polygenic risk scores for breast cancer," J. Human Genet, vol. 66, pp. 1–11, Oct. 2020, doi: 10.1038/s10038-020-00832-7. [PubMed: 33303975]

[3]. Fergus P, Montanez A, Abdulaimma B, Lisboa P, Chalmers C, and Pineles B, "Utilizing deep learning and genome wide association studies for epistatic-driven preterm birth classification in African-American women," IEEE/ACM Trans. Comput. Biol. Bioinf, vol. 17, no. 2, pp. 668–678, Apr. 2018, doi: 10.1109/TCBB.2018.2868667.

[4]. Ho Thanh Lam L, Le NH, Van Tuan L, Tran Ban H, Nguyen Khanh Hung T, Nguyen NTK, Huu Dang L, and Le NQK, "Machine learning model for identifying antioxidant proteins using features calculated from primary sequences," Biology, vol. 9, no. 10, p. 325, Oct. 2020, doi: 10.3390/biology9100325.

[5]. Do DT and Le NQK, "Using extreme gradient boosting to identify origin of replication in Saccharomyces cerevisiae via hybrid features," Genomics, vol. 112, no. 3, pp. 2445–2451, May 2020, doi: 10.1016/j.ygeno.2020.01.017. [PubMed: 31987913]

[6]. Baltres A, Al Masry Z, Zemouri R, Valmary-Degano S, Arnould L, Zerhouni N, and Devalland C, "Prediction of oncotype DX recurrence score using deep multi-layer perceptrons in estrogen receptor-positive, HER2-negative breast cancer," Breast Cancer, vol. 27, no. 5, pp. 1007–1016, Sep. 2020, doi: 10.1007/s12282-020-01100-4. [PubMed: 32385567]

[7]. Molnar C, Interpretable machine learning. A Guide for Making Black Box Models Explainable. Lulu.com, 2020. [Online]. Available: https://christophm.github.io/interpretable-ml-book/ and https://books.google.com/books?id=jBm3DwAAQBAJ&source=gbs_navlinks_s

[8]. Cordell HJ, "Epistasis: What it means, what it doesn't mean, and statistical methods to detect it in humans," Proc. Hum. Mol. Genet, vol. 11, no. 20, pp. 2463–2468, Oct. 2002, doi: 10.1093/hmg/11.20.2463.

[9]. Phillips PC, "Epistasis—The essential role of gene interactions in the structure and evolution of genetic systems," Nature Rev. Genet, vol. 9, no. 11, pp. 855–867, Nov. 2008, doi: 10.1038/nrg2452. [PubMed: 18852697]

[10]. Simonyan K, Vedaldi A, and Zisserman A, "Deep inside convolutional networks: Visualising image classification models and saliency maps," in Proc. Workshop Int. Conf. Learn. Represent Banff, AB, Canada: Citeseer, 2014. [Online]. Available: https://sites.google.com/site/representationlearning2014/workshop-proceedings

[11]. Shrikumar A, Greenside P, and Kundaje A, "Learning important features through propagating activation differences," in Proc. Int. Conf. Mach. Learn, Jul. 2017, pp. 3145–3153. Accessed: Nov. 11, 2019. [Online]. Available: http://proceedings.mlr.press/v70/shrikumar17a.html

[12]. Ribeiro MT, Singh S, and Guestrin C, "'Why should I trust you?': Explaining the predictions of any classifier," in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, Aug. 2016, pp. 1135–1144, doi: 10.1145/2939672.2939778.

[13]. Lundberg SM and Lee S-I, "A unified approach to interpreting model predictions," in Proc. 31st Int. Conf. Neural Inf. Process. Syst. (NIPS), Long Beach, CA, USA. Red Hook, NY, USA: Curran, 2017, pp. 4768–4777.

[14]. Dandl S, Molnar C, Binder M, and Bischl B, "Multi-objective counterfactual explanations," in Proc. Int. Conf. Parallel Problem Solving Nature Cham, Switzerland: Springer, Sep. 2020, pp. 448–469.

[15]. Chen J, Song L, Wainwright M, and Jordan M, "Learning to explain: An information-theoretic perspective on model interpretation," in Proc. 35th Int. erence Mach. Learn, Jul. 2018, pp. 883–892. Accessed: Nov. 4, 2021. [Online]. Available: https://proceedings.mlr.press/v80/chen18j.html

[16]. Cui T, Marttinen P, and Kaski S, "Learning global pairwise interactions with Bayesian neural networks," in Proc. 24th Eur. Conf. Artif. Intell, vol. 325, 2020, pp. 1087–1094, doi: 10.3233/FAIA200205.

[17]. Sorokina D, Caruana R, and Riedewald M, “Additive groves of regression trees,” in Proc. 18th Eur. Conf. Mach. Learn, Berlin, Germany, Sep. 2007, pp. 323–334, doi: 10.1007/978-3-540-74958-5_31.

[18]. Tsang M, Cheng D, and Liu Y, “Detecting statistical interactions from neural network weights,” in Proc. Int. Conf. Learn. Represent. (ICLR), Vancouver, BC, Canada, 2018.

[19]. Pace RK and Barry R, “Sparse spatial autoregressions,” Statist. Probab. Lett, vol. 33, pp. 291–297, May 1997, doi: 10.1016/S0167-7152(96)00140-X.

[20]. Amos CI, Dennis J, Wang Z, Byun J, Schumacher FR, Gayther SA, Casey G, Hunter DJ, Sellers TA, Gruber SB, and Dunning AM, “The OncoArray consortium: A network for understanding the genetic architecture of common cancers,” Cancer Epidemiol. Biomarkers Prevention, vol. 26, no. 1, pp. 126–135, Jan. 2017, doi: 10.1158/1055-9965.EPI-16-0106.

[21]. Purcell S, Neale B, Todd-Brown K, Thomas L, Ferreira MAR, Bender D, Maller J, Sklar P, De Bakker PI, Daly MJ, and Sham PC, “PLINK: A tool set for whole-genome association and population-based linkage analyses,” Amer. J. Hum. Genet, vol. 81, no. 3, pp. 559–575, Sep. 2007, doi: 10.1086/519795. [PubMed: 17701901]

[22]. Rivandi M, Martens JWM, and Hollestelle A, “Elucidating the underlying functional mechanisms of breast cancer susceptibility through post-GWAS analyses,” Frontiers Genet., vol. 9, p. 280, Aug. 2018. Accessed: Feb. 1, 2022. [Online]. Available: https://www.frontiersin.org/article/10.3389/fgene.2018.00280

[23]. Cardoso SR, Gillespie A, Haider S, and Fletcher O, “Functional annotation of breast cancer risk loci: Current progress and future directions,” Brit. J. Cancer, vol. 2021, pp. 1–13, Nov. 2021, doi: 10.1038/s41416-021-01612-6.

[24]. Shaker OG and Senousy MA, “Association of SNP-SNP interactions between RANKL, OPG, CHI3L1, and VDR genes with breast cancer risk in Egyptian women,” Clin. Breast Cancer, vol. 19, no. 1, pp. e220–e238, Feb. 2019, doi: 10.1016/j.clbc.2018.09.004. [PubMed: 30309792]

[25]. van de Haar J, Canisius S, Yu MK, Voest EE, Wessels LFA, and Ideker T, “Identifying epistasis in cancer genomes: A delicate affair,” Cell, vol.177, no.6, pp. 1375–1383, May2019, doi:10.1016/j.cell.2019.05.005. [PubMed: 31150618]

[26]. Wang L, Ingle J, and Weinshilboum R, “Pharmacogenomic discovery to function and mechanism: Breast cancer as a case study,” Clin. Pharmacol. Therapeutics, vol. 103, no. 2, pp. 243–252, Feb. 2018, doi: 10.1002/cpt.915.

[27]. Gao M, Quan Y, Zhou X-H, and Zhang H-Y, “PheWAS-based systems genetics methods for anti-breast cancer drug discovery,” Genes, vol. 10, no. 2, p. 154, Feb. 2019, doi: 10.3390/genes10020154.

[28]. Gonçalves E, Segura-Cabrera A, Pacini C, Picco G, Behan FM, Jaaks P, Coker EA, van der Meer D, Barthorpe A, Lightfoot H, and Mironenko T, “Drug mechanism-of-action discovery through the integration of pharmacological and CRISPR screens,” Mol. Syst. Biol, vol. 16, no. 7, Jul. 2020, doi: 10.15252/msb.20199405.

[29]. Wu L, Yao L, Zhang H, Ouyang T, Li J, Wang T, Fan Z, Fan T, Lin B, Yin CC, and Xie Y, “A genome-wide association study identifies WT1 variant with better response to 5-fluorouracil, pirarubicin and cyclophosphamide neoadjuvant chemotherapy in breast cancer patients,” Oncotarget, vol. 7, no. 4, pp. 5042–5052, Nov. 2015, doi: 10.18632/oncotarget.5837.

[30]. Zhao X, Li J, Liu Z, and Powers S, “Combinatorial CRISPR/Cas9 screening reveals epistatic networks of interacting tumor suppressor genes and therapeutic targets in human breast cancer,” Cancer Res, vol. 81, no. 24, pp. 6090–6105, Dec. 2021, doi: 10.1158/0008-5472.CAN-21-2555. [PubMed: 34561273]

[31]. Velasco-Ruiz A, Nuñez-Torres R, Pita G, Wildiers H, Lambrechts D, Hatse S, Delombaerde D, Van Brussel T, Alonso MR, Alvarez N, Herraez B, Vulsteke C, Zamora P, Lopez-Fernandez T, and Gonzalez-Neira A, “POLRMT as a novel susceptibility gene for cardiotoxicity in epirubicin treatment of breast cancer patients,” Pharmaceutics, vol. 13, no. 11, p. 1942, Nov. 2021, doi: 10.3390/pharmaceutics13111942. [PubMed: 34834357]

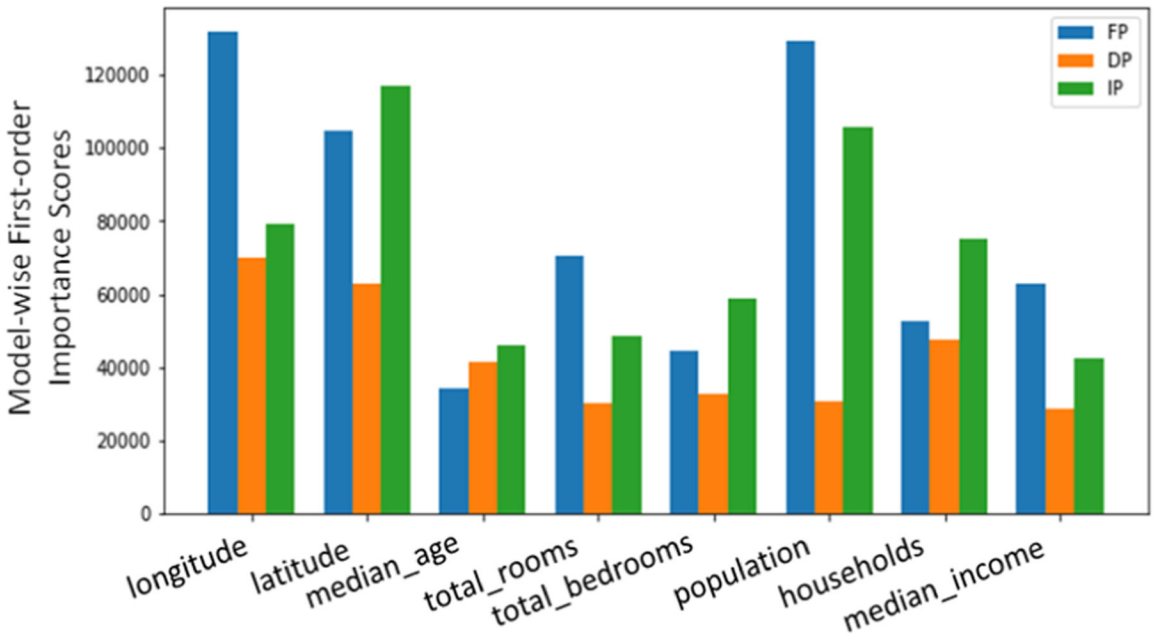

FIGURE 1.
First-order model-wise interpretation. The three bars of a feature represented the FP, IP and DP scores of this feature in the LINA model.

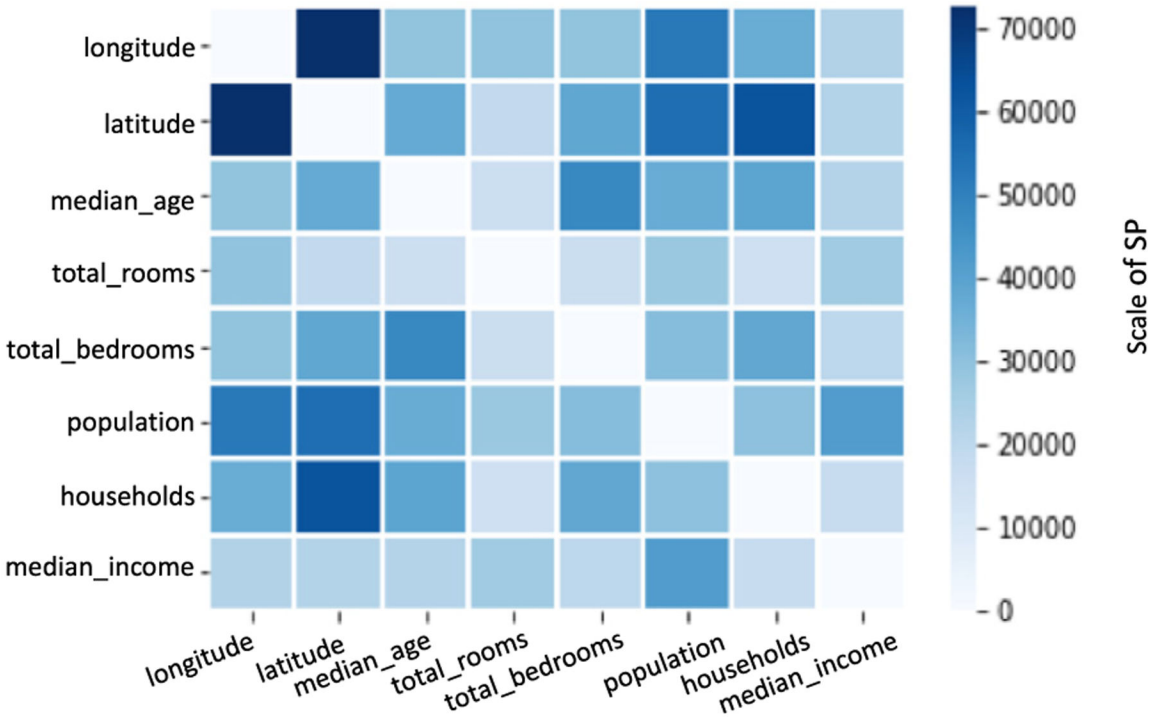

FIGURE 2.

Second-order model-wise interpretation. The second-order model-wise importance scores (SP) are undirected between two features and are shown in a symmetric matrix as a heatmap. The importance scores for the feature self-interactions are set to zero in the diagonal of the matrix.

TABLE 1.

Benchmarking of the first-order interpretation performance using five synthetic datasets (F1~F5)*

Methods \ Datasets	F1	F2	F3	F4	F5	Average
LINA DP	**1.00±0.00**	0.88±0.03	0.25±0.07	0.65±0.05	0.92±0.03	0.74±0.04
LINA IP	**1.00±0.00**	0.92±0.03	0.69±0.01	0.84±0.03	0.96±0.03	0.88±0.02
LINA FP	**1.00±0.00**	0.97±0.02	**1.00±0.00**	**0.91±0.04**	**1.00±0.00**	**0.98±0.01**
DeepLift	0.99±0.01	**1.00±0.00**	0.95±0.03	0.83±0.12	**1.00±0.00**	0.95±0.03
Saliency	**1.00±0.00**	0.90±0.01	**1.00±0.00**	0.76±0.11	**1.00±0.00**	0.93±0.03
Grad*Input	**1.00±0.00**	**1.00±0.00**	0.85±0.08	0.78±0.12	**1.00±0.00**	0.93±0.04
L2X	0.59±0.06	0.41±0.07	0.15±0.11	0.30±0.08	0.5±0.03	0.39±0.07
LIME	−0.72±0.0	−0.52±0.08	−0.14±0.07	−0.57±0.05	−0.3±0.06	−0.45±0.05

* The best Spearman correlation coefficient for each synthetic dataset is highlighted in bold

TABLE 2.

Precision of the second-order interpretation by LINA SP, NID and GEH in ten synthetic datasets (F6 F10).

Total Features	Datasets	NID	GEH	LINA SP
13 features	F6-A	44.5% ±0.2%	50.0%±0.2%	**61.8%±0.2%**
	F7-A	**98.0%±0.1%**	41.0%±0.2%	92.0%±0.1%
	F8-A	80.6%±0.2%	48.8%±0.4%	**85.0%±0.2%**
	F9-A	62.2%±0.4%	41.4%±0.3%	**70.0±0.3%**
	F10-A	56.7%±0.3%	75.0%±0.5%	**91.7%±0.3%**
	Average	68.4%±0.2%	51.2%±0.3%	**80.1±0.2%**
100 features	F6-B	51.8%±0.2%	18.1%±1.0%	**52.7%±0.3%**
	F7-B	44.0%±0.2%	28.8%±0.4%	**90.0%±0.0%**
	F8-B	76.3%±0.1%	47.9%±0.2%	**80%.0±0.3%**
	F9-B	40.0%±0.3%	41.8%±0.2%	**51.7%±0.3%**
	F10-B	**66.6%±0.0%**	10.4%±1.0%	50.0%±0.1%
	Average	55.7%±0.2%	29.4%±0.6%	64.9%±0.2%

* The best precision for each dataset is highlighted in bold

TABLE 3.

Performance benchmarking of the first-order interpretation for predictive genomics.

Methods	LINA FP	Saliency	grad*Input	DeepLift	LIME	L2X
# SNPs at 0.1% FDR	**127**	35	75	75	9	0
# SNPsatl%FDR	**158**	35	88	85	9	0
# SNPs at 5% FDR	**255**	57	122	119	9	0
FDR at top-100 SNP	**0.0%**	7.5%	3.0%	2.0%	16.3%	N/A
FDR at top-200 SNP	**1.5%**	16.2%	9.3%	9.3%	20.5%	N/A

TABLE 4.

Performance benchmarking of the second-order interpretation for predictive genomics.

Methods	LINA SP	NID	GEH
# SNP pairs at 0.1% FDR	**583**	415	0
# SNP pairs at 1% FDR	**1040**	504	0
# SNP pairs at 5% FDR	**2887**	810	0
FDR at top-1000 SNP pairs	**0.9%**	10.5%	N/A
FDR at top-2000 SNP pairs	**3.0%**	31.8%	N/A