Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

SPECIAL ISSUE PAPER

VtNet: a Neural Network with Variable Importance Assessment

Lixiang Zhang | Lin Lin | Jia Li*

[1]Department of Statistics, The Pennsylvania State University, University Park, PA 16802, U.S.A

Correspondence
*Jia Li, Department of Statistics, The Pennsylvania State University, State College, PA 16802, U.S.A Email: jol2@psu.edu

The architectures of many neural networks rely heavily on the underlying grid associated with the variables, for instance, the lattice of pixels in an image. For general biomedical data without a grid structure, the multi-layer perceptron (MLP) and deep belief network (DBN) are often used. However, in these networks, variables are treated homogeneously in the sense of network structure; and it is difficult to assess their individual importance. In this paper, we propose a novel neural network called Variable-block tree Net (VtNet) whose architecture is determined by an underlying tree with each node corresponding to a subset of variables. The tree is learned from the data to best capture the causal relationships among the variables. VtNet contains an LSTM-like cell for every tree node. The input and forget gates of each cell control the information flow through the node, and they are used to define a significance score for the variables. To validate the defined significance score, VtNet is trained using smaller trees with variables of low scores removed. Hypothesis tests are conducted to show that variables of higher scores influence classification more strongly. Comparison is made with the variable importance score defined in Random Forest from the aspect of variable selection. Our experiments demonstrate that VtNet is highly competitive in classification accuracy and can often improve accuracy by removing variables with low significance scores.

KEYWORDS:
VtNet, Variable-block, Causal Discovery Tree, LSTM-like Cell, Significance Score

1 | INTRODUCTION

Deep neural networks (DNN) have achieved phenomenal success in a broad spectrum of predictive data analysis problems, for instance, in computer vision (Voulodimos, Doulamis, Doulamis, & Protopapadakis 2018), natural language processing (Otter, Medina, & Kalita 2020) and speech recognition (Nassif, Shahin, Attili, Azzeh, & Shaalan 2019). In those fields, many neural network architectures have been designed for sequential or imagery data, which rely heavily on the underlying grid structure of the variables, e.g., the lattice of pixels in an image, the ordered appearance of words. Neural networks have also been designed for general graph data, specifically, each input instance being a graph with potentially varying topology. Examples include the recursive neural network (RvNN) (Socher et al. 2013) and graph autoencoder (GAE) (Cao, Lu, & Xu 2016). A comprehensive review of graph neural networks is referred to (Z. Wu et al. 2020). However, for general multivariate data such as those in the biomedical areas, the data are neither features on a grid nor graphs. In this case, available DNNs are much limited. The commonly used architectures are Multi-Layer Perceptron (MLP) and Deep Belief Networks (DBN) (G. Hinton, Osindero, & Teh 2006; G. E. Hinton 2009). They treat covariates homogeneously by fully-connected networks, and the importance of any covariate for predicting the output is opaque (Wankhede 2014). Although techniques have been developed to reduce the complexity of MLP, they do not directly address the issue of assessing the importance of variables.

One of the core tasks in biomedical studies is to identify the underlying causal relations and make use of them (Fergusson, Boden, & Horwood 2009; Kleinberg & Hripcsak 2011; Mente, de Koning, Shannon, & Anand 2009; Tyrrell et al. 2016; Williamson 2019). With the causal relationships

among covariates revealed, the consequences of actions can be better predicted (Hoyer, Janzing, Mooij, Peters, & Schölkopf 2009; Judea 2000), and new insights about health and disease can be gained (Cooper et al. 2015; Sun et al. 2015). While the "gold-standard" approach to infer causality is by randomization, it is often infeasible due to logistic, economic, and/or ethical constraints. On the other hand, with the availability of large amounts of observational data, e.g., The Cancer Genome Atlas (TCGA) (Network 2008), National Cancer Database (NCDB) (Bilimoria, Stewart, Winchester, & Ko 2008; Winchester, Stewart, Bura, & Scott Jones 2004), Surveillance, Epidemiology, and End Results Program (SEER) (Duggan, Anderson, Altekruse, Penberthy, & Sherman 2016), efforts have been devoted to deriving causality from observational data. Causal discovery among variables is a powerful technique for inferring causal structure from observational data without randomized assignment (Glymour, Zhang, & Spirtes 2019). A basic idea of causal discovery is to represent the causal relationship between variables using a directed acyclic graph. Various methods and algorithms have been developed, for example, the constraint and score-based methods (Yu, Li, & Liu 2016) and the Functional Causal Models algorithms (K. Zhang, Wang, Zhang, & Schölkopf 2015).

Motivated by the need to understand the role of variables in many biomedical data analysis tasks, we hereby develop a deep learning approach with the capacity to effectively assess variable importance and in the meanwhile achieve high classification accuracy. In a nutshell, our approach exploits causal discovery to build a directed graph for the variables, which is then leveraged to design a new DNN with reduced complexity and readiness for evaluating variable importance. Specifically, we develop a neural network called Variable-block tree Net (VtNet) for classification.

In VtNet, the variables are partitioned into blocks which correspond to nodes in the causal graph. A Long Short-Term Memory (LSTM)-like cell is used to model each node (Gers, Schmidhuber, & Cummins 1999; Hochreiter & Schmidhuber 1997). The causal graph is then approximated by spanning trees, which determine the overall structure of VtNet. By the nature of MLP, the weight matrices of the network cannot pinpoint the role of a variable. In contrast, the input and forget gates in an LSTM-like cell control the information flow, which reflects the importance of the current input. Therefore a significance score can be defined for variables in any node using the input and forget gates of the corresponding cell. A node with a low significance score means that the input variables of this node contribute little to the entire information flow and are expected to have a low influence on classification. As a comparison to the conventional LSTM popular for sequential data, the LSTM-like cells in VtNet do not have "time-wise" repetitive weights because the variables in each node are of different meanings and often different dimensions.

The rest of the paper is organized as follows. In Section 2, we review related work. Our proposed methods are presented in Section 3. Experiments and comparisons with baseline methods are described in Section 4. We also demonstrate the effectiveness of the variable significance score from several perspectives, such as hypothesis testing, correlation analysis, and variable selection. Finally, we conclude and discuss future work in Section 5.

2 | RELATED WORK

There is a growing interest in interpreting models trained by DNNs or other machine learning methods. Recent works include locally approximating the model around an individual prediction (Guo, Huang, Tao, Xing, & Lin 2018; Lundberg & Lee 2017; Ribeiro, Singh, & Guestrin 2016), explaining deep features trained from DNN using semantic features and original quantitative features (Paul et al. 2019), and assessing the importance of variables based on DNN, e.g., the recently developed RATE (Ish-Horowicz, Udwin, Flaxman, Filippi, & Crawford 2019) and CXPlain (Schwab & Karlen 2019) and other previous works (Shrikumar, Greenside, & Kundaje 2017; Simonyan, Vedaldi, & Zisserman 2013; Sundararajan, Taly, & Yan 2017). See Roscher, Bohn, Duarte, and Garcke (2020) for a survey on explainable machine learning. For machine learning methods other than DNN, how to reveal the roles of variables has also been explored (Lundberg et al. 2020 2018). Our work to assess the importance of original variables is in the category of "explaining variables". Quantifying the importance of variables can help researchers gain new insights or verify existing knowledge about certain measurements. Furthermore, the importance scores can be used to select variables. Although we leverage the structure of VtNet to define an easily computed significance score, our goal is not to explain the importance of variables based upon any given DNN, a difference from existing work. Instead, we aim at building new models that are relatively easy to explain and in the meanwhile accurate in classification. We also note that VtNet is motivated by biomedical data for which causal discovery is meaningful, but it is not suitable for images or audio signals which possess intrinsic underlying graph structures.

Our approach of constructing trees via causal discovery bears some similarity with the method of Tree Augmented Naive Bayes (TAN) (Friedman, Geiger, & Goldszmidt 1997). To determine the tree structure, one approach in TAN reduces the problem of constructing a maximum likelihood tree to one of finding the maximum weighted spanning tree in a graph. The edge weights are mutual information between variables (Vergara & Estévez 2014). The tree obtained as such is undirected. If a directed tree is desired, a node in the tree is randomly chosen as the root, and the choice of the root has no effect on the likelihood. In contrast, VtNet employs causal discovery which naturally yields a directed graph. The causal scores serve as the edge weights. A directed maximum weighted spanning tree is then sought out. Details can be found in Section 3.3. We have experimented with the method of TAN to build the tree graph for VtNet, but the result shows that few choices of the root node can provide

performance comparable to that based on causal discovery. Even when we applied an ensemble method to multiple randomly generated trees, the performance is still inferior to that of causal discovery.

Our proposed model is related to LSTM, which is a special Recurrent Neural Network (RNN) (Pascanu, Mikolov, & Bengio 2013; Williams & Zipser 1989) architecture. It contains a repetitive neural network module at any time position. Cascaded as a chain, these modules are "recurrent" since the weight matrices in each module are shared across the chain. We define an LSTM-like cell for each node in the causal graph, but these cells do not share weights. In Section 3.4, we explain why this structure enables the definition of variable significance scores. The LSTM cell has been used to model nodes in a tree rather than in a chain. In particular, (Tai, Socher, & Manning 2015; Zhu, Sobihani, & Guo 2015) have developed Tree-LSTM for semantic analysis in natural language processing. Although VtNet is structurally similar to Tree-LSTM, there are important differences. The input to Tree-LSTM contains instances each represented by a graph. The graph topology can vary, but the variables in each graph node are homogeneous in their meanings. As a result, the LSTM cells in Tree-LSTM are recursive (that is, identical). For VtNet, the input data are not graphs, while a causal graph is learned from the data to capture relationships among variables. Due to the different nature of input data, the LSTM-like cells in VtNet are unique for each node. In addition, the information flow in VtNet is from the root to the leaf nodes (top-down), while that in Tree-LSTM is the reverse (bottom-up).

Finally, we point out that deep learning has attracted growing interest in the statistics community. For example, (Tang, Ma, Waljee, & Zhu in press) proposed a semi-supervised joint learning method for classifying longitudinal clinical events. It is demonstrated that their proposed method outperforms the purely supervised method and the existing two-step semi-supervised methods. For subspace classification, (H. Wu, Fan, & Lv 2020) explored whether the DNN statistically mimics the two-step procedure of clustering followed by classification. (Y. Yuan, Deng, Zhang, & Qu 2020) introduced from a statistical perspective the general structures and applications of various DNNs, including convolutional neural networks and generative adversarial networks.

3 | MODEL CONSTRUCTION AND VARIABLE IMPORTANCE

3.1 | Notations

Denote a random vector X by $(X_1, X_2, ..., X_p)^T \in \mathbb{R}^p$ and the ith sample or realization of it by $(x_{i1}, x_{i2}, ..., x_{ip})^T \in \mathbb{R}^p$. Moreover, the data matrix $\mathbb{X} = (x_1, x_2, ..., x_p) \in \mathbb{R}^{n\times p}$, where x_j is the jth column of $\mathbb{X}$, containing values of the jth variable across all sample points. We use the terms feature and variable exchangeably. A variable block is a subset of the p features, e.g., (X_1, X_3). Suppose we partition the p-dimensional random vector X into V variable blocks, indexed by $v = 1, 2, ..., V$. Let the number of variables in the vth block be p_v, a.k.a., the dimension of the vth variable block. We have $\sum_{v=1}^{V} p_v = p$. Denote the sub-vector containing variables in the vth variable block by $X^{(v)}$. If we reorder variables in X according to the order of the variable blocks, we get the random vector $\widetilde{X} = (X^{(1)^T}, X^{(2)^T} ... X^{(V)^T})^T \in \mathbb{R}^p$. For brevity of notation, we assume without loss of generality $X^{(1)} = (X_1, X_2, ..., X_{p_1})^T \in \mathbb{R}^{p_1}$ and $X^{(v)} = (X_{m_v+1}, X_{m_v+2}, ..., X_{m_v+p_v})^T \in \mathbb{R}^{p_v}$, where $m_v = \sum_{i=1}^{v-1} p_i$, for $v = 2, ..., V$. Then we simply have $X = (X^{(1)^T}, X^{(2)^T} ... X^{(V)^T})^T \in \mathbb{R}^p$.

3.2 | Variable blocks

For sequential or spatial data, variables can usually be considered as attributes of nodes on a graph. For example, pixel-wise features in an image are attributes of nodes on a regular two-dimensional grid. In another word, the indices for different variables are not symbolic but correspond with positions on a grid. For non-sequential and non-spatial biomedical data, the indices of variables are symbolic and can be permuted without changing the nature of the data. In such cases, some generic DNNs such as MLP are used. In MLP, all variables of the input data are fully connected in the hidden layers via a weight matrix W and a bias vector b. However, underlying causal relationships between variables usually exist in biomedical data. Such relationships enable us to construct special architectures of neural networks, allowing lower network complexity and better interpretation for the roles of variables.

By considering variable blocks instead of only individual variables in each node, we can more effectively model the interaction among variables within the same node without compromising computational cost. Furthermore, if each node contains only one variable, the causal relationship graph can become too big, and the cells with one-dimensional input may require intensive pre-training (Li, Zhao, Huang, & Gong 2014). For the interest of variable selection, group-based selection has been much explored in statistics (M. Yuan & Lin 2006), although less so in the literature of neural networks (May, Dandy, & Maier 2011; G. P. Zhang 2000). We use a simple scheme to generate the variable blocks. We randomly select one seed variable and compute its correlation (or mutual information) with every other variable. Variables with the highest correlation with the seed variable are grouped with it to form a variable block. The size of the block can be decided by thresholding the correlation coefficients or by an upper bound on the number of variables permitted in one block. After one block is formed, the same process is applied to the remaining

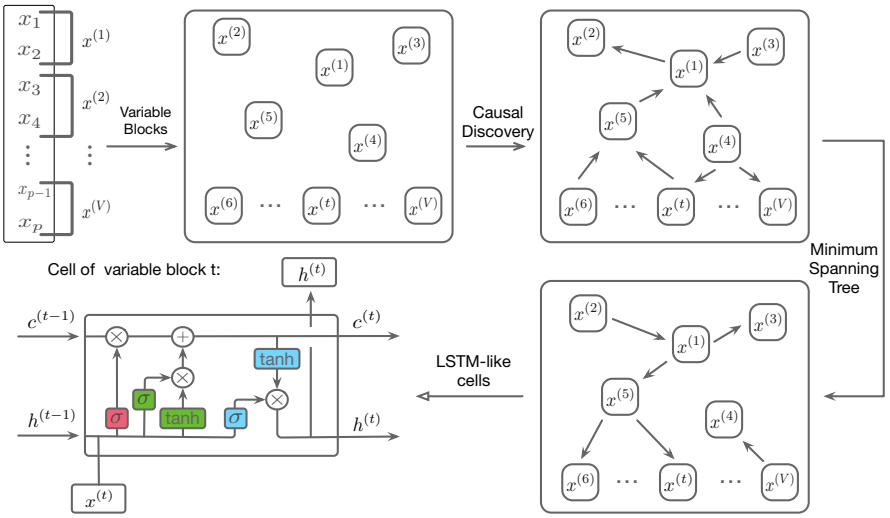

FIGURE 1 VtNet work flow. Individual Variables, $x_1, \ldots, x_p$, are first divided into variable blocks, $x^{(1)}, \ldots, x^{(V)}$. A directed causal graph is built on them by causal discovery. The graph will be further reduced to a minimum spanning tree, and variables are connected using the causal directed link. The structure of the neural network, VtNet, is determined by the tree, and LSTM-like cells are used to model every node (i.e., variable block). The red unit indicates the forget gate $f^{(t)}$; the green units are the input gate $i^{(t)}$ and input proposal $a^{(t)}$; and the blue square unit indicates the output gate $o^{(t)}$. The parameter matrices, e.g., $W_i^{(t)}$, $W_f^{(t)}$, and $W_o^{(t)}$, are not shared across the cells.

variables to form another block, so on and so forth. We finally obtain a partition of all variables as shown in Figure 1. Our experiments show that the difference between using correlation and mutual information is negligible. We use correlation because it is faster to compute.

3.3 | Causal discovery tree

Biomedical data analysis is often concerned with discovering and modeling causal relationships between variables, and various computational methods have been developed. Understanding causal relationships is especially important for predictive biomedical analysis, as it enables making predictions under interventions. Those well-studied causal discovery methods can represent the underlying causal knowledge as a directed graphical causal model, as shown in Figure 1. We adopt the Causal Generative Neural Networks (CGNN) proposed in (Goudet et al. 2017) to model the directed causal graph, in particular, using the implementation described in (Kalainathan, Goudet, & Dutta 2020). To the best of our knowledge, there is no causal discovery method designed specifically for groups of variables instead of individual variables. But the idea of grouping highly correlated variables and representing them by a single variable has been explored (Coumans, Claassen, & Terwijn 2017). In our approach, when each variable block has a quite small number of variables (fewer than 3 in practice), we randomly pick one variable from every block as the representative to construct the directed causal graph. When the variable block size is big, we use summary statistics such as mean value to represent the block. As information flow in a directed causal graph is not completely in one direction, we approximate the causal graph by a minimum spanning tree using the Chu–Liu/Edmonds' algorithm (Chu 1965; Edmonds 1967). The weight of each edge is defined as the negative of the causal score. Therefore the minimum spanning tree seeks the simplest structure of the largest cumulative causal score. If the directed causal graph has multiple nodes without an input or causal node, multiple trees will be obtained from the directed graph. Variables in these trees will be modeled separately at first, and then combined at the end by additional layers in the network. Details will be explained shortly when we present the network architecture.

3.4 | LSTM-like cell

The architecture of the cell we used to model each variable block is essentially that of LSTM, as shown in Figure 1. For the tth cell, it is typically composed of a memory cell $c^{(t)}$, a hidden state $h^{(t)}$, and 3 gates. The 3 gates are input gate $i^{(t)}$, output gate $o^{(t)}$ and forget gate $f^{(t)}$. The main difference here from a typical LSTM cell is that the parameters in each cell are not duplicates of one set of parameters. Instead, each cell has input data of a unique variable block and hence a unique set of parameters. Without loss of generality, suppose the input variable block to the tth cell is $X^{(t)}$ with realization $x^{(t)}$. The notation $\odot$ means element-wise multiplication, and σ is the sigmoid activation function with range $[0, 1]$, applied to each element. Then the tth cell is defined as:

$$i^{(t)} = \sigma(W_i^{(t)} h^{(t-1)} + U_i^{(t)} x^{(t)} + b_i^{(t)}),$$

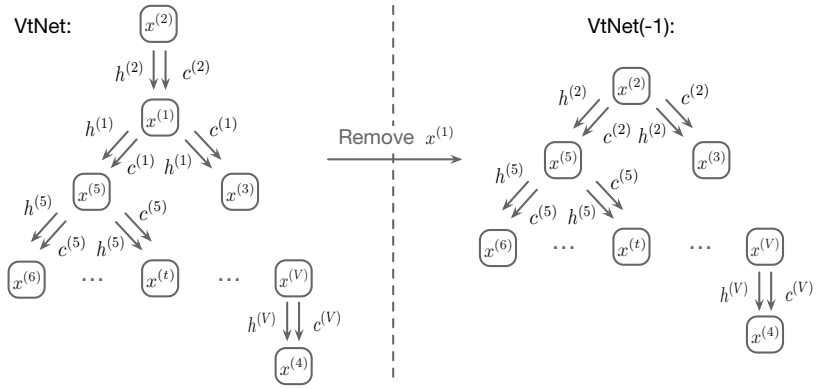

FIGURE 2 VtNet architecture and the shrinking technique. The tree structure is discovered by causal discovery. Each node is modeled by a LSTM-like cell, and passes its memory $c^{(t)}$ and hidden state $h^{(t)}$ to all the child nodes. When removing the node of the lowest significance score, for example $x^{(1)}$, all its child nodes are re-linked to its parent node $x^{(2)}$. The new architecture is called VtNet(-1).

$$
\begin{aligned}
f^{(t)} &= \sigma(W_f^{(t)} h^{(t-1)} + U_f^{(t)} x^{(t)} + b_f^{(t)}),\\
o^{(t)} &= \sigma(W_o^{(t)} h^{(t-1)} + U_o^{(t)} x^{(t)} + b_o^{(t)}),\\
a^{(t)} &= \tanh(W_a^{(t)} h^{(t-1)} + U_a^{(t)} x^{(t)} + b_a^{(t)}),\\
c^{(t)} &= c^{(t-1)} \odot f^{(t)} + i^{(t)} \odot a^{(t)},\\
h^{(t)} &= o^{(t)} \odot \tanh(c^{(t)}).
\end{aligned}
$$

For each sample, suppose $x^{(t)} \in \mathbb{R}^{p_t}$ and $h^{(t-1)} \in \mathbb{R}^{q}, \forall t$, then $U_i^{(t)}, U_f^{(t)}, U_o^{(t)}, U_a^{(t)} \in \mathbb{R}^{q \times p_t}$, $W_i^{(t)}, W_f^{(t)}, W_o^{(t)}, W_a^{(t)} \in \mathbb{R}^{q \times q}$ and $b_i^{(t)}, b_f^{(t)}, b_o^{(t)}, b_a^{(t)}, i_t^{(t)}, f_t^{(t)}, o_t^{(t)}, a_t^{(t)}, c_t^{(t)} \in \mathbb{R}^{q}$. In practice, we set the state size $q = \min(20, \min\{2 * p_t, t = 1, 2, ..., V\})$. Same as in the LSTM cell, information is delivered across the entire tree by the memory $c^{(t)}$ and hidden state $h^{(t)}$. The way of connecting these LSTM-cells into a graph structure will be described next.

The memory cell $c^{(t)}$ and hidden state $h^{(t)}$ summarize the information up to the tth node and are regulated by the gates to receive new information or erase irrelevant information. The forget gate $f^{(t)}$ receives $h^{(t-1)}$, the information from the previous module, and the new input data $x^{(t)}$. We can view $f^{(t)}$ as a proportion to control the usage of memory $c^{(t-1)}$ in the update $c^{(t)}$ in the next cell. A higher value of $f^{(t)}$ results in stronger influence of $c^{(t-1)}$ on $c^{(t)}$. If the effect of the ancestor nodes is negligible given the new input, $f^{(t)}$ approaches 0, or figuratively, the forget gate closes. Besides past memory, the other part of $c^{(t)}$ is based on the current $i^{(t)}$ and $a^{(t)}$, which are the input gate and input proposal respectively. The input proposal $a^{(t)}$ encodes the new information at t, while $i^{(t)} \in (0, 1)$ controls the proportion of $a^{(t)}$ that will be added in $c^{(t)}$. In summary, the forget gate $f^{(t)}$ and input gate $i^{(t)}$ can be viewed as two probability matrices, which reflect the relative importance of current input $x^{(t)}$. We are thus inspired to define a significance score for variables in the node based on the quantities at the gates.

3.5 | VtNet architecture and interpretation of variable importance

Suppose a tree structure has been determined by causal discovery (Figure 1). The corresponding VtNet architecture is illustrated in the left panel in Figure 2. Every parent node passes the same information through memory $c^{(t)}$ and hidden state $h^{(t)}$ to all its child nodes. A softmax layer is added after the information has passed through all the leaf nodes, and the hidden state $h^{(t)}$ of every leaf node is used as input to this last layer. The softmax layer outputs the probability of a data point belonging to each class. Mini-batch (Cotter, Shamir, Srebro, & Sridharan 2011) and Adam optimizer (Kingma & Ba 2014) are used to train the VtNet. As aforementioned, there can be multiple trees generated based on the causal graph. Currently, the leaf nodes of all the trees are connected to the softmax layer.

To interpret quantitatively the working scheme of VtNet, we assign each node in the tree a significance score based on the input gate and forget gate of the LSTM-like cell. Recall that the input gate $i^{(t)}$ or forget gate $f^{(t)}$ is a vector for each data point with elements in the range $[0, 1]$. We view these values as weights that control the information flow. Denote the matrix containing $i^{(t)}$'s across all the training data points by $(i^{(t)})$, and similarly define the notation $(f^{(t)})$. At any node, a small value in $(f^{(t)})$ indicates that the proportion of previous information to forget is large. If the value at the input gate is big in the meantime, this means that the information of this variable block is important relative to the previous information. To account for the effects from both the input and forget gates, we define the significance score as follows. Denote the significance score of variable block t by $\mathcal{S}_t$. Let $||\cdot||_F$ denote the Frobenius norm of matrix. Then

$$\mathcal{S}_t \triangleq \frac{||(i^{(t)})||_F}{||(f^{(t)})||_F}.$$

TABLE 1 Summary of the 8 datsets after pre-processing, where #VB represents number of variable blocks used.

Dataset	Size	Dim	#VB	Classes	Dataset	Size	Dim	#VB	Classes
SKCM	388	30	15	2	Yan	124	483	15	7
STAD	393	24	12	2	Pollen-high	301	1767	15	11
KIRC	430	24	12	2	Pollen-low	301	1593	15	11
LUAD	455	20	10	2	Deng	317	276	15	9

Note that when $||(\mathsf{f}^{(\mathsf{t})})||_{\mathsf{F}} = 0$, we define $\mathcal{S}_{\mathsf{t}}$ to be $+\infty$ as the previous information is negligible comparing to current variables, which rarely happens in practice. Moreover, 1 is a meaningful threshold for the significance score, as a significance score greater than 1 means the current input information is more important than the previous information. The effectiveness and validity of this definition will be examined in Section 4.

If it is necessary to identify important individual variables, we can define variable blocks each containing a single variable when the data dimension is low. However, if the dimension is high, this approach will generate an overwhelmingly large number of nodes for causal discovery. We would thus suggest using our approach as a screening tool. Specifically, via VtNet, we can identify blocks of variables that are important in the sense of the whole block. Once the block-wise selection of variables has narrowed down the set of variables to consider, we can examine the importance of individual variables by combinatorial approaches, e.g., leave-one-out.

To perform variable selection based on the significance score $\mathcal{S}_{\mathsf{t}}$, we simply shrink the tree graph of VtNet by removing nodes in the order of lowest significance score first. This operation is illustrated in Figure 2. When removing a node, its descendants are not removed (different from pruning). Instead, the removed node is bypassed, that is, in the new tree, its child nodes become the children of its parent. We use the terminology VtNet(-k) to mean the network trained on the tree with k nodes removed. We can remove multiple nodes, one in each round, until a criterion is met. Deciding when to stop removing nodes is a trade-off between model complexity and accuracy. Currently, we suggest to threshold the significance score at 1 because a score greater than 1 indicates that the input variables of this node have more influential information to pass down comparing with the previous information the node receives. Experimental results on real datasets are provided in Section 4.

4 | EXPERIMENTS

We now evaluate the classification performance of VtNet on 8 gold standard biomedical datasets and design hypothesis testing to validate the significance scores defined for variables. More specifically, 4 datasets are obtained from The Cancer Genome Atlas (TCGA). They are respectively the Skin Cutaneous Melanoma (SKCM), Stomach Adenocarcinoma (STAD), Kidney Renal Clear Cell Carcinoma (KIRC), and Lung Adenocarcinoma (LUAD) (National Cancer Institute n.d.) datasets. All their response variables are the patient's survival status: survival or death. The other 4 are single-cell data with pre-determined cell labels/types. They are named after the papers where they first appeared: Yan data (Yan et al. 2013), Pollen-high data, Pollen-low data (Pollen et al. 2014), and Deng data (Deng, Ramsköld, Reinius, & Sandberg 2014). The basic information about these pre-processed datasets is given in Table 1. We removed missing data and uninformative features from the TCGA datasets. For the single-cell data, each variable is the expression level of a particular gene. The genes with variances smaller than the average variance are filtered out. Table 1 shows the size and dimension of each dataset after pre-processing.

For each dataset, the number of variables in every block is set to be roughly equal. The number of variable blocks for any of the survival analysis datasets is determined by assuming 2 variables in each block; and the number of blocks is set to 15 for all the single-cell datasets. We also experimented with different numbers of variable blocks and found rather small differences in classification performance. In Table 2, results under VtNet* and VtNet** are obtained from VtNet with the numbers of variable blocks other than the default setting. Specifically, for the TCGA datasets, a variable block contains at most 3 variables in VtNet*, and at most 4 variables in VtNet**. For the single-cell datasets, VtNet* has 12 variable blocks, and VtNet** has 10 variable blocks. To evaluate the performance, 30% of the data are used for testing, and the remaining 70% for training. Classification accuracy is computed on the test data. For every dataset, we repeat the experiments 10 times with different random splits of training and test data.

In Table 2, we report the average performance over the 10 runs and the standard deviation (values in the parenthesis). For the case of binary classification (all the TCGA datasets), the average AUC (area under the ROC curve) is also evaluated. For comparison, we use as baselines a few popular classifiers in biomedical data analysis, in particular, MLP, logistic regression with $\mathcal{L}2$ penalty (LG), random forest (RF) (Breiman 2001) and support vector machine (SVM) (Suykens & Vandewalle 1999) with radial basis function kernel. Moreover, we experimented with DBN and found that a well-trained MLP with refining techniques like dropout usually outperforms a simple DBN, which is not surprising since DBN can be viewed

TABLE 2 Results on 8 datasets for TCGA binary classification and cell type classification. The highest accuracy/AUC for each dataset is marked out in bold.

	LG	RF	SVM	MLP	VtNet	VtNet(-1)	VtNet(-2)	VtNet(-3)	VtNet*	VtNet**
Dataset	Acc. (sd)	Acc. (sd)	Acc. (sd)	Acc. (sd)	Acc. (sd)	Acc. (sd)	Acc. (sd)	Acc. (sd)	Acc. (sd)	Acc. (sd)
SKCM	0.611 (0.08)	0.641 (0.07)	0.710 (0.03)	0.767 (0.02)	0.797 (0.02)	**0.803** (0.03)	0.796 (0.03)	0.795 (0.02)	0.773 (0.04)	0.789 (0.03)
STAD	0.620 (0.03)	0.653 (0.04)	0.645 (0.05)	0.653 (0.03)	0.669 (0.04)	0.666 (0.04)	0.677 (0.04)	**0.678** (0.05)	0.663 (0.04)	0.664 (0.04)
KIRC	0.648 (0.03)	0.696 (0.08)	0.749 (0.05)	0.771 (0.03)	0.780 (0.03)	0.773 (0.03)			**0.789** (0.04)	0.772 (0.04)
LUAD	0.668 (0.02)	0.615 (0.06)	0.728 (0.02)	0.729 (0.03)	0.733 (0.03)	0.738 (0.03)	**0.741** (0.03)		0.740 (0.04)	0.734 (0.02)
Yan	0.803 (0.10)	0.853 (0.04)	0.224 (0.04)	0.931 (0.04)	**0.947** (0.08)				0.945 (0.06)	0.936 (0.06)
Pollen-high	0.957 (0.02)	0.965 (0.02)	0.162 (0.04)	**0.974** (0.02)	0.965 (0.02)				0.962 (0.01)	0.973 (0.02)
Pollen-low	0.941 (0.02)	0.934 (0.02)	0.162 (0.04)	0.960 (0.02)	0.953 (0.01)				0.946 (0.02)	**0.962** (0.01)
Deng	0.726 (0.08)	0.793 (0.07)	0.204 (0.02)	0.882 (0.02)	0.888 (0.02)				**0.891** (0.03)	0.887 (0.03)
Dataset	AUC (sd)	AUC (sd)	AUC (sd)	AUC (sd)	AUC (sd)	AUC (sd)	AUC (sd)	AUC (sd)	AUC (sd)	AUC (sd)
SKCM	0.668 (0.04)	0.752 (0.05)	0.824 (0.03)	0.841 (0.03)	0.845 (0.03)	**0.847** (0.04)	0.845 (0.04)	0.846 (0.04)	0.838 (0.03)	0.846 (0.03)
STAD	0.647 (0.05)	0.680 (0.04)	0.668 (0.05)	0.681 (0.05)	0.698 (0.06)	0.698 (0.06)	0.697 (0.06)	0.698 (0.06)	0.700 (0.05)	**0.705** (0.04)
KIRC	0.816 (0.04)	0.792 (0.07)	0.851 (0.04)	**0.871** (0.03)	0.857 (0.04)	0.857 (0.03)			0.853 (0.04)	0.852 (0.03)
LUAD	0.721 (0.04)	0.556 (0.10)	0.727 (0.04)	0.732 (0.05)	0.741 (0.04)	**0.743** (0.04)	0.741 (0.04)		0.739 (0.04)	0.736 (0.04)

as a special initialization technique for MLP. For brevity, we only list the results of MLP but not DBN. To examine variable selection based on the significance score, we remove the node of the lowest significance score one at a time until all the nodes have a significance score greater than 1.

4.1 | Classification performance

Table 2 shows the classification results for the 8 datasets. VtNet(-1) means that the model is trained with one node in the original tree removed. That node must have a significance score below 1 according to the original VtNet as well as the lowest score among all the nodes. If no node with a score below 1 exists in the original VtNet, we only report the result for VtNet. Similarly, if VtNet(-1) has a node with the lowest significance score below 1, that node will be further removed and we obtain VtNet(-2). For every change in the tree structure, the VtNet is retrained.

The results show that MLP and VtNet achieve similar average accuracy or AUC's, but they outperform LG, RF, and SVM in most cases. Importantly, comparing the results of VtNet(-1), VtNet(-2), and VtNet(-3) with VtNet, we see that with the nodes of the lowest significance scores removed, the classification performance remains very close to the original model or sometimes is even better. For SKCM, VtNet(-1) achieves the best average accuracy and best average AUC among all the models.

TABLE 3 Results of hypothesis testing and correlation analysis to validate the significance score of variables.

	SKCM	STAD	KIRC	LUAD
Accuracy test p-value	0.064	0.098	0.001	0.002
AUC test p-value	0.090	0.001	0.002	0.047
Accuracy correlation	-0.357	-0.241	-0.152	-0.419
AUC correlation	-0.350	-0.611	-0.472	-0.276

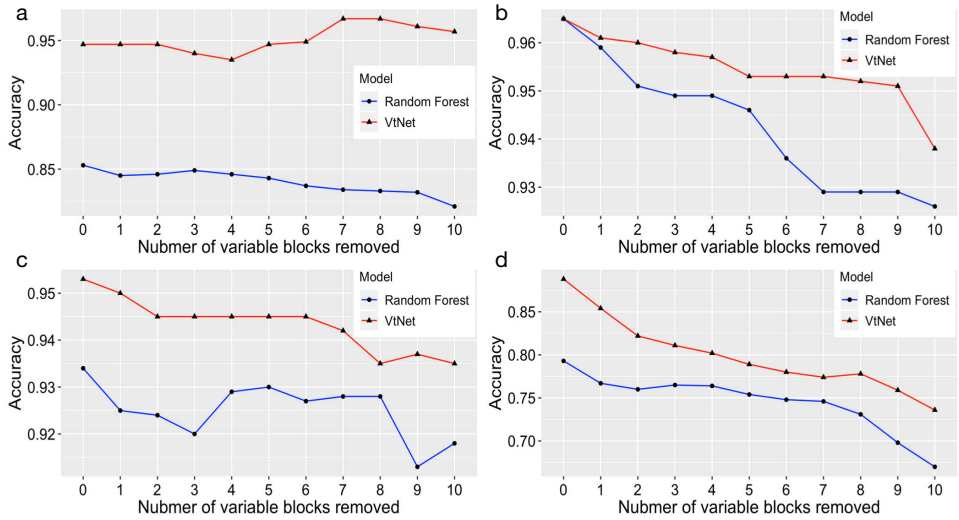

FIGURE 3 Comparison of variable selection between VtNet and RF on single-cell datasets. (a) Yan data. The performance of VtNet(-k), k $= 0, \ldots, 10$, is substantially better than RF. VtNet(-k), k $= 6, \ldots, 10$ achieves higher accuracy than VtNet(-0), showcasing the potential to enhance classification by selecting variables. (b) Pollen-high data. At any reduced dimension, the accuracy of RF decreases more comparing with the original dimension. (c) Pollen-low data. (d) Deng data. For the last two datasets, the performance gap between VtNet and RF varies considerably across the dimensions, with VtNet always outperforming RF.

4.2 | Hypothesis testing to validate the significance score

In order to validate the definition of the significance score, we designed experiments to conduct hypothesis testing and correlation analysis, the idea similar to that of Bickel (2020), in which hypothesis testing is used to evaluate the performance of DNNs. After training VtNet, we compute the significance score of each node and divide the nodes into two groups, those with scores < 1 (Group 1) and those > 1 (Group 2). For the datasets that have more than 4 nodes with scores < 1, specifically, SKCM, STAD, KIRC, and LUAD, the numbers of variables in these two groups are balanced. We formally test whether VtNets obtained by randomly removing one node from Group 1 achieve better classification accuracy than those of VtNets obtained by removing one node from Group 2 (in the sense of the distribution).

For each group, we repeat the experiment 20 times, each time randomly removing one node. We recorded the accuracy and AUC of each experiment. As no distributional assumption is made about the values of the accuracy (or AUC), a non-parametric rank test is used to test whether the VtNets obtained by removing nodes in the two groups respectively perform equally well. The Mann-Whitney U test (Mann & Whitney 1947) is carried out on both accuracy and AUC. The alternative hypothesis is that the VtNets with nodes in Group 1 (lower significance scores) removed perform better. In the meanwhile, we calculate the correlation between the significance score of the removed node and the accuracy (or AUC). The results are provided in Table 3. At a significance level $\alpha = 0.1$, the null hypothesis of all the tests on either accuracy or AUC would be rejected. In addition, all the correlations between the significance score of the removed node and the accuracy (or AUC) are negative. It is evident that the significance score hereby defined reflects well the importance of a variable for classification.

4.3 | Select and interpret the roles of variables

Variable selection for classification is much pursued in gene expression studies. A score to quantify the importance of variables not only enables us to quickly find a subset of genes with little or no compromise in classification accuracy but also helps to reveal more pinpointed relationships between variables and classes. For example, we may find that certain genes are valuable for distinguishing some classes, but not so relevant for some other classes. Understanding such relationships between genes and classes can be useful for diagnostics purposes when only part of the classes are in consideration for a particular population (Tanner et al. 2008).

Here we examine the usefulness of the significance score $\mathcal{S}_t$ from two aspects. First, we want to see what insights can be gained when variables with low significance scores are removed. Second, we investigate whether this score can enable effective selection of variables. In this regard, we compare variable selection based respectively on our $\mathcal{S}_t$ and the importance score provided by RF. For both VtNet and RF, the scores are readily available as a byproduct of the trained models. Therefore these two approaches are most natural to compare. For the first task, we discuss Deng data in details. For the second task, we show the results on all the single-cell datasets.

VtNet achieves the best accuracy among all the methods on Deng data, while all nodes have significance scores over 1. This result indicates that all the variables contribute substantially to classification. For a thorough analysis, we removed the variable blocks (a total of 15) one at a time until we have removed 10 blocks which contain more than 65% of all the genes. A great decline of accuracy exists between VtNet(-1) (0.854) and VtNet(-2) (0.822). We find that 60% of the increase in the error rate is caused by 30% of samples in class 2 which are misclassified as class 3. This observation indicates that the variables removed influence mostly the distinction of class 2 and 3. Another big gap in accuracy is between VtNet(-9) (0.760) and VtNet(-10) (0.730). Before removing the 10th variable block, test samples from classes 1, 3, and 9 are all correctly classified, which account for respectively 3.8%, 4.4%, 7.3% of the single-cell population. If the researchers are especially interested in distinguishing these classes, the first 9 variable blocks can be skipped. Similar analysis shows that for Pollen-high data, classes 1-5 can be distinguished without the first 10 variable blocks.

One popular variable selection approach based on RF is to rank variables by the Gini-based variable importance (Genuer, Poggi, & Tuleau-Malot 2010). Let us refer to the VtNet with k variable blocks removed as VtNet(-k), $k = 0, \ldots, 10$. Suppose the number of variables removed in VtNet(-k) is $\tilde{d}_k$, $k = 1, \ldots, 10$. To compare RF with VtNet(-k), we remove correspondingly $\tilde{d}_k$ variables with the smallest Gini-based importance scores and retrain the RF. There is a slight amount of randomness in the execution of RF and VtNet due to the bootstrap samples used in RF and the Mini-batch technique for training a neural net. We thus run the experiment 20 times for each algorithm under any setup and report the average accuracy. The comparison for all the single-cell datasets is shown in Figure 3. For the Yan and Pollen-high data, the drop in accuracy when more variables are removed is considerably faster for RF than VtNet. For the other two datasets, the overall drop is similar. Interestingly, in the case of the Yan data, the performance of VtNet(-k), $k = 6, \ldots, 10$, is superior to that of the original VtNet(-0). The boost of performance by variable selection is not observed with RF. At lower dimensions, the performance gap between VtNet and RF becomes wider for the Yan and Pollen-high data.

5 | CONCLUSIONS AND FUTURE WORK

We have developed a novel neural network architecture, namely, VtNet, aiming at biomedical data classification. The architecture is determined by an underlying tree that best captures the causal relationships among the variables, while the tree nodes are variable blocks. The tree graph is learned from the data, and hence dynamic. Experimental results show that VtNet achieves better classification accuracy than MLP for seven out of the eight datasets, and better accuracy than RF for seven out of the eight datasets (tied on the other one). Moreover, VtNet enables us to define an easily computed significance score to interpret the role of each variable block. The effectiveness of this score is substantiated by hypothesis testing and by comparing the performance of variable selection with RF. One potential future work is to develop an ensemble version of VtNet. It would also be interesting to compare the VtNet significance score with more computationally intensive variable importance scores developed for more general DNNs.

DATA AVAILABILITY

TCGA datasets are downloaded using R package TCGAretriever (Fantini 2019), and single-cell datasets are from (Deng et al. 2014; Pollen et al. 2014; Yan et al. 2013).

ACKNOWLEDGMENTS

This research work is partially supported by NSF grant DMS-2013905.

References

Bickel, D. R. (2020). Testing prediction algorithms as null hypotheses: Application to assessing the performance of deep neural networks. *Stat*, *9*(1), e270.

Bilimoria, K. Y., Stewart, A. K., Winchester, D. P., & Ko, C. Y. (2008). The national cancer data base: a powerful initiative to improve cancer care in the united states. *Annals of surgical oncology*, *15*(3), 683–690.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Cao, S., Lu, W., & Xu, Q. (2016). Deep neural networks for learning graph representations. In *Aaai* (Vol. 16, pp. 1145–1152).

Chu, Y.-J. (1965). On the shortest arborescence of a directed graph. *Scientia Sinica*, *14*, 1396–1400.

Cooper, G. F., Bahar, I., Becich, M. J., Benos, P. V., Berg, J., Espino, J. U., … Lee, A. V. (2015). The center for causal discovery of biomedical knowledge from big data. *Journal of the American Medical Informatics Association*, *22*(6), 1132–1136.

Cotter, A., Shamir, O., Srebro, N., & Sridharan, K. (2011). Better mini-batch algorithms via accelerated gradient methods. In *Advances in neural information processing systems* (pp. 1647–1655).

Coumans, V., Claassen, T., & Terwijn, S. (2017). Causal discovery algorithms and real world systems.

Deng, Q., Ramsköld, D., Reinius, B., & Sandberg, R. (2014). Single-cell rna-seq reveals dynamic, random monoallelic gene expression in mammalian cells. *Science*, *343*(6167), 193–196.

Duggan, M. A., Anderson, W. F., Altekruse, S., Penberthy, L., & Sherman, M. E. (2016). The surveillance, epidemiology and end results (seer) program and pathology: towards strengthening the critical relationship. *The American journal of surgical pathology*, *40*(12), e94.

Edmonds, J. (1967). Optimum branchings. *Journal of Research of the national Bureau of Standards B*, *71*(4), 233–240.

Fantini, D. (2019). Tcgaretriever: Retrieve genomic and clinical data from tcga [Computer software manual]. Retrieved from `https://CRAN.R-project.org/package=TCGAretriever` R package version 1.5.

Fergusson, D. M., Boden, J. M., & Horwood, L. J. (2009). Tests of causal links between alcohol abuse or dependence and major depression. *Archives of general psychiatry*, *66*(3), 260–266.

Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian network classifiers. *Machine learning*, *29*(2-3), 131–163.

Genuer, R., Poggi, J.-M., & Tuleau-Malot, C. (2010). Variable selection using random forests. *Pattern recognition letters*, *31*(14), 2225–2236.

Gers, F., Schmidhuber, J., & Cummins, F. (1999). Learning to forget: Continual prediction with LSTM.

Glymour, C., Zhang, K., & Spirtes, P. (2019). Review of causal discovery methods based on graphical models. *Frontiers in Genetics*, *10*.

Goudet, O., Kalainathan, D., Caillou, P., Guyon, I., Lopez-Paz, D., & Sebag, M. (2017). Causal generative neural networks. *arXiv preprint arXiv:1711.08936*.

Guo, W., Huang, S., Tao, Y., Xing, X., & Lin, L. (2018). Explaining deep learning models–a bayesian non-parametric approach. In *Advances in neural information processing systems* (pp. 4514–4524).

Hinton, G., Osindero, S., & Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, *18*(7), 1527–1554.

Hinton, G. E. (2009). Deep belief networks. *Scholarpedia*, *4*(5), 5947.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Hoyer, P. O., Janzing, D., Mooij, J. M., Peters, J., & Schölkopf, B. (2009). Nonlinear causal discovery with additive noise models. In *Advances in neural information processing systems* (pp. 689–696).

Ish-Horowicz, J., Udwin, D., Flaxman, S., Filippi, S., & Crawford, L. (2019). Interpreting deep neural networks through variable importance. *arXiv preprint arXiv:1901.09839*.

Judea, P. (2000). Causality: models, reasoning, and inference. *Cambridge University Press. ISBN 0*, *521*(77362), 8.

Kalainathan, D., Goudet, O., & Dutta, R. (2020). Causal discovery toolbox: uncovering causal relationships in python. *Journal of Machine Learning Research*, *21*(37), 1–5.

Kingma, D., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kleinberg, S., & Hripcsak, G. (2011). A review of causal inference for biomedical informatics. *Journal of biomedical informatics*, *44*(6), 1102–1112.

Li, J., Zhao, R., Huang, J.-T., & Gong, Y. (2014). Learning small-size dnn with output-distribution-based criteria. In *Fifteenth annual conference of the international speech communication association.*

Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., … Lee, S.-I. (2020). From local explanations to global understanding with

explainable ai for trees. *Nature machine intelligence*, *2*(1), 2522–5839.

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in neural information processing systems* (pp. 4765–4774).

Lundberg, S. M., Nair, B., Vavilala, M. S., Horibe, M., Eisses, M. J., Adams, T., … Kim, J. (2018). Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. *Nature biomedical engineering*, *2*(10), 749–760.

Mann, H. B., & Whitney, D. R. (1947). On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, 50–60.

May, R., Dandy, G., & Maier, H. (2011). Review of input variable selection methods for artificial neural networks. *Artificial neural networks-methodological advances and biomedical applications*, *10*, 16004.

Mente, A., de Koning, L., Shannon, H. S., & Anand, S. S. (2009). A systematic review of the evidence supporting a causal link between dietary factors and coronary heart disease. *Archives of internal medicine*, *169*(7), 659–669.

Nassif, A. B., Shahin, I., Attili, I., Azzeh, M., & Shaalan, K. (2019). Speech recognition using deep neural networks: A systematic review. *IEEE Access*, *7*, 19143–19165.

National Cancer Institute. (n.d.). *Cancer genome atlas*. `https://www.cancer.gov/about-nci/organization/ccg/research/structural-genomics/tcga`. Accessed: 2020-09-03.

Network, C. G. A. R. (2008). Comprehensive genomic characterization defines human glioblastoma genes and core pathways. *Nature*, *455*(7216), 1061.

Otter, D. W., Medina, J. R., & Kalita, J. K. (2020). A survey of the usages of deep learning for natural language processing. *IEEE Transactions on Neural Networks and Learning Systems*.

Pascanu, R., Mikolov, T., & Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *International conference on machine learning* (pp. 1310–1318).

Paul, R., Schabath, M., Balagurunathan, Y., Liu, Y., Li, Q., Gillies, R., … Goldgof, D. B. (2019). Explaining deep features using radiologist-defined semantic features and traditional quantitative features. *Tomography*, *5*(1), 192.

Pollen, A., Nowakowski, T., Shuga, J., Wang, X., Leyrat, A., Lui, J., … Chen, P. (2014). Low-coverage single-cell mrna sequencing reveals cellular heterogeneity and activated signaling pathways in developing cerebral cortex. *Nature Biotechnology*, *32*(10), 1053.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 1135–1144).

Roscher, R., Bohn, B., Duarte, M. F., & Garcke, J. (2020). Explainable machine learning for scientific insights and discoveries. *IEEE Access*, *8*, 42200–42216.

Schwab, P., & Karlen, W. (2019). Cxplain: Causal explanations for model interpretation under uncertainty. In *Advances in neural information processing systems* (pp. 10220–10230).

Shrikumar, A., Greenside, P., & Kundaje, A. (2017). Learning important features through propagating activation differences. In *Proceedings of the 34th international conference on machine learning-volume 70* (pp. 3145–3153).

Simonyan, K., Vedaldi, A., & Zisserman, A. (2013). Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., & Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing* (pp. 1631–1642).

Sun, Y., Li, J., Liu, J., Chow, C., Sun, B., & Wang, R. (2015). Using causal discovery for feature selection in multivariate numerical time series. *Machine Learning*, *101*(1-3), 377–395.

Sundararajan, M., Taly, A., & Yan, Q. (2017). Axiomatic attribution for deep networks. In *Proceedings of the 34th international conference on machine learning-volume 70* (pp. 3319–3328).

Suykens, J., & Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural Processing Letters*, *9*(3), 293–300.

Tai, K. S., Socher, R., & Manning, C. D. (2015). Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*.

Tang, W., Ma, J., Waljee, A. K., & Zhu, J. (in press). Semi-supervised joint learning for longitudinal clinical events classification using neural network models. *Stat*, e305.

Tanner, S. D., Bandura, D. R., Ornatsky, O., Baranov, V. I., Nitz, M., & Winnik, M. (2008). Flow cytometer with mass spectrometer detection for massively multiplexed single-cell biomarker assay. *Pure and Applied Chemistry*, *80*(12), 2627–2641.

Tyrrell, J., Richmond, R. C., Palmer, T. M., Feenstra, B., Rangarajan, J., Metrustry, S., … De Silva, N. M. G. (2016). Genetic evidence for causal relationships between maternal obesity-related traits and birth weight. *Jama*, *315*(11), 1129–1140.

Vergara, J. R., & Estévez, P. A. (2014). A review of feature selection methods based on mutual information. *Neural computing and applications*,

24(1), 175–186.

Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, *2018*.

Wankhede, S. B. (2014). Analytical study of neural network techniques: Som, mlp and classifier-a survey. *IOSR Journal of Computer Engineering*, *16*(3), 86–92.

Williams, R., & Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, *1*(2), 270–280.

Williamson, J. (2019). Establishing causal claims in medicine. *International Studies in the Philosophy of Science*, *32*(1), 33–61.

Winchester, D. P., Stewart, A. K., Bura, C., & Scott Jones, R. (2004). The national cancer data base: a clinical surveillance and quality improvement tool. *Journal of surgical oncology*, *85*(1), 1–3.

Wu, H., Fan, Y., & Lv, J. (2020). Statistical insights into deep neural network learning in subspace classification. *Stat*, *9*(1), e273.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Philip, S. Y. (2020). A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*.

Yan, L., Yang, M., Guo, H., Yang, L., Wu, J., Li, R., … Yan, J. (2013). Single-cell rna-seq profiling of human preimplantation embryos and embryonic stem cells. *Nature structural & molecular biology*, *20*(9), 1131.

Yu, K., Li, J., & Liu, L. (2016). A review on algorithms for constraint-based causal discovery. *arXiv preprint arXiv:1611.03977*.

Yuan, M., & Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *68*(1), 49–67.

Yuan, Y., Deng, Y., Zhang, Y., & Qu, A. (2020). Deep learning from a statistical perspective. *Stat*, *9*(1), e294.

Zhang, G. P. (2000). Neural networks for classification: a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *30*(4), 451–462.

Zhang, K., Wang, Z., Zhang, J., & Schölkopf, B. (2015). On estimation of functional causal models: general results and application to the post-nonlinear causal model. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *7*(2), 1–22.

Zhu, X., Sobihani, P., & Guo, H. (2015). Long short-term memory over recursive structures. In *International conference on machine learning* (pp. 1604–1612).

How to cite this article: Zhang L., Lin. L, and Li. J (2020), VtNet: a Neural Network with Variable Importance Assessment, *Stat*, *2017;00:1–6*.