Effective Interpretable Learning for Large-Scale Categorical Data

Yishuo Zhang[1†], Nayyar Zaidi[1*†], Jiahui Zhou[2], Tao Wang[4], Gang Li[1]

[1]School of Information Technology, Deakin University, Melbourne, VIC, Australia.

[2]Asia-Pacific Academy of Economics and Management, University of Macau, Macau, China.

[4]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China.

*Corresponding author(s). E-mail(s): nayyar.zaidi@deakin.edu.au;
Contributing authors: zhangyis@deakin.edu.au;
kaemihara@outlook.com; wangtao@iie.ac.cn; gang.li@deakin.edu.au;
†These authors contributed equally to this work.

Abstract

Large scale categorical datasets are ubiquitous in machine learning and the success of most deployed machine learning models rely on how effectively the features are engineered. For large-scale datasets, parametric methods are generally used, among which three strategies for feature engineering are quite common. The first strategy focuses on managing the breadth (or width) of a network, e.g., generalized linear models (aka. `wide learning`). The second strategy focuses on the depth of a network, e.g., Artificial Neural networks or `ANN` (aka. `deep learning`). The third strategy relies on factorizing the interaction terms, e.g., Factorization Machines (aka. `factorized learning`). Each of these strategies brings its own advantages and disadvantages. Recently, it has been shown that for categorical data, combination of various strategies leads to excellent results. For example, `WD`-Learning, `xdeepFM`, etc., leads to state-of-the-art results. Following the trend, in this work, we have proposed another learning framework – `WBDF`-Learning, based on the combination of `wide`, `deep`, `factorization`, and a newly introduced component named `Broad Interaction network` (`BIN`). `BIN` is in the form of a Bayesian network classifier whose structure is learned apriori, and parameters are learned by optimizing a joint objective function along with `wide`, `deep` and `factorized` parts. We denote the learning of `BIN` parameters as `broad learning`. Additionally, the parameters of `BIN` are constrained to be actual probabilities – therefore, it is extremely interpretable. Furthermore, one can sample or generate data from `BIN`, which can facilitate learning and provides a framework for *knowledge-guided machine learning*. We demonstrate that our proposed framework possesses the resilience to maintain excellent classification performance when confronted with biased datasets. We evaluate the efficacy of our framework in terms of classification performance on various benchmark large-scale categorical datasets and compare against state-of-the-art methods. It is shown that, `WBDF` framework a) exhibits superior performance on classification tasks, b) boasts outstanding interpretability and c) demonstrates exceptional resilience and effectiveness in scenarios involving skewed distributions.

Keywords: Low-bias Models, Large Categorical Datasets, Feature Engineering, Interpretable Models, Discriminative Bayesian network Models

1 Introduction

Feature engineering is the key to building better machine learning models in the era of big data [1], [2]. The ever-increasing volume of datasets in today's world motivates the need for building models with low bias, such that higher-order interactions among features, if present – can be modeled correctly. Note, for larger quantities of data, the variance component of the error tends to be zero, and the bias component is the one that generally derives the error [3]. This is one of the reasons behind the success of parametric models such as deep Artificial Neural networks ($\texttt{ANN}^d$) models (d characterizes the depth of the model), Higher-order Logistic Regression ($\texttt{LR}^n$) models (n characterizes the order of the features considered in the model, e.g., $n = 1$ for linear, $n = 2$ for quadratic, etc.), Factorization Machine ($\texttt{FM}^m$) models (where like n in higher-order logistic regression, m characterizes the order of the features considered in the model, e.g., $m = 1$ for linear, $m = 2$ for quadratic, etc.), and non-parametric models such as Random Forest (`RF`) [4], Gradient Boosting Decision Trees, (`GBT`) etc. [5]. Generally, for large-scale datasets, parametric methods are preferred as non-parametric methods like `GBT` require loading entire data into the memory. $\texttt{ANN}^d$ relies on the strategy of adding layers to build deep models, which gives them the capacity to engineer features. We refer to this form of learning as `deep`-learning [1]. *Generalized Linear Models* (`GLM`), represent another category of parametric models which have been proven to be effective for large datasets, though computationally inefficient [5]. We refer to this form of learning as `wide`-learning, as the model complexity can be controlled through the width of the input layer. In general, capturing higher than quadratic-level interactions is a challenge for such models. $\texttt{FM}^m$ represents the latest category of parametric models, which relies on factorizing the interaction among the variables such that the obtained final non-linear model is linear in the input parameters [6]. Factorization Machines and its variants have been the models of choice when

[1] Various layers of `ANN` (e.g., `dense, convolution, recurrence`) serves as feature engineering modules for modeling higher-order feature interactions and hence leading to a low-bias model.

learning from extremely large quantities of categorical datasets. We refer to this form of learning as Factorized-learning.

It has been shown that for large-scale categorical data, combination of various strategies leads to excellent results. E.g., WD-Learning is an end-to-end framework combining wide and deep learning [7]. xdeepFM, the existing state-of-the-art (SOTA) model, is the combination of deep and factorization strategies [5]. However, these frameworks do have some drawbacks:

- Capturing higher-order interactions in the model can be difficult. E.g., in the case of WD-Learning, for moderate-size datasets, obtaining all possible cubic or higher-order features is next to impossible. Most state-of-the-art (SOTA) models rely on deep component to engineer higher-order features.
- Existing SOTA frameworks have limited interpretable capabilities. Though one can interpret the parameters of the wide part, since the parameters are free parameters (and not probabilities), interpretation can be difficult. Any form of interpretation in models such as xdeepFM is not possible.
- The incorporation of human knowledge is quite limited. For example, one can make use of prior knowledge that some feature combinations never occur – leading to a form of feature selection, but this is limited to order-1 or order-2 features only in WD learning. Incorporation of such knowledge is not possible in xdeepFM.
- Learning from biased datasets can be challenging for existing state-of-the-art (SOTA) frameworks. In fact, there is no mechanism in these frameworks that mitigate or eliminate the impact of bias present in the dataset.

How to obtain SOTA model with the capabilities of a) constructing higher-order explicit features, b) superior interpretability, and c) knowledge-guided learning for bias correctness and other related benefits has been the main motivation of this work.

How can one incorporate knowledge in machine learning models? Well, a simple solution is that of the Bayesian network. A Bayesian network is a directed acyclic graph, that depicts the dependence of features in a problem. The model is excellent for incorporating expert knowledge, which is generally done through the specification of the structure of a Bayesian network, as well as related probabilities. Bayesian networks are also super-interpretable, and we will discuss later that they can also formulate higher-order interactions, hence leading to a framework that can incorporate cubic or higher-order interactions easily. Moreover, since the Bayesian network can sample and generate synthetic data, knowledge-guided learning is achievable as one can repeatedly sample a desirable dataset (e.g., an un-biased dataset) during or prior to training.

So, are Bayesian networks a panacea? Well, the assumptions encoded by the structure of the Bayesian network can be incorrect, which can limit the efficacy of the model. A wide or deep model, on the other hand, has fewer assumptions than Bayesian network [2]. How can one get the benefits of both Bayesian networks as well as wide and deep learning? A simple strategy is that of the ensemble, where Bayesian networks, wide and deep constitute individual models. In fact, ensembling of disparate models has a long history in machine learning [8] and has shown to be quite effective.

[2] A wide model has all possible order-n terms, and a deep model given sufficient depth (d) can be a universal approximater.

One drawback of ensembling is that the individual models are trained separately and their collaborations only occur when combining their predictions. *Is there a way, we can integrate Bayesian networks with* `wide` *and* `deep` *models in a single model within an end-to-end learning framework (i.e., avoiding ensemble strategy)? In the following, we will show that we can do this by learning the parameters of Bayesian networks by optimizing a similar loss function that is typically optimized by* `wide` *and* `deep` *models. This will provide us a way to integrate Bayesian network with other parametric models including not only* `wide` *and* `deep`*, but also* `factorized` *models as well.*

In this work, we denote a Bayesian network that is trained by optimizing `conditional log-likelihood` (a discriminative objective function) as `Broad Interaction network` (`BIN`). Of course, the parameters of `BIN` are constrained to be actual probabilities, and once they are learned, they allow the importance of high-order feature interactions to be observed clearly, which adds interpretable capabilities to the model [3]. In this work, we have proposed a new framework that integrates `BIN` with `wide` and `deep` models. To the best of our knowledge, this is the first work that integrates Bayesian networks in an end-to-end fashion in a deep learning framework. The model is designed to be applied to extremely large categorical data. One property of categorical data is that they can be extremely sparse. For example, a `city` feature can be a list of all the cities in the world and can be either 1 or 0. It has been shown that on these sparse categorical datasets, `factorized` models, e.g., `Factorization Machines` can be quite effective, as they can leverage the present values to learn a weight for values that are not present in the data. Therefore, to better model categorical data, we have proposed to integrate `BIN` with not only `wide` and `deep` models, but also with `factorized` model resulting in our proposed framework *wide, broad, deep and factorized learning* (`WBDF`-Learning framework). The term *broad* denotes the `BIN` component of the framework and learning the parameters of `BIN` is denoted as `broad learning`.

About the term '`Broad`' – The term `broad` learning has been introduced before in [10], where it is described as fast and accurate learning without a deep structure and is different to our work. On the other hand, terms `Wide`, `Deep` and `Factorized` learning are well-known terms in machine learning. To define `broad`-learning, let us formalize `wide` learning first. The term wide is used in machine learning research for models such as quadratic, cubic or higher-order linear or logistic regression. The term was first used in [7], where authors proposed an ensemble of quadratic logistic regression and deep learning models. They characterized quadratic logistic regression as a `wide` model. We define `wide` learning as:

Definition 1. *`Wide` Learning incorporates learning with all possible n-level interaction features in the data.*

Based on the above definition, $\texttt{LR}^n$ is a form of `wide` learning. We define `broad`-learning as:

Definition 2. *`Broad` Learning incorporates learning with a subset of all possible n-level interaction features, where the subset is either specified by an expert, obtained based on some pre-specified metric or learned from another data source.*

[3] Note, in `BIN`, one can interpret the model parameters during the training as well. This is in contrast to `LIME` [9] – another popular interpretation method widely used in industry, which offers only the post-training explanation.

Based on the above definition, $\texttt{BN}^k$ is an example of broad learning. Another example of broad learning is higher-order feature selection based on measures such as *mutual information* [11], followed by a simple linear model.

On the inclusion of both 'Wide' and 'Broad' Components – As we discussed earlier, Broad learning such as BIN is not a panacea, and subset of feature-interactions selected by the model can be incorrect (after all, we are using some metric to select a subset of interactions from all possible interactions). The main benefit of using Broad learning is because of its ability to scale to higher feature interactions. However, the inclusion of wide component can result in better results, of course at the expense of the computational cost of the size of the model. Typically, we can afford a smaller wide part and a much bigger broad part.

On the interpretability of prediction in WBDF Framework – We will see that only the BIN component in our framework is interpretable, however, all the four components contribute to the prediction. So the interpretability that we get from BIN depends on how much the broad component is contributing to the actual prediction. We will show later in the experiments that the performance and contribution of the broad component is usually higher than the other components. Secondly, we will be using the attention layer that is built on top of the output of each component. This layer is also interpretable and one can easily determine the contribution of each of the components, and the interpretation analysis can be based on how relevant broad component is in the overall prediction. In summary, much of the interpretation capability of WBDF is due to its broad component, as well as attention mechanism. Attention can help in interpreting the extent of broad component in final prediction, whereas each of the parameter of BIN can be interpreted to determine the contribution of each feature. Of course, explaining model's output in terms of all four components, i.e., wide, deep, factorized and broad is highly desirable, but is a challenging endeavour – one that is left as a future direction of this work.

On knowledge-guided learning of WBDF framework – In the last few years, knowledge-guided machine learning has gained a lot of traction [12]. The idea is to facilitate traditional learning by incorporating additional/auxiliary information (or knowledge) about the problem during the training stage. It has been shown that inclusion of such knowledge can improve traditional machine learning models. E.g., one strategy is to obtain a physical model of the process under study and obtain some additional data based on this physical model – the obtained data is used in conjunction with existing training data. The additional information that is present in this newly obtained data can improve classification but it can also correct for any bias that is present in the original data. E.g., a bank computing the credit score of its customer might be faced with data that consists of customers only from one city – however, as part of knowledge-guided machine learning, they can generate data for other cities which can mitigate some effects of bias. In WBDF framework, as we mentioned earlier, BIN has an advantage, that it can be trained prior with sufficient data, hence acting as the knowledge system. This is equivalent to structure learning of the Bayesian network, where a human in the loop can help specify or verify the structure of the network. Now, one can sample from this model, and the resulting data can be used in augmentation during the training of WBDF. In a nutsell, by leveraging the

BIN's sampling function capability, we can learn and train with some expert domain knowledge, generate additional data through sampling, expand the original (biased) dataset, and ultimately improve classification performance.

On the importance of Bayesian network structure – No doubt, BIN is an important component of our proposed WBDF framework. What if the assumptions encoded by the Bayesian network in BIN are incorrect? Well, structure learning is a crucial aspect of BIN, and it is important that a correct (or a desirable) structure is specified. In this work, we have constrained ourselves to restricted Bayesian network structure learning, i.e., a structure is learned prior to learning the parameters of BIN component. Note, a restricted Bayesian network uses information such as mutual information or conditional mutual information to determine the structure, such as TAN [13] or KDB [14], etc. This structure has shown to be extremely effective, leading to a scalable model which leads to state-of-the-art results on massive datasets [15]. How to incorporate structure learning with-in BIN is a challenging problem. There are several recent advances such as [16] and [17] in structure learning, that we believe can be integrated in BIN. However, this is left as a future work.

The contributions of this work are as follows:

1. We propose an elegant framework that constitutes Wide, Broad, Deep and Factorized components, to obtain a low-bias model for large volumes of data. The integration is done through an interpretable *Attention* mechanism.
2. By comparing on two standard CTR prediction datasets, 10 large categorical datasets and two synthetic datasets, we demonstrate that the WBDF-Learning framework has better performance than SOTA models such as FNN, CrossNet, xdeepFM, etc., as well as superior training time and a faster convergence profile. On standard CTR dataset Criteo, our method leads to state-of-the-art results based on the experimental settings in [18].
3. We study WBDF framework under knowledge-guided learning scenario. Under this scenario, BIN is used to sample a new dataset to correct for some artificially introduced bias. We compare the performance of vanilla WBDF and WBDF with bias correction through data sampling capability of BIN.
4. We evaluate the interpretability of our model and demonstrate its effectiveness by comparing it with the popular LIME model.

The rest of the paper is organized as follows. We will discuss related work in Section 2, and then present the proposed learning framework in Section 3. The experimental evaluation of our proposed framework is conducted in Section 4. We conclude in Section 5 with pointers to future works.

2 Related Works

WD-Learning is the classical hybrid model, which achieves both memorization and generalization in one model by jointly training a wide linear model component and a deep neural network component [19].

FNN model is a variant of Factorization Machines (FM) model, and can be considered as an FM-initialized ANN [20]. The idea is simple – an FM which is trained prior to

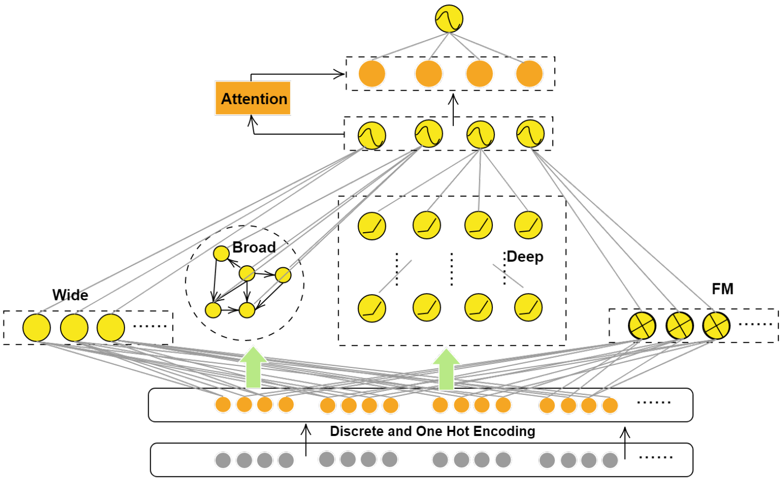

Fig. 1: Illustration of the WBDF framework.

training an ANN, is used as the representation of each categorical feature in the data. The structure of FNN allows the bottom layer to exploit spatially local correlation among the features. The supervised-learning embedding layer within the FNN using factorization machines reduces the dimension from sparse features to dense continuous features. After the embedding layer, multiple dense layers are built.

DeepCrossNet (DCN) [21] has a slightly different architecture compared to FNN. The CrossNet (cross-network) component in DeepCrossNet intends to explicitly generate higher-order feature by a cross operation, which is a special type of higher-order feature interaction by having the hidden layer output as the scalar product of the input layer. Repeated hidden layers can conduct the higher-order interactions according to existing ones and preserve the interactions from previous layers. Then the output of CrossNet is concatenated with a dense ANN to obtain the final prediction as DCN.

xdeepFM [5] is the recent SOTA model which achieves the most accurate performance. It has two levels of feature interaction namely, bit-wise and vector-wise. The bit-wise is the feature interaction that happens on the value of the feature directly, whereas, the vector-wise is the feature interaction that happens on the dimension of the feature matrix. The interacted object in vector-wise feature interaction is not the single value of the feature but the vector from the embedding matrix of the feature. The Compressed Interaction network (CIN) is the main component that can handle the vector-wise interaction. The xdeepFM combines ANN with an explicit vector-wise fashion compressed interaction network to learn the higher order features. It can be seen that xdeepFM is based on CrossNet but at a bit-wise level operation, and it applies the feature interaction at the vector-wise level.

3 WBDF Learning

Our proposed WBDF framework is composed of wide, deep, factorized and broad components, followed by an Attention layer, resulting in an effective yet interpretable end-to-end system. An overview of the entire framework is given in Figure 1. In the following subsections, let us discuss various components of the framework.

3.1 The Wide,Deep and Factorized Components

The wide component of WBDF-Learning framework is simply a linear model with the form:

$$\mathcal{O}_{\text{wide}}(y|\mathbf{x}) = \sigma_w(\mathbf{W}^\top \tilde{\mathbf{x}} + \mathbf{c}), \tag{1}$$

where σ_w denotes the Softmax for the Wide component. The parameter $\mathbf{W}$ is the parameters of the wide network, and $\mathbf{c}$ is the bias term [4]. The input data $\tilde{\mathbf{x}}$ is the transformed features from the input features – $\mathbf{x}$. The cross-product features that are computed can be expressed as: $\tilde{\mathbf{x}} = \phi_2(\mathbf{x}) = \prod_{i=1}^{p} \prod_{j=i}^{p} [x_i x_j]$, where $\phi_2(.)$ represents a quadratic transformation of the input data. In general, the wide part can be as wide as possible, e.g., an order-n transformation can be written as:

$$\tilde{\mathbf{x}} = \phi_n(\mathbf{x}) = \prod_{i=1}^{p} \prod_{j=i}^{p} \cdots \prod_{k=n-1}^{p} [x_i x_j \ldots x_k]. \tag{2}$$

As discussed in Section 1, going beyond $n = 2$ is not trivial on even moderate-size datasets, as it significantly increases the time and space complexity. In our proposed formulation of WBDF-Learning, the usage of the wide component is limited to $n = 2$, similar to other SOTA frameworks. Note that the feature transformation of Equation 2 is preceded by discretization of numeric features, leading to all categorical features in the dataset. Therefore, $x_i x_j \ldots x_k$ denotes the cross-product of n categorical features.

The Deep component of our WBDF-learning framework consists of a typical feed-forward deep neural network with the input embedding vector $\hat{\mathbf{x}}$. An embedding of each input feature is learned as part of the network, hence we define this transformation of input features as: $\hat{\mathbf{x}} = \Phi_{\text{Embedding-Layer}}(\mathbf{x})$. In general, the Deep component can be as deep as possible, e.g., a depth-**d** network can be defined as:

$$\mathcal{O}_{\text{deep}}(y|\mathbf{x}) \;=\; \sigma_d(\mathbf{D}^{\mathbf{d}} \mathbf{h}^{\mathbf{d-1}}). \tag{3}$$

Here σ_d denotes the activation function at layer d, and $\mathbf{D}^d$ denotes the parameters to be learned at that layer. $\mathbf{h}^d$ denotes the output of the hidden layer d [5]. Just like the wide component, the input data is first discretized leading to all categorical features in the data. Note that typical ANN can handle numeric features. However, for leveraging embedding layers in ANN and to be consistent with the wide component, deep component of our WBDF framework takes only categorical features.

[4] Note, in the following discussion, we will subsume parameter $\mathbf{c}$ in $\mathbf{W}$ for sake of simplicity.

[5] Note, generally deep models have several other layers such as batch-normalization, dropout etc., and we have only shown dense layers here.

A typical `factorized` model takes the following form:

$$\mathrm{P}(y|\mathbf{x}) = \sigma_f(\sum_{i=1}^{p}\sum_{j=i+1}^{p}\langle F_i^2, F_j^2\rangle x_i x_j + \sum_{i=1}^{p} F_i^1 x_i + \mathbf{c}).$$

Here, F^1 and F^2 denotes the linear and quadratic parameters of the model. However, our `WBDF` framework relies on the recent advancements in `factorized` learning, and utilize *Compressed Interaction network* (`CIN`) from [5], which performs a layer-wise operation on the data to obtain feature maps:

$$\mathbf{x}_{h,*}^{k} = \sum_{i=1}^{H_{k-1}}\sum_{j=1}^{p} F_{i,j}^{k,h}(\mathbf{x}_{i,*}^{k-1} \circ \mathbf{x}_{j,*}^{0}).$$

Note, p denotes the number features, $1 \leq h \leq H_k$ and H_k denotes the number of embedding feature vectors in k-th layer. Precisely, k-th layer here indicates the k-th hidden layer of the `CIN` and the $k \in [1, m]$. In rest of the paper, we donate the m as the total layer size in `CIN`. The features maps are then pooled as: $p_i^k = \sum_{j=1}^{D} \mathbf{x}_{i,j}^k$ for $i \in [1, H_k]$, where D is the dimension of embedding. A pooling vector of size H_k is generated as: $\mathbf{p} = [p_1^k, \ldots p_{H_k}^k]$ for layer k, and for all layers as: $\mathbf{p}^+ = [\mathbf{p}^1, \ldots \mathbf{p}^T]$. Here T denotes the depth of the network or the number of layers. We write the `factorized` component in `WBDF` as:

$$\mathcal{O}_{\text{fact}}(y|\mathbf{x}) = \sigma_f(F^o \mathbf{p}^+), \tag{4}$$

where σ_f denotes the `Softmax`, and F^o are its parameters.

3.2 The `Broad` Component

We denote the `broad` component of our model as `Broad Interaction network` (`BIN`). It is based on a restricted Bayesian network and exploits the trick of [22] by learning a separate set of parameters ($\mathbf{B}$ in our framework) by optimizing a discriminative objective function, i.e., `conditional log-likelihood (CLL)`. This discriminative training of parameters gives us the capability to use a Bayesian network alongside a `Wide` and `Deep` components in an end-to-end system. A Bayesian network is a directed acyclic graph. There are two component of a Bayesian network – structure and the parameters. The first stage of Bayesian network involves structure learning, which is followed by parameter learning. Structure learning incorporates learning the structure that is a directed edge is added among the features of the dataset. Structure learning problem comprises of adding, deleting and reversing edges such that some criteria score is optimized. Such approaches to structure learning are called un-restricted approaches. Restricted approaches on the other hand rely on simple heuristics such as mutual information or conditional mutual information to determine an optimal structure. In `BIN`, the structure is learned via an un-restricted structure learning. In a way – `BIN` has no control over the structure learning (the structure is given to `BIN`), it only plays the role of the second stage of Bayesian network learning – i.e., the parameter learning.

What does the parameters of Bayesian network look like? Well, let us make use of over-parameterization trick of [22], and define our Bayesian network (`broad`) model as:

$$\mathcal{O}_{\text{broad}}(y|\mathbf{x}) \propto \theta_y^{b_y} \prod_{i=1}^{p} \theta_{x_i|y,\Pi_i(\mathbf{x})}^{b_{x_i|y,\Pi_i(\mathbf{x})}}, \tag{5}$$

which is parameterized by two set of parameters: $b_. \in \mathbf{B}$ and $\theta_. \in \Theta$. The subscript $x_i|y, \Pi(\mathbf{x})$ denotes the parameters correspond to following feature interaction: attribute i, class attribute with value y and set of attribute values returned by function $\Pi()$. Note, $\Pi()$ is a function that return the parents of each attribute based on a Bayesian network's structure, and it can be seen that we have been able to learn a weight for various high-lever interactions in the data. Note, structure learning of Bayesian network is basically the specification of $\Pi()$ function, where we specify the parents of each attribute. In structure learning, one can limit the maximum number of parents an attribute can take. This is specified by the parameter k that controls the complexity of the model.

Now that we have established the structure learning component of Bayesian network in Equation 5, let us focus on the parameter learning in Equation 5. There are two set of parameters – Θ and $\mathbf{B}$. The parameter Θ constitutes of parameters which are learned by optimizing the log-likelihood (`LL`) of the data, and hence are equal to the actual empirical probabilities. The parameter Θ can be called as the generative parameters. The parameter $\mathbf{B}$ is optimized by optimizing the conditional log-likelihood (`CLL`), and often described as the discriminative parameters. The parameter $\mathbf{B}$ is optimized as part of (discriminative) training of `BIN` in `WBDF` framework. Important to note that unlike Θ, which are set of probabilities – the parameter $\mathbf{B}$ is not constrained, and can take on any value. We can write conditional probabilities in succinct form as:

$$\mathcal{O}_{\text{broad}}(y|\mathbf{x}) = \sigma_b(\mathbf{B}^T \log \Theta). \tag{6}$$

where σ_b denotes the `Softmax`. Again, Θ in Equation 6 is a generative parameter and is learned prior to training a `BIN`, whereas, $\mathbf{B}$ is a discriminative parameter and is learned by optimizing the objective function of `WBDF`, i.e., learned alongside the parameters of `wide`, `deep` and `factorized` models.

It can be seen that Equation 6 is over-parameterized. Rather than learning the parameter Θ apriori, by optimizing the `LL` objective function, we can learn it by optimizing the `CLL` objective function instead. Note, the parameter Θ are actual probabilities, and therefore, one will have to enforce the probability constraints during the optimization. The conditional probabilities in this case represent a slight variant of Equation 6, and we define it as the output of our `Broad Interaction network`, which differentiates our formulation with that of [22]:

$$\mathcal{O}_{\text{broad}}(y|\mathbf{x}) = \sigma_b(\log \Theta), \tag{7}$$
$$\text{with } \Theta \text{ satisfying probability constraints.}$$

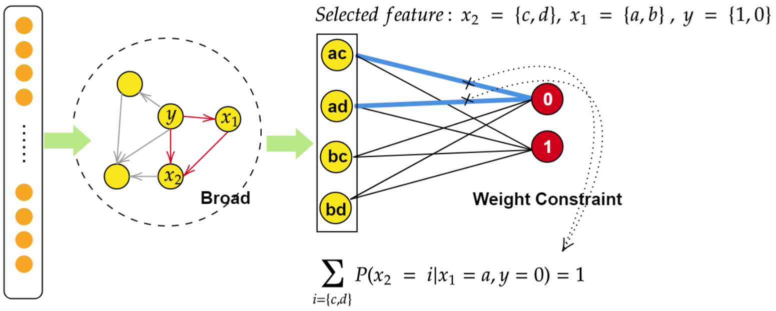
Selected feature: $x_2 = \{c,d\}$, $x_1 = \{a,b\}$, $y = \{1,0\}$
ac
ad
bc
bd
0
1
y
x_1
x_2
Broad
Weight Constraint
$\sum_{i=\{c,d\}} P(x_2 = i|x_1 = a, y = 0) = 1$

Fig. 2: Illustration of the `Broad Interaction network` with weight constraints.

Now, Θ in Equation 7 is a discriminative parameter and is learned by optimizing the objective function of `WBDF`, i.e., learned alongside the parameters of `wide`, `deep` and `factorized` models.

One can use `BIN` in our framework by either optimizing Equation 6 or 7. *Note, these two variants lead to similar results (in terms of the classification performance) as we show later in ablation studies, but optimizing Equation 7 leads to slower training time as it maintains the probability constraints over its parameters. It, however, leads to lesser number of parameters in the model, with an easy interpretation of parameters to be optimized, i.e., Θ – and hence is used as the default setting.*

3.3 `WBDF` Training

`WBDF` learning framework offers an elegant combination of `Wide`, `Broad`, `Deep` and `Factorized` components, in which the parameters of each component are trained in an end-to-end fashion. Our joint optimization will update the weights of all four components simultaneously connected together via `Attention` layer, by back-propagating the error while optimizing a single objective function. The use of `Attention` in `WBDF` is motivated from its enormous success in NLP and related domains and serves two purposes:

- Firstly, the `Attention` layer before the final output layer can provide the component-level interpretation. And, therefore, the importance of each component in `WBDF` can be easily determined.
- Secondly, the `Attention` can work as the gate for controlling the influence of different components during the training. Therefore it automatically decides which component to trust more and hence wight more in order to optimize the objective function.

We define the `Attention` layer in `WBDF` framework as:

$$\begin{aligned} \mathbf{z} &= \mathcal{O}_{\text{wide}}(y|\mathbf{x}) \oplus \mathcal{O}_{\text{broad}}(y|\mathbf{x}) \oplus \mathcal{O}_{\text{deep}}(y|\mathbf{x}) \oplus \mathcal{O}_{\text{factorized}}(y|\mathbf{x}), \\ \tilde{y} &= \text{Softmax}(\text{RELU}(h.\mathbf{z} + h_0)), \end{aligned} \tag{8}$$

Algorithm 1 WBDF Algorithm

```
Input  : $\mathcal{D}, n, k, d, m$
Output: Learned $\mathbf{W}, \mathbf{B}, \mathbf{F}, \mathbf{D}, \Theta, h$
Discertize numeric features in the data.
Compute MI and Conditional MI on the $\mathcal{D}$ and learn the structure of Bayesian network.
Initialize $\mathbf{W}, \mathbf{B}, \mathbf{F}, \mathbf{D}, \Theta, h$ to appropriate initializers.
for iteration $k \subset K$ do
    Calculate $\mathcal{O}_{\text{wide}}(y|\mathbf{x})$ ;                // Equation 1
    Calculate $\mathcal{O}_{\text{deep}}(y|\mathbf{x})$ ;                // Equation 3
    Calculate $\mathcal{O}_{\text{factorized}}(y|\mathbf{x})$ ;          // Equation 4
    Calculate $\mathcal{O}_{\text{broad}}(y|\mathbf{x})$ ;               // Equation 7
    Calculate $\tilde{y}$ ;                                              // Equation 8
    $\mathbf{g}^k \leftarrow \nabla_{\mathbf{W},\mathbf{B},\mathbf{F},\mathbf{D},\Theta,h}\mathcal{L}(y, \tilde{y})$
     $[\mathbf{W}^{k+1}, \mathbf{B}^{k+1}, \mathbf{F}^{k+1}, \mathbf{D}^{k+1}, \Theta^{k+1}, h^{k+1}] \leftarrow [\mathbf{W}^k, \mathbf{B}^k, \mathbf{D}^k, \Theta^k, h^k] + \eta\mathbf{g}^k$ ;
     // $\eta$:StepSize
end
return $\boldsymbol{W}, \boldsymbol{B}, \boldsymbol{F}, \boldsymbol{D}, \Theta, h$
```

where $\oplus$ denotes concatenation and h and h_0 denotes the parameters of the Attention layer Based on Equation (8), the objective function of WBDF-Learning framework can be written as:

$$\min_{\mathbf{W},\mathbf{B},\mathbf{D},\mathbf{F},\Theta,h} \quad \mathcal{L}(y, \tilde{y}) = -(y \log(\tilde{y}) + (1 - y) \log(1 - \tilde{y})), \tag{9}$$

Here, $\mathcal{L}$ denotes the objective function – and is typically the cross-entropy loss, which is optimized via standard gradient-descent based optimization algorithm such as adam solver. Note, for sake of completeness, we have included both Θ, and $\mathbf{B}$ for broad component. As we discussed in Section 3.2, one can learn Θ apriori, and can only learn $\mathbf{B}$ (Equation 6) during the optimization of Equation 9. Other option is – one can ignore $\mathbf{B}$, and learn Θ only (Equation 7) during the optimization of Equation 9.

We present the pseudo-code of WBDF-Learning framework in Algorithm 1.

3.4 Knowledge-guided WBDF

Let us in this section discuss how WBDF can integrate knowledge-guided machine learning through its capacity to generate unbiased data. Note, knowledge-guided machine learning is a vast area with many applications, however, we have only considered the application of bias-correctness as a representative application in this work. As we mentioned earlier, deep, wide and factorized classification models do not possess the capacity to correct for biased data – they can only be trained using the given (biased) dataset. Therefore, with biased datasets, the training outcome of these models can be sub-optimal. The usual remediation involves resorting to other generative models for data augmentation outside the framework. Our broad component, however, presents a

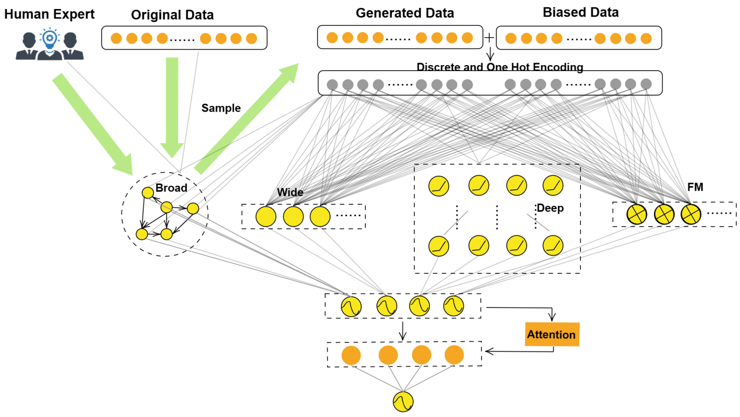

Fig. 3: Illustration of knowledge-guided application of WBDF.

solution to generate data with-in the learning framework. This capability enables WBDF to correct for the bias in original data by either learning Bayesian network structure from another data source, or by involving an expert in the loop in structure learning [6]. Afterward, the WBDF can be employed to fit the generated data – i.e., it is trained on the mixture of the biased dataset along with the un-biased generated dataset. This is illustrated in Figure 3 – where we show the framework of the WBDF under knowledge-guided learning. Here, `Original data` signifies the comprehensive knowledge of the entire domain, whereas `Biased data` represents the given biased training data. If no `Original data` is present, another form of structure learning is through expert in the loop – where a human expert can help learn the structure, from which data can be sampled.

Unlike the proposed WBDF framework, when dealing with `Biased data`, the `broad` component is trained using the complete domain knowledge, hence obtaining a synthetic dataset – while `wide, deep` and `factorized` are trained with mixture of `Generated data` and `Biased data`. The framework does this by setting the `is-learnable` parameter for each component during the training process.

4 Experiments

In this section, we will empirically evaluate the efficacy of our proposed WBDF-Learning framework by comparing its performance against other related methods on standard datasets, as well as on synthetic datasets. In addition, we have constructed a contrived knowledge-guided scenario featuring biased distributions and measured the lift on accuracy afterwards.

In particular, we are interested in finding answers to the following research questions:

[6] Note, in the experiments, we will use an ingenious strategy based on feature importance from tree-based models, to generate bias data, as for most of datasets used in our work, we do not have the expert available or presence of the un-biased version of the datasets.

Table 1: Statistics of the Datasets

Dataset	**Instance**	**features**	**Class**	**Source**
Criteo	45M	39	2	CTR
Avazu	40M	23	2	CTR
Higgs	1M	28	2	UCI
Sussy	1M	18	2	UCI
KDD99	1M	41	5	UCI
NSL-KDD99	1M	41	5	UCI
Pokerhand-4	1M	9	8	Synthetic
Pokerhand-6	1M	13	8	Synthetic
Pokerhand	1M	11	10	UCI
Covtype	581K	55	7	UCI
Localization	164K	7	11	UCI
Adult	48K	14	2	UCI

- **RQ1:** How well does WBDF-Learning framework perform over different datasets when compared to its individual components?
- **RQ2:** How does the proposed WBDF-Learning framework perform against other low-bias SOTA models which rely on effective feature engineering, such as WD, xdeepFM, etc. especially on CTR datasets.
- **RQ3:** What is the effect of introducing the broad component? E.g., what interpretation capabilities does it bring?
- **RQ4:** What is the effectiveness of WBDF in knowledge-guided machine learning scenario handling biased data?

Before embarking on the explanation of our results, let us start by explaining our experiments settings.

4.1 Experiment Setup

We perform three different types of experiments. The first experiment tests the classification effect of our proposed WBDF model. For this we chose several datasets (details of these datasets are given in Table 1). We have used a total of 12 datasets in this work. Out of 12, there are 10 UCI datasets. There are two popular industry bench-marking datasets Criteo and Avazu mostly used for *click-through-rate* (CTR) prediction. Additionally, we have used 2 Synthetic datasets.

The second set of experiments study the interpretation of our proposed WBDF framework. We have made use of one dataset from Table 1 – Adult – and compared the interpretation of WBDF model with that of state-of-the-art model LIME.

The third set of experiments study the knowledge-guided capability of our framework. For this, we have used four datasets from Table 1. Note, for each dataset, we adopted several techniques to construct a biased dataset.

4.1.1 On Creation of Synthetic Datasets

The two synthetic datasets are variants of Pokerhand dataset. The motivation for synthesizing these two datasets is to create a dataset that requires a low-bias model. We will use the performance of these two datasets to determine how effective the model's feature engineering capability is. Two synthesized datasets are based on standard Pokerhand by following below 4 steps:

1. Version of the synthetic Pokerhand is specified.
2. Rules of each class for Pokerhand are identified. E.g., Full house is not available for four-hand synthetic Pokerhand-4.
3. The cards with respective hands are uniformly sampled.
4. In each sampling round, the cards in hand are checked by the rule of each class from Pokerhand datasets. Once checked, they are stored on other cards.

4.1.2 Methods used in Comparison

We compare with the following methods:

- **W**: High-order logistics regression ($\mathtt{LR}^n$) with $n = 2$.
- **B**: Discriminative k-dependence Bayesian classifier ($\mathtt{BN}^k$) with $k = 3$ [22].
- **D**: Deep Neural network ($\mathtt{ANN}^d$) with $d = 3$.
- **F**: Compressed Interaction network (CIN) [5] with a layer size of 3 ($m = 3$).
- **LR**: Typical logistics regression model [23] – a linear model, incldued only for benchmarking.
- **WD**: Typical Wide and Deep model [7] with quadratic Wide component and a Deep component with $d = 3$.
- **xdeepFM**: One of SOTA methods for CTR prediction. Layer of size 3 is used with CIN.
- **FNN**: Factorization machine based artificial neural network [20], which incorporates order-2 feature interactions.
- **DCN**: CrossNet model [21] with default size of cross-layer to be 3.
- **WBDF**: WBDF framework based on Algorithm 1. We use $n = 2$ for wide (wide component is equivalent to **W**), $d = 3$ for deep (deep component is equivalent to **D**) and $m = 3$ for factorized (factorized component is equivalent to **F**). The broad component k is set to 3 for all datasets (unless stated otherwise), note broad component is equivalent to **B**.
- **ONN** [24], **FiBiNet** [25] and **IPNN** [26]: Three SOTA models that are specialized for CTR prediction problem on Criteo and Avazu.

In the experiments, we have tried to be systematic in terms of the comparison. E.g., we used $m = 2$ for FNN, as it is the default choice in almost all studies. For WBDF, we have used $m = 3$ as the default choice for factorized component. This is because, WBDF utilizes CIN from [5] as the factorized component, where $m = 3$ is default option in most studies. It is important ot note that an ensemble model of the form: W+B+D+F is just an WBDF model with-out the attention mechanism. So it is not a competitor but just a variant of our proposed model. In our initial experiments, we have seen that attention improves the performance of W+B+D+F model, and hence we have used attention mechanism as the default setting in WBDF.

4.1.3 On relationship between wide and broad component's size

It is important to note that, $n = 1$ and $k = 0$ encompasses linear interactions, and are equivalent models (for comparison). Whereas, $n = 2$ and $k = 1$ encompasses quadratic interactions and are equivalent, whereas $n = 3$ and $k = 2$ are equivalent as they encompass cubic interactions. In general, for a wide model with size n, the equivalent broad model will have size $k = n - 1$.

4.1.4 On our Evaluation Strategy

We have used the standard settings (hyper-parameters) of each of the method which is used in the existing literature for comparison.

Our evaluation strategy is based on cross-validation, i.e., each method is tested on each dataset using 5 rounds of 2-fold cross validation, and averaged accuracy and AUC results are reported. For some datasets, we use 10% of the train data for validation (if needed by the method), resulting in a final train:validation:test (TVT) ratio of $4 : 1 : 5$.

To compare results on standard CTR datasets, we follow the evaluation design strategy of [18], i.e., a train:validation:test (TVT) ratio of $8 : 1 : 1$ is used for two CTR datasets.

4.2 WBDF vs. WD/W/B/D/F

Let us in this section compare the accuracy of WBDF with that of W, B, D and F models. The results are given in Table 2. We are interested in **RQ1**, i.e., determining if the joint learning framework leads to performance better than its constituent parts trained separately. We will also compare the performance of WBDF with WD-learning, and hence partly answer **RQ2**.

One can draw following conclusions from Table 2:

- It is encouraging to see that WBDF outperforms all other compared models including its four constituent components, which demonstrates its capability as an effective model for large data quantities.
- The difference in the performance of WBDF and with WD on Pokerhand and two synthetic datasets is massive – almost 15% improvement. This demonstrates the power of capturing higher-order interactions with Broad component. Note, Pokerhand is a dataset where each instance represents a poker *hand*, which has order-5 dependencies. Pokerhand-4 and Pokerhand-6 have order-4 and order-6 dependencies respectively.

It can be seen that WBDF is superior on all datasets, however, it is far more effective on multi-class datasets – there is a difference of over 8% in improvement of WBDF and the next best, i.e., WD model.

4.3 WBDF vs. xdeepFM and other SOTA Models

Let us compare the performance of WBDF with competing low-bias models such as FNN, DCN and xdeepFM models, to find an answer to **RQ2**. For the sake of completeness

Table 2: Comparison of the average accuracy of WBDF with its components, alongside WD-learning framework, on 12 standard datasets.

Dataset	WBDF	WD	B	W	D	F
Criteo	**81.65**	79.73	78.91	76.12	75.61	73.67
Avazu	**82.12**	81.17	81.30	80.19	81.33	80.22
Higgs	**89.38**	88.11	87.63	78.10	85.10	77.76
Sussy	**87.65**	86.30	87.11	80.10	83.75	78.93
KDD99	**98.37**	94.10	92.90	89.80	93.20	89.86
NSL-KDD99	**96.55**	90.20	88.71	87.10	87.70	87.30
Pokerhand-4	**96.39**	83.12	82.33	78.25	80.15	77.19
Pokerhand-6	**91.62**	79.95	80.35	75.98	79.27	75.79
Pokerhand	**95.29**	80.70	81.10	76.90	79.70	76.69
Covtype	**91.26**	88.20	87.96	83.20	84.90	83.09
Localization	**72.09**	65.01	66.70	63.80	64.20	63.67
Adult	**87.57**	83.50	85.70	81.70	82.68	80.96
Avg - Binary Class	**84.748**	83.075	83.528	78.382	81.594	77.908
Avg - Multi class	**91.653**	83.041	82.864	79.291	81.303	79.084
Avg - Overall	**88.775**	83.055	83.141	78.912	81.424	78.594

Table 3: Comparison of the average accuracy performance of WBDF and with competing models. TVT split of 4:1:5 (repeated 5 times).

Dataset	WBDF	xdeepFM	DCN	FNN	LR
Criteo	**81.65**	81.52	81.46	80.83	76.12
Avazu	**82.12**	82.10	82.01	81.99	78.67
Higgs	89.38	**89.71**	89.25	88.13	75.22
Sussy	**87.65**	87.59	87.32	86.91	77.67
KDD99	**98.37**	97.22	96.90	93.91	87.93
NSL-KDD99	**96.55**	96.37	95.10	90.53	86.10
Pokerhand-4	**96.39**	95.71	84.61	84.12	72.67
Pokerhand-6	**91.62**	90.07	82.73	81.79	68.93
Pokerhand	**95.29**	94.11	83.91	83.60	73.11
Covtype	**91.26**	91.12	91.03	90.86	75.27
Localization	72.09	**73.56**	67.12	65.83	61.87
Adult	**87.57**	87.16	85.92	84.73	78.91
Avg - Binary Class	**84.748**	84.706	84.202	83.712	76.051
Avg - Multi class	**91.653**	91.162	85.914	84.377	75.126
Avg - Overall	**88.775**	88.473	85.201	84.101	75.511

(and establishing a baseline), we have also presented the results with logistic regression model. Since each dataset is different, it will be interesting to see whether there is a single model that can outperform others on all datasets. We present the accuracy results in Table 3, from which it can be seen that WBDF-learning framework outperform all other baseline models on 10 out of 12 datasets. The only two losses are on Higgs and Localization to xdeepFM model. Considering that our WBDF wins the most over SOTA models, we find these result very encouraging. Overall, there is a 3%

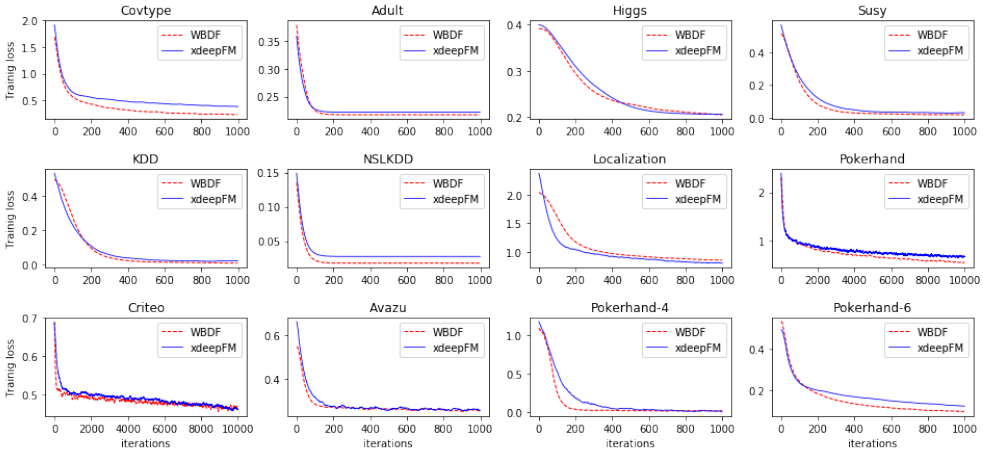

Fig. 4: Comparison of the `training-loss` convergence profile of `WBDF` and `xdeepFM`.

difference in the performance of `WBDF` over `DCN` and `FNN` and a difference of 10% over `LR`. It can be seen that `WBDF` is far more effective on multi-class datasets with a performance improvement of 0.37% over `xdeepFM`. Given the scale of these datasets, again, such an improvement is significant. **On the two `CTR` datasets `Criteo` and `Avazu`, the `WBDF` outperforms other models.** We will discuss the performance difference in terms of `AUC` later in the section. These experimental findings dictate that `WBDF` has superior feature engineering resulting in a low-bias model compared to `SOTA` models.

4.3.1 Convergence Analysis

The comparison of the convergence of `training-loss` of `WBDF` and `xdeepFM` models on 12 datasets is shown in Figure 4. We claim that a better convergence profile is one that asymptote to a better point in the optimization space, and also converges faster. Note, in previous section, we already have established that `WBDF` has a better performance in terms of `accuracy` comparing to `xdeepFM`. According to the convergence plots in Figure 4, it can be seen that `WBDF` converges not only much faster but asymptote to a lower point on most of the datasets. These results are extremely encouraging as they demonstrate that `WBDF` not only has a better performance in terms of accuracy but also in terms of convergence. Furthermore, we compare the training time between `WBDF` and the `xdeepFM` in Figure 5, where it can be seen that `WBDF`'s better convergence leads to faster training time as compared to `xdeepFM`. Please note that the training time of `WBDF` includes the structure learning of the Bayesian network. It is important to note that given the (massive) size of datasets, a small improvement in model's accuracy can result in significant business value. Also, the main benefit of `WBDF` as compared to `xdeepFM` is due to its interpretation and knowledge-guided nature. The fact, that it has a similar or better performance both in terms of accuracy and training time than `xdeepFM` and other state-of-the-art models is an added advantage.

Table 4: Comparison of the AUC performance of WBDF and with SOTA models on CTR datasets based on evaluation strategy of [18] using a TVT ratio of 8:1:1 (repeated 2 times).

Dataset	WBDF	xdeepFM	ONN	FIBINET	DCN	IPNN
Criteo	**81.58**	81.43 ***	81.48 *	81.31	81.44 **	81.42
Avazu	79.85	79.33	**79.92** *	79.52 **	79.31	79.44 ***

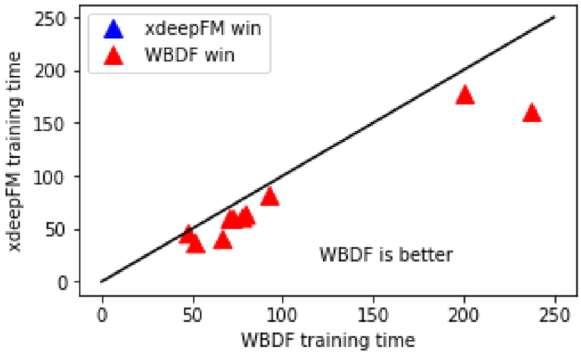

Fig. 5: Training time comparison between WBDF and xdeepFM.

4.4 AUC Comparison on Criteo and Avazu

Let us use the evaluation strategy of [18] and compare the performance of WBDF with SOTA methods that are specialized for CTR prediction datasets. We report the AUC results in Table 4. We use the notation of (*), (**) and (***) to denote the current best, second-best and third-best methods respectively, according to study in [18] [7]. It can be seen that WBDF leads to better than current-best performance on Criteo dataset, whereas, it leads to second-best performance on Avazu. **Given the specialized nature of competing baseline methods, we find these results extremely encouraging, which demonstrates that WBDF has the potential to be an effective model. In the following, we will discuss the interpretability of WBDF, which is a feature missing from all the competing baseline methods.**

4.5 Interpretation

In this section, we will answer **RQ3** by demonstrating how the Broad component of WBDF opens the room for a better interpretation and a step towards an explainable model in a deep learning framework. We conjecture that having this component also opens the door for adding in auxiliary information (learned either from other sources or through human experts) in a deep learning model.

[7] Note, we have used [18] as it is a reliable benchmark study. We have not used the leader board of https://paperswithcode.com/sota/click-through-rate-prediction-on-criteo, as we found that different papers have used different versions of data as well as evaluation strategy.

4.5.1 Interpretation from BIN

As the learned parameters (Θ) of BIN are actually conditional probabilities of the form $\mathrm{P}(x_i|y, \Pi(x_i))$, they are super interpretable, during and after the training. E.g., during training, one can interpret the importance of features for determining the value of class or predicting class. In this work, we used the following measure of feature-set importance:

$$\mathcal{I}^{y}_{x_i,\Pi(x_i)} \propto \frac{\mathrm{P}(y|x_i, \Pi(x_i))}{\mathrm{P}(y)}. \tag{10}$$

The conditional probabilities $\mathrm{P}(y|x_i, \Pi(x_i))$ can be obtained by conducting the inferencing on the broad component BIN [27]. Now, higher the score – $\mathcal{I}_{x_i,\Pi(x_i)}$, higher the contribution of that feature-interaction towards the prediction of class value. Note, we assume conditional independence (among features) when doing this analysis. In practice, this is not true, however, many algorithms assume such independence, when determining the importance of a feature (or feature-interaction) for class prediction [28]. We claim that one can do all the interpretation during the training of WBDF. On the contrary, popular state-of-the-art models such as LIME works by doing an explanatory analysis once the model is trained.

Interpretability of a model is generally observed by identifying features or feature combinations that play a role in determining the output of the model. In order to demonstrate the interpretation capabilities of WBDF, we randomly selected two instances from Adult dataset and listed feature interactions ranked by importance based on the probabilites from BIN in Table 5 [8]. The reason for selecting Adult dataset is that features are easily interpretable. Most of the other datasets in our analysis lacks meaningful features. The top 3 high-order features with higher conditional probability score are marked as bold, for which BIN has given a higher score for predicting class income $>$ 50K. Next, we tested the same two instances via LIME to explain the feature contribution. It can be seen from Figure 6 that the top 3 high-order features with higher contribution via LIME are the same as those in Table 5 for both instances. It is encouraging to see that BIN can offer the same interpretability capability that LIME (and its variants) offers only after the training. The similar trend was observed for other datasets – we have only included two examples in this analysis, due to space constraints.

4.6 Knowledge-guided Machine Learning

To study knowledge-guided machine learning capability of WBDF – we employ four datasets (Adult, Sussy, Higgs and KDD99). Our strategy in this experiment is to first train WBDF with biased dataset. We are interested in comparing this model's performance with following model:

- We utilize the Broad component of WBDF to generate synthetic data. The structure of the Broad component in this model has to be either a) specified by some expert that can provide an un-biased perspective through structure learning, or b) structure can be learned on some related un-biased data.

[8] Note. the instance 1 selected is married, female, USA, civilian, $40-55$, professor, high-doctorate, and the instance 2 selected is not-married, male, USA, civilian, $0-25$, Repair, Some-college

Table 5: Illustration of interpretability of BIN.

Feature ($\Pi(x_i)$, x_i)	$\mathcal{I}^{y>50k}_{x_i,\Pi(x_i)}$	$\mathcal{I}^{y<=50k}_{x_i,\Pi(x_i)}$
Age=[40,55],Marital-status=Married_CIV	**0.82**	0.17
Age=[40,55],Workclass=Private	**0.61**	0.2
Occupation = Prof, Education-num =[14.5,inf]	**0.71**	0.02
Age=[40,55],Hours-per-week = [-inf,34.5]	0.2	0.06
Age=[40,55],Education = Doc	0.16	0.01
Marital-status=Married_CIV, Relationship = Own-child	0.27	0.1
Marital-status=Married_CIV, Race = White	0.22	0.07
Relationship = Own-child, Sex = Female	**0.51**	0.23
Sex = Female, Nationality = USA	**0.36**	0.18
Relationship = Own-child, Capital_gain = [-inf,57]	0.02	0.01
Relationship = Own-child, Capital_loss = [-inf,1551]	0.012	0.01
Probability	**0.915**	0.085
Age=[-inf,25],Marital-status=Never-married	0.14	**0.74**
Age=[-inf,25],Workclass=Private	0.39	**0.86**
Occupation = Craft-repair, Education-num =[8.5,10.5]	0.37	0.17
Age=[-inf,25],Hours-per-week = [39.5,41.5]	0.55	**0.35**
Age=[-inf,25],Education = Some-College	0.27	0.1
Marital-status=Never-married, Relationship = not_in_family	0.12	**0.42**
Marital-status=Never-married, Race = White	0.31	0.01
Relationship = not_in_family, Sex = Male	0.09	**0.19**
Sex = Male, Nationality = USA	0.01	0.02
Relationship = not_in_family, Capital_gain = [-inf,57]	0.01	0.01
Relationship = not_in_family, Capital_loss = [-inf,1551]	0.01	0.01
Probability	0.283	**0.717**

- The un-biased dataset generated this way is combined with existing biased datasets, and WBDF model is trained. We call this model the knowledge-guided WBDF model.

We aim to measure the boost in the performance of knowledge-guided WBDF variant with that of standard WBDF model by computing the lift in the accuracy [9].

The important question is how to produce a biased dataset? Because, none of the datasets we have used in this work has an associated expert who can provide the true structure, and also none of the datasets have a related un-biased version available. To address these issues, we used two different strategies:

- The original dataset is firstly trained (where input features are represented in one-hot encoding format) with tree-based ensemble model such as Random Forest. Then the feature importance is obtained from the trained Random Forest. The most important feature-values in the dataset towards to the classification performance are identified. By removing a percentage of instances containing the most important feature-values – a biased dataset is obtained.
- The second method for acquiring the biased dataset is introduced by leveraging the Shapley Additive explanations (SHAP) values [29]. The SHAP values can identify influential feature-values according to the cooperative gaming theory. Specifically, they can quantify the contribution of each feature towards the prediction outcome for a specific instance, considering both its own value and its

[9] The lift of the accuracy is the improvement in percentage between knowledge-guided WBDF and standard vanilla WBDF

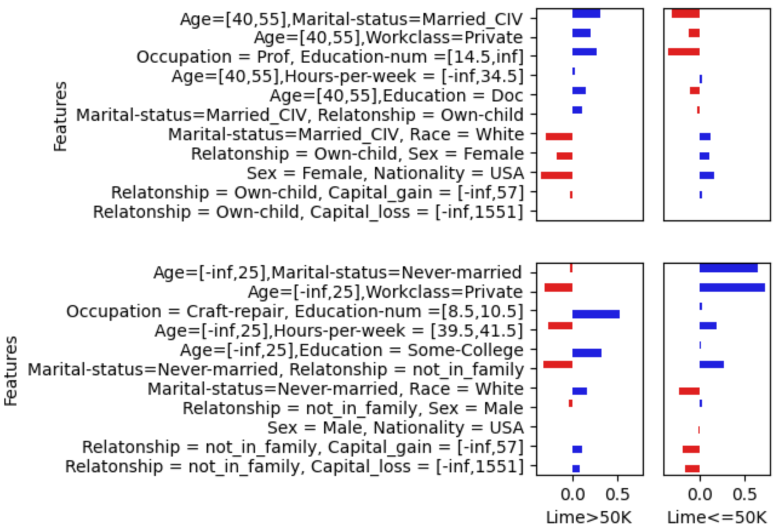

Fig. 6: Interpretation of results on selected instances from `LIME` (first feature is the $\Pi(x_i)$ and second feature is x_i).

interactions with other features. Therefore, it provides a more nuanced understanding of the contribution of the feature-value in the original dataset in complex models. By leveraging `SHAP` values, the biased dataset can be obtained by removing the instances which contain these important feature-values.

In the knowledge-guided learning experiment, the correspondingly used dataset is divided into training and testing subsets with 80% and 20% ratio. The `Broad` component, i.e., the structure and parameter learning are trained from the original 80% of the data – denoted as `Original Data`. The `broad` component is later used to generate samples which are equal in size to `Original Data` – denoted as `Generated Data`. Additionally, the training set is used to obtain the biased dataset – denoted as `Biased Data`. The testing set is used as the hold-out for evaluation purpose in this experiment.

The `Generated data` is blended with the `biased data` to fit a `WBDF` model for knowledge-guided machine learning – denoted as knowledge-guided `WBDF`. Table 6 and Table 7 shows the comparative results of the knowledge-guided `WBDF`. It can be seen that the knowledge-guided `WBDF` outperforms vanilla `WBDF` that is trained on `biased data`. Overall, it can be seen that knowledge-guided `WBDF` can obtain an average of over 10% lift in the accuracy. This is very encouraging as it demonstrates that `WBDF` can not only compete with the state-of-the-art models such as `xdeepFM` in terms of classification performance, but it can also provide a solution for classifying with biased dataset through knowledge-guided machine learning. What also stands out in the Table 6 and Table 7 is the excellent results on `Generated data` – i.e., `WBDF` trained on data generated from `Broad` component.

Table 6: Comparison of accuracies on four datasets where biased data is generated with `Ensemble tree-based feature importance` approach.

Dataset	Biased Data WBDF	Generated Data– WBDF	Generated + Biased Data Knowledge-guided WBDF	Lift
Adult	77.46	81.42	**81.68**	**5.45%**
Sussy	61.59	74.69	**78.59**	**27.6%**
Higgs	74.76	88.84	**88.91**	**18.93%**
KDD99	88.30	88.82	**90.96**	**2.14%**
Average	75.53	83.44	**85.04**	**12.59%**

Table 7: Comparison of accuracies on four datasets where biased data is generated with `SHAP-based feature importance` approach.

Dataset	Biased Data WBDF	Generated Data– WBDF	Generated + Biased Data Knowledge-guided WBDF	Lift
Adult	81.58	81.42	**83.46**	**2.30%**
Sussy	70.91	74.69	**77.13**	**8.77%**
Higgs	85.78	88.84	**89.62**	**3.84%**
KDD99	78.04	88.82	**91.90**	**17.76%**
Average	79.08	83.44	**85.53**	**8.16%**

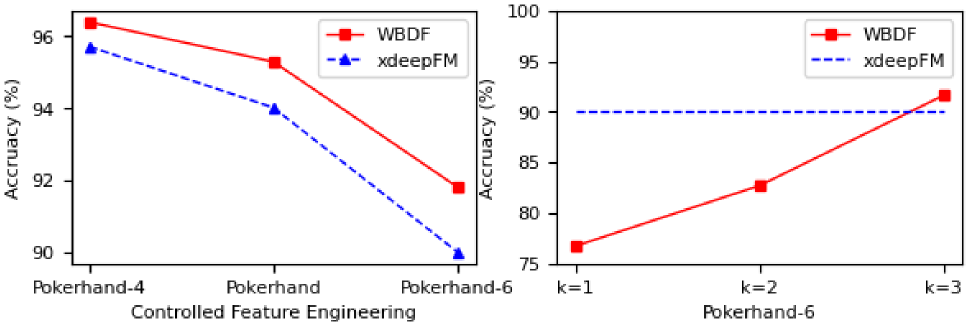

Fig. 7: (Left) Performance variation of `WBDF` and `xdeepFM` with different synthetic datasets with controlled interactions; (Right) Effect of k on `WBDF`'s performance demonstrated on synthetic dataset.

4.7 Ablation studies

4.7.1 Controlled Feature Engineering and k

In this section, we compare the performance of `WBDF` with varying level of feature interactions. We make use of 3 synthetic `Pokerhand` datasets, such that it has 4, 5 and 6 level interactions. The results are shown on left-hand-side in Figure 7. We also plot the performance of `xdeepFM` for comparison. It can be seen that `WBDF` has far more superior feature engineering capability resulting in much better performance on these synthetic datasets.

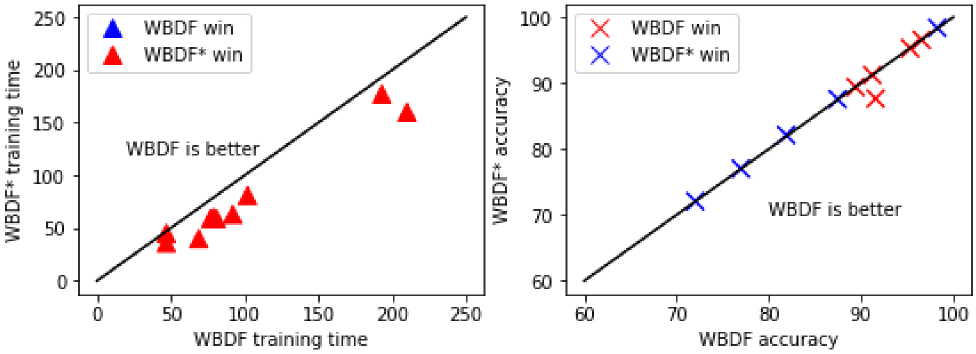

Fig. 8: Accuracy and Training-time comparison of `WBDF` and `WBDF`$*$.

We also study the role of parameter k in our `WBDF` model on one of the synthetic data (`Pokerhand-6`). The results are shown in right-hand-side of Figure 7. As expected, the higher the value of k, the better the results are. For sake of completeness, again, we have plotted `xdeepFM` performance on this dataset. It can be seen that `WBDF` with $k = 1$ and $k = 2$ has inferior performance than `xdeepFM`. However, $k = 2$ leads to superior performance – highlighting the importance of `BIN` in `WBDF` framework.

4.7.2 On `BIN`'s Form

The output of `BIN` in `WBDF` is based on Equation 7, where it learns parameters Θ directly. One can be interested in the output of the form of Equation 6, i.e., $\sigma_b(\mathbf{B}^T \log \Theta)$. Note, in this case, we aim to pre-learn the parameter Θ (optimizing log-likelihood), and learn $\mathbf{B}$ as part of `WBDF` training. We call this version `WBDF`$*$. We conducted the ablation study on these two forms of `BIN`. It can be seen from Figure 8 that both forms of `BIN` lead to similar results, though `WBDF`$*$ is much faster in training as compared to `WBDF`. This is expected, as `WBDF`$*$ does not have to fulfil the probability constraints during the discriminative training process. The downside of this speed is the lack of interpretation in `WBDF*` model.

4.7.3 On the role of `Attention`

In this section, we extracted the `attention` score from `WBDF` during training on `Adult` dataset. As we discussed earlier, one useful trait of `attention` is that it leads to excellent interpretability, as it explains which component is contributing more towards the final prediction. As the attention score is highly instance-specific, we have used different portions of the data to extract an average attention score (denoted as `30%data,50%data,70%data,100%data`). Figure 9 shows the extracted attention score where the darker color represents a higher score. The x-axis represents the component-wise score for each of the two classes. E.g, W_1 represent the attention score of `wide` component for first class, and W_2 represents the attention score of `wide` component for the second class. We present results during two stages of the training, that is: `epoch` $= 50,$ `epoch` $= 200$). It can be seen that the attention scores are consistent

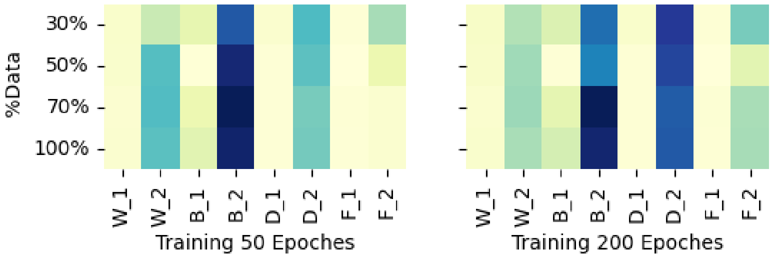

Fig. 9: Attention score over the various component in WBDF during training.

during the two training stages. Whereas, components B_2 and D_2 are the one with highest weights, and hence played a dominant role in classification. One can also see that deep component needs a lot more data (larger number of epochs) to gain confidence and play a role in deciding the class label, as it is more dominant at 200 epochs instead of 50. However, the broad component remains consistent from the start of training till finishing (attention scores do not change from 50 to 200 epochs). It is worth noting that class 2 has higher weights than class 1 (i.e., W_2 > W_1, B_2 > B_1, D_2 > D_1 and F_2 > F_1) in Figure 9 – this is because on adult dataset, the class distribution is skewed in favour of class 2. Nonetheless, the focus should be on the relative weights of each component, and it can be seen that broad component plays the biggest role in making a prediction.

5 Conclusion

In this paper, we have presented a model for addressing the growing need of building a low-bias model for extremely large quantities of data, that is also interpretable. We have shown that our proposed WBDF model based on an end-to-end learning of wide, deep, factorized learning, and a newly formulated broad interaction network, can lead to **a)** better classification performance than SOTA models (hence with superior feature engineering capability); **b)** faster training time, better convergence profile; **c)** offers interpretability as good as SOTA frameworks, especially during the training time; **d)** capability to do knowledge-guided machine learning, as demonstrated by the scenario of bias-correctness. With ever-increasing scale of today's datasets, we believe that WBDF offers an excellent learning framework. In future, we plan to study alternative models of broad-learning, including unrestricted Bayesian networks, as well as incorporating capability to handle numeric features.

6 Code

The code for experiments conducted in this paper is available at: https://anonymous.4open.science/r/wbdlearning-464D/.

7 Conflict of Interest Statement

Authors do not have any funding information to report. Authors do have any actual, perceived or potential conflict of interests (financial or non-financial) to disclose as well.

References

[1] Lian, J., Zhang, F., Xie, X., Sun, G.: Restaurant survival analysis with heterogeneous information. In: The World Wide Web Conference (2017)

[2] Nargesian, F., Samulowitz, H., Khurana, U., Khalil, E.B., Turaga, D.S.: Learning feature engineering for classification. In: Ijcai, pp. 2529–2535 (2017)

[3] Martınez, A.M., Webb, G.I., Chen, S., Zaidi, N.A.: Scalable learning of bayesian network classifiers. Journal of Machine Learning Research, 1–35 (2016)

[4] Breiman, L.: Random forests. Machine learning **45**(1), 5–32 (2001)

[5] Lian, J., Zhou, X., Zhang, F., Chen, Z., Xie, X., Sun, G.: xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In: Proceedings of the 24th ACM SIGKDD (2018)

[6] Rendle, S.: Factorization machines. In: ICDM, pp. 995–1000 (2010)

[7] Heng-Tze, C., Koc, L., Harmsen, J., Shaked, T., Chandra, T.: Wide and deep learning for recommender systems. In: arXiv:1606.07792 (2016)

[8] Dong, X., Yu, Z., Cao, W., al.: A survey on ensemble learning. Front. Comput. Sci (2020) https://doi.org/10.1007/s11704-019-8208-z

[9] Garreau, D., Luxburg, U.: Explaining the explainer: A first theoretical analysis of lime. In: International Conference on Artificial Intelligence and Statistics, pp. 1287–1296 (2020). PMLR

[10] Chen, C.P., Liu, Z.: Broad learning system: An effective and efficient incremental learning system without the need for deep architecture. IEEE transactions on neural networks and learning systems (2017)

[11] Zaidi, N.A., Petitjean, F., Webb, G.I.: Efficient and effective accelerated hierarchical higher-order logistic regression for large data quantities. In: SIAM International Conference on Data Mining (2018)

[12] Karpatne, A., Kannan, R., Kumar, V.: Knowledge Guided Machine Learning: Accelerating Discovery Using Scientific Knowledge and Data. CRC Press, ??? (2022)

[13] Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. Machine

Learning **29**(2), 131–163 (1997)

[14] Sahami, M.: Learning limited dependence bayesian classifiers. In: Proceedings of the Second International Conference on KD and DM (1996)

[15] Martinez, A., Chen, S., Webb, G.I., Zaidi, N.A.: Scalable learning of bayesian network classifiers. Journal of Machine Learning Research (2015)

[16] Ng, I., Zhu, S., Fang, Z., Li, H., Chen, Z., Wang, J.: Masked Gradient-Based Causal Structure Learning, pp. 424–432. https://doi.org/10.1137/1.9781611977172.48

[17] Deleu, T., Góis, A., Emezue, C., Rankawat, M., Lacoste-Julien, S., Bauer, S., Bengio, Y.: Bayesian Structure Learning with Generative Flow Networks (2022)

[18] Zhu, J., Liu, J., Yang, S., Zhang, Q., He, X.: Open benchmarking for click-through rate prediction. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pp. 2759–2769 (2021)

[19] Cheng, H.-T., al.: Wide & Deep Learning for Recommender Systems. arXiv (2016). https://doi.org/10.48550/ARXIV.1606.07792

[20] Zhang, W., Du, T., Wang, J.: Deep learning over multi-field categorical data: A case study on user response prediction. CoRR **abs/1601.02376** (2016)

[21] Wang, R., Fu, B., Fu, G., Wang, M.: Deep & cross network for ad click predictions. In: Proceedings of the ADKDD'17, pp. 1–7 (2017)

[22] Zaidi, N.A., Webb, G.I., Carman, M.J., Petitjean, F., Cerquides, J.: Efficient parameter learning of bayesian network classifiers. Machine Learning **106**, 1289–1329 (2017)

[23] Jain, A.K., Mao, J., Mohiuddin, K.M.: Artificial neural networks: A tutorial. Computer (1996)

[24] Zhang, Q., Li, J., Jia, Q., Wang, C., Zhu, J., Wang, Z., He, X.: Unbert: User-news matching bert for news recommendation. In: IJCAI, pp. 3356–3362 (2021)

[25] Huang, T., Zhang, Z., Zhang, J.: Fibinet: combining feature importance and bilinear feature interaction for click-through rate prediction. In: Proceedings of the 13th ACM Conference on Recommender Systems, pp. 169–177 (2019)

[26] Qu, Y., Cai, H., Ren, K., Zhang, W., Yu, Y., Wen, Y., Wang, J.: Product-based neural networks for user response prediction. In: 2016 IEEE 16th International Conference on Data Mining (ICDM), pp. 1149–1154 (2016). IEEE

[27] Chavira, M., Darwiche, A.: Compiling bayesian networks using variable elimination. In: IJCAI, vol. 2443 (2007)

[28] Kraskov, A., Stögbauer, H., Grassberger, P.: Estimating mutual information. Physical review E (2004)

[29] Lundberg, S.M., Lee, S.-I.: A unified approach to interpreting model predictions. Advances in neural information processing systems **30** (2017)