Improving Extended Tree Augmented Naïve Classifier with GAN

Hao Zuo, Haoran Wang, Yun Zhou*

***Abstract*—Extended Tree Augmented Naïve Classifier (ETAN) is a recently proposed model which relax the independence assumption between input features. However, it is usually challenging to train a good ETAN, because it needs more training data. To address this problem, in this paper, we introduce the Generative Adversarial Network (GAN), which has the ability to model the probability distribution of original data and generate synthetic data that close to the realistic data. Moreover, the generated data can also enhance the original correlation and independence among features and class. We name our method as GAN-ETAN, and experiments on 6 real world datasets show the GAN-ETAN can improve the classification performance of original ETAN in some extent.**

***Index Terms*—Tree Augmented Naïve Classifier, Generative Adversarial Network, Data synthesis, Classification performance.**

I. Introduction

PREDICTING a class based on some features is the classification problem. Bayesian classification uses the prior probabilities inferred from original data to calculate and predict the posterior probabilities of class. The Naïve Bayes classifier [1] sets the conditional independence assumption of features: for known class, all features are assumed to be independent of each other. The Naïve Bayes learns the joint probability distribution of features and class, and then calculates the category with the maximum posterior probability. However, this assumption is usually difficult to satisfy in real-world tasks and there will be a large error in the actual prediction. In order to address this problem, an improvement method, tree-augmented Naïve Bayes (TAN) [2], relaxs the independence assumption using a more complex graph. TAN calculates the conditional mutual information between two features and constructs a complete graph based on a maximum weighted spanning tree. Conditional mutual information portrays the relevance of different features in the case of known class, TAN preserves the dependencies between strongly correlated features with the maximum weighted spanning tree algorithm. However, Extended TAN (ETAN) [3] uses an absolutely different strategy than TAN. ETAN allows features without the class as parent, multiple features with only the class as parent, and features completely disconnected. This contains almost all the possibilities of the graph. But ETAN usually challenging to train, because the limited dataset cannot satisfy the construction of more complex networks. Thus we must search for improvements from data.

In this paper we develop a new framework Extended Tree Augmented Naïve Classifier with GAN (GAN-ETAN) that combines ETAN with Generative Adversarial Network (GAN) [4]. GAN is a neural network architecture of deep learning, which consists of a generator network and a discriminator network. The generator generates data from random noise, and the discriminator is used to discriminate whether the generated data corresponds to the original data distribution. Generator and discriminator play against each other. After iterative training, the final generator generates synthetic data with the distribution of the original data. The synthetic data enhances the strong correlation and improve the independence of features and class. Especially, features without class, multiple features with only class and features completely disconnected. GAN-ETAN uses the synthetic data for structure learning through the Bayesian Dirichlet likelihood equivalent uniform (BDeu) score [5], [6], [7]. When applied to the same dataset with the benchmarking suite, GAN-ETAN performs significantly better than ETAN self.

This paper is divided as follows. Section II introduces the related work. Section III outlines the basic information on BN and GAN. Section IV presents our method, GAN-ETAN. Section V empirical experiments and discussion of results. Finally, Section VI concludes the paper and suggests possible future work.

II. Related work

During the past decade, there are many studies on extended tree augmented Naïve classifier. These studies focus on finding the optimal Bayesian structure and constructing the Bayes optimal classifier. ETAN significantly removes the constraints between features and class and expands the possibility of Bayesian structure, but it also increases the difficulty of finding the optimal structure. ETAN didn't find a stable and reliable algorithm.

Consequently, Aaron Meehan and Cassio P. de Campos (2015) [8] claimed that averaged extended tree augmented Naïve Bayes (AETAN) which combined the Averaging One-dependent Estimator (AODE) [9] and the Extended Tree Augmented Naïve Bayes (ETAN) into a single classifier. It adds the possibility of a super attribute on top of the existing ETAN algorithm and then infers all the graphs in an average model. This averaging usually has the effect of increasing the accuracy of classification while reducing the impact that any single graph has on the overall result, therefore, AETAN also needs a larger computational time. Bojan Mihaljevic

Manuscript received March 28, 2023; revised June 14, 2023.This work was supported in part by the Nature Science Foundation of China [grant number 62276272], the Science and Technology Innovation Program of Hunan Province [grant number 2021RC3076] and Training Program for Excellent Young Innovators of Changsha [grant number KQ2009009]

Hao Zuo is a postgraduate student of Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha, 410000, China, E-mail: haozuo17@163.com

Haoran Wang is a manager of National University of Defense Technology, Changsha, 410000, China, E-mail: wanghaoran@nudt.edu.cn

Yun Zhou is a professor of Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha, 410000, China, E-mail: zhouyun@nudt.edu.cn

et al. (2018) [10] extended and further proved that there exists minimal class-focused DAGs (C-DAGs, presented by Acid et al. 2005 [11]) (MC-DAGs) that are equivalent to the complete DAGs, and design an adaptive e greedy equivalence search (GES) to access all equivalence classes containing MC-DAGs, so as to obtain the best Bayesian classification. This method reduces the search space of DAGs. Gao Xiaoguang et al. (2019) [12] proposed a derivative approach of Bayesian estimation called constrained Bayesian estimation (CBE) in which the expert judgments are introduced with Dirichlet priors. CBE constructs a posteriori distribution based on a Dirichlet prior and then takes the expectation of the posterior distribution as the parameter estimate. CBE has a considerable modeling effect on sparse training data.

Furthermore, Shouta Sugahara et al. (2020) [13] used the sub-bagging method to reduce the posterior standard error of Bayesian network structure learning and improve the classification accuracy. This method has a highly accurate classification effect under small data samples. And because of the asymptotic consistency of sub-bagging, it is still effective for large data. Faced with the same posterior accuracy problem, Shouta Sugahara and Maomi Ueno (2021) [14] further proposed an exact learning augmented Naïve Bayes classifier (ANB), it restricts class variables to have no superclass. When a class variable has many parents, the estimation of the conditional probability parameters of the class variable becomes unstable because the number of parent variable configurations becomes large. Thus, ABN solved this problem by directly changing the relationship between a class variable and parents. This method also can effectively reduce the posterior standard error of Bayesian network structure learning. In the latest research, Shouta Sugahara et al. (2022) [15] found that when the data itself has an extended Bayesian network structure, recursive autonomy identification (RAI) with Bayes factor can efficiently learn Bayesian networks. This method decomposes the whole structure into local structures and tests the conditional independence with Bayes factor. All of above methods optimize the algorithm from the perspective of reducing the posterior error.

III. Preliminaries

A. Bayesian networks

Consider a joint distribution $P(X_1, X_2, \ldots, X_n)$ containing n variables that, using the chain rule, can be written as

$$\begin{aligned} &P\left(X_1, X_2, \cdots, X_n\right) \\ &= P\left(X_1\right) P\left(X_2 \mid X_1\right) \cdots P\left(X_n \mid X_1, X_2, \cdots, X_{n-1}\right) \\ &= \prod_{i=1}^{n} P\left(X_i \mid X_1, X_2, \cdots, X_{i-1}\right) \end{aligned} \tag{1}$$

For any X_i, if there exists $pi\left(X_i\right) \subseteq \left\{X_1, \cdots, X_{i-1}\right\}$ such that, given $pi(X_i)$, X_i is conditionally independent of the other variables in $\{X_1, X_2, \ldots, X_{i-1}\}$

$$P\left(X_i \mid X_1, X_2, \ldots, X_{i-1}\right) = P\left(X_i \mid \pi\left(X_i\right)\right) \tag{2}$$

So there is

$$P\left(X_1, \ldots, X_n\right) = \prod_{i=1}^{n} P\left(X_i \mid \pi\left(X_i\right)\right) \tag{3}$$

This gives a decomposition of the joint distribution, where $P\left(X_i \mid \pi\left(X_i\right)\right)$ is the marginal distribution $P(X_i)$ when $\pi\left(X_i\right) = \emptyset$. The distribution of variable X_i is directly dependent on the value of $\pi\left(X_i\right)$. Pearl (1988) [16] proposed to construct a directed graph to represent these dependency and independence relations in the following way. (1) Represent each variable as a node. (2) For each node X_i, a directed edge is drawn from each node in $\pi\left(X_i\right)$ to X_i.

Bayesian networks can be understood on both qualitative and quantitative levels. At the qualitative level, it describes the dependency and independence relationships between variables using a directed acyclic graph. At the quantitative level, it portrays the dependence of variables on their parent nodes using conditional probability distributions.

B. BN structure learning

In the Bayesian model selection framework, the model structure $\mathcal{G}$ and model parameters $\theta_{\mathcal{G}}$ are treated as random variables. The possible values of the variable $\mathcal{G}$ include all directed acyclic graphs with $X_1, X_2, \cdots, X_n$ as a node. Given $\mathcal{G}$, the possible values of the variable $\theta_{\mathcal{G}}$ are all the values of the parameters corresponding to $\mathcal{G}$. The structure prior, expressed as $P_{\mathcal{G}}$, is the prior knowledge about the structure $\mathcal{G}$. The parameter prior $P(\theta_{\mathcal{G}} \mid \mathcal{G})$, is the prior knowledge about the parameters $\mathcal{G}$. The observations are expressed as $D = (D_1, D_2, \ldots, D_m)$. The posterior probability distribution is $P(\mathcal{G}, \theta_{\mathcal{G}} \mid D)$.

The score function is used to evaluate how well the structure fits the data sample, the better the fit, the higher the score. Cooper and Herskovits (1992) [17] first proposed a Bayesian score function for BN structure learning, called CH score or K2 score function. Later, Heckerman et al. (1995) [18] gave the equation a more reliable theoretical basis by proposing the BD score function. When the structural prior information obeyed the uniform distribution, the corresponding scoring function is the Bayesian Dirichlet equivalence uniform (BDeu). The BDeu score is represented as

$$\begin{aligned} &P(D \mid \mathcal{G}) \\ &= \prod_{i=0}^{n} \prod_{j=1}^{q\pi_i} \frac{\Gamma\left(\frac{\alpha}{q\pi_i}\right)}{\Gamma\left(\frac{\alpha}{q\pi_i} + N_{\pi_i=j}\right)} \prod_{k=1}^{rX_i} \frac{\Gamma\left(\frac{\alpha}{rX_i q\pi_i} + N_{X_i=k,\pi_i=j}\right)}{\Gamma\left(\frac{\alpha}{rX_i q\pi_i}\right)} \end{aligned} \tag{4}$$

where α is a hyperparameter.

C. Generative adversarial network

Generative adversarial network (GAN) is a new architecture in the field of deep learning. It includes two neural networks. One is the generative network, which is used to generate data; The other is the discriminative network, which is used to judge whether the generated data is true or false. They are transmitted through the same intermediate vector space. The structure of GAN is shown in Figure 1.

GAN calculation structure inputs the random Gaussian noise and the original data into the generator and the discriminator respectively, and the two networks are trained alternately. The specific process is as follows: First, input the real tag data into the discrimination network and train the discrimination network; Then, the random noise vector with the same dimension as the feature space is input into the generation network to generate a new vector.

ISBN: 978-988-14049-4-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

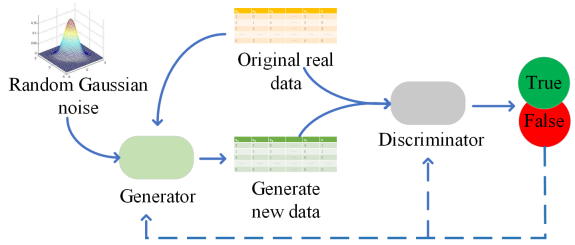

Fig. 1. the GAN Calculation Structure

The vector space dimension is the same as the real data, which is input into the trained discrimination network, and the network is trained using the discrimination results; Then the generated data are mixed with real data and input into the discriminant network to train the discriminant network; Finally, the generation network and the discrimination network compete and promote each other to form a generation confrontation network that can fit the real data distribution. The objective function of GAN is:

$$\begin{aligned}&\min_G \max_D V(D,G) = \\ &\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1-D(G(\boldsymbol{z})))]\end{aligned} \tag{5}$$

Generative adversarial networks are widely used in the field of computer vision, including image translation [19], [20], sequential data generation [[21], [22], [23], anomaly detection [24], etc. However, the method in this paper uses GAN to generate discrete table data, and combines it with ETAN. This is a novel idea.

IV. Proposed Method

In this section, we aim to explain our method from two dimensions: mathematical logic and structure. In terms of mathematical logic, we first analyze the mathematical method of ETAN, and then compare the mathematical support of our model, GAN-ETAN. In terms of structure modeling, we will give the model architecture and algorithm of GAN-ETAN.

A. Extended Tree Augmented Naïve Classifier

A Bayesian network can be defined as a triple $(\mathcal{G}, \mathcal{X}, \mathcal{P})$, where $\mathcal{G}$ represents a directed acyclic graph (DAG) consisting of nodes (variables $\mathcal{X}$) and edges ($E_{\mathcal{G}}$). And $\mathcal{X}$ is a set of variables X_i that are features in the data, $\mathcal{P}$ is a collection of conditional mass functions $p(X_i \mid \Pi_i)$, where pi_i denotes the parents of X_i in the graph. The primary goal of training a Bayesian classifier is to derive a DAG of the Bayesian structure from the data. Structure learning algorithms are mainly classified as constraint-based learning, scoring function based learning and hybrid learning algorithms. ETAN takes scoring function based method, supposes the function as s_D. Therefore, the objective function is

$$\mathcal{G}^* = \text{argmax}_{\mathcal{G}\in\mathcal{G}}\, s_D(\mathcal{G}) \tag{6}$$

Since the structure being scored is additive, the complete DAG can be obtained by summing up the parts

$$s_D(\mathcal{G}) = \sum_{i=0}^{n} s_D\left(X_i, \Pi_i\right) \tag{7}$$

If x_0 is taken as the root node representing the classification, then the likelihood equivalence of the scoring function can be expressed as follows.

$$\begin{aligned}&s_D\left(X_i, \{X_0, X_j\}\right) + s_D\left(X_j, \{X_0\}\right) \\ &= s_D\left(X_j, \{X_0, X_i\}\right) + s_D\left(X_i, \{X_0\}\right)\end{aligned} \tag{8}$$

According to the above likelihood equivalence, it is given that the weights of the connecting edges of two representing attribute nodes are equal with different direction. The edge weight is denoted as w.

$$\begin{aligned}w\left(X_i, X_j\right) &= -\left(s_D\left(X_i, \{X_0, X_j\}\right) - s_D\left(X_i, \{X_0\}\right)\right) \\ &= w\left(X_j, X_i\right)\end{aligned} \tag{9}$$

If x_1 be the only node without a feature as parent

$$\begin{aligned}&\max_{\mathcal{G}\in\mathcal{G}_{\text{TAN}}} s_D(\mathcal{G}) \\ &= \max_{\Pi_i':\forall i>1}\left(\sum_{i=2}^{n} s_D\left(X_i, \left\{X_0, X_{\Pi_i'}\right\}\right) + s_D\left(X_1, \{X_0\}\right)\right) \\ &= s_D\left(X_1, \{X_0\}\right) - \min_{\Pi_i':\forall i>1}\left(-\sum_{i=2}^{n} s_D\left(X_i, \left\{X_0, X_{\Pi_i'}\right\}\right)\right) \\ &= \sum_{i=1}^{n} s_D\left(X_i, \{X_0\}\right) - \min_{\Pi_i':\forall i>1}\sum_{i=2}^{n} w\left(X_i, X_{\Pi_i'}\right)\end{aligned} \tag{10}$$

Thus, if $w\left(X_i, X_j\right) \geq 0$, that is, $s_D\left(X_i, \{X_0, X_j\}\right) \leq s_D\left(X_i, \{X_0\}\right)$, remove those edges (x_i, x_j) and run the minimum spanning tree algorithm over this reduced graph. In the setting of etan, we have $|\Pi_i| \leq 1$, or $|\Pi_i| = 2$ and $\Pi_i \supseteq \{X_0\}$.

The following relations among subsets of DAGs hold.

$$\begin{aligned}s_D\left(\mathcal{G}^*_{ETAN}\right) &\geq s_D\left(\mathcal{G}^*_{\text{TAN}}\right) \\ s_D\left(\mathcal{G}^*_{ETAN}\right) &\geq s_D\left(\mathcal{G}^*_{\text{Naïve}}\right)\end{aligned} \tag{11}$$

B. Extended Tree Augmented Naïve Classifier with GAN

Our goal is to train the GAN to generate new data $\hat{X}$ ($\hat{X}$ has the same features as X but different sampling amounts) with a distribution $\hat{P}$ similar to that of the ETAN labeled dataset X. The new data $\hat{X}$ and the original data X were then combined to reconstruct the training dataset for training the ETAN classifier. Our approach provides high quality data from the data side to train superior classification networks.

In this method, JS divergence is used to measure the distance between the generated data and the original data. Before introducing the Jensen–Shannon (JS) divergence we first understand the Kullback–Leibler (KL) divergence. KL divergence describes the distance between the probability distribution $\hat{P}$ obtained by evaluation training and the target distribution $P(X)$, which can be expressed as

$$\begin{aligned}D_{KL}(\hat{P}\|P) &= -\sum_{x\in X}\hat{P}(x)\log\frac{1}{\hat{P}(x)} + \sum_{x\in X}\hat{P}(x)\log\frac{1}{P(x)} \\ &= \sum_{x\in X}\hat{P}(x)\log\frac{\hat{P}(x)}{P(x)}\end{aligned} \tag{12}$$

ISBN: 978-988-14049-4-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

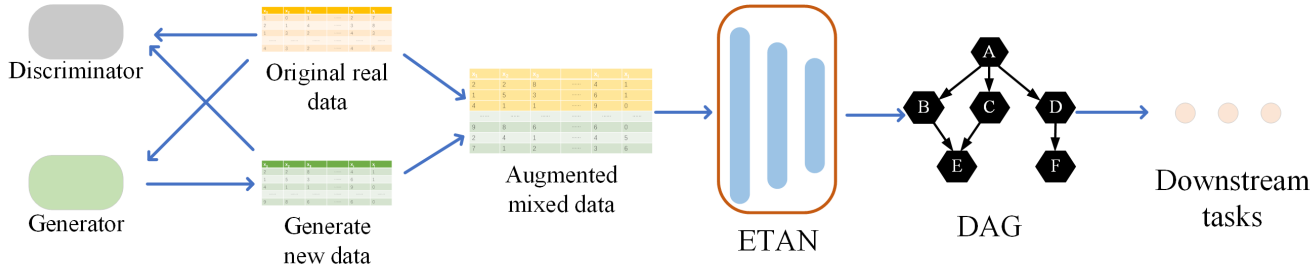

Fig. 2. the GAN-ETAN Structure Model

Since the relative position of $\hat{P}(x)$ and $P(x)$ in the logarithm term in the KL divergence determines that the KL divergence is actually asymmetric, i.e $\mathrm{KL}(\hat{P}\|P) \neq \mathrm{KL}(P\|\hat{P})$. Since the KL divergence is not symmetrical, we can still start from the perspective of the reference frame, so we can just average the distances calculated in all reference frames. This is the idea of JS divergence, specifically defined as

$$D_{JS}(\hat{P}\|P) = \frac{1}{2}D_{KL}\left(\hat{P}\|\frac{\hat{P}+P}{2}\right)+\frac{1}{2}D_{KL}\left(P\|\frac{\hat{P}+P}{2}\right) \tag{13}$$

Next we begin the derivation of the optimization of the GAN with the goal of minimizing the JS divergence between the distribution of the generated new data $\hat{P}$ and the distribution of the real data P. As we introduced in the preliminaries, the optimization of GAN is first to train the discriminator D with a fixed generator G, and the goal is to obtain the optimal discriminator D^*. Then, with the optimal discriminator D^*, the objective function of the generator is updated for training, and the goal is to obtain the optimal generator G^*. (1) When the generator G is fixed, the optimal discriminator D^* is

$$D_G^*(x) = \frac{P(x)}{P(x)+\hat{P}(x)} \tag{14}$$

(2) In the case of the optimal discriminator D^*, the objective function of the generator G becomes:

$$\begin{aligned}
&C(G) = V\left(G, D_G^*\right) \\
&= \mathbb{E}_{x\sim P(x)}\left[\log D_G^*(x)\right] + \mathbb{E}_{x\sim \hat{P}(x)}\left[\log\left(1-D_G^*(x)\right)\right] \\
&= \mathbb{E}_{x\sim P(x)}\left[\log\frac{P(x)}{P(x)+\hat{P}(x)}\right] \\
&+ \mathbb{E}_{x\sim \hat{P}(x)}\left[\log\frac{\hat{P}(x)}{P(x)+\hat{P}(x)}\right] \\
&= \mathbb{E}_{x\sim P(x)}\left[\log\frac{P(x)}{\frac{1}{2}\left(P(x)+\hat{P}(x)\right)}\right] \\
&+ \mathbb{E}_{x\sim \hat{P}(x)}\left[\log\frac{\hat{P}(x)}{\frac{1}{2}\left(P(x)+\hat{P}(x)\right)}\right] - \log 4 \\
&= 2D_{JS}(\hat{P}\|P) - \log 4
\end{aligned} \tag{15}$$

After the above optimization of the GAN, we obtain new data $\hat{X}$ with distribution $\hat{P}$ similar to the original data X with distribution P. In the following we demonstrate that mixing and sampling the new data $\hat{X}$ with the original data X as a new dataset $\tilde{X}$ has a boosting effect on the construction of the ETAN classifier.

$$\tilde{X} = \gamma\hat{X} + X \tag{16}$$

Where γ is a hyperparameter. We make analysis in experiment.

Using mixed data $\tilde{X}$ to construct ETAN to calculate the BDEu scores of the Bayesian network would be better than the scores obtained using only the original data X. This demonstrats that our method can construct a better Bayesian network than ETAN, and naturally superior to TAN classifiers and plain Bayesian classifiers.

$$\begin{aligned}
s_D\left(\mathcal{G}^*_{GAN-ETAN}\right) &\geq s_D\left(\mathcal{G}^*_{\text{ETAN}}\right) \\
s_D\left(\mathcal{G}^*_{GAN-ETAN}\right) &\geq s_D\left(\mathcal{G}^*_{\text{TAN}}\right) \\
s_D\left(\mathcal{G}^*_{GAN-ETAN}\right) &\geq s_D\left(\mathcal{G}^*_{\text{Naïve}}\right)
\end{aligned} \tag{17}$$

Then we construct the optimal Bayesian network as

$$\mathcal{G}^* = \mathrm{argmax}_{\mathcal{G}\in\mathcal{G}_{GAETAN}}\, s_D(\mathcal{G}) \tag{18}$$

Algorithm 1: Training generative adversarial nets with stochastic gradient descent.

Input: X, random Gaussian noise z
Output: $\hat{X}$
for number of training iterations **do**
  for k steps **do**
    Sample minibatch of m noise samples $\left\{z^{(1)},\ldots,z^{(m)}\right\}$
    Sample minibatch of m examples $\left\{x^{(1)},\ldots,x^{(m)}\right\}$ from X
    Update the discriminator by ascending its stochastic gradient:
    $\nabla_{\theta_d}\frac{1}{m}\sum_{i=1}^{m}\left[\log D\left(\boldsymbol{x}^{(i)}\right)+\log\left(1-D\left(G\left(z^{(i)}\right)\right)\right)\right]$
  end
  Sample minibatch of m noise samples $\left\{z^{(1)},\ldots,z^{(m)}\right\}$
  Update the generator by decending its stochastic gradient:
  $\nabla_{\theta_g}\frac{1}{m}\sum_{i=1}^{m}\log\left(1-D\left(G\left(z^{(i)}\right)\right)\right)$
end
$\hat{X} = G^*(z)$ $\tilde{X} = \gamma\hat{X} + X$
return $\tilde{X}$

ISBN: 978-988-14049-4-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

C. GAN-ETAN's structure and algorithm

This section mainly introduces the Extended Tree Augmented Naïve Classifier with GAN. Firsly, fig2 gives a schematic of the structure of GAN-ETAN. It has two main components, one of which is the use of generative adversarial networks (GAN) to generate new data that can be fitted to the original data distribution. The other part is to use the augmented data for ETAN to find the optimal directed acyclic graph (DAG) and then perform downstream tasks such as classification.

GAN-ETAN structure model goes from left to right, in turn. Firstly, the original real data were used to iteratively train the generative adversarial network until new data fitting the distribution of the original data were generated. Then, the original data and the new data were mixed according to probability sampling to obtain the enhanced mixture data. Finally, the augmented mixture data is used for ETAN to obtain the optimal DAG, and the DAG is used for downstream tasks such as classification.

Next we give the algorithm of GAN-ETAN. Algorithm 1 is to alternately train the discriminator and generator in the generative adversarial network. After training, mixing the new data generated by probability sampling with the original data to obtain the augmented data.

Algorithm 2 describes the process of ETAN to construct the optimal directed acyclic graph using the BDeu scoring function. Take the augmented mixed data obtained above as input. Algorithm 2 is improved to use Hill-Climb Search algorithm to compute the directed maximum spanning tree.

Algorithm 2: Searching for optimal DAG with ETAN.

Input: $\tilde{X}$
Output: optimal DAG: $\mathcal{G}^*$
$S_D(\mathcal{G}_{data}) = BDeuSocre(data)$
$S^* \leftarrow -\infty$
for $x_i \in \tilde{X} \backslash \{x_0\}$ **do**
  classAsParent $[x_i] \leftarrow S_D(x_i, \{x_0\}) > S_D(x_i, \emptyset)$
  in $\leftarrow$ Hill-Climb Search(x_i)
  $\mathcal{G} \leftarrow$ buildGraph ($\tilde{X}$, x_i, in, classAsParent)
  if $s_D(\mathcal{G}) > s^*$ **then**
  $\mathcal{G}^* \leftarrow \mathcal{G}$
  $s^* \leftarrow s_D(\mathcal{G})$
end
return $\mathcal{G}^*$

V. Experiments

We conduct experiments for different algorithms on both citation and datasets to verify the effectiveness of the proposed method. Experimental results show that our proposed method has improved effect.

A. Datasets

In order to have a comprehensive evaluation, we adopt two samll networks (< 20 nodes, ASIA, SURVEY), three medium networks ($20 - 50$ nodes, ALARM, BARLEY and INSURANCE) and one large network ($50 - 100$ nodes, HAILFINDER). These datasets have been commonly used to evaluate methods on Bayes structure learning tasks. The statistics of all experimental datasets are shown in Table I. These datasets are all taken from Bayesian Network Repository: https://www.bnlearn.com/bnrepository/.

TABLE I
DETAILS OF SIX EXPERIMENTAL NETWORKS.

	Small Networks		Medium Networks			Large Network
	ASIA	SURVEY	ALARM	BARLEY	INSURANCE	HAILFINDER
# Nodes	8	6	37	20	27	56
# Arcs	8	6	46	25	52	66
# Parameters	18	21	509	230	1008	2656

B. Experiments Setup

Parameter Settings: We set the amount of data in the six original datasets to 200 for all. The number of generated new data is also 200. When using GAN to synthesize new data, the batch size of training is 10 and the epoch of training is 200. The number of network layers in GAN is related to the number of variables in the dataset. $ETAN^*$ uses the hill-climbing algorithm.

Evaluation Metrics: We employ structural hamming distance (SHD) and AUC value to measure the performance of GAN-ETAN. The SHD value represents the distance between the true DAG and the learned DAG, which is the total number of edges that are distinct in the two graphs. The closer SHD value to 0 the better. The AUC value, which stands for area under the ROC curve, has been widely used in previous tasks. The closer AUC value to 1 the better.

Computing Infrastructures: We implement our proposed framework upon PyTorch and pyAgrum. All the experiments are trained on a personal computer server with Windows 11, an NVIDIA GeForce RTX 3070Ti (16GB memory) GPU, an 12th Gen Intel(R) Core(TM) i9-12900H CPU.

C. Overall Performance

We evaluate the performance of the proposed framework by comparing it with the traditional method. The comparison of SHD and AUC scores is demonstrated in Table II. According to the evaluation results, we make the following observations further.

TABLE II
DETAILS OF EXPERIMENT RESULTS.

		Small Networks		Medium Networks			Large Network
		ASIA	SURVEY	ALARM	BARLEY	INSURANCE	HAILFINDER
SHD	$ETAN^*$	12	5	72	87	48	**67**
	GAN-ETAN	**7**	5	**59**	**83**	**43**	71
AUC	$ETAN^*$	0.3375	0.5835	**0.7515**	**0.5685**	0.6405	**0.6755**
	GAN-ETAN	**0.625**	**0.611**	0.7415	0.534	**0.643**	0.6395

From the above results, we can see that our method GAN-ETAN improves significantly on small networks, slightly on medium networks, and is difficult to improve on large networks. The possible reason is that large and medium networks are more complex, and there is a greater demand for data volume. The slight improvement on these complex networks indicates that our method is still considerable and effective.

ISBN: 978-988-14049-4-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

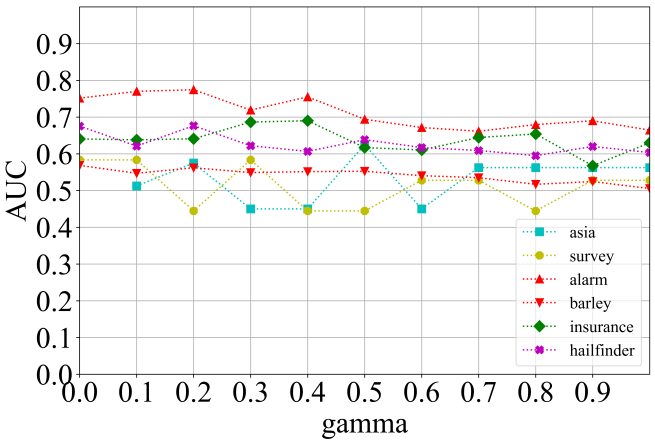

Fig. 3. the Experimental Results of Hyper-parameter

D. Parameters Analysis

In this section, we investigate the effects of the hyper-parameter γ in our proposed GAN-ETAN framework. Experimental results of the AUC scores for all six networks are reported in Figure 3, which suggest that the reasonable choice of γ is around 0.4-0.5.

VI. Conclusion

We propose a novel framework, GAN-ETAN, which combines Extended Tree Augmented Naïve Classifier (ETAN) with Generative Adversarial Network (GAN). GAN-ETAN enhances the original correlation and independence among features and class. GAN-ETAN can build more accurate direct acyclic graphs. The experimental results show that GAN-ETAN can improve the classification performance of original ETAN in some extent.

References

[1] P. Domingos and M. Pazzani, "On the optimality of the simple bayesian classifier under zero-one loss," Machine learning, vol. 29, pp. 103–130, 1997.
[2] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," Machine learning, vol. 29, pp. 131–163, 1997.
[3] C. P. de Campos, M. Cuccu, G. Corani, and M. Zaffalon, "Extended tree augmented naive classifier," in Probabilistic Graphical Models: 7th European Workshop, PGM 2014, Utrecht, The Netherlands, September 17-19, 2014. Proceedings 7. Springer, 2014, pp. 176–189.
[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," Journal of Japan Society for Fuzzy Theory and Intelligent Informatics, 2014.
[5] W. Buntine, "Theory refinement on bayesian networks," Elsevier eBooks, 1991.
[6] G. F. Cooper and E. H. Herskovits, "A bayesian method for the induction of probabilistic networks from data," Machine Learning, 1992.
[7] D. Heckerman, D. Geiger, and D. M. Chickering, "Learning bayesian networks: the combination of knowledge and statistical data," Machine Learning, 1994.
[8] A. Meehan and C. P. de Campos, "Averaged extended tree augmented naive classifier," Entropy, 2015.
[9] G. I. Webb, J. R. Boughton, and Z. Wang, "Not so naive bayes: aggregating one-dependence estimators," Machine Learning, 2005.
[10] B. Mihaljević, C. Bielza, and P. Larrañaga, "Learning bayesian network classifiers with completed partially directed acyclic graphs," Probabilistic Graphical Models, 2018.
[11] S. Acid, L. M. de Campos, and J. G. Castellano, "Learning bayesian network classifiers: Searching in a space of partially directed acyclic graphs," Machine Learning, 2005.
[12] G. Xiao-guang, Y. Yu, and G. Zhigao, "Learning bayesian networks by constrained bayesian estimation," Chinese Journal of Systems Engineering and Electronics, 2019.
[13] S. Sugahara, I. Aomi, and M. Ueno, "Bayesian network model averaging classifiers by subbagging." Entropy, 2020.
[14] S. Sugahara and M. Ueno, "Exact learning augmented naive bayes classifier," arXiv: Learning, 2021.
[15] S. Sugahara, W. Kishida, K. Kato, and M. Ueno, "Recursive autonomy identification-based learning of augmented naive bayes classifiers," in International Conference on Probabilistic Graphical Models. PMLR, 2022, pp. 265–276.
[16] J. Pearl, Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan kaufmann, 1988.
[17] G. F. Cooper and E. Herskovits, "A bayesian method for the induction of probabilistic networks from data," Machine learning, vol. 9, pp. 309–347, 1992.
[18] D. Heckerman, D. Geiger, and D. M. Chickering, "Learning bayesian networks: The combination of knowledge and statistical data," Machine learning, vol. 20, pp. 197–243, 1995.
[19] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," arXiv: Computer Vision and Pattern Recognition, 2016.
[20] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2223–2232.
[21] L. Yu, W. Zhang, J. Wang, and Y. Yu, "Seqgan: Sequence generative adversarial nets with policy gradient," arXiv: Learning, 2016.
[22] G. L. Guimaraes, B. Sanchez-Lengeling, P. L. C. Farias, and A. Aspuru-Guzik, "Objective-reinforced generative adversarial networks (organ) for sequence generation models." arXiv: Machine Learning, 2017.
[23] C.-C. Hsu, H.-T. Hwang, Y.-C. Wu, Y. Tsao, and H.-M. Wang, "Voice conversion from unaligned corpora using variational autoencoding wasserstein generative adversarial networks," Cornell University - arXiv, 2017.
[24] D. Li, D. Chen, J. Goh, and S.-k. Ng, "Anomaly detection with generative adversarial networks for multivariate time series," arXiv preprint arXiv:1809.04758, 2018.

ISBN: 978-988-14049-4-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)