IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 12, 2021, accepted January 3, 2022, date of publication January 11, 2022, date of current version January 31, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3141776

A Hybrid GAN-Based Approach to Solve Imbalanced Data Problem in Recommendation Systems

WAFA SHAFQAT AND YUNG-CHEOL BYUN
Department of Computer Engineering, Jeju National University, Jeju 63243, South Korea

Corresponding author: Yung-Cheol Byun (ycb@jejunu.ac.kr)

This work was supported in part by the Korea Institute for Advancement of Technology (KIAT) Grant through the Korea Government [Ministry of Trade, Industry & Energy (MOTIE)] (The Establishment Project of Industry-University Fusion District) under Grant P0016977; and in part by the Ministry of Small and Medium-Sized Enterprises (SMEs) and Startups (MSS), South Korea, under the ''Regional Specialized Industry Development Program (Research and Development)'' supervised by the Korea Institute for Advancement of Technology (KIAT), under Grant S3091627.

ABSTRACT With the advent of information technology, the amount of online data generation has been massive. Recommendation systems have become an effective tool in filtering information and solving the problem of information overload. Machine learning algorithms to build these recommendation systems require well-balanced data in terms of class distribution, but real-world datasets are mostly imbalanced in nature. Imbalanced data imposes a classifier to focus more on the majority class, neglecting other classes of interests and thus hindering the predictive performance of any classification model. There exist many traditional techniques for oversampling minority classes. Still, generative adversarial networks (GAN) have been showing excellent results in generating realistic synthetic tabular data that keeps the probability distribution of the original data intact. In this paper, we propose a hybrid GAN approach to solve the data imbalance problem to enhance recommendation systems' performance. We implemented conditional Wasserstein GAN with gradient penalty to generate tabular data containing both numerical and categorical values. We also augmented auxiliary classifier loss to enforce the model to explicitly generate data belonging to the minority class. We designed the discriminator architecture with the concept of PacGAN to receive m-packed samples as input instead of a single input. This inclusion of the PacGAN architecture eliminated the mode collapse problem in our proposed model. We did a two-fold evaluation of our model. Firstly based on the quality of the generated data and secondly on how different recommendation models perform using the generated data compared to original data.

INDEX TERMS GAN, imbalanced data, oversampling, synthetic data, recommendation systems, condition GAN, WGAN-GP, PacGAN.

I. INTRODUCTION

An enormous amount of data is daily generated at a large scale in every domain. Research says [1] digital data has grown nine times in volume in the past five years. Also, according to the current statistics, the world will obtain more than 35 trillion gigabytes of digital data or even more by 2023 [2]. eCommerce is being increasingly populated with new customers. Hence, researchers are constantly trying to enhance the user experience and develop a market that is beneficial for both the customer and the company. Recommendation systems have been a leading-edge technology in extracting meaningful information from the widely collected unstructured data in every field. Recommendation systems organize this abundance of data and help in customers' decision-making by narrowing down the options and analyzing every customer's personal preferences; that saves a lot of time and has the power to promote unexplored products or places that could benefit the business.

Imbalanced data can be a problem where the data used for training is not distributed equally for every category or class. Training machine learning algorithms with imbalanced data can give a training accuracy of more than 90%. Still, when we try to test our model, our model outputs poor

The associate editor coordinating the review of this manuscript and approving it for publication was Yin Zhang.

results or results favoring the majority class. In many cases, the group of interest might be the minority class [3], so training models become very difficult. The imbalance data problem can be solved in three ways: algorithm-level methods, data-level methods, and hybrid approaches [4]. The algorithm-level method uses weight schema to modify the latent learner for reducing bias towards the majority class [4]. In the data-level techniques, various data sampling methods are used to handle data imbalance problems. Hybrid methods are the combination of the above two methods. Researchers are continuously trying to enhance the accuracy of recommendation systems. In a latest work, Cai *et al.* [5] a recommendation system built on the basis of item-to-item transition patterns in every category. They proposed a collaborative sequential recommendation model and demonstrated that category-wise item-to-item transitions are better indicators of next items than general item-to-item transitions. In another work, Chen *et al.* [6] focused on the co-occurrence of items and developed a recommendation system that is modeled both on user-item and item-item interaction. The authors claimed that a co-occurrence neural network is better capable of encoding highly descriptive features compared to other baseline approaches. Jiang *et al.* [7] proposed a content recommendation model that dynamically learns knowledge features and gives an overview of past user preferences along with knowledge graphs. The authors also propose a self-attention-based knowledge representation that learns semantics from triples to improve the performance of recommendation systems.

The main contribution of our work is to build a hybrid generative adversarial networks (GAN) [8] based approach for solving data imbalance problems to enhance recommendation systems' performance. We name our proposed architecture as (CWGAN-GP-PacGAN). The implementation process through which we achieve our goal is as follows:

- We develop a hybrid GAN approach that learns the distribution of data and conditions the generator on the minority class through conditional GAN. Our model is trained to generate tabular data consisting of both categorical and numerical data.
- We train the model with the loss function of Wasserstein GAN with gradient penalty (WGAN-GP). We also augment auxiliary classifier loss to explicitly condition the generator and discriminator to generate data that belongs to the minority class.
- We inherit the concept of PacGAN to design the architecture of our discriminator. We provide m-packed samples of data as input to the discriminator to eliminate the mode collapse problem.
- We present a thorough performance evaluation section where we evaluate the generated synthetic data for oversampling and experiment the generated data with various traditional and recent models to see how each model performs with the balanced dataset.

II. RELATED WORKS

In machine learning, class imbalance or data imbalance is extensively considered one of the most significant challenging issues. Therefore, various oversampling approaches have been proposed to cater to data imbalance problems in different fields. Random over-sampling (ROS) is the most naïve technique that randomly duplicates minority class instances to increase the sample size. Many studies have used various variations of the ROS technique before training the model. For example, in a study presented in [9], a new distributional oversampling approach is designed to classify the text data. It takes advantage of the distributional characteristics of the terms in the data and generates new documents of the minority class. Similarly, in [10], a wrapper-based random oversampling is used to preprocess the data to handle the class imbalance problem. In this approach, different regions are identified that are appropriate for sampling, and the Genetic Algorithm (GA) works as an underlined model for discovering optimal regions.

Another technique, named SMOTE proposed in [11], has gained substantial popularity. It uses k-NN with a random distance to generate data in the neighbors of minority class data. Different variations of SMOTE have been introduced with specific improvements, e.g., in [12], a Borderline SMOTE is presented in which only those instances are oversampled that lies near the borderline of the minority class. Another variation of SMOTE, named CE-SMOTE by [13], introduced the concept of clustering consistency index to figure out the cluster boundary minority examples. After this, these minority example instances are over-sampled to augment the original data. In another variation, SMOTE was also combined with a boosting technique in [14] named SMOTEBoost that generated synthetic data from the minority class where the boosting weights were indirectly changed and hence considered the skewed distributions.

On the other hand, GANs (Generative Adversarial Networks) are becoming widely successful as an alternative approach for imbalanced data handling. GAN, initially proposed by [8], is a famous data augmentation approach that creates new data by learning the actual data distribution. Since then, many variations of GAN have been proposed in different domains. In [15], GAN and its variant are used to create synthetic data of fake transactions. The author uses WGAN and conditional WGAN for the purpose of generating synthetic data of fake transactions. But, the performance could have been enhanced with an improved version of WGAN i.e. WGAN-GP. A novel GAN-based anomaly detection technique based on an autoencoder and two different discriminators is presented in [16]. An end-to-end model named ITCGAN is proposed in [17] for imbalanced traffic categorization. ITCGAN generates traffic samples for minority classes to simultaneously balance the original traffic and train the optimal classifier. GAN's are widely used for synthetic data generation and oversampling data to handle imbalanced data problems in various domains. GAN is used to produce images, but GAN's are now being used to generate tabular or time-series data. Adversarial networks can now handle complicated data structures in tabular data. The authors in [18] proposed a three-layer GAN architecture to solve the class imbalance problem through minority class oversampling. The three-tier architecture consists of a convex generator, a multiclass classifier network, and a discriminator. The author mentioned in their work that the GAMO architecture has a tendency to overfit and can be improved through the use of hybrid architectures of GAN. Also, the quality of the generated images can be improved even with less training samples if hybrid architectures are used. Koivu *et al.* [19] proposed a neural-network-based activation-specific GAN to generate synthetic data to enhance prediction probability. Conditional GAN is one of the famous approaches for class imbalance problems since the generator can be conditioned with the class to be oversampled.The authors of actGAN indicate that first-trimester biophysical measurement data is important for improving the detection rate, but the authors couldn't use it due to unavailability. The detection rate can be enhanced by generating these data belonging to the minority class. The authors use WGAN-GP architecture but without automatic class conditioning. The performance and relevance of the model could have been improved with the inclusion of class conditioning. The authors in [20] proposed a minority class oversampling technique that uses conditional generative adversarial networks to restore proper training data distribution. The author proposes the oversampling technique for binary class imbalanced data and does not handle the multiclass imbalanced problem. In our work, we take the concept of hybrid GAN architecture to generate synthetic click sequence data and oversample the minority class to enhance the performance of recommendation systems. The following sections will overview our proposed work, the dataset used, and our generated synthetic data performance evaluation.

III. DATA

We explain the data we have used for our experiment in this section. Our primary goal in this paper is to solve the imbalance problem in the dataset and oversample the data of the minority class through generating synthetic data using a hybrid GAN model. Therefore, we have spent quality time first analyzing and preprocessing the data to extract meaningful and relevant information from the dataset. We have acquired the input data for the GAN and recommendation model from an online shopping mall in Jeju, South Korea, known as the eJeju mall. The data section is further subdivided into click data of users, personal data, and product detail data. The dataset contains information about the IP address of the user's; product accessed or clicked time, user's browsing period, URL of the accessed page by the user, ID of the product clicked, name of the product, and whether the product has been saved in the cart, or just listed as an item. The eJeju mall data consists of 294894 records over one year (2019-2020). The total number of unique customers is 51386 with 7701 purchased products, 244533 unique clicks, and 42630 visits for item description. In table 1 we have presented the click sequence length for minority classes. The minimum size of the total number of clicked products is 3 with the total number of records as mentioned in table 1. The maximum size for click sequence in a single session varied from 15 to 40. We have also mentioned the details about how many rows were discarded, the number of relevant records, and expanded records.

TABLE 1. Characteristics of the dataset.

Experimental Data	Click Sequence Length for Minority Classes		
	3 - 15	3 - 25	3- 40
Total number of rows	294894	294894	294894
Rows discarded (length<minimum)	1440	2596	2321
Relevant rows	293454	292298	292573
Expanded rows	3013	3013	3013

A. CLICK DATA OF USER'S

The most important data here in this dataset is the user click data, where the user's click history, such as previously clicked items or the clicked items in one session, is stored. This data helps in understanding the behavioral pattern of each user. Also, through this data, we can explore the most clicked or purchased items and different click patterns or sequences of various users. We extracted information such as the number of unique clicks, total number of purchased items per user, and total number of clicks per session.

B. USER'S PERSONAL DATA

The dataset also provides us the IP address of each user, date, and time of clicked items per session. IP address helped us extract the location information of every customer, and from the date and time information, we could create more sessions or extract more information about sessions. We derived session ID and number of sessions from the user's data.

C. PRODUCT DETAIL DATA

In this dataset, we have product IDs, product names, and product click types. Since we are working on enhancing the performance of product recommendation systems, we considered these crucial data while labeling minority and majority classes for our GAN models for generating synthetic data.

IV. METHODOLOGY

This section presents the detailed implementation process of our proposed methodology for solving the data imbalance problem in recommendation systems using hybrid GAN models. As shown in Fig. 1, the process initiates with taking eJeju mall online shopping raw data as input. As explained in the data section, the raw data has time and date information, IP address, click details, action type, item ID, item type, and item name information. The preprocessing unit identifies the underlying patterns of regularity and irregularity of

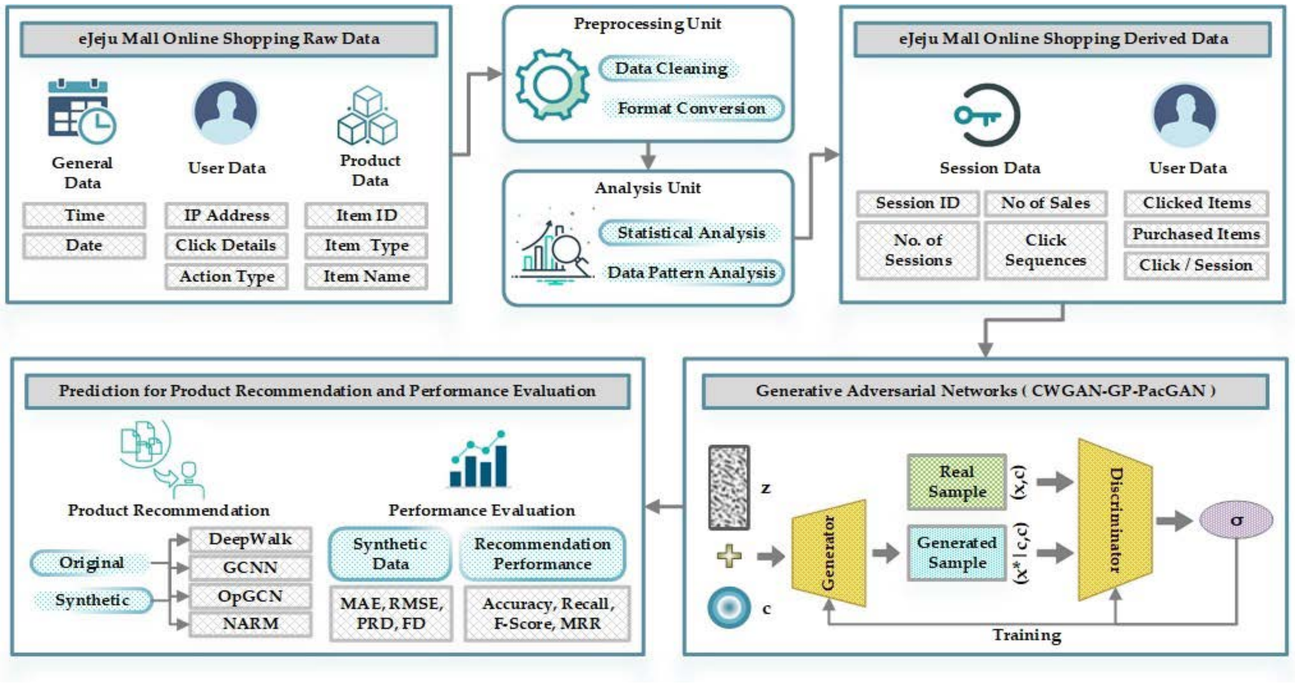

FIGURE 1. **Overview of the proposed model.**

data. We understand the data structure and implement data cleansing and format conversion in this preprocessing phase. Then we send the preprocessed data to the analysis unit. We perform statistical analysis such as unique IPs, clicks per product, unique sessions, sales per user and session, etc. We then perform data pattern analysis, filtrate the relevant data, and discard the rest responsible for anomalies and irregularities.

After preprocessing and research, we obtain the derived data, then passed to the proposed hybrid GAN model. The hybrid model includes the architecture of conditional GAN [18], WGAN-GP [19], and PacGAN [20] to generate synthetic data and oversample the minority classes to enhance the performance of recommendation models. We have implemented conditional GAN architecture that helps us explicitly condition the minority class, i.e., items that have been clicked more but bought less. Our idea behind targeting this scenario is that even if popular items might not be bestsellers, they might be less priced and affordable, in which case it is necessary to draw customers' attention to those products. Often, people do not try to buy products that are less priced but buy products that belong to the best seller category and might be comparatively high priced. In such cases, those neglected items remain unexplored and unreviewed. We use the loss function of WGAN-GP and integrate an auxiliary classifier (AC) loss to generate samples that evidently belong to the given minority class. We introduce the concept of sampled input to discriminator as in PacGAN to eliminate the model collapse problem of GAN architecture and improve performance. We then move on to the performance evaluation phase, where we evaluate the synthetic data's quality and performance of recommendation systems when using the synthetic data.

A. CWGAN-GP-PACGAN

Goodfellow *et al.* [8] introduced generative adversarial networks (GAN) in the year 2014. Since then, GAN has been widely used and is still being developed in different forms for different applications. GAN is a machine learning framework with two neural networks known as the generator (G) and the discriminator (D) network that competes against each other in a zero-sum game. The generator is fed with random input noise variable n_z that retains P_d's data distribution over the original data distribution x. The discriminator receives samples generated from the generator and also the actual data. The Work of the discriminator is to correctly identify whether the data is from the real or the generated samples. The generator tries to fool the discriminator, and meanwhile, the discriminator works to distinguish the real samples from the fake ones accurately. At a particular point, the generator generates realistic samples and cannot be distinguished from real samples. Therefore, the generator tries minimizing $log(1 - D(G(z)))$, whereas the min-max function can be formulated as in equation 1.

$$\min_G \max_D F(G, D) = \mathbb{E}_{x \sim P_d}[logD(x)] + \mathbb{E}_{z \sim n_z}[log(1 - D(G(z)))] \qquad (1)$$

Typically, GAN's are trained with the loss function of Vanilla GAN that is hard to train and might not converge to Nash equilibrium. Using the loss function of Vanilla GAN, it becomes difficult to assess whether the generator is still trying to learn, has finished converging, or has stopped learning and collapsed. It also leads to mode collapse, where the generator maps all possible values to a single output. Wasserstein GAN [21] solves all these problems by optimizing Wasserstein-1 distance, also called the Earth Mover distance, instead of the Jensen-Shannon Divergence (JSD) used in Vanilla GAN. The objective of Wasserstein GAN is to build a powerful discriminator that would output a significant gradient for the generator even if the generator's performance is poor. With slight alteration in the GAN objective, Wasserstein-1 distance can be defined as in equation 2 with D being a k-Lipschitz function.

$$\min_G \max_D \mathbb{E}_{x\sim P_d}[D(x)] - \mathbb{E}_{z\sim n_z}[D(G(z))] \tag{2}$$

Wasserstein GAN requires weight-clipping of the critic in a compact space $[-c, c]$, where the critic is the optimal discriminator. Although Wasserstein GAN performs better than the Vanilla GAN, it leads to optimization difficulties and vanishing or exploding gradient problems. Gulrajani *et al.* [22] proposed a different way of imposing the Lipschitz constraint and introduced WGAN-GP, a Wasserstein GAN with a gradient penalty. The gradient penalty in the WGAN-GP architecture complies with the condition of the 1-Lipschitz and replaces the weight-clipping mechanism in Wasserstein GAN. WGAN-GP proposes that if a differentiable function has gradients with the norm at most one everywhere, it can be considered as 1-Lipschitz [22]. So, the objective of WGAN-GP would be a combination of the original critic loss and the gradient penalty, as shown in equations 3 and 4.

$$OriginalCriticLoss = \mathbb{E}_{\tilde{x}\sim P_d}[D(\tilde{x})] - \mathbb{E}_{x\sim P_r}[D(G(x))] \tag{3}$$

$$GradientPenalty = \lambda \mathbb{E}_{\hat{x}\sim P_{\hat{x}}}[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2] \tag{4}$$

where $\hat{x} \sim P_{\hat{x}}$ defines random samples, P_d and P_r denote data, and generator distribution and λ, according to [22] and our experiments, has been selected as 10. Although it is computationally demanding to compute gradient penalty, it is easier to train WGAN-GP.

Conditional GAN, known as CGAN [23], generates the output of a specific class by conditioning the generator on class labels. On the other hand, the discriminator is prepared to ensure that the generator does not ignore the conditioned class label, making the training process firm. In this work, we first train CGAN to estimate the distribution of our data that helps us explicitly sample minority class, i.e., click data of items that are visited and clicked a lot of times but have been purchased less. We try to oversample the click sequences of this minority class by conditioning the generator on the minority class. We obtain the category from the derived data, such as whether the item has just been visited or bought. We also generate the frequency of the categories, and according to the frequency distribution, we divide the data into majority and minority classes. We then append the minority class to the input of both the generator and the discriminator. We use the loss function of WGAN-GP due to its advantages mentioned above. We also integrate an auxiliary classifier A_C loss to ensure that the generator explicitly generates data belonging to the minority class. We use different networks for the discriminator and the A_C, unlike AC-GAN [24]. This architecture helps us to simultaneously compute the A_C loss as well as condition the discriminator. So our GAN objective could be defined as in equation 5.

$$\begin{aligned}\min_G \max_D\ & \mathbb{E}_{\tilde{x}\sim P_d}[D(\tilde{x})] - \mathbb{E}_{x\sim P_r}[D(G(x))] \\ & + \lambda \mathbb{E}_{\hat{x}\sim P_{\hat{x}}}[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2] \\ & + \lambda_{A_C} \mathbb{E}_{x\sim P_r}[[B_{CE}(A_C(G(x)))]\end{aligned} \tag{5}$$

where A_C is our defined network for the auxiliary classifier, B_{CE} defines the binary cross-entropy between the original class labels and the class's predicted probability, either 0 or 1, since it has two categories, majority, and minority. We continuously update the scale factor λ_{A_C} of the A_C loss to eight percent of the absolute value of $D(G(x))$ to ensure that the generator's primary objective would be to reduce the Wasserstein loss.

B. NETWORK ARCHITECTURE

This section will describe the architecture of the generator, discriminator, and auxiliary classifier.

The generator is provided with the latent noise and a condition, i.e., class label, to generate synthetic data. The input is then passed parallel through the hidden and cross layers. Cross layers precisely compute the feature interaction and help model the relationship between variables. Cross layers increase the variation of noise in the generator. The final output from the last hidden and cross layers is concatenated and passed as input to the output layer. Each categorical column generates a single output. For generating output for the numerical column, each categorical column is embedded, passed through the hidden layer, and served as input to the numerical output layer. Many a time, numerical columns are correlated with the categorical column. So, we follow the approach of self-conditioning, where the output from the generator for the numerical column is conditioned on its own output for the categorical columns. This helps in capturing the relationships between numerical and categorical columns. In Fig. 2, we give the overview of the generator architecture where $C_1, \ldots, C_n$ are categorical columns, $E_1, \ldots, E_n$ are the embedding layers, H_N is the hidden layer, Num stands for numerical columns, Out_c and Out_N represent output for categorical and numerical columns.

As shown in Fig. 3, the discriminator receives the embedded categorical columns, the numerical columns combined with noise, and a class label as input. We perform the embedding by sending the one-vector of the original data through the embedding layers. The embedding layers reduce the dimension and represent the input in a better way to the discriminator. We also add Gaussian noise to the numerical columns before passing it to the hidden and cross layers. For the discriminator, we use the concept of

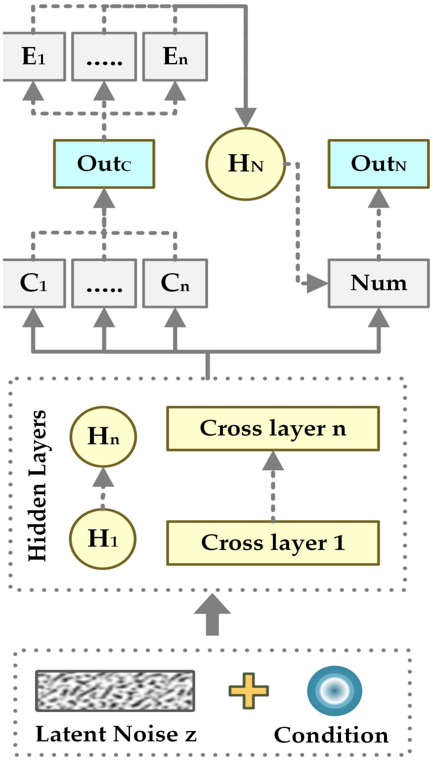

FIGURE 2. Architecture of the generator.

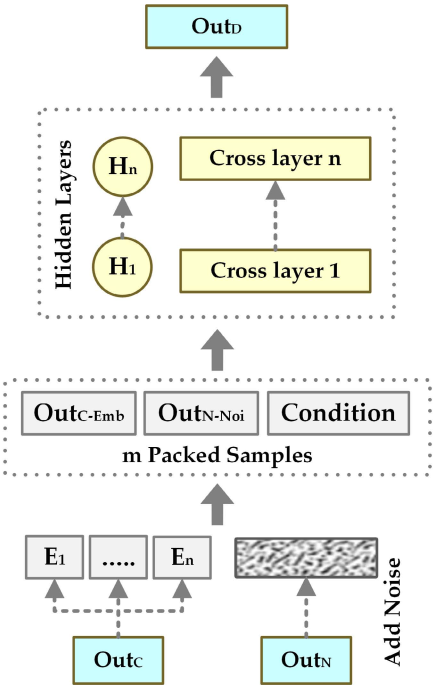

FIGURE 3. Architecture of the discriminator.

PacGAN [25], where the discriminator is provided with m samples packed jointly from either generated data or actual data instead of being provided a single input from the actual or generated data. These samples are taken independently from either the real or the generated distribution. In our work, we pack ten samples. The discriminator contains L fully connected layers, with every layer containing N nodes. The discriminator generates a binary output, so the $(L + 1)th$ layer has a single node fully connected to the previous layer. We compute the computational complexity of the parameter update of a single minibatch where every minibatch contains S samples. The backpropagation is done by the matrix-vector multiplication in each hidden layer with complexity $O(N^2)$ for every input sample. Therefore, the total minibatch update complexity is $O(SLN^2)$. So if there are M packed samples, the per-minibatch complexity grows to $O((L + M)SN^2)$. So, complexity increases with an increased sample size. We experimented with 5, 10, 15, and 20 sample sizes, among which the optimal solution was 10 packed samples that generated good results.

So the input to the discriminator is m or 10 packed samples of categorical and numerical from real or generated distribution along with the condition. The inclusion of the PacGAN concept helps our model to eliminate the mode collapse problem. So these m packed samples are then passed to the next layer, which runs hidden and crosses layers parallel. Since layer normalization normalizes the activation along the feature direction instead of the mini-batch direction, layer normalization after each hidden layer is used to speed up the learning process and stabilize the training process. The discriminator at the final layer generates a scalar output and does not use any activation function.

In Fig. 4, we present the architecture of auxiliary classifier that almost resembles the discriminator architecture except for the fact that we do not add any noise to the numerical columns, we do not take any condition or class label in the auxiliary classifier, and at the final layer, we implement a sigmoid activation function to get the probability of a class label. So first, each categorical column is embedded, and then along with the numerical data, it is passed to the hidden and cross layers in parallel. Output from the last layer of the hidden and cross layers are then passed through sigmoid activation, and finally, the auxiliary classifier outputs the probability for a class label.

C. DATA MODELING

We have modeled our GAN architecture to handle both numerical and categorical data. We have used min-max scaling for numerical data and did not apply any activation function for the generator's output. The purpose of not using any activation function is to let the generator output and constrain values within the training data range. We also add noise to the discriminator input for numerical data, and our noise is sampled from a uniform distribution. We have used uniform distribution because sampling from normal distribution can often take extreme values, which, when passed through the generator, outputs extreme values for the numerical output which is not desired.

We have used one-hot encoding and Gumbel-softmax [26], [27] activation functions for categorical data. We added

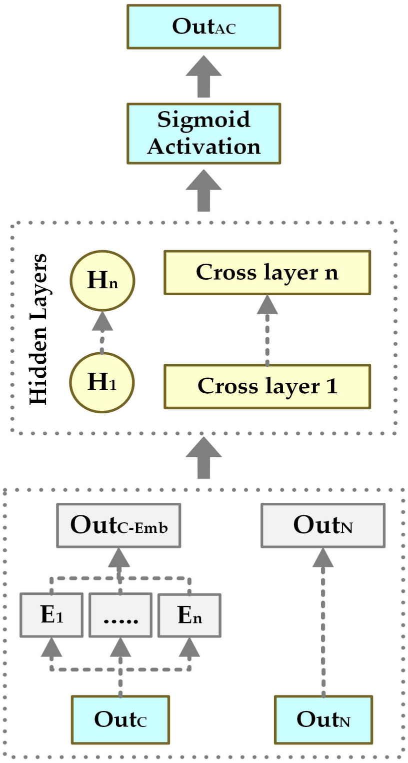

FIGURE 4. Architecture of the auxiliary classifier (AC).

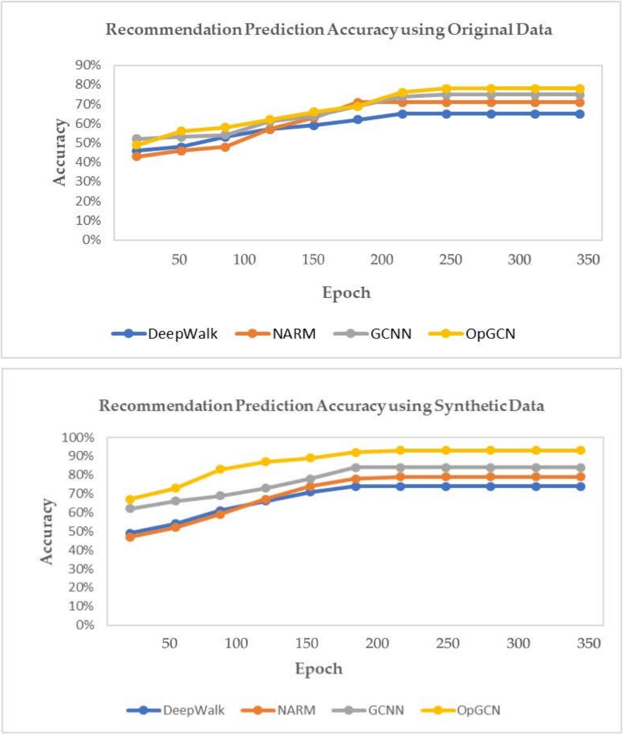

FIGURE 5. Product recommendation prediction accuracy with generated synthetic data.

noise from the Gumbel distribution to the logits. Let us assume a vector v that defines the log probabilities that are not normalized for every category C. Then, the Gumbel-softmax is applied to every vector v_i, as shown in equation 6

$$Gumbel - softmax(v_i) = \frac{\exp\left((v_i + G_i)/\tau\right)}{\sum_{j=1}^{C} \exp\left((v_j + G_j)/\tau\right)} \quad for\ i = 1, \ldots, C \tag{6}$$

where $G_1, \ldots, G_C$ are i.i.d from Gumbel(0,1). τ is the parameter for temperature, which controls the output diversity. We chose the temperature $\tau = 0.58$, i.e., less than one, to enforce the generator to connect the latent noise input's explicit values with each categories.

V. PERFORMANCE EVALUATION

This section presents the detailed evaluation of our proposed model and compares its performance with other existing models. We measure the performance of our proposed model based on two objectives. First is the quality of the synthetic data, and second is the performance of recommendation systems using the generated synthetic data. We have compared the quality of our generated synthetic data with other traditional oversampling techniques like SMOTE [11], B-SMOTE [12], SMOTENC [28] and ADASYN [29]. We also quantified our performance and compared it with recent GAN approaches like TGAN [30], CTGAN [31], TGAN-skip [32], TGAN-WGAN-GP [33], WCGAN-GP [34], and TGAN-skip-WGAN-GP [32]. For synthetic data evaluation, we used metrics like mean correlation coefficient, and root means square error (RMSE), Fréchet Distance (FD), mean absolute error (MAE), mirror column association, and percent root mean square difference (PRD). For the performance of recommendation systems, we used evaluation metrics like accuracy, recall, F-score, and mean reciprocal rank (MRR). This section also presents the mean and standard deviation comparison of original and synthetic data.

In Fig. 5, we present the prediction accuracy of recommendation models such as DeepWalk, NARM, GCNN, and OpGCN using original and data and generated synthetic data from our proposed model. As we can see, for most models, especially OpGCN, the accuracy increases considerably when synthetic data from our proposed model is being used. The original and synthetic datasets were split into 70% training and 30% testing. While training, we used 10% of the training dataset for validation. In Fig. 5, the prediction accuracy of the recommendation model on the training dataset has been shown. We present the prediction accuracy for the test dataset in Fig. 6 which shows that the model does not overfit with our mentioned dataset.

A. SYNTHETIC DATA

Correlation between the original and generated data for oversampling is an important metric to evaluate the quality of the synthetic data. Pearson's correlation coefficient that ranges from −1 to 1 has been used in this work to quantify the correlation between two variables. Values between 0 to

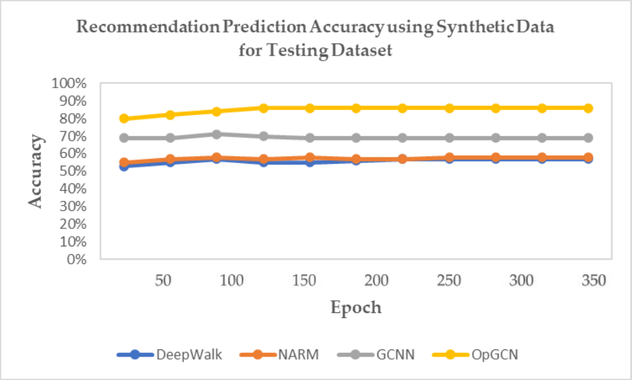

FIGURE 6. Product recommendation prediction accuracy with generated synthetic data for test dataset.

0.3 or 0 to −0.3 represents negligible correlation, whereas values between 0.3 to 0.5 or −0.3 to −0.5 and 0.5 to 0.7 or −0.5 to -0.7 depicts low and moderate correlation. Values ranging in 0.7 to 0.9 or −0.7 and −0.9 and 0.9 to 1 or -0.9 to 1 represent highly or extensively correlated. Zero represents no relationship, and the positive and negative values represent a direct or indirect relationship. The mean correlation coefficient can be formulated in equation 7, where O and S signify original and synthetic data.

$$Coef = \frac{\sum_{i=1}^{n}(O_i - O)(S_i - S)}{\sqrt{[\sum_{i=1}^{n}(O_i - \bar{O})][\sum_{i=1}^{n}(S_i - \bar{S})]}} \quad (7)$$

Through the RMSE value, we measure the stability of our proposed model and quantify the difference between the original and the generated data. RMSE can be formulated, as in equation 8.

$$RMSE = \sqrt{n\sum_{i=1}^{n}(O_i - S_i)^2} \quad (8)$$

Frechet distance finds the similarity between ordering and position of points along the curves. Let us assume that $O_R = a_1, a_2, a_3, \ldots, a_R$ represents the order of points along the original curve and $S_Q = d_1, d_2, d_3, \ldots, d_Q$ shows the order of points along the generated data curve, then we can measure the sequence length $\|l\|$, as in equation 9:

$$\|l\| = \max_{i=1,\ldots,n} l(a_{s_i}, d_{t_i}) \quad (9)$$

where euclidean distance is signified through l and a_{s_i} and d_{t_i} measures the order of sequence points. So, we formulate the Fréchet Distance as in equation (10) [35].

$$FD(R, Q) = \min \|l\| \quad (10)$$

Mean absolute error calculates the average absolute error between the original and the generated data, as shown in equation 11.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|O_i - S_i| \quad (11)$$

We compute the mirror column association to find the relationship of columns between the original and the generated dataset. The greater the value, the higher is the association between columns which indicates better results.

We compute the distortion between the original and generated data through the percent root mean square difference, as shown in equation 12.

$$PRD = \sqrt{100\frac{\sum_{i=1}^{n}(O_i - S_i)^2}{\sum_{i=1}^{n}(O_i)^2}} \quad (12)$$

Computation of mean and standard deviation is another good approach for visual evaluation of the correlation between original and synthetic data. In Fig. 7, the green line represents the original data, and the plotted points are the synthetic or generated data. The closer the points to the green line, the greater is the correlation between the original and the synthetic data. We have compared the mean and standard deviation of original and synthetic data generated through TGAN-WGAN-GP, WCGAN-GP, TGAN-skip-WGAN-GP, and our proposed model CWGAN-GP-PacGAN. Results indicate that the generated data through our proposed model is closest to the diagonal, indicating better performance than other GAN models.

In table 2, we have presented the result comparison of different models. We have evaluated each model based on their generated synthetic data for oversampling. We have used our dataset to measure the performance of each model. The correlation coefficient for our proposed model is the highest, and it produces much less error than other models. Column-wise association is also remarkably good for the generated data through our proposed model CWGAN-GP-PacGAN.

B. RECOMMENDATION PREDICTION

We use the synthetic or generated data from our proposed GAN model to evaluate the performance of recommendation models. We consider the synthetic data of our dataset generated through various models and compare the performance of different recommendation models such as DeepWalk, NARM, GCNN, and OpGCNN.

The evaluation metrics that we considered are recall, F-score, MRR, DCG, and NDCG. We find the correct predictions through recall value, so greater or more significant recall values are an indication of much better results. F-score measures the accuracy of the recommendation or prediction model, so the higher the value for F-score, the better the model's performance. As we can see in table 3 and table 4 the recall and F-score value for our proposed model CWGAN-GP-PacGAN is the highest, indicating better results as compared to other models. MRR is the mean reciprocal rank that measures the first or the most relevant items. We also present the value for MRR@5 and MRR@20. There was a slight improvement when we considered MRR@5. We experimented and provided the results for discounted cumulative gain (DCG@5) and normalized discounted cumulative gain for GCNN and OpGCN recommendation

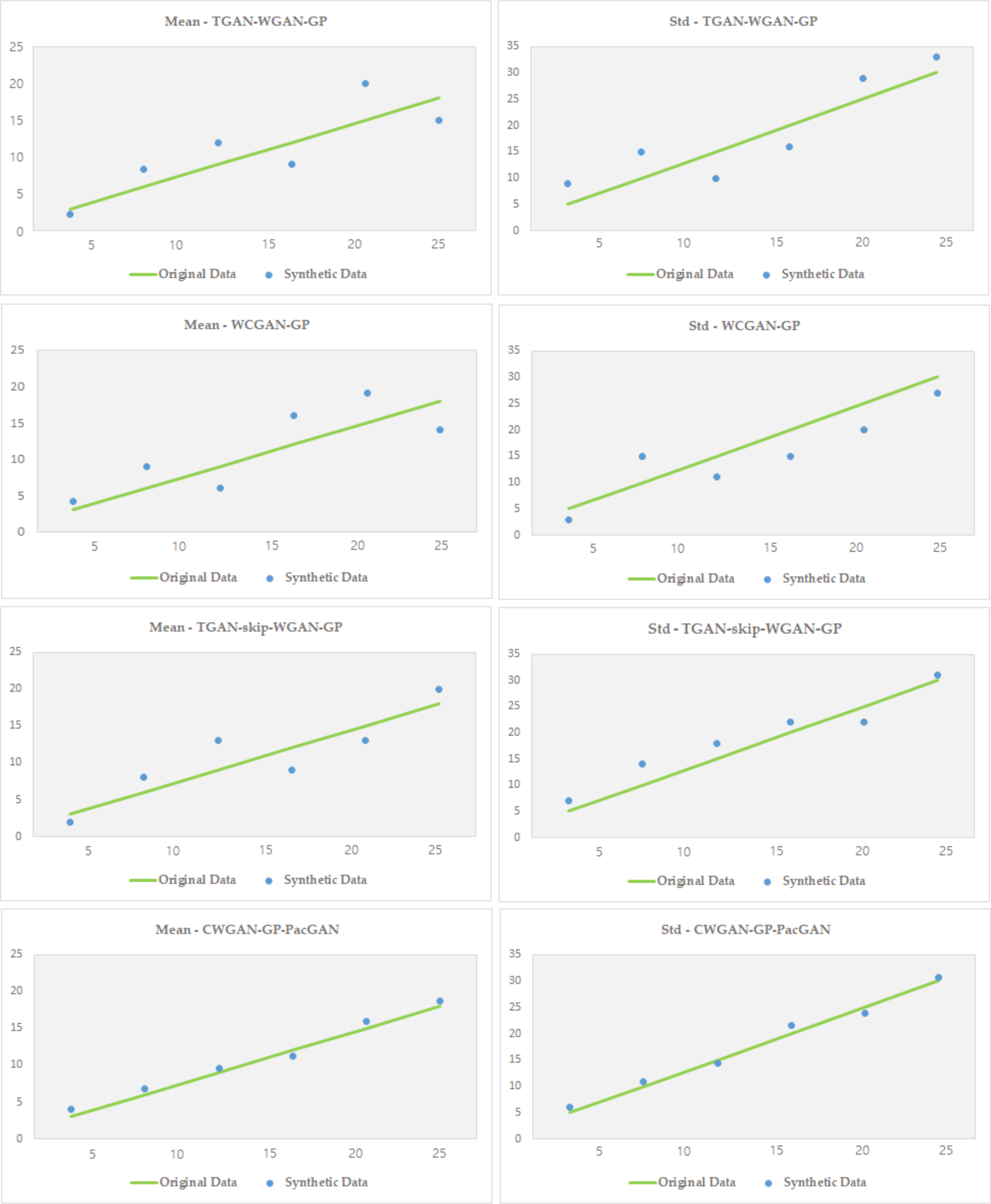

FIGURE 7. **Comparison of mean and standard deviation of original and synthetic data for different GAN models.**

models on the original dataset as well as synthetic dataset generated by various models. Every recommendation system has a relevance score, and cumulative gain is computed by summing all the relevance scores in a recommendation

TABLE 2. Model and quality evaluation of generated synthetic data.

Model	Mean Correlation Coefficient	RMSE	FD	MAE	Mirror Column Association	PRD
SMOTE	0.621	0.81	0.83	0.79	0.597	86.9
B-SMOTE	0.644	0.74	0.76	0.72	0.636	82.7
SMOTENC	0.639	0.76	0.79	0.74	0.601	84.2
ADASYN	0.589	0.84	0.86	0.83	0.573	88.1
TGAN	0.782	0.74	0.69	0.71	0.726	76.4
CTGAN	0.811	0.75	0.71	0.68	0.734	71.5
TGAN-skip	0.838	0.62	0.67	0.62	0.767	68.7
TGAN-WGAN-GP	0.763	0.74	0.74	0.73	0.718	77.1
WCGAN-GP	0.882	0.59	0.63	0.57	0.830	64.6
TGAN-skip-WGAN-GP	0.927	0.53	0.59	0.51	0.894	58.9
CWGAN-GP-PacGAN	0.946	0.49	0.56	0.46	0.941	53.1

TABLE 3. Performance comparison of DeepWalk and NARM model for product recommendation with original and synthetic data obtained from different models.

Model	DeepWalk			NARM		
	Recall@20	F-Score	MRR@20	Recall@20	F-Score	MRR@20
Original Data	25.62	0.581	11.37	29.39	0.632	13.41
SMOTE	30.87	0.614	16.73	35.71	0.657	18.99
B-SMOTE	36.51	0.657	21.03	39.66	0.683	21.72
SMOTENC	33.44	0.629	19.55	37.52	0.661	20.88
ADASYN	28.91	0.561	13.72	31.36	0.642	15.63
TGAN	45.82	0.692	23.91	42.82	0.717	27.89
CTGAN	47.77	0.713	28.75	46.71	0.736	29.01
TGAN-skip	51.63	0.756	32.10	56.90	0.773	31.67
TGAN-WGAN-GP	42.7	0.681	21.77	40.91	0.695	26.72
WCGAN-GP	56.39	0.793	36.92	59.01	0.814	38.91
TGAN-skip-WGAN-GP	59.71	0.825	38.66	66.88	0.849	41.02
CWGAN-GP-PacGAN	63.48	0.836	41.37	69.94	0.862	44.91

TABLE 4. Performance comparison of GCNN and OpGCN model for product recommendation with original and synthetic data obtained from different models.

Model	GCNN			OpGCN		
	Recall@20	F-Score	MRR@20	Recall@20	F-Score	MRR@20
Original Data	31.81	0.641	15.83	35.67	0.663	21.88
SMOTE	36.63	0.692	19.02	39.01	0.723	26.71
B-SMOTE	39.10	0.713	23.73	42.36	0.774	29.82
SMOTENC	37.21	0.702	20.17	40.72	0.752	27.33
ADASYN	33.64	0.664	17.82	36.78	0.690	23.47
TGAN	41.72	0.733	27.33	46.35	0.792	34.60
CTGAN	49.85	0.749	33.91	51.24	0.836	35.19
TGAN-skip	56.71	0.791	37.80	59.01	0.851	39.02
TGAN-WGAN-GP	40.10	0.714	25.68	43.47	0.763	32.39
WCGAN-GP	67.63	0.825	39.02	68.91	0.890	44.61
TGAN-skip-WGAN-GP	73.66	0.863	45.81	77.82	0.924	49.02
CWGAN-GP-PacGAN	78.91	0.881	47.63	84.53	0.936	53.48

set. DCG considers the position and discounts the relevance score. It divides the relevance score with the log of the corresponding position. NDCG measures the ratio between recommended order DCG and the actual order DCG. Table 3, table 4 and table 5 show the performance comparison of DeepWalk, NARM, GCNN, and OpGCN model for product recommendation prediction considering both original and synthetic data generated from different models.

C. ABLATION STUDY

To demonstrate the importance of the auxiliary classier in our proposed methodology, we provide the ablation study in table 6. First we implement our proposed auxiliary classifier combined with the loss function of Vanilla GAN instead of using WGAN-GP, then we perform experiments with CWGAN-GP-PacGAN without the auxiliary classifier and we compare the results of these two experiments with our proposed model containing the auxiliary classifier. We measure the mean correlation coefficient, RMSE, FD, MAE, mirror column association, and PRD of the generated synthetic data. From the experiments, we can see that the auxiliary classifier with Vanilla GAN performs slightly better than WGAN-GP without an auxiliary classifier, but the inclusion of auxiliary classifier with the CWGAN-GP-PacGAN architecture improves the performance significantly. Results show that the inclusion of auxiliary classifier enhances the

TABLE 5. Performance comparison of GCNN and OpGCN model for product recommendation based on DCG@5, NDCG@5 and MRR@5.

Model	GCNN			OpGCN		
	DCG@5	NDCG@5	MRR@5	DCG@5	NDCG@5	MRR@5
Original Data	17.84	0.871	16.34	21.67	0.863	22.65
SMOTE	21.99	0.904	19.15	25.82	0.881	26.89
B-SMOTE	19.75	0.895	21.91	32.68	0.903	30.02
SMOTEENC	21.35	0.909	21.82	28.90	0.889	25.91
ADASYN	18.20	0.882	18.66	25.61	0.880	22.45
TGAN	26.81	0.915	25.81	33.12	0.917	34.66
CTGAN	39.00	0.962	37.90	37.91	0.934	36.72
TGAN-skip	30.64	0.936	28.72	40.04	0.945	39.11
TGAN-WGAN-GP	33.72	0.953	30.85	37.68	0.930	35.72
WCGAN-GP	37.06	0.959	39.01	45.79	0.972	47.81
TGAN-skip-WGAN-GP	44.71	0.982	46.83	50.22	0.988	49.77
CWGAN-GP-PacGAN	50.62	0.991	49.04	54.31	0.993	54.10

TABLE 6. Ablation study results for evaluating auxiliary classifier. AC denotes auxiliary classifier.

Metrics	AC with Vanilla GAN loss	Without AC	CWGAN-GP-PacGAN
Mean Correlation Coefficient	0.889	0.867	0.946
RMSE	0.62	0.63	0.49
FD	0.61	0.64	0.56
MAE	0.55	0.57	0.46
Mirror Column Association	0.872	0.843	0.941
PRD	63.9	64.2	53.1

performance and reduces the error rate in generating synthetic data remarkably.

VI. CONCLUSION

In this work, we propose a novel hybrid GAN model that combines the advantages of PacGAN into the architecture of conditional Wasserstein GAN to oversample the minority classes in online shopping tabular data to enhance the accuracy and reduce the error rate of recommendation systems. Our proposed model combines the auxiliary classifier loss and the Wasserstein loss with gradient penalty to handle categorical and numerical data. We introduce the concept of sampled inputs in discriminator as in PacGAN into the discriminator architecture of conditional Wasserstein GAN. The introduction of sampled discriminator as PacGAN and the cross-entropy loss of AC helps in eliminating the mode collapse problem and increases the convergence rate to a reasonable extent. We have used three architectures for developing the generator, discriminator, and auxiliary classifier. It differs from the AC-GAN in that in AC-GAN, the discriminator and the AC share similar parameters in the hidden layer. Our model uses different architecture for both the network that helps us condition the discriminator and use the AC loss simultaneously.

We preprocess and analyze the data to extract meaningful information in our proposed architecture, then feed it to the GAN model. We also perform product recommendations using the generated oversampled data through various prediction models such as DeepWalk, NARM, GCNN, and OpGCN. We divide our performance evaluation section into two portions. We first evaluate the quality of the generated synthetic data based on different evaluation metrics and compare its quality with synthetic data obtained from other traditional and GAN models. Secondly, we use data generated from our proposed model and other models for product recommendations. We then evaluate how multiple recommendation systems perform using our generated synthetic data.

We have also presented the visual representation of how our generated oversampling data correlates to the original data to a greater extent than other existing models. The performance evaluation section demonstrated that our proposed WCGAN-GP-PacGAN generates better quality synthetic data and considerably enhances recommendation systems' performance. Our model can be further developed and tested for various other publicly available datasets in the future. We aim to test our model on open-access datasets and modify the configuration of the GAN model accordingly.

REFERENCES

[1] J. Gantz and D. Reinsel, "Extracting value from chaos," *IDC iView*, vol. 1142, pp. 1–12, Jun. 2011.

[2] J. Gantz and D. Reinsel, "The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east," *IDC iView, IDC Analyze Future*, vol. 2007, no. 2012, pp. 1–16, 2012.

[3] W. Wei, J. Li, L. Cao, Y. Ou, and J. Chen, "Effective detection of sophisticated online banking fraud on extremely imbalanced data," *World Wide Web*, vol. 16, no. 4, pp. 449–475, 2012.

[4] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *J. Big Data*, vol. 6, no. 1, pp. 1–54, Dec. 2019.

[5] R. Cai, J. Wu, A. San, C. Wang, and H. Wang, "Category-aware collaborative sequential recommendation," in *Proc. 44th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2021, pp. 388–397.

[6] M. Chen, Y. Li, and X. Zhou, "CoNet: Co-occurrence neural networks for recommendation," *Future Gener. Comput. Syst.*, vol. 124, pp. 308–314, Nov. 2021.

[7] Z. Jiang, C. Chi, and Y. Zhan, "Let knowledge make recommendations for you," *IEEE Access*, vol. 9, pp. 118194–118204, 2021.

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 2672–2680.

[9] A. Moreo, A. Esuli, and F. Sebastiani, "Distributional random oversampling for imbalanced text classification," in *Proc. 39th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2016, pp. 805–808.

[10] A. Ghazikhani, H. S. Yazdi, and R. Monsefi, "Class imbalance handling using wrapper-based random oversampling," in *Proc. 20th Iranian Conf. Electr. Eng. (ICEE)*, May 2012, pp. 611–616.

[11] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, no. 28, pp. 321–357, Jun. 2006.

[12] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-SMOTE: A new oversampling method in imbalanced data sets learning," in *Proc. Int. Conf. Intell. Comput.* Berlin, Germany: Springer, 2005, pp. 878–887.

[13] S. Chen, G. Guo, and L. Chen, "A new over-sampling method based on cluster ensembles," in *Proc. IEEE 24th Int. Conf. Adv. Inf. Netw. Appl. Workshops*, Apr. 2010, pp. 599–604.

[14] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, "SMOTEBoost: Improving prediction of the minority class in boosting," in *Proc. Eur. Conf. Princ. Data Mining Knowl. Discovery*. Berlin, Germany: Springer, 2003, pp. 107–119.

[15] H. Ba, "Improving detection of credit card fraudulent transactions using generative adversarial networks," 2019, *arXiv:1907.03355*.

[16] J. Kim, K. Jeong, H. Choi, and K. Seo, "Gan-based anomaly detection in imbalance problems," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 128–145.

[17] Y. Guo, G. Xiong, Z. Li, J. Shi, M. Cui, and G. Gou, "Combating imbalance in network traffic classification using GAN based oversampling," in *Proc. IFIP Netw. Conf. (IFIP Networking)*, Jun. 2021, pp. 1–9.

[18] S. S. Mullick, S. Datta, and S. Das, "Generative adversarial minority oversampling," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1695–1704.

[19] A. Koivu, M. Sairanen, A. Airola, and T. Pahikkala, "Synthetic minority oversampling of vital statistics data with generative adversarial networks," *J. Amer. Med. Inform. Assoc.*, vol. 27, no. 11, pp. 1667–1674, Nov. 2020.

[20] G. Douzas and F. Bacao, "Effective data generation for imbalanced learning using conditional generative adversarial networks," *Expert Syst. Appl.*, vol. 91, pp. 464–471, Jan. 2018.

[21] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 214–223.

[22] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of Wasserstein GANs," 2017, *arXiv:1704.00028*.

[23] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*.

[24] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 2642–2651.

[25] Z. Lin, A. Khetan, G. Fanti, and S. Oh, "PacGAN: The power of two samples in generative adversarial networks," *IEEE J. Sel. Areas Inf. Theory*, vol. 1, no. 1, pp. 324–335, May 2020, doi: 10.1109/JSAIT.2020.2983071.

[26] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," 2016, *arXiv:1611.01144*.

[27] C. J. Maddison, A. Mnih, and Y. Whye Teh, "The concrete distribution: A continuous relaxation of discrete random variables," 2016, *arXiv:1611.00712*.

[28] M. Mukherjee and M. Khushi, "SMOTE-ENC: A novel SMOTE-based method to generate synthetic data for nominal and continuous features," *Appl. Syst. Innov.*, vol. 4, no. 1, p. 18, Mar. 2021.

[29] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IEEE World Congr. Comput. Intell.)*, Jun. 2008, pp. 1322–1328.

[30] L. Xu and K. Veeramachaneni, "Synthesizing tabular data using generative adversarial networks," 2018, *arXiv:1811.11264*.

[31] L. Xu, "Synthesizing tabular data using conditional GAN," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, 2020.

[32] B. Brenninkmeijer, A. de Vries, E. Marchiori, and Y. Hille, "On the generation and evaluation of tabular data using GANs," Ph.D. dissertation, Dept. Data Sci., Radboud Univ., Nijmegen, The Netherlands, 2019.

[33] X. Wei, B. Gong, Z. Liu, W. Lu, and L. Wang, "Improving the improved training of Wasserstein GANs: A consistency term and its dual effect," 2018, *arXiv:1803.01541*.

[34] M. Zheng, T. Li, R. Zhu, Y. Tang, M. Tang, L. Lin, and Z. Ma, "Conditional Wasserstein generative adversarial network-gradient penalty-based approach to alleviating imbalanced data classification," *Inf. Sci.*, vol. 512, pp. 1009–1023, Feb. 2020.

[35] F. Zhu, F. Ye, Y. Fu, Q. Liu, and B. Shen, "Electrocardiogram generation with a bidirectional LSTM-CNN generative adversarial network," *Sci. Rep.*, vol. 9, no. 1, pp. 1–11, Dec. 2019.

WAFA SHAFQAT received the B.S. degree in computer sciences from FAST-NUCES, Islamabad, the M.S. degree from Sangmyung University, Cheonan Campus, under the supervision of Prof. Hyun-Chul Kim, and the Ph.D. degree from the Machine Learning Laboratory (ML Lab), Jeju National University (JNU), under the supervision of Prof. Yung-Cheol Byun. She successfully defended her Ph.D. thesis, in December 2019. She joined JNU as a Faculty Member and a Researcher. She was a member of the Network Data Science Laboratory, Sangmyung University. Her research interests include machine learning applications, recommendation systems, optimization, and natural language processing.

YUNG-CHEOL BYUN received the B.S. degree from Jeju National University, in 1993 and the M.S. and Ph.D. degrees from Yonsei University, in 1995 and 2001, respectively. He worked as a Special Lecturer at Samsung Electronics and SDS, from 1998 to 2001. From 2001 to 2003, he was a Senior Researcher at the Electronics and Telecommunications Research Institute (ETRI). In 2003, he was promoted to Assistant Professor with Jeju National University, where he is a Full Professor with the Department of Computer Engineering. His research interests include AI and machine learning, pattern recognition, blockchain and deep learning-based applications, big data and knowledge discovery, time-series data analysis and prediction, image processing, medical image applications, and recommendation systems.