POLİTEKNİK DERGİSİ

JOURNAL of POLYTECHNIC

ISSN: 1302-0900 (PRINT), ISSN: 2147-9429 (ONLINE)

URL: http://dergipark.org.tr/politeknik

Comparative analysis of ad click behavior prediction using GAN-augmented data and traditional machine learning techniques

GAN-artırılmış veri ve geleneksel makine öğrenimi teknikleri kullanılarak reklam tıklama davranışı tahmininin karşılaştırmalı analizi

***Yazar(lar) (Author(s))**: Amel Sulaiman Mandan[1], Oktay YILDIZ[2]*

ORCID[1]: 0000-0002-9850-2118

ORCID[2]: 0000-0001-9155-7426

To cite to this article: Mandan A. S. and Yıldız O., “Comparative Analysis of Ad Click Behavior Prediction Using GAN-Augmented Data and Traditional Machine Learning Techniques”, ***Journal of Polytechnic***, 28(2): 683-695, (2025).

Bu makaleye şu şekilde atıfta bulunabilirsiniz: Mandan A. S. Ve Yıldız O., “Comparative Analysis of Ad Click Behavior Prediction Using GAN-Augmented Data and Traditional Machine Learning Techniques”, ***Politeknik Dergisi***, 28(2): 683-695, (2025).

Erişim linki (To link to this article): http://dergipark.org.tr/politeknik/archive

DOI: 10.2339/politeknik.1525138

Comparative Analysis of Ad Click Behavior Prediction Using GAN-Augmented Data and Traditional Machine Learning Techniques

Highlights

- *User demographic and online activity data from Kaggle were used for click-through rate prediction.*
- *Generative Adversarial Networks were employed for data augmentation to improve model performance.*
- *Six machine learning models, including KNN, Random Forest, and Gradient Boosting, were tested with and without GAN-based data augmentation.*
- *GAN-based augmentation significantly improved accuracy and sensitivity, with a notable 20% performance boost in the KNN model, demonstrating the effectiveness of GANs in enhancing predictive accuracy for click-through rate in e-commerce applications.*

Graphical Abstract

In this study, user demographic and online activity data from Kaggle were used to predict click-through rates using GAN-based data augmentation techniques. Six machine learning algorithms were tested with and without data augmentation. The impact of GAN on model performance, including improvements in sensitivity and accuracy, was investigated experimentally.

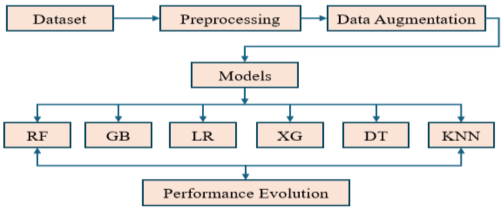


Figure. Steps of research process.

Aim

This study aims to compare the effectiveness of Generative Adversarial Networks based data augmentation methods with traditional machine learning techniques in predicting ad click behavior.

Design & Methodology

The study utilized user demographic and online activity data obtained from Kaggle, with data augmentation performed using GAN. Six different machine learning algorithms were compared, both with and without data augmentation.

Originality

This research explores how Generative Adversarial Networks based data augmentation techniques can improve performance in predicting click-through rates. There are limited studies in the literature examining the effectiveness of such augmentation methods.

Findings

GAN-based data augmentation improved the sensitivity and specificity of all models used, with a 20% improvement specifically observed in the KNN model. Data augmentation with GANs provided a notable performance boost across all models.

Conclusion

The GAN-based data augmentation method enhanced the performance of machine learning models, resulting in higher accuracy rates. This approach offers an effective solution for predicting click-through rates and highlights the importance of data augmentation techniques in future research.

Declaration of Ethical Standards

The author(s) of this article declare that the materials and methods used in this study do not require ethical committee permission and/or legal-special permission.

Comparative Analysis of Ad Click Behavior Prediction Using GAN-Augmented Data and Traditional Machine Learning Techniques

Araştırma Makalesi / Research Article

Amel Sulaiman MANDAN[1*], Oktay YILDIZ[2]

[1]Computer Science Department, Informatics Institute, Gazi University, Ankara, Turkey

[1]Computer Science Department, Kirkuk University, Kirkuk, Iraq

[2]Department of Computer Engineering, Faculty of Engineering, Gazi University, Ankara, Turkey

(Geliş/Received : 30.07.2024 ; Kabul/Accepted : 27.08.2024 ; Erken Görünüm/Early View : 11.10.2024)

ABSTRACT

In e-commerce, predicting click-through rates (CTR) is crucial to anticipating user behavior. User historical data can be used to extract interests and enhance CTR prediction, leading to higher accuracy. In this study, a generative adversarial network (GAN) has been used to tackle the issue of an insufficient dataset for click-through rates. Furthermore, six different machine learning algorithms have been assessed for predicting ad click behavior. For the experimental study, we obtained user demographic and online activity data from Kaggle, along with a binary label indicating ad clicks. To enhance the model's performance, we employed a GAN for data augmentation and generated additional training examples. We compared the machine-learning algorithm's outcomes with and without GAN-based data augmentation to evaluate its predicted accuracy. According to the findings, most algorithms have increased sensitivity and specificity after utilizing GAN to augment the data, indicating that the generated data has improved their ability to accurately distinguish positive and negative events. GAN-based data augmentation boosted all models to varying degrees, according to the findings.

Keywords: Click-Through Rate (CTR), Generative Adversarial Network (GAN), Data Augmentation, Machine Learning.

GAN-Artırılmış Veri ve Geleneksel Makine Öğrenimi Teknikleri Kullanılarak Reklam Tıklama Davranışı Tahmininin Karşılaştırmalı Analizi

ÖZ

E-ticarette, kullanıcı davranışını öngörmek için tıklama oranlarının (TO) tahmin edilmesi önemlidir. Yüksek doğruluklu ilgi alanlarının çıkarılması ve TO tahmini için kullanıcıların geçmiş verileri kullanılabilir. Bu çalışmada, yetersiz ya da dengesiz veri kümelerinde reklam tıklama davranışının tahmini için Üretken Çekişmeli Ağlar (ÜÇA) kullanılmıştır. Çalışmada altı farklı makine öğrenmesi algoritmasının reklam tıklama davranışını tahmin etmedeki etkinliği değerlendirilmiştir. Gerçekleştirilen deneysel çalışmada, Kaggle'dan elde edilen kullanıcı demografik ve çevrimiçi aktivite verileri ve reklam tıklama etiketini gösteren bir veri kümesi kullanılmıştır. Modelin performansını artırmak amacıyla veri artırma yapılmış, bunun için ÜÇA kullanılmıştır. Tahmin doğruluğunu değerlendirmek için makine öğrenimi algoritmalarının ÜÇA temelli veri artırma ve veri artırma olmaksızın elde edilen sonuçlar karşılaştırılmıştır. Elde edilen sonuçlarda, hassasiyet ve özgüllük değerlerinin artığı, oluşturulan verilerin modellerin olumlu ve olumsuz olayları doğru bir şekilde ayırt etme yeteneklerini geliştirdiği gösterilmiştir. Bulgulara göre GAN tabanlı veri artırma, tüm modelleri farklı derecede güçlendirmiştir.

Anahtar Kelimeler: Tıklama Oranı (TO), Üretken Çekişmeli Ağlar (ÜÇA), Veri Arttırma, Makine Öğrenimi.

1. INTRODUCTION

Online advertising has changed significantly with innovations like search advertising, social media platforms, and mobile technology. These have enabled businesses to target engaged audiences and develop various ad formats, resulting in greater efficiency and personalization [1]. Display advertising uses banner ads containing text, photos, videos, and motion to target specific spots on a website or app. Its aim is to draw in new users and promote industry services. Global spending on display advertising reached $164.6 billion in 2022 (Guttmann, 2022). To assess the value of digital advertising, an accurate estimation of the click-through rate (CTR) is essential. CTR is the ratio of clicks on an ad to the total number of times it was shown online, and it measures the success of digital marketing. A high CTR shows higher audience engagement. Data augmentation enhances prediction models, which is essential for correct estimation. Transformation based data augmentation increases training occurrences, balances class distributions, minimizes noise, and explores feature space to reduce overfitting. These data augmentation advances improve model performance and prediction efficiency, enhancing machine learning algorithms' accuracy and effectiveness [2]. Estimating ad engagement is important for academics and businesses.

*Sorumlu Yazar (Corresponding Author)

e-posta: asulaiman.salihi@gazi.edu.tr

Public datasets and studies on "CTR prediction" have been conducted to improve click prediction accuracy. Researchers use various techniques, including:

In a study [3], users' interests were found to be dynamic and influenced by their engagement order. Recommender systems typically rely on historical user-item interactions to predict preferences, but the "Comprehensive Present-Interest Network (CIPN)" model was introduced as a solution to this problem. The CIPN has two parts: one for current interest and one for comprehensive interest in the interaction sequence. A new MLP was also created to improve model training. Public and industrial datasets were used in the experiments, showing that the CPIN with both comprehensive and current interest performed better than either interest alone. The authors [4] introduced the Recurrent Interaction Network (RIN), which enhances the structure of recurrent neural networks (RNNs) using matrix multiplication techniques to describe explicit interactions. They also employed a convolutional neural network (CNN) to find nonlinear links between features, allowing for the learning of various feature interaction orders. The RIN was integrated with a conventional DNN in DRIN to learn feature interactions overtly and implicitly, leading to successful experimental results. The suggested RIN outperforms other models and is reliable for feature interaction based on matrix multiplication. The research [5] utilized both the attention mechanism and the Gated Recurrent Unit (GRU) to enhance the prediction of click-through rates. To identify concealed relationships among non-temporal features, they introduced a stacked autoencoder in the feature interaction module. The study [6] presented an algorithm that uses a combination of CNN and LSTM neural networks for click-through rate prediction in advertising. By using CNN for feature extraction and LSTM for time series analysis, the proposed model outperforms single structure networks in terms of prediction accuracy. [7] Used historical data to improve click-through rate prediction. The authors suggested the ICE approach to increase attention via dynamic interactions. To accommodate user interest levels and candidate concentration, a unique adaptive interest attention unit was created. The ICE-DEN model predicts click-through rates. An embedding layer captures low-dimensional user features, while mini-batch perception regularization and the Dice activation function train deep learning networks with many variables. The Amazon, MovieLens, and Taobao datasets yielded 90.89%, 84.49%, and 92.88% accuracy for the ICE-DEN model. Various techniques have been proposed to predict CTR with good results, with the goal of minimizing logloss, or RMSE, across all training samples. However, these techniques often overlook regional details in favor of collecting global data on user click activity. The article [8] proposes retrieval-based factorization machines (RFM) as a method for predicting CTR. RFM integrates global knowledge obtained from factorization machines (FM) with local data based on neighboring samples. The authors also use clustering to optimize neighbor retrieval in smaller sections of the training set. Experiments on the Frappe, MovieLens, and Criteo datasets show that the RFM model outperforms other models in terms of metrics such as RMSE, ROC AUC, and accuracy. A new CTR prediction framework called MSMC was proposed by [9], which uses salient and diverse semantic feature encoders to include feature relevance and semantic information. MSMC applies attentive modules to encode features and predict CTR with higher-order interactions, outperforming current state-of-the-art methods on two public datasets. In [10], a model combining logistic regression (LR) with stochastic gradient ascent (SGA) was proposed to predict clicks in sponsored searches. The article also compared the time efficiency of SGA and BGA methods in creating a classifier model and evaluated their accuracy. The authors proposed LSTMcp and LSTMip models in [11] to predict user clicks and interests, respectively. The models utilize deep LSTM networks to learn latent features from users' temporal sequence of page visits, considering temporal information for better predictions. In [12], a highly accurate CTR prediction model called the Dual-View Attention Network (DVAN) was introduced. DVAN utilizes both user and item views from advertisement history logs and uses a universal pairwise channel unit to establish domain relationships. It adapts its representation from coarse to fine. [13] Proposed the Dual-View Attention Network (DVAN) to predict CTR by considering user and item correlations. DVAN uses coarse and fine attention modules to identify relevant user-item interactions and create global and local data for prediction. The model outperformed existing approaches on four datasets, according to AUC and log loss metrics. [14] proposed the Deep Field Relation Neural Network (DFRNN) model for CTR prediction, which uses deep neural networks to simulate feature interactions and models 2-order feature interactions. DFRNN outperformed classic FM models and recent deep models like PNN and DeepFM in terms of AUC and log loss on real datasets. Researchers in [15] proposed a new method, multi-view feature transfer (MFT), employing transfer learning to estimate click-through rates by extracting pertinent information from non-relevant ads. MFT categorizes data and generates view clusters by merging feature vectors with common features. Six classifiers were evaluated on five datasets, with MFT performing the best and GAN having the highest AUC value. The authors proposed a joint learning model that combines residual networks to probe feature interactions and a neural attention network to understand second-order feature interactions. The model outperforms conventional neural networks on the Criteo and Avazu datasets, as shown by LogLoss and AUC metrics and previous state-of-the-art studies [16]. [17] Introduced a novel approach based on capsules to predict CTR and CVR by capturing the diverse interests of users. The model uses a modified dynamic routing technique, attention mechanism, and weighted loss function to emphasize distinctions between capsules. The model's explainability was demonstrated by a correlation matrix. The ACN method outperformed prior state-of-the-art techniques on public and commercial datasets. [18] proposed a causality-based CTR prediction model called Causal-GNN that combines feature, user, and ad graph representations within the GNN framework. The model captures high-order feature graph representations using GraphFwFM and obtains user and ad graph representations using GraphSAGE. Causal-GNN achieves superior AUC and logloss values compared to other methods on three public datasets, and GraphFwFM captures high-order representations effectively on the causal feature graph. In [19] a model named HoAFM was introduced to explore high-order feature interactions explicitly and rapidly by refining feature representations and employing a bit-wise sparse attention mechanism. The authors compared their model with recent deep learning-based models, including NFM, PIN, and DeepFM, and demonstrated that HoAFM achieved better performance on the Criteo and Avazu datasets. HoAFM's lightweight settings help alleviate overfitting compared to xDeepFM, which lacks confirmation of the efficacy of high-order patterns. [20] Proposed RILKE, a novel approach that uses locally kernel embedding to address sparsity, and RTILKE, an enhanced version that incorporates unsupervised transfer learning to tackle the issue of imbalance in advertising data. The research evaluated seven methodologies and five datasets and found that RTILKE outperformed other algorithms, including RILKE, in predicting CTR in online advertising, resulting in improved advertising response prediction. [21] presented the AutoFT framework to improve CTR prediction accuracy for a new target domain by automatically integrating parameters from a pre-existing model. The Gumbel-Softmax technique is used to co-train lightweight policy networks with the target domain. The approach can be applied to various deep CTR models and has been demonstrated to outperform other methods in extensive offline experiments. According to [22], a two-layer neural network has been proposed for CTR prediction that is more accurate and scalable than any individual CTR model. This model is well-suited for use in real-time recommender systems and can be created through a model distillation framework. The authors suggest that any CTR model can be added to the ensemble using this methodology and can be distilled into any neural architecture.Data augmentation in the ad click prediction field was explored in this study to produce synthetic samples that may be added to training data to boost the prediction model's performance while working with limited data. The fundamental goal is to take advantage of the resources that are now available to automatically create new data sets and to provide possible solutions for a variety of issues that are associated with machine learning. The paper was organized as follows: Section 2 offered an in-depth explanation of the theoretical background and prediction method. Section 3 described the experimental study, while Section 4 presented and analyzed the experimental results. Section 5 was focused on discussing the findings and presenting the conclusion.

2. THEORETICAL BACKGROUND

This section outlines the main aspects of the study, which are critical for predicting ad click behavior based on user demographic and online activity data. These components include preprocessing, data augmentation, machine learning algorithms, and evaluation metrics.

2.1. Preprocessing

In data analysis and machine learning, data preprocessing is a pivotal step that transforms raw data into a usable format that is organized, free of errors, inconsistencies, and missing values. To achieve accurate and meaningful results, one must follow several crucial steps in the data preprocessing process. Data cleansing is the first step in data preprocessing. The first step of data preprocessing involves identifying and correcting errors and inconsistencies to ensure accuracy, followed by transforming the data into a suitable format for analysis or machine learning purposes. Scaling, normalization, and feature selection are common techniques in data transformation, ensuring the data's easy interpretation and potential for insights. In this study, we employed manual feature selection to eliminate irrelevant features, and we normalized numerical features within a range of -1 to 1 to prepare the data for analysis. Standardizing the data was critical to ensure accurate and reliable results due to variations in the value range of each feature, which could skew results if not standardized before further analysis.

2.2. Data Augmentation

Data augmentation is a process commonly used in machine learning and computer vision to boost the size of a training dataset by producing additional variations of the original data The theory behind data augmentation is that by expanding the quantity and variety of the data sets, the model will be better able to adapt to new, unknown data, consequently improving its accuracy. When working with restricted datasets, data augmentation is particularly useful since it helps minimize overfitting by providing the model with more different cases to learn from. This helps the model perform better overall. Applications in computer vision, such as image categorization, object recognition, and segmentation, often make use of it [23].

The Generative Adversarial Networks (GANs) is a machine-learning technique that may produce new, realistic data from a training sample as shown in figure 1.

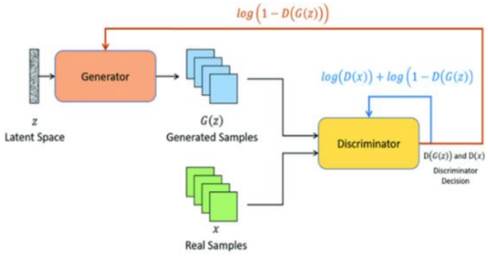


Figure 1. Generative Adversarial Networks

GANs need two distinct networks to function: a generator network and a discriminator network. The generator network takes in random noise like an input and attempts to reproduce the training set's actual data as output. On the other hand, the discriminator network attempts to identify the difference between actual and faked data when supplied both types of inputs. Two networks experience what is known as adversarial training together. We train the discriminator network to accurately identify authentic data as authentic and fake data as fake. The discriminator network trains to recognize genuine data, while the generator network produces fake data to trick the discriminator into believing it is real. The generator network improves its ability to create realistic data as the two networks continue to learn from each other, while the discriminator network becomes better at distinguishing between genuine and fabricated data. Ultimately, the generator network can generate fresh, high-quality data that closely resembles the actual data from the training set. The GAN training objective is defined as follows:

$$V_{GAN}\ (D, G) = E_{x\sim pdata\ (x)}\ [\log\ (D\ (x))] + E_{z\sim pz\ (z)}\ [\log\ (1 - D\ (G\ (z)))] \quad (1)$$

The equation (1) in the GAN model defines two loss functions: -log(D(x) for the discriminator network, and log(1-D(G(z))) for the generator network. Because these are two distinct networks, separate optimizers for G and D are required. The discriminator's objective is to maximize the cost function log(D(x)), while the generator aims to minimize the cost function log(1-D(G(z))).

Several applications, such as image generation, text generation, and video fabrication, have effectively used GANs. They serve as an efficient method for generating new data and can enhance a diverse range of machine learning applications [24].

Machine learning Algorithm

2.2.1. Random forest

The random forest algorithm is a supervised learning method that can perform both classification and regression tasks by combining many individual trees. The algorithm generates a forest where each tree predicts a class based on features, and the final prediction is made by selecting the class with the most votes across all the trees. Studies such as [25] indicate that augmenting the number of trees in the forest can enhance the accuracy of the random forest classifier. This algorithm uses a mathematical formula that involves combining multiple decision trees to form an ensemble model. The formula changes with the number of trees in the forest. The following formula mathematically represents the random forest classifier:

$$n_{ij} = W_I C_j - W_{left(j)} C_{left(j)} - W_{right(j)} C_{right(j)} \quad (2)$$

Where***:***

ni sub(j) denotes the significance of node j

w sub(j) refers to the weighted number of samples that arrive at node j

C sub(j) represents the impurity value of node j

Left(j) signifies the left child node obtained after the split at node j

Right(j) represents the right child node obtained after the split at node j.

2.2.2. Gradient boosting

Gradient Boosting is an ensemble learning approach used to solve regression and classification problems in machine learning. It adds new models to the ensemble and trains them to correct previous models' errors using gradient descent optimization. The base models are decision trees, each shallow with few nodes and leaves. The algorithm's key parameters are tree quantity, learning rate, maximum depth, and minimum data for node splitting. Gradient Boosting is popular due to its ability to handle various data types and provide high predictive accuracy, but it may overfit and is expensive without proper tuning [26]. The mathematical formula for Gradient Boosting is two parts. In Gradient Boosting, there are two functions to consider: the objective function and the prediction function. The objective function, which consists of a loss function and a regularization term, optimizes the model parameters during training.

$$obj = \Sigma\ L\ (y, \hat{y}) + \Omega(f) \quad (3)$$

Where obj is the objective function, y is the true label, ŷ is the predicted label, L is the loss function, f is the decision tree, and Ω is the regularization term. The prediction function in Gradient Boosting combines the predictions of multiple decision trees to make the final prediction. It can be written as:

$$\hat{y} = \Sigma\ f(x) \quad (4)$$

Where ŷ is the predicted label, f is the individual decision tree, and x is the input data [26].

2.2.3. Logistic regression

In binary classification tasks, the machine learning algorithm known as logistic regression models the probability of an input belonging to one of two classes based on a set of input features. It can accept independent variables of any type and use coefficients to improve training data observations. The logistic function is used to transform linear regression results into probabilities between 0 and 1. Key factors for optimizing the model include the solver approach and regularization parameter. Logistic Regression's simplicity, interpretability, and ability to handle large datasets make it widely used in various domains, including finance, marketing, healthcare, and social sciences [27]. Logistic regression can be expressed mathematically as follows:

$$p(z) = \frac{1}{1+e^{-z}} \quad (5)$$

The equation uses a linear combination of input features denoted by 'z', with 'p' indicating positive class probability and 'e' representing the mathematical constant 2.71828. The z is a linear combination of the input features:

$$z = b0 + b1x1 + b2x2 + ... + bnxn \quad (6)$$

Where b0 is the intercept, b1 to bn are the coefficients of the input features x1 to xn, respectively [28].

2.2.4. XGBoost

XGBoost is a machine learning library that implements the gradient boosting technique. It was created by Tianqi Chen in 2014 and has since gained popularity as a highly effective algorithm for working with structured data [29]. XGBoost is a gradient boosting algorithm that employs decision trees and regularization to avoid overfitting. It can handle large datasets with tens of thousands of features and is suitable for both regression and classification problems. XGBoost can automatically prune decision trees and supports custom loss functions and evaluation metrics. Its speed, accuracy, and flexibility have made it popular in various domains, including recommendation systems, fraud detection, image classification, and NLP [26]. The mathematical formula for XGBoost can be broken down into two parts. The first part is the Objective Function: in XGBoost is designed to optimize the model parameters during training. It is a sum of two terms: a loss function and a regularization term.

$$Obj = L(y, \hat{y}) + \Omega(\hat{W}) \quad (7)$$

The XGBoost algorithm includes an objective function (obj) that uses the true label (y), predicted label (ŷ), loss function (L), set of model parameters ($\hat{W}$), and regularization term (Ω). The prediction function combines multiple decision trees to make the final prediction. It can be written as:

$$\hat{y} = \sum f(x, T) \quad (8)$$

Where ŷ is the predicted label, f is the individual decision tree, x is the input data, and T is the tree's set of splitting rules [30].

2.2.5. Decision tree

Decision trees are a kind of classification approach that may be used with both categorical and numerical data. A decision tree is a type of construction that resembles a tree. When working with medical datasets, a decision tree is a fundamental and widely used technique to help make decisions. A tree graphical display of alternative answers to a choice depending on specific conditions is easy to construct and analyze since the data is organized into a tree structure. As the name implies, a decision tree starts with a single node or root and then branches out into several responses, just like a tree [31].

ID3 algorithm computes entropy of class and attributes, calculates information gain for each attribute using equations 9, 10, and 11, and selects attribute with highest information gain as root node, considering it to be the most informative attribute. This process is repeated until all attributes are incorporated into the tree [32].

$$\mathrm{Info} = -\sum_{i=1}^{m} \mathrm{Pi} * \log_2(\mathrm{Pi}) \quad (9)$$

$$\mathrm{Infon}_A\,(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} * \mathrm{info}\,(D_j) \quad (10)$$

$$\mathrm{Gain}(A) = \mathrm{info}\,(D) - \mathrm{info}_A(D) \quad (11)$$

2.2.6. K-Nearest neighbors

KNN is a supervised machine learning approach that used for classification and regression. It is one of the most simple and quickest classification methods available. The closer two samples are to each other, the higher the probability of their connection, as similar items tend to group together. The k parameter indicates how many neighbors there are for a certain data point. The next step involves calculating distance functions to determine the distance between the new data point and the samples in the data set. Based on its distance values, the new data point assign to the class of k neighbors. Then it will label accordingly [33].

As stated by [34] the distance metric used in KNN can vary, but commonly used distance measures are Euclidean distance and Manhattan distance. Suppose we have a test data point xi with features x1, x2, ..., xn and a training data point xj with features y1, y2, ..., yn. The Euclidean distance between these two points is calculated as:

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(y_i - x_i)^2} \quad (12)$$

And the Manhattan distance between these two points is calculated as:

$$d = \sum_{i=1}^{n} |x_i - y_i| \quad (13)$$

2.2.7. Evolution Metrics

The present study employed a range of machine learning algorithms, namely Random Forest, Gradient Boosting, Logistic Regression, XGBoost, Decision Trees, and K-Nearest Neighbor, to address a classification problem. The objective was to forecast the outcome of a target variable utilizing input features. We used performance metrics as showed in Table 1 to evaluate the algorithms' classification performance.

Table 1. Performance Metrics

Metric	Formula
Precision	Tp / (Tp + Fp)
Sensitivity (Recall)	Tp / (Tp + Fn)
Specificity	Tn / (Tn + Fp)
F1-Score	2 * (Precision * Sensitivity) / (Precision + Sensitivity)
Accuracy	(Tp + Tn) / (Tp + Tn + Fp + Fn)

Where:

- TP: True Positive (correctly predicted positive instances).
- FP: False Positive (incorrectly predicted positive instances).
- TN: True Negative (correctly predicted negative instances).
- FN: False Negative (incorrectly predicted negative instances).

We evaluated and compared the classification performance of each algorithm using these performance metrics [27]. The goal of this study was to evaluate the effectiveness of using a GAN for data augmentation and a machine learning algorithm to predict ad click behavior based on user demographics and online activity data. This study aims to improve ad campaign performance and increase user engagement with advertisements for advertisers and marketers.

3. EXPERIMENTAL STUDY

This study aimed to enhance the performance of a machine learning algorithms for predicting user clicks in online advertising by utilizing Generative Adversarial Networks (GANs) to augment the dataset. The GAN-generated samples represented the original dataset's distribution and introduced novel data points to identify previously unseen patterns. Following the acquisition of the dataset, the initial phase involves preprocessing, which involves cleaning, normalizing, and transforming the data to ensure its suitability for analysis. After that, data augmentation is carried out, involving enhancing the quality of the dataset by including of altered replicas of the data or the generation of novel synthetic data based on the existing data. Next, we employ various machine learning models such as Random Forest (RF), Gradient Boosting (GB), Logistic Regression (LR), XGBoost (XG), Decision Tree (DT), and K-Nearest Neighbors (KNN). The last stage, performance review, examines the efficacy and accuracy of the models using a range of measurements. Our results revealed a significant improvement in accuracy when comparing the model's performance with and without data augmentation. The findings suggest that data augmentation based on GAN is an effective technique for enhancing the accuracy of machine learning models in online advertising click prediction. Future research could further explore the use of GANs in other domains to improve machine learning algorithm performance.

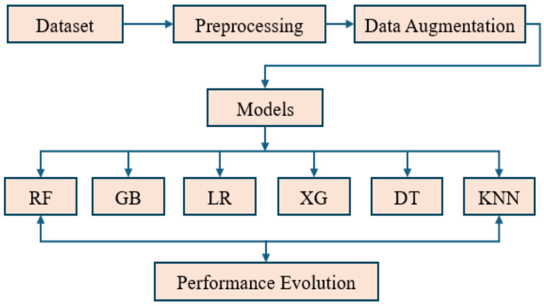


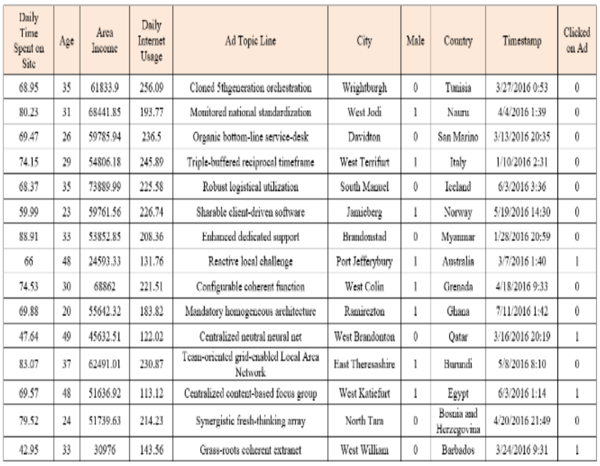
Figure 2. Steps of research process.

3.1. Dataset

To conduct this research, we used a Kaggle dataset that is available to the public. There is a total of 1000 instances gathered from the real world, each having 8 distinct attributes as follow:

regarding the consumer behavior data, the average, maximum, and minimum daily time spent on a particular website, known as "Daily Time Spent on Site", were found to be 65, 91.43, and 32.60 minutes. The corresponding age distribution of the customers was characterized by an average age of 36 years, a maximum age of 61 years, and a minimum age of 19 years. Additionally, the income level of the geographical area in which the customers reside, referred to as "Area Income", was observed to have an average value of 55000, a maximum value of 79484, and a minimum value of 13996. Lastly, the average time, maximum time, and minimum time that consumers spent on the internet daily, known as "Daily Internet Usage", were recorded to be 180, 269, and 104 minutes respectively. The dataset also included information on the city, gender, and country of the consumers as illustrated in Table 2.

Table 2. Dataset Sample.



Daily Time Spent on Site	Age	Area Income	Daily Internet Usage	Ad Topic Line	City	Male	Country	Timestamp	Clicked on Ad
68.95	35	61833.9	256.09	Cloned 5thgeneration orchestration	Wrightburgh	0	Tunisia	3/27/2016 0:53	0
80.23	31	68441.85	193.77	Monitored national standardization	West Jodi	1	Nauru	4/4/2016 1:39	0
69.47	26	59785.94	236.5	Organic bottom-line service-desk	Davidton	0	San Marino	3/13/2016 20:35	0
74.15	29	54806.18	245.89	Triple-buffered reciprocal timeframe	West Terrifurt	1	Italy	1/10/2016 2:31	0
68.37	35	73889.99	225.58	Robust logistical utilization	South Manuel	0	Iceland	6/3/2016 3:36	0
59.99	23	59761.56	226.74	Sharable client-driven software	Jamieberg	1	Norway	5/19/2016 14:30	0
88.91	33	53852.85	208.36	Enhanced dedicated support	Brandonstad	0	Myanmar	1/28/2016 20:59	0
66	48	24593.33	131.76	Reactive local challenge	Port Jefferybury	1	Australia	3/7/2016 1:40	1
74.53	30	68862	221.51	Configurable coherent function	West Colin	1	Grenada	4/18/2016 9:33	0
69.88	20	55642.32	183.82	Mandatory homogeneous architecture	Ramirezton	1	Ghana	7/11/2016 1:42	0
47.64	49	45632.51	122.02	Centralized neutral neural net	West Brandonton	0	Qatar	3/16/2016 20:19	1
83.07	37	62491.01	230.87	Team-oriented grid-enabled Local Area Network	East Theresashire	1	Burundi	5/8/2016 8:10	0
69.57	48	51636.92	113.12	Centralized content-based focus group	West Katiefurt	1	Egypt	6/3/2016 1:14	1
79.52	24	51739.63	214.23	Synergistic fresh-thinking array	North Tara	0	Bosnia and Herzegovina	4/20/2016 21:49	0
42.95	33	30976	143.56	Grass-roots coherent extranet	West William	0	Barbados	3/24/2016 9:31	1

4. EXPERIMENTATION and RESULT ANALYSIS

The following section provides an analysis of the outcomes obtained from six machine learning models implemented for user behavior prediction. We split the data set into training (%70) and testing (%30) sets for this analysis. In our study, we exclusively utilized the numerical features of the dataset to ensure the robustness and precision of our analytical models. The primary objective of this investigation is to assess if data augmentation can enhance the model's performance. The research involves the use of various supervised machine learning techniques, including Random Forest, Gradient Boosting, Logistic Regression, XGBoost, Decision Tree, and K-Nearest Neighbors. The findings are presented below.

4.1. User Behavior Prediction without Data Augmentation

4.1.1. Random forest

The confusion matrix for Random Forest (RF) model in Fig 3 showed that it accurately identified 292 out of the total samples, with 141 true positives and 151 true negatives. However, the model did have 3 false negatives and 5 false positives. The RF model demonstrated an overall accuracy of 97%, with a sensitivity of 0.98 and a specificity of 0.96. These results indicate that the model correctly identified 98% of non-click cases and 96% of click cases, respectively. Moreover, the precision metric revealed that 98% of all instances predicted by the model related to the correct class. Importantly, the F1-score of 0.97 for both classes showed that the RF algorithm achieved a well-balanced precision and recall. We used default parameters for RF algorithm. Table 3 presents these results.

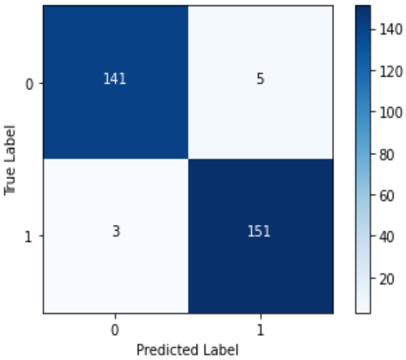


Figure 3. RF Confusion Matrix

4.1.2. Gradient boosting

The GB model correctly recognized 138 click instances and 152 non-click instances out of 300 samples, indicating proper user behaviour prediction as illustrated in Fig 4. It incorrectly classified 2 click instances as non-clicks and 8 non-click instances as clicks. The classification report shows precision of 0.99 for non-clicks and 0.95 for clicks, with high recall for both classes 0.95 for non-clicks and 0.99 for clicks. The F1-score for both classes is 0.97, indicating a satisfactory balance between precision and recall. The GB model also has a specificity of 0.94 and a sensitivity of 0.98, showing its accuracy in recognizing non-click cases and click instances, as showed in Table 3.

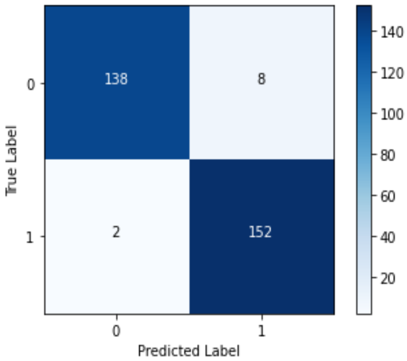


Figure 4. GB Confusion Matrix

4.1.3. Logistic regression

The LR model accurately predicted 290 out of 300 instances with 6 false negatives and 4 false positives, as shown in the confusion matrix Fig 5. It achieved a sensitivity of 0.96 and specificity of 0.97, indicating its ability to identify click and non-click instances. The precision for non-clicks is 0.96 and for clicks is 0.97, while the recall for both is 0.96. The F1 score for both classes is 0.97, indicating a balance between precision and recall. Overall, the LR model efficiently classified cases into appropriate classes as shown in Table 3.

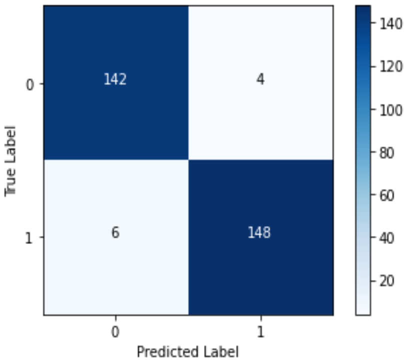


Figure 5. LR Confusion Matrix

4.1.4. XGBoost

The XGB model accurately classified 288 cases out of the complete dataset, with 139 TP and 149 TN predictions, but it also had 7 FP and 5 FN predictions. This is shown in the confusion matrix in Fig 6. As demonstrated in Table 3 the model achieved a sensitivity of 0.96 and a specificity of 0.95, indicating its ability to recognize click and non-click cases. The model's accuracy was 96%, and it had a precision of 0. 96 for click samples. The recall for non-clicks was 0.95 and for clicks was 0.97, indicating that the model correctly classified 95% of non-click cases and 97% of actual click cases as clicks. The F1-score for both non-click and click labels was 0.96, demonstrating a balance between precision and recall.

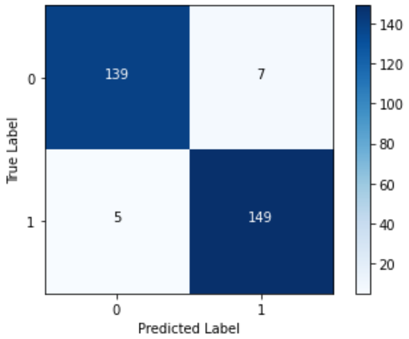


Figure 6. XGB Confusion

4.1.5. Decision tree

The DT model accurately identified 284 out of the total samples, with 135 TP and 149 TN predictions, but also had 11 FP and 5 FN predictions according to the confusion matrix in Fig 7. The model's sensitivity was 0.96, indicating its ability to recognize 96% of click cases, and its specificity was 0.92, indicating its ability to recognize 92% of non-click cases. The models For non-click instances, precision was 0.96, while for click instances, it was 0.93. It had a recall of 0.92 for non-click and 0.97 for click classes, correctly identifying 92% of non-click and 97% of click

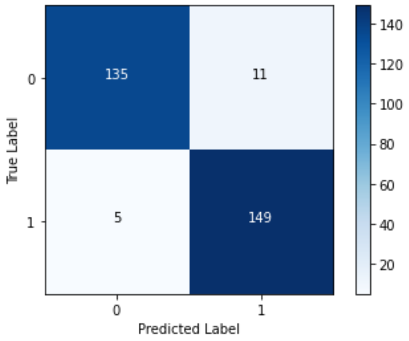


Figure 7. DT Confusion Matrix

instances. The F1-score was 94% for non-click and 95% for click classes. These results were obtained by using default parameters for DT as demonstrated in Table 3.

4.1.6. KNN

Based on the results in confusion matrix Figure 8 and Table 3. The model correctly classified 222 out of the total samples. The model correctly identified 113 out of 146 actual negative instances, demonstrating a specificity of 77%. The sensitivity for actual positive cases was 70%, correctly identifying 109 out of 154. The KNN classifier achieved an accuracy of 74%, with a precision of 0.72 for class 0 and 0.77 for class 1. The recall for class 0 was 0.77, and for class 1 was 0.71. Both classes had an F1 score of 74%. These results were obtained by adjusting the k value to 3.

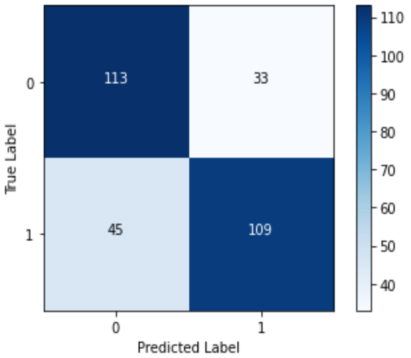


Figure. 8. KNN Confusion Matrix

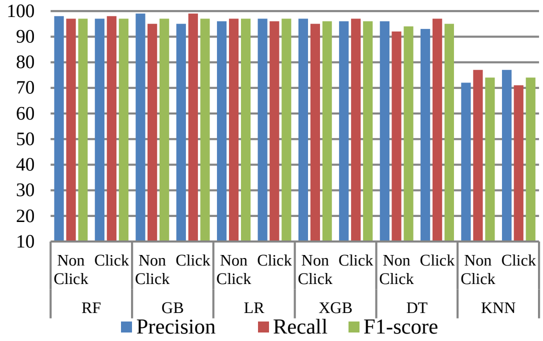


Figure 9. Evaluation Metrics for Models without Data Augmentation

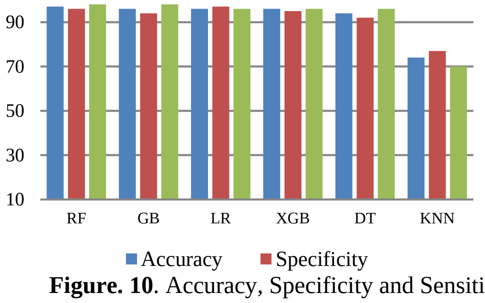


Figure. 10. Accuracy, Specificity and Sensitivity For Models Without Data Augmentation

Table 3. Evaluation Metrics Resultes for Models without Data Augmentation

Model	Acc.	Spec.	Sens.	Class (Non- click=0, click=1)	Precision	Recall	F-Score
RF	97%	96%	98%	0	98%	97%	97%
				1	97%	98%	97%
GB	96%	94%	98%	0	99%	95%	97%
				1	95%	99%	97%
LR	96%	97%	96%	0	96%	97%	97%
				1	97%	96%	97%
XGB	96%	95%	96%	0	97%	95%	96%
				1	96%	97%	96%
DT	94%	92%	96%	0	96%	94%	95%
				1	93%	97%	95%
KNN	74%	77%	70%	0	72%	77%	74%
				1	77%	71%	74%

4.2. User Behaviour Prediction with Data Augmentation

4.2.1. Random forest

After augmenting the dataset using the GAN algorithm as demonstrated in Table 4. The model accurately classified 759 non-click cases and 727 click instances out of an overall sample size of 1500 according to the confusion matrix in Fig 11. The Random Forest model obtained 99% accuracy, 98% sensitivity, and 99% specificity. Additionally, the RF model attained a precision, recall, and F1-score of 99 % for both classes as shown in Table 5.

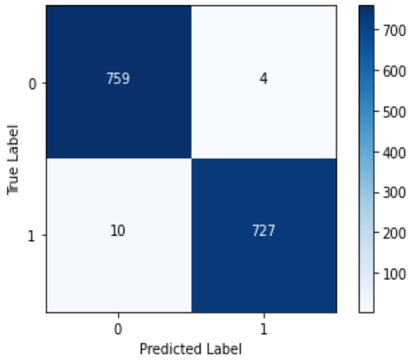


Figure 11. RF Confusion Matrix

Table 4. Augmented Dataset Sample

Daily Time Spent on Site	Ag e	Area Income	Daily Internet Usage	Mal e	Clicked on Ad
81.98	40	65461.92	229.22	0	0
66.01	23	34127.21	151.95	0	1
61.57	53	35253.98	125.94	1	1
53.3	34	44893.71	111.94	0	1
34.87	40	59621.02	200.23	0	1
43.6	38	20856.54	170.49	0	1
77.88	37	55353.41	254.57	0	0
75.83	27	67516.07	200.59	0	0

4.2.2. Gradient boosting

With an accuracy score of 0.99, the GB model successfully identified 99% of testing data. According to the confusion matrix in Fig 12, the model accurately predicted 759 click cases and 726 non-click instances. In addition, the model has a sensitivity of 0.98 and a specificity of 0.99, indicating that it.

Accurately detected 98% of click cases and 99% of non-click instances. The model also did well in correctly classifying occurrences into their respective classes, as evidenced by its high precision, recall, and F1-score of 99% for both classes. The GB model trained on the enhanced dataset using GAN appears to be highly accurate and reliable in identifying click and non-click cases as illustrated in Table 5.

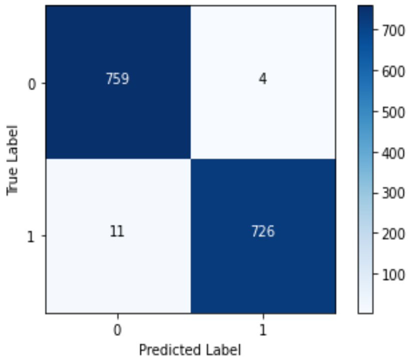


Figure. 12. GB Confusion Matrix

4.2.3. Logistic regression

The LR model achieved 97% accuracy, correctly identifying 748 clicks and 710 non-clicks out of the total samples as illustrated in confusion matrix in Fig 13. The sensitivity and specificity of the model for non-click class is 0.96 and 0.98, respectively. The precision for non-click class is 97% and for click class is 98%. The recall for click instances is 96% and for non-click instances is 98%. Both classes have an F1 score of 97%. As demonstrated in Table 5.

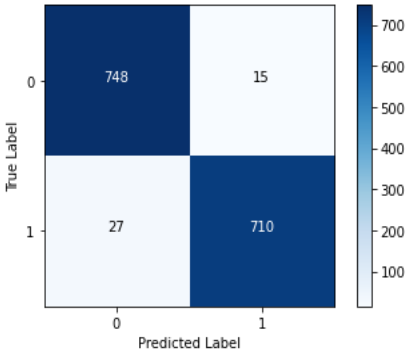


Figure 13. LR Confusion Matrix

4.2.4. XGBoost

The XGB model achieved an accuracy of 0.98 when tested on the enriched dataset using GAN technology. This indicates that it properly detected 98% of the instances. The confusion matrix in Fig 14 demonstrates that the model successfully recognised 756 click occurrences and 728 non-click instances, respectively. The model's sensitivity and specificity were 0.98 and 0.99, suggesting that it correctly detected 98% of click cases and 99% of non-click instances. In addition, the model performed well in correctly classifying instances into their respective classes, as seen by the high values of precision and recall, as well as an F1-score that was 99% for both classes as showed in Table 5.

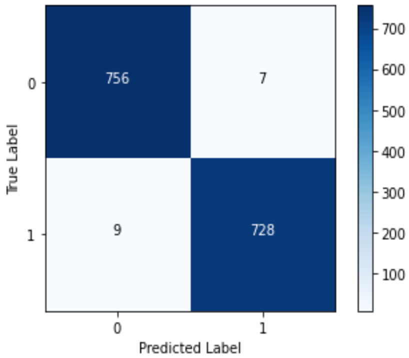


Figure. 14. XGB Confusion Matrix

4.2.5. Decision tree

As confusion matrix show in Fig 15, the algorithm accurately identified 758 instances in which the user did not click and 722 instances in which the user did click by making 758 TP predictions and 722 TN predictions. It obtained an accuracy of 98%, implying that it correctly identified 98% of the occurrences. The results show that the model correctly detected 97% of click cases and 99% of non-click instances, with a sensitivity and specificity of 97% and 99%, respectively. For the non-click class, the model scored an F1-score of 99% along with precision and recall scores of 98% and 99% respectively. The model achieved a precision of 99%, a recall of 98%, and an F1-score of 99% for the click class.

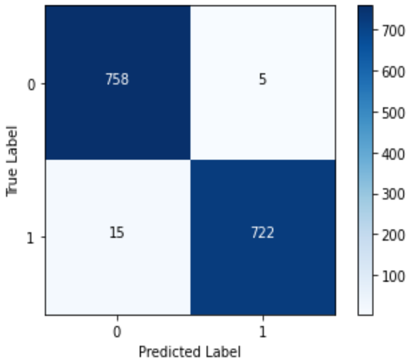


Figure. 15. DT Confusion Matrix

4.2.6. KNN

The K-Nearest Neighbours (KNN) model had an accuracy of 0.94, correctly identifying 94% of cases. It successfully detected 730 click cases and 694 non-click instances with 730 true positive (TP) predictions and 694 true negative (TN) predictions as demonstrated in Fig 16. The model had a sensitivity of 0.94 for click instances and a specificity of 0.95 for non-click cases. It also had recall, F1-score, and precision values of 96%, 94%, and 94%, respectively, for non-click occurrences and 94%, 95%, and 94%, respectively, for click instances. Table 5 shows that we obtained these results by adjusting the k value to 3.

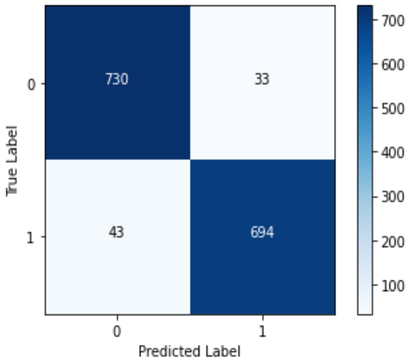


Figure. 16. KNN Confusion Matrix

5. CONCLUSION

The incorporation of GAN as a data complement has resulted in substantial improvements in the performance of all models as illustrated in Fig 19-24, as revealed by the accuracy results presented in Table 3. Prior to GAN, the KNN model had the lowest accuracy score of 74%, while the RF model had the highest score of 97%, as shown in Table 3. However, Table 5 demonstrates that all models have benefited from using GAN for data augmentation. The RF and GB models achieved the highest accuracy score of 99%, suggesting that they have learned more diverse and comprehensive patterns in the data, leading to superior performance.

Table 5. Evaluation Metrics Results for Models with Data Augmentation

Model	Acc.	Spec.	Sens.	Class (Non- click=0, click=1)	Precision	Recall	F-Score
RF	99%	99%	98%	0	99%	99%	99%
				1	99%	99%	99%
GB	99%	99%	98%	0	99%	99%	99%
				1	99%	99%	99%
LR	97%	98%	96%	0	97%	98%	97%
				1	98%	96%	97%
XGB	98%	99%	98%	0	99%	99%	99%
				1	99%	99%	99%
DT	98%	99%	97%	0	98%	99%	99%
				1	99%	98%	99%
KNN	94%	95%	94%	0	94%	96%	95%
				1	95%	94%	95%

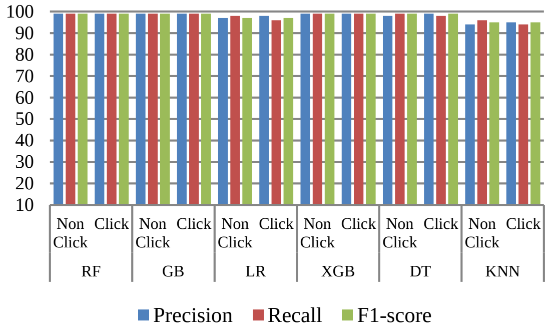


Figure 17. Evaluation Metrics for Models With Data Augmentation

Both XGB and DT models also exhibited significant improvement in accuracy, scoring 98%. The KNN model showed the most substantial improvement in accuracy. The training loss graph in Fig 24 indicates beneficial knowledge acquisition because it starts off high then regulates. The validation loss decreases to just over the training loss, indicating that the model is performing effectively and has minimal overfitting.

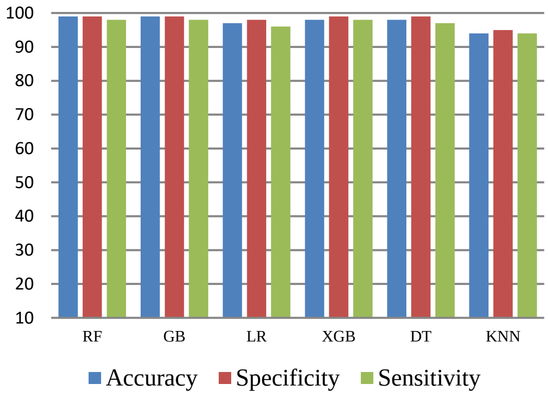


Figure 18. Accuracy, Specificity and Sensitivity for Models with Data Augmentation

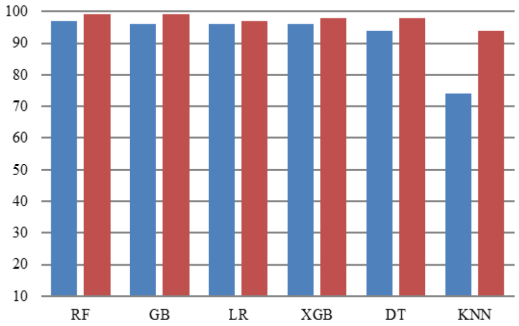


Figure 19. Comparison of Accuracy with And without Data Augmentation

Adding GAN-based data to existing data has made most models more sensitive and specific, which means that the generated data has improved their ability to correctly identify positive and negative instances. Overall, incorporating GAN-based data augmentation has considerably improved model performance, albeit to varying degrees depending on the model. This research proposes a new methodology for predicting click behaviour on online ads using a GAN-augmented dataset. We trained and validated the six machine learning models on both the original and augmented datasets using various performance metrics, including accuracy, sensitivity, specificity, F1-score, precision, and recall. The results demonstrate that generating new data with GAN has significantly improved the models' ability to distinguish between positive and negative occurrences. For instance, the accuracy of the RF and GB models both increased to 99%, having previously been at 97% and 96%, respectively. The KNN model exhibited the most significant improvement, with a 20% increase in accuracy compared to the other models. Most models also showed improvements in sensitivity and specificity due to the larger dataset.

Our study enhanced the dataset used to train machine learning models with data augmentation. We recognize that incremental data augmentation may be advantageous. Increasing augmentation gradually and evaluating its effect on model performance may help find a balance that improves model accuracy and minimizes overfitting.

To further understand how data augmentation affects model performance, future research could use a similar gradual method. The results of this study demonstrate that using GAN-based data augmentation significantly improves the performance of machine learning models in predicting user behavior. Specifically, several models' accuracy increased significantly after using GAN to generate additional data. This finding suggests that GAN-based data augmentation is a promising approach for user behavior analysis and prediction.

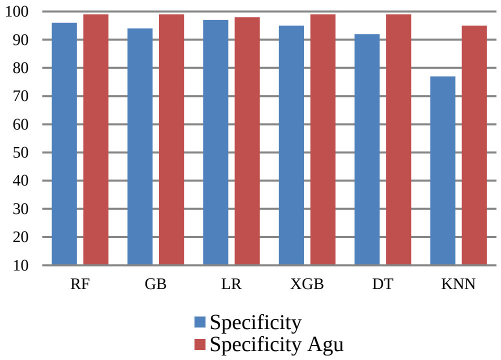


Figure 20. Comparison of Specificity with and Without Data Augmentation

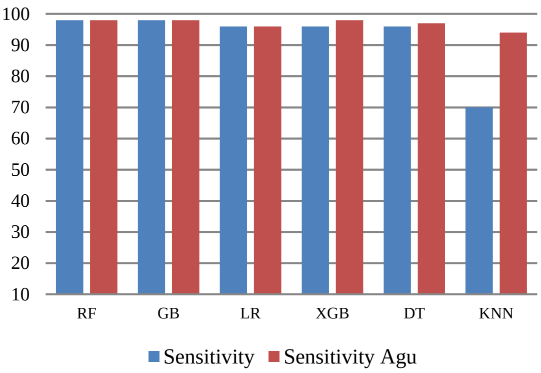


Figure 21. Comparison of sensitivity with and Without Data Augmentation

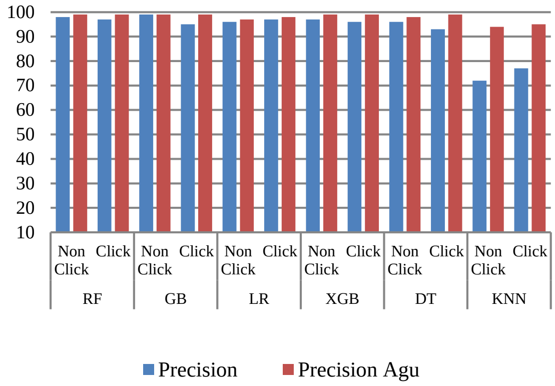


Figure 22. Comparison of precision with and Without Data Augmentation

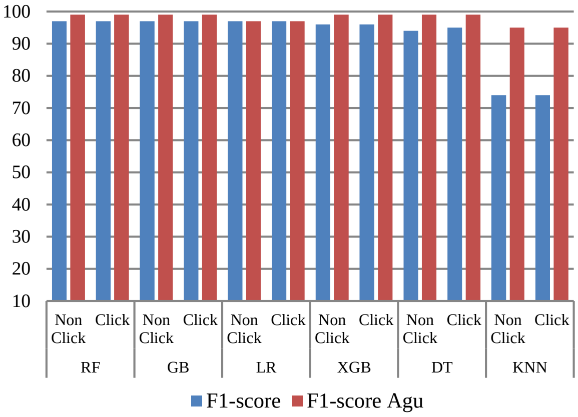


Figure 23. Comparison of F1-score with and Without Data Augmentation

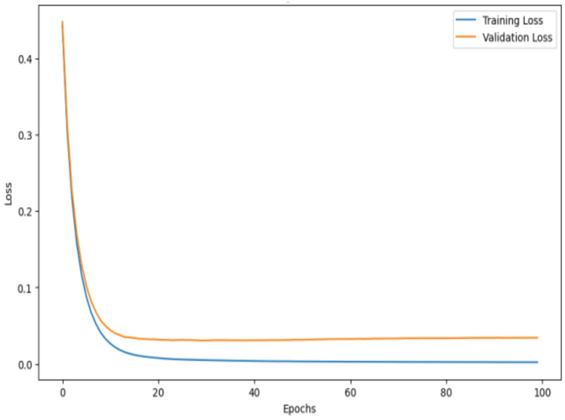


Figure 24. Training Loss Graph

AUTHORS' CONTRIBUTIONS

Amel Sulaiman MANDAN: Theoretical research, simulation study, and interpretation of results.
Oktay YILDIZ: Theoretical research and interpretation of results.

CONFLICT OF INTEREST

There is no conflict of interest in this study.

DECLARATION OF ETHICAL STANDARDS

The authors of this article state that the materials and methods employed in this study do not necessitate ethical committee approval or legal-special authorization.

REFERENCE

[1] Liu-Thompkins Yuping, "A Decade of Online Advertising Research: What We Learned and What We Need to Know," ***Journal of Advertising***, pp. 1-13, (2018).

[2] Y., & Zhai, P. Yang, "Click-through rate prediction in online advertising: A literature review," ***Information Processing & Management***, p. 59, (2022).

[3] Hong, Ziang, Xiong, Jinjie, You, Xiaolin, Wu, Min, Xia Wenxing, "CPIN: Comprehensive present-interest network for CTR prediction," ***Expert Systems With Applications***, (2021).

[4] Zhao Xudong, Xu Xinying, Han Xiaoxia, Ren Jinchang, Li Xingbing, Xie Jun, "DRIN: Deep Recurrent Interaction Network for click-through," ***Information Sciences***, (2022).

[5] WeiKang He, Yu Zhu, Jianghu Zhu, Yunpeng Xiao, "A click-through rate model of e-commerce based on user interest and temporal behavior," ***Expert Systems With Applications***, (2022).

[6] Danqing Zhu, "Advertising Click-Through Rate Prediction Based on CNN-LSTM Neural Network," ***Computational Intelligence and Neuroscience***, (2021).

[7] Liqing Qiu, Cheng'ai Sun, Qingyu Yang, Caixia Jing, "ICE-DEN: A click-through rate prediction method based on interest contribution extraction of dynamic attention intensity," ***Knowledge-Based Systems***, (2022).

[8] Y., Wang, S., Huang, Y., Zhao, X., Zhao, W., Duan, Y., & Wang, X. Tang, "Retrieval-Based Factorization Machines for Human Click Behavior Prediction," ***Computational Intelligence and Neuroscience***, (2022).

[9] J., Ma, C., Zhong, C., Zhao, P., & Mu, X. Zhang, "Multi-scale and multi-channel neural network for click-through rate prediction," ***Neurocomputing***, (2022).

[10] Dhanani, Keyur Rana Jenish, "Logistic Regression with Stochastic Gradient Ascent to Estimate Click Through Rate," in ***Information and Communication Technology for Sustainable Development: Proceedings of ICT4SD***, Singapore, p. 319,(2018).

[11] Gharibshah, Xingquan Zhu,Arthur Hainline, Michael Conway Zhabiz, "Deep Learning for User Interest and Response Prediction in Online Display Advertising," ***Data Science and Engineering***, (2020).

[12] K., Huang, Q., Zhang, F. E., & Lu, J. Song, "Coarse-to-fine: A dual-view attention network for click-through rate prediction," ***Knowledge-Based Systems***, p. 216, (2021).

[13] K., Huang, Q., Zhang, F. E., Lu, J. Song, "Coarse-to-fine: A dual-view attention network for click-through rate prediction," ***Knowledge-Based Systems***, (2021).

[14] D., Wang, Z., Zhang, L., Zou, J., Li, Q., Chen, Y., Sheng, W. Zou, "Deep Field Relation Neural Network for click-through rate prediction," ***Information Sciences***, pp. 128-139, (2021).

[15] D., Xu, R., Xu, X., Xie, Y. Jiang, "Multi-view feature transfer for click-through rate prediction," **Information Sciences**, pp. 961-976, (2021).

[16] M., Cai, S., Lai, Z., Qiu, L., Hu, Z., Ding, Y. Liu, "A joint learning model for click-through prediction in display advertising," ***Neurocomputing***, pp. 206-219, (2021).

[17] D., Hu, B., Chen, Q., Wang, X., Qi, Q., Wang, L., Liu, H. Li, "Attentive capsule network for click-through rate and conversion rate prediction in online advertising," ***Knowledge-Based Systems***, p. 106522, (2021).

[18] P., Yang, Y., Zhang, C. Zhai, "Causality-based CTR prediction using graph neural networks," ***Information Processing & Management***, p. 103137, (2023).

[19] Z., Wang, X., He, X., Huang, X., Chua, T. S. Tao, "HoAFM: A High-order Attentive Factorization Machine for CTR prediction," ***Information Processing and Management***, p. 102076, (2020).

[20] Y., Jiang, D., Wang, X., Xu, R. Xie, "Robust transfer integrated locally kernel embedding for click-through rate prediction," ***Information Sciences***, pp. 190-203, (2019).

[21] X., Liu, Q., Su, R., Tang, R., Liu, Z., He, X., Yang, J. Yang, "Click-through rate prediction using transfer learning with fine-tuned parameters," ***Information Sciences***, pp. 188-200, (2022).

[22] A., Shetty, S. D. Jose, "DistilledCTR: Accurate and scalable CTR prediction model through model distillation," ***Expert Systems with Applications***, p. 116474, (2022).

[23] Alhassan Mumuni and Fuseini Mumuni, "Data augmentation: A comprehensive survey of modern approaches," ***Array***, (2022).

[24] A., Mittal, M., & Battineni, G. Aggarwal, "Generative adversarial network: An overview of theory and applications.," ***International Journal of Information Management Data Insights***, (2021).

[25] S Xu et al., "Cardiovascular risk prediction method based on CFS subset evaluation and random forest classification framework," ***in 2nd international conference on big data analysis***, pp. 28–32,(2017).

[26] C., Csörgő, A., & Martínez-Muñoz, G. Bentéjac, "comparative analysis of gradient boosting algorithms," ***Artificial Intelligence Review***, pp. 1937-1967, (2021).

[27] K., Patel, H., Sanghvi, D., & Shah, M. Shah, "comparative analysis of logistic regression, random forest and KNN models for the text classification," ***Augmented Human Research***, pp. 1-16, (2020).

[28] J., & Rana, K. Dhanani, "Logistic Regression with Stochastic Gradient Ascent to Estimate Click Through Rate," ***Information and Communication Technology for Sustainable Development***, pp. 319-326, (2018).

[29] Tianqi Chen and Carlos Guestrin, "XGBoost: A Scalable Tree Boosting System," in KDD '16: ***The 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining***, San Francisco California, USA, pp. 785–794,(2016).

[30] C., Csörgő, A., Martínez-Muñoz, G. Bentéjac, "A comparative analysis of gradient boosting algorithms," ***Artificial Intelligence Review***, pp. 1937-1967, (2021).

[31] Charbuty B. and Abdulazeez A., "Classification Based on Decision Tree Algorithm for Machine Learning," ***Journal of Applied Science and Technology Trends***, pp. 20-28, (2021).

[32] I. D., Sun, Y., Wang, Z. Mienye, "Prediction performance of improved decision tree-based algorithms: a review," ***Procedia Manufacturing***, pp. 698-703, (2019).

[33] Shuangjie Li, Kaixiang Zhang, Qianru Chen, Shuqin Wang, and Shaoqiang Zhang, "Feature Selection for High Dimensional Data Using Weighted K-Nearest Neighbors and Genetic Algorithm," ***IEEE Access***, pp. 139512 - 139528, (2020).

[34] F., Araghinejad, S. Modaresi, "A comparative assessment of support vector machines, probabilistic neural networks, and K-nearest neighbor algorithms for water quality classification," ***Water resources management***, pp. 4095-4111, (2014).

[35] A. Internet advertising spending worldwide from 2007 to 2024, by format Guttmann. Statista.[Online].https://www.statista.com/statistics/276671/global-internet-advertising-expenditure-by-type/ (2021)