National College of Ireland

Automatic Test Data Generation in Banking Applications Using Deep Learning

MSc Research Project
Data Analytics

Taniya Bagh
Student ID: x22120821

School of Computing
National College of Ireland

Supervisor: Sasirekha Palaniswamy

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Taniya Bagh

Student ID: x22120831

Programme: Data Analytics **Year:** 2023

Module: Msc Data Analytics

Supervisor: Sasirekha Palaniswamy

Submission Due Date: 01/02/2023

Project Title: Automatic Test Data Generation in Banking Applications Using Deep Learning

Word Count: 8757 **Page Count** 22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Taniya Bagh

Date: 14 December 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Automatic Test Data Generation in Banking Applications Using Deep Learning

Taniya Bagh
X22120831

Abstract

It is well known that data associated with the Banking domain are huge in volume. With such large numbers come concerns regarding security and Privacy. These concerns are addressed with proper testing of the applications associated with the domain. In the present scenario, with the growth of technology, there is a rise in complexity in terms of handling and testing the software. Testing in the banking domain is one of the most crucial parts of the industry since it ensures not only the functionality of an organization but also responsible for reliability and security of the data indulged with the association. Therefore, robust testing is necessary to achieve accurate results which further assists in identifying issues and vulnerabilities that can potentially harm the financial data which leads to security breaches, and help them rectify the issues, safeguarding the sensitive data. In this paper, the Application of various deep learning techniques such as Generative Adversarial Networks, Variational Encoders, Recurrent Neural Networks, and other Hybrid models will be utilized to produce synthetic transaction data and the performance of the model will be evaluated by calculating statistical values such as MSE, MAE and RMSE.

1 Introduction

1.1 Background and Motivation

There are numerous yet significant challenges associated with test data generation when it comes to the banking domain. In the present scenario, manual approaches for generating test data are opted which are not only tedious but can potentially hinder the performance evaluation of a software. Another approach that's opted for by most of the big organizations is testing with obfuscated data which are highly confidential and susceptible to security breaches. Handling such a huge volume of data and maintaining data privacy and security can be cumbersome and significant at the same time. The test data should be adaptable to put up with dynamic systems of the banking domain and continuous updates of the application associated with it. Maintaining a balance between the issues faced by the organization while testing is necessary and to overcome such difficulties organization needs to generate reliable data that will be secure, accurate and maintain the privacy of the confidential data. These measurements help in the improvement of the efficacy of the testing system and will ensure the smooth functionality of the banking applications.
Deep learning models have the ability to learn from historical data, identify the hidden layers, and create synthetic data that mimics real-world scenarios is a benefit when it comes to mimicking the data related to financial operations and implementing them for test data generation can help boost the efficiency as well as the productiveness of the testing of

software applications in the banking domain. The data generated using conventional methods has several setbacks concerning the security, diversity, reliability, and privacy of the data. Additionally, there is another yet major setback for testing of software application is that it potentially lacks the details related to real-life scenarios. These issues can be addressed using deep learning algorithms such as generative adversarial networks commonly known as GANs, Recurrent neural Networks (RNN), variational encoders (VAE), and many more which can produce synthetic data that are complex and realistic, resembling actual banking data. Adapting deep learning methodologies contributes to the automation of the test data generation. The automation of test data generation will not only lead to speed up the testing process but will also help in the reduction of manual intervention. An additional advantage of opting for deep learning models is that they can keep up with the dynamic systems that include the continuous update of software, ensuring keeping the system's reliability and effectiveness on point. These processes will lead to achieving a reliable yet efficient system that will adhere to all the rules and regulations set for the organization while maintaining the integrity of the banking data. This paper will focus on addressing all the limitations faced by banking organizations in terms of test data generation and overcoming them by the implementation of different deep-learning models.

1.2 Problem Statement and Research Question

Using inadequate data for testing purposes can lead to various malfunctions which further affect the performance of the software applications. These vulnerabilities can potentially affect the growth of the organization's financial status and can take a toll on reputation too and affect the customers/clients associated with the organization. Several studies and research have proved that the application of automation in software testing is more efficient as compared to traditional methods since it minimizes the risk of issues occurring through manual intervention. Although there are ample benefits of automatic test data generation over manual methods, most organizations opt for manual approaches till date and there is room for improvements that need to be addressed.

The proposed research paper focuses on identifying and providing analysis on the question that states – "To what extent do deep learning techniques influence the applications associated with the banking domain through generating automatic test data that will mimic the real-scenario data and will be diverse in nature keeping the significant requirements such as data privacy, regulatory compliance, and security intact?"

1.3 Objective

Automation of test data generation using deep learning algorithms will not only increase the reliability of the banking applications but will contribute to the security and privacy of the data as well, which serves the objective of this paper. Introducing deep learning models such as GANs, RNN, VAE, and GMM for automatic test data generation that will mimic real-world data and assist in mitigating above stated problems. The hybrid models formed using these models have also shown a significant rise in terms of the accuracy of the synthetic data generation which will be discussed and proved further.

1.4 Structure of the Report

The report structure comprises of following sections. Section 1 consists of a brief overview of the topic related to the research paper which includes the limitations caused by manual methods, the problem statement that needs to be addressed which led to the research question,

and the purpose or the objective of the paper that will be worked upon. Section 2 comprises of brief literature review that was carried out to describe the past work done using similar strategies to achieve different goals in the field of automation. The next section, section 3 includes the methodology which includes the work done to achieve the objective of the research. Section 4 comprises of design specification of the opted models and Section 5 contains the implementation of the selected deep learning models. Section 6 and 7 consists of the evaluation of the performance of the models and later consists of a conclusion and future work.

2 Related Work

In the recent era, every organization works with big data to serve different purposes that are beneficial to the organization as well as the end user. The banking sector is one such domain that has a tremendous amount of data to handle and work with. For several years, continuous studies have been carried out to integrate and address the pros and cons associated with the sector. Continuous evaluation has taken place to bring evolution in the same field.

2.1 Test data generation using Conventional techniques and transition towards automation due to its limitation in banking applications.

Another example of the advancement of new technologies in this sector is e-banking and with technologies comes the risk of cyber-security threats which makes testing of software in an adequate manner necessary. Software should be made threat-free (such as Microsoft SDL tool) throughout its life cycle since the performance of the software acts as a bridge between the threat and the application. There are several tools and approaches carried out to build threat-free models that will provide security to the data (Abdallah, 2010). With continuous development in the field of banking, comes different challenges that are necessary to address to avoid vulnerabilities such as fraud, security breaches, and under-performance of the software. In a study carried out in (Murthy, 2018), addresses the limitations that occurred due to manual practices carried out to generate test data which is further used for testing applications. In recent eras, the whole world has been glued to mobile phones and that brought mobile applications into the trend. The banking sector is not behind in the race and has introduced several mobile banking applications that have leveraged many good things and have made life easier. With good comes bad, and although life has gotten easier but with that, there has been a tremendous increase in the cases of online thefts, phishing, frauds, breaches, etc. Numerous reasons cause these issues such as inadequate testing of the applications (manually tested) due to a shortage of tools that fail to detect data-related vulnerabilities, manual intervention, and many more. Studies are carried out to address these shortcomings and propose mitigations to overcome these issues (Sen Chen, 2020). These papers gave insights regarding the traditional methodologies used for application/software testing and the limitations faced by it.

To overcome such barriers, the banking sector has started to shift testing and operations from a manual approach to automation. In the study carried out by (Akin, 2018) has enlisted various benefits achieved as a result of the transition from manual regression testing to automated regression testing. Regression testing is carried out to evaluate the performance of

a code after it has been updated as per the requirements of the organization thus making it one of the integral parts since it evaluates the performance and the functionality of the updated code, ensuring that the code is delivered without any vulnerabilities or bugs. Implementing automation to this process has resulted in many benefits such as faster execution of tests, accurate test coverage, minimal human intervention leading to a reduction in bugs/defects, and faster and more accurate results. Many challenges were faced by the authors while implementing this transition, but they were able to overcome them by automating the system using the JAVA programming language. Moreover, this implementation helped them in the cost-cutting mechanism by saving the work of several staff members (count to 48). Microservices are the architecture that enables features present in the application to work independently without any worries about the impact on the other factors which further facilitates the creation and maintenance of the external ecosystem. In (Ding, 2020), the authors enlisted the issues received while creating test cases using manual techniques. There were significant discrepancies noticed in the functionality of the manually generated test cases and they were failing to achieve the objectives as mentioned in the test requirements. As a result, studies were carried out to improve the functionality by automating the test case generation using various algorithms namely Automated Efficient Test Generator known as AETG, and another algorithm named as Pairwise. The architecture of the framework was created in a way that all the layers (testing layers) present in the framework such as data, test case, adaption layer, analysis, and execution worked together to give the results. This adaptation was implemented to achieve the maximum efficiency out of the test cases so that it can contribute to enhancement in the capability of the testing of software. The papers mentioned above gave an overview of the importance of the application of automation in testing using machine learning and deep learning algorithms and understanding current technologies used by organizations to overcome the limitations.

The increase in the volume of data has led to an increase in the implementation of deep learning algorithms for data management, data extraction (useful), and analysis of the quality of data. Similarly, this approach has been included in banking applications to achieve various set goals and requirements. One such implementation is mentioned in (Wang, 2020), where authors have carried out the study and application of deep learning algorithms for image detection in credit card systems. The objective of the study was to detect the image orientation received by scanning the images while issuing a credit card from the bank. For image classification, they used deep CNNs, and the opted model was VGG16. Various synthetic data were generated for training the model and then the trained model was again used for the prediction of certain parts (contextual) which helped the model to produce more accurate and efficient results (Shilpa, 2018). Banking domains contain data that are highly confidential and are most susceptible to breaches and threats. All the information of an organization is stored in the databases. Therefore, it is necessary to keep its safety at utmost importance. Any attacks on the database can lead to data exploitation, the exposing of personal information that can be sensitive, and other hazardous outcomes. To avoid such unforeseen situations, (Ashlam, 2022) introduced a framework that is constructed using multi-phase algorithms using deep learning techniques such as clustering, CNN, KNN, etc to avoid threats caused by SQL injection attacks. The evaluation of the performance of the models was carried out by calculating accuracy, F1 Score, and recall. These papers helped to

gain knowledge about the application of different deep learning algorithms in different applications of banking that will further assist in achieving the set goal of this research paper.

2.2 Application of Deep Learning algorithms for test data generation.

Testing is necessary for every organization whether it be software, medical, or any other field. In (M. Cinquini, 2021), the authors describe the importance of synthetic data generation in Software testing and the impact caused because of its limitations, therefore they created a baseline mode using NCDA to identify the nonlinearity in the data. The efficiency of the generation of synthetic data is further enhanced using a generative method named GENCDA. Though the model created gave decent results, it was concluded that the efficiency of the models and the data can be increased using deep learning techniques. For this research paper various other research papers, assisted in electing some of the deep learning models that will further assist in the achievement of the set goal.

RNN

In the present era, everything is shifting towards automation and robots are the best example of evolution in the field of technology that mimics human behavior, this makes the application of deep learning leveraging in the field of robotics. Testing on real-world datasets is carried out using various neural networks such as recurrent neural networks on sequential data. GAN and VAE are popular techniques due to their ability to create synthesis data which should be mirror images of the real data. CRNN technique was used by (Martinelli, 2023) to generate synthetic data for sensors (UWB and UHF-ID). In (R. S. Bhowmick, 2020) authors introduce a word or spelling corrector for a specified Indian language (Hindi and Bengali), they used LSTM and its type to achieve their objective and the end model was used to compare with RNN (since it is vastly used for language processing). The accuracy of the model achieved is 80%. One of the major aspects of AI is Text mining, it holds significant value in the analysis department as well as the business world therefore various deep learning methods are used to make it more efficient. In (A. C. Pandey, 2019) deep learning methods such as RNN and LSTM were used to generate a synthetic dataset with the help of hyper parameter tuning efficient data were generated with higher accuracy of the model.

GAN

The application of deep learning is widely used in image recognition and therefore there are numerous methodologies present that can be used to generate fake thermal facial data samples. To evaluate the quality of the image generated using different techniques, a sample of Synthetic data consisting of thermal facial data was created by (Corcoran, 2021) using StyleGAN which was developed by NVIDIA to produce high-quality synthetic data. Other techniques were used such as Wide ResNet CNN which proved to be the most effective technique. In the field of medical science, there is continuous growth in terms of technology and science. Any limitations can be a roadblock for deep learning associated with the field. One such limitation was captured in the radiology images, the lack of annotation was proving to be a hindrance in the path of in-depth knowledge, therefore, a Deep Convolutional Generative Adversarial Network was used to create X-ray and MRI synthetic data which was of high-resolution in (S. Shetty, 2023) and evaluation of the generated data indicated the rise

in the accuracy in image classification process by 4-5%. Air pollution is increasing day by day and many researchers are working to apply deep learning techniques to achieve efficient prediction of air quality but due to inconsistent data, it has become nearly impossible to achieve. Therefore, the paper (Le, 2021) proposed a type of GAN model named AirGAN which will learn the distribution of real sequence and added unsupervised adversarial loss will help the model to distinguish between real and synthetic data. There was a significant drop in the values of MSE from 0.024 to 0.015. The growth in technology has demanded growth in applications of AI/ML as well. To enhance the data quality different deep learning models such as GAN and HMA are used and the accuracy of the model is calculated based on the three Vs that is Variety, Veracity, and Volume of the data(A. Sharma, 2023).

Hybrid Models

Hybrid models consisting of VAE and GAN (VAE) were used to generate photovoltaic samples. The efficiency of the encoder was increased by the application of hybrid deep neural networks HDD with layers consisting of LSTM which upon evaluation gave exceptionally good outcomes (D. A. Rosa de Jesús, 2021). In (B. Hariharan, 2022), authors tried to carry out image-to-image classification using deep convolution GAN (DCGAN) to achieve high-resolution images with other specifications intact. To improve the efficiency of the model, StyleGAN is used, so the quality of the synthetic data gets enhanced. A hybrid model consists of GRU, LSTM, and a Neural network that is fully connected are used to create a model for wind power generation prediction. Upon evaluation, the proposed model provided results with 10% higher accuracy than that of NN(M. A. Hossain, 2020). A hybrid model consisting of ANN and Monte Carlo for probabilistic analysis was built to create synthetic data that conveys wind scenarios to design the hybrid energy systems (J. Chen and J. Zhao, 2021).

3 Research Methodology

The fundamentals of Knowledge discovery in the databases or KDD is the methodology that has been opted, keeping the objective of the thesis paper in mind that will assist in determining the working of the models based on how closely each model generates automated test data which will closely resemble the data feed while training the models (real-world data). This approach includes data extraction which extracts useful information from the huge volume of raw data and further utilizes this data to gain insights, develop deep learning models for the automatic generation of test data, and evaluate the performance of the built models as shown in Figure 1.

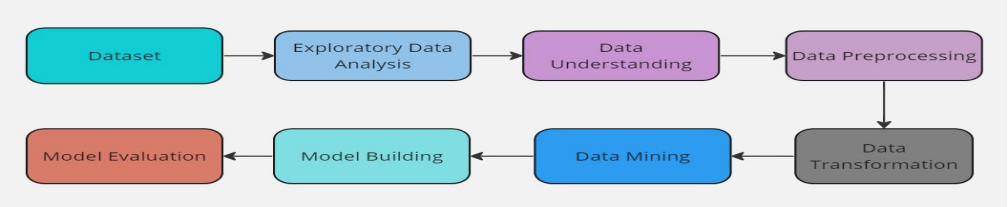

Figure 1: KDD Methodology for Automatic Test Data Generation

3.1 Data Collection and Specification

The dataset [1]that has been utilized for carrying out the work to achieve the goal of this research paper has been collected from the Kaggle Website under the name "Bank Customer Segmentation (1M+ Transactions)" which contains details of transactions from an Indian bank and is available for the public in csv file format (bank_transactions.csv). The dataset comprised nine fields which consisted of transaction ID, Customer ID, Customer Gender, Customer Location, Customer account balance, Transaction time, and Transaction amount in INR. Two of the fields namely Customer DOB and Transaction date are encrypted and have been handled in the code. There are various types of transactions present in the dataset such as credit, debit, and also contains the customer's profile which will assist in achieving a robust model.

3.2 EDA (Exploratory Data Analysis)

This process is carried out to gain the insights and patterns a data possesses. It helps in achieving in-depth knowledge of the data. With the help of visualization and statistical analysis, we can identify the nature of the data, outliers that can hinder the performance of the model, and the relationship between the features. This is an important step since this process helps in decision-making and achieving accurate results which will further assist in the building of high high-performing model that will serve the objective of the project. Data cleaning is done by handling missing, null values (using dropna() function), detecting, and removing outliers using the interquartile range, and removing any noise present in the dataset to ensure clean data to proceed further with the implementation of the model. Visualization using bar plots, pair plots, correlation matrices, etc was carried out to get in-depth knowledge of the data. Figure 2 demonstrates the bar plot with the mean and median of the numerical feature. It can be seen that the transaction time has the highest value of the median whereas the customer account balance field has the highest value of mean.

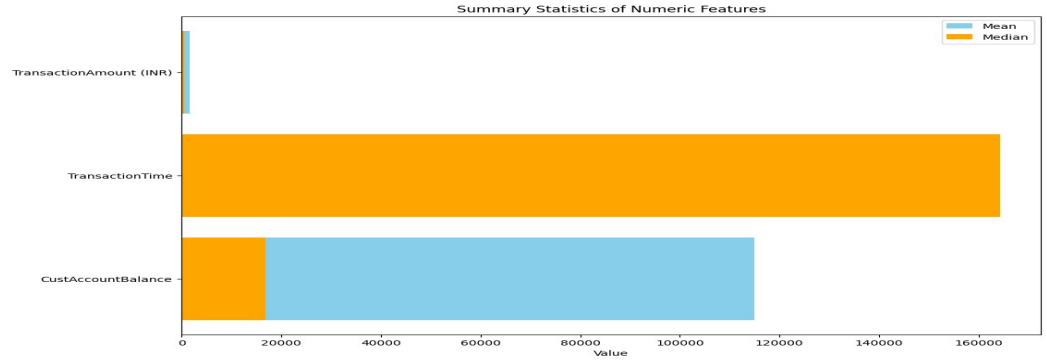

Figure 2 Statistical Summary of Numerical Features

Figure 3 demonstrates the count of the males and females present in the data. The bar plot showcases the count of males is much higher than that of females and it was carried out to gain clarity of the data which will further assist with the feature selection method.

[1] https://www.kaggle.com/datasets/shivamb/bank-customer-segmentation

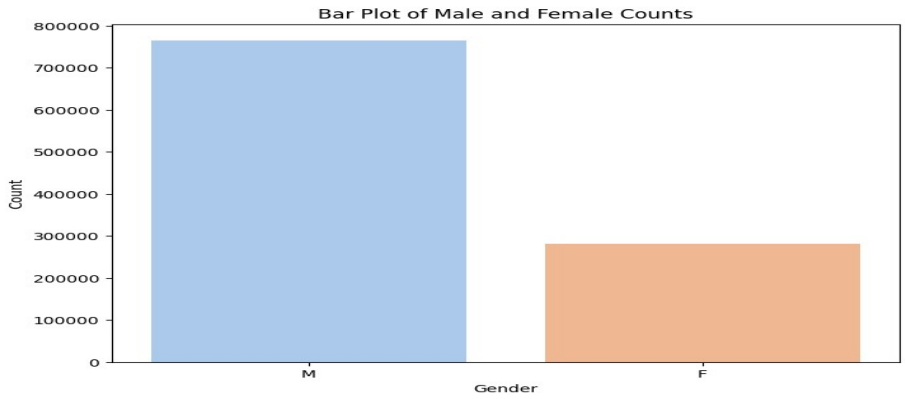

Figure 3 Count of Male and Female

A pair plot was also plotted for all the features as shown in Figure 4. After gaining insights from the features and understanding their nature through visualization we move forward toward our next process which is Data preprocessing.

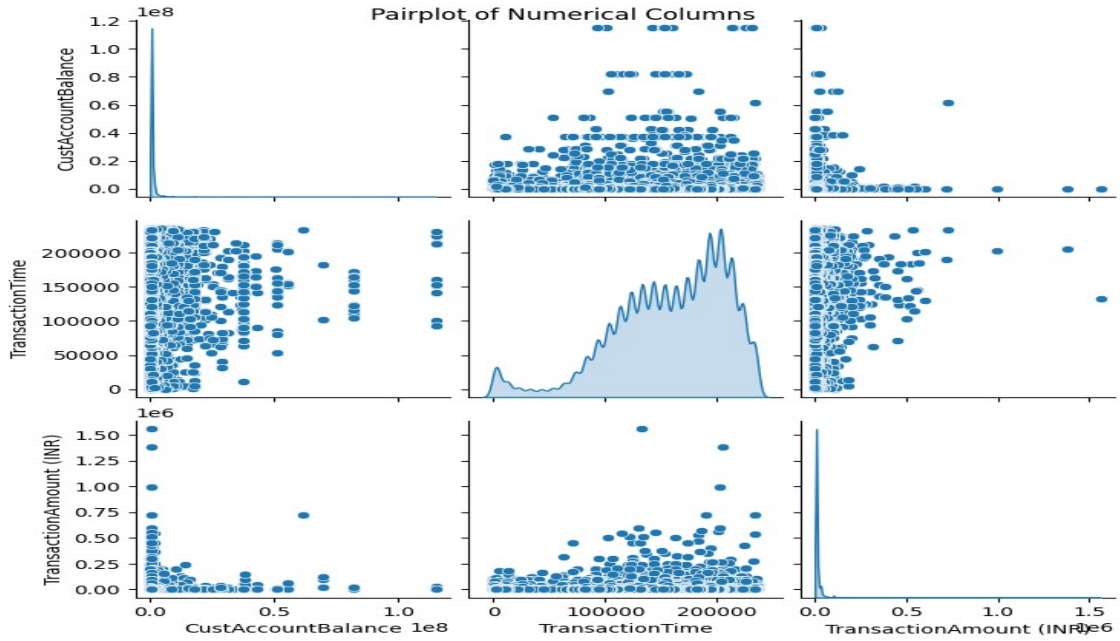

Figure 4 Pairplot for all the features.

3.3 Data Preprocessing

Different approaches of data preprocessing were opted by different models. Initially, to achieve simple models and gain more insights into synthetic data generation mechanisms, feature selection was used to extract only numerical fields for models such as RNN and GAN. While proceeding toward complexity all the features were utilized while model building. The data is split into train and test data in the ratio of 80:20 that is 80% training set and 20% test data using the scikit-learn library. However, the method of the data split varies. For the models GAN and the RNN model, data was split considering the target variable in

Y(CustGender) and other variables in X and further gets split into train and test variables whereas, for hybrid models, the whole data set gets split into two parts train set and test set.

3.4 Data Transformation

The fields present in the dataset are both numerical and categorical. The numerical features undergo normalization and standardization to ensure that all the values are consistent and will not potentially hinder the learning process of the model that has to be implemented. Categorical features are converted into numerical fields using a label encoder so that all the features can be addressed while building models for synthetic data generation. One of the fields that is Customer DOB was transformed into the Age field and the negative values present in the field were set to 0 so that no junk values could hinder the pre-training of the models. Later this transformed field was used in building the GAN and RNN model. While building the models, the standardscaler() function was used to fit the training set as that of real data, and after the synthetic data were generated inverse-transform method present in the scaler function was utilized to convert generated samples to the original scale that matches real data.

3.5 Data Mining

There are numerous approaches mentioned for generating synthetic data both through the application of machine learning as well as deep learning techniques but upon deep analysis with the reference of the research papers, it was seen that GAN, RNN, and VAE were the top high-performing models which delivered synthetic data that resembles the original data. Therefore, in this paper, four models will be evaluated based on their performance. These models are two basic models (GAN and RNN) and two hybrid models (Ensembled VAE and Hybrid of GAN and VAE known as VAEGAN).

3.6 Model Evaluation

The performance of the models needs to be evaluated to achieve the efficiency and reliability of the model created. The evaluation can be done both by visualization and calculating the statistical values. In this paper, the accuracy of the models is calculated using statistical values such as mean squared error (MSE), mean absolute error (MAE), and Root mean squared error (RMSE). For GAN, the mean and standard deviation of real values and synthetic data were calculated to check the quality of the data generated and will further assist in assessing the capturing capability of the GAN model when it was trained on real data. Visualization is also included in assessing the quality of the synthetic data by plotting the distribution and bar plots of real data vs synthetic data generated to understand. Moreover, these plots assist in understanding the capability of the models to generate data that will resemble the real data on which these models were trained on. The MSE, MAE, and RMSE can be calculated with the equation mentioned in (1), (2) and (3).

$$MSE = {}^{1}/_{n} \sum_{i=1}^{n} (yi - yi')^{2} \quad (1)$$

$$MAE = {}^{1}/_{n} \sum_{i=1}^{n} | yi - yi' | \quad (2)$$

$$RMSE = \sqrt{MSE} \quad (3)$$

Where,

- *n* indicates the total observation numbers.
- *yi* indicates the actual value for *i* observations.
- *yi'* indicates the predicted value for *i* observations.

4 Design Specification

A framework that is 3-tier in nature (Figure 5) has been opted for the implementation of this thesis paper. The first tier or layer is named Tier Data, in this layer data is stored in different sources(in this case it is google colab pro) that can be retrieved and accessed for carrying out data extraction, data transformation, and pre-processing steps. The clean and retrieved data goes to the second tier named Application tier. In this layer, different deep learning models are applied to achieve the objective of the proposed paper which is the automatic generation of synthetic data that will mimic the real data which was again used by the models for training. After the generation of synthetic data comes the third part of the architecture which is the evaluation tier, in this layer statistical evaluation and visualization are performed to evaluate the performance of the models and the quality of the synthetic data.

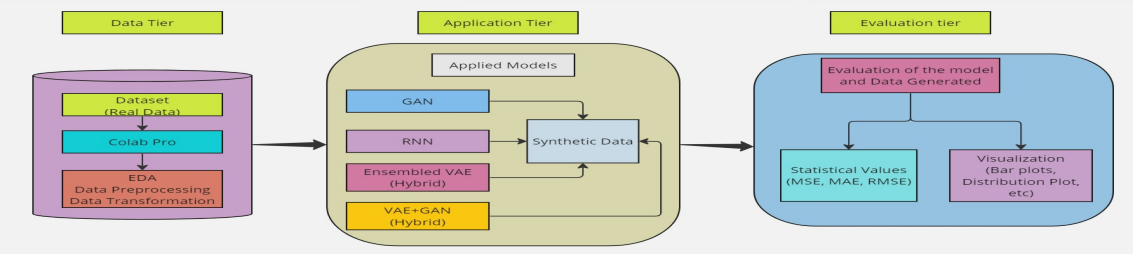

Figure 5: 3-Tier Architecture

4.1 Neural Network Architecture:

Inspired by the human brain mechanism, machine learning introduced an architecture that is supposed to process data as done by a human brain. This architecture was named as Neural network, a part of an artificial neural network (ANN). It contains layers that are interconnected and named nodes or neurons. The raw data (input) data is fed to the input layer and then gets processed through these hidden layers. The data gets mathematically computed in these hidden layers. After computation, the data is sent to the output layer from where we receive the anticipated result as shown in Figure 6. In the figure, the connections depict the weights which help in determining the input's strength and influence on the output produced by the neurons. The weights in the neural networks are set while training the model depending on the nature of the input data provided so that the desired output can be achieved

with accurate predictions. It has various importance on its own such as it can handle huge volumes of data even with the complexity which makes it a good fit for tasks that involve image classification, NLPs, etc. It can be moulded as per the requirements due to its flexible architecture. The output received from the model created using neural networks tends to give more accurate results as compared to other machine learning models. The most important advantage of this architecture is that it doesn't need manual intervention to learn and provide the output. Various other architectures that are on high demand are comprised using neural networks such as Convolutional neural networks, Deep Neural Networks, RNN, and GAN.

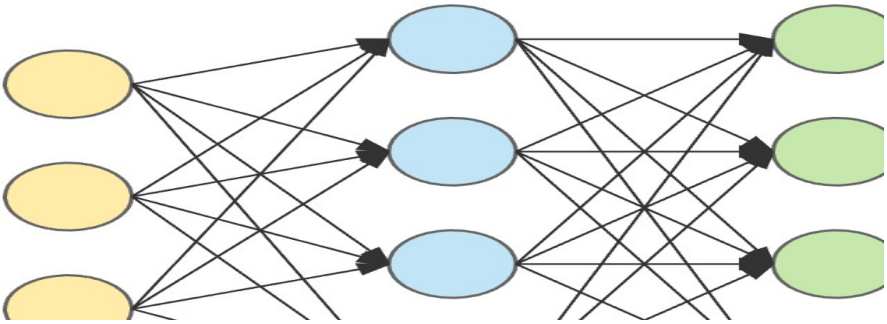

Figure 6[2] Neural Network Architecture

5 Implementation

Various software and technologies were used to work and achieve the proposed objective that has been mentioned in this paper. Figure 7 represents a pictorial representation of all the software, frameworks, and technologies that were utilized while working on this thesis paper. The software platform that was preferred for the implementation of this project is Google Colab Pro. Since the dataset contained quite a large number of data (1+ million data). Computing deep learning algorithms such as GAN, RNN, and Hybrid models needs longer runtime which comes in handy with the colab software and faster results are achieved. The programming language used for the implementation of this work is Python. It has a wide collection of libraries and frameworks within it such as Tensorflow, Keras, Pytorch,Pandas, Numpy, Matplotlib, Scikit-learn, and many more. With the use of these resources and the simplicity of the language, building the machine learning models becomes easier. For result visualization and Exploratory Data Analysis libraries such as Matplotlib and Seaborn. For building all the deep learning models the framework used is TensorFlow, Keras (high-level API for Tensorflow library), and Pytorch . Table 1 consists of the configuration setup that is necessary for carrying out the work.

[2] https://d14b9ctw0m6fid.cloudfront.net/ugblog/wpcontent/uploads/2020/05/1_3fA77_mLNiJTSgZFhYnU0Q.png

Figure 7 Technologies and Software used.

Table 1: Configuration Setup

IDE	Google Colab Pro
Programming Language	Python
Modules	Keras, TensorFlow, Matplotlib,Pandas, Numpy
Computation	CPU
Number of CPU	1
Type	Intel(R) Xeon(R) CPU @ 2.20GHz

5.1 Generative Adversarial Network (GAN) Model Implementation:

GANs are one of the most used deep learning techniques for generating synthetic data. GANs are constructed using two neural networks which increase their capability to learn and produce synthetic data that resembles the distribution of real data. Figure 8 demonstrates its architecture. It comprises a Generator and a Discriminator. For the synthesis of synthetic data, the generator is fed with random input(noise) from latent space. The discriminator is fed both the real data from the dataset and synthetic data produced to check if it can discriminate the difference between the two samples. Adversarial works with both samples continuously leading to improved quality of the samples received by the generator. The generator used in the model consists of 3 dense layers (256, 512, and 1024). Each dense layer is followed by LeakyRelu activation function which works to include non-linearity. For providing the stabilization and fastening up the training of the neural networks, batch normalization has been included in the layers. To keep the generated data within a specified range ([-1,1]), the tanh activation function has been utilized in the output layer. While training the model, the data is modified iteratively which leads to updating both the networks (Discriminator and Generator). The discriminator is trained using training data that is retrieved from the real dataset and the synthetic data that was received from the generator.

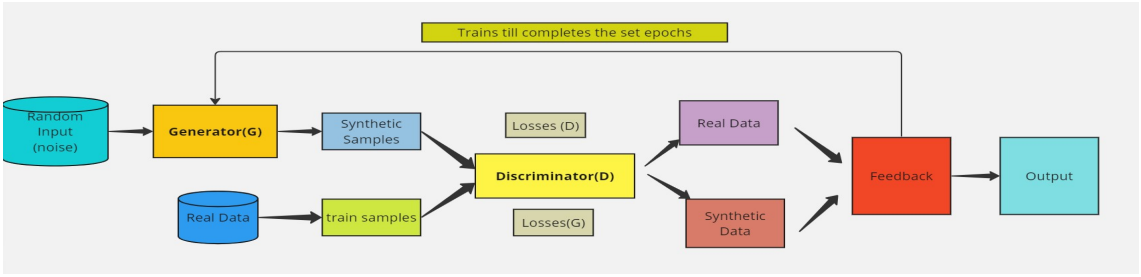

Figure 8 Architecture of GAN

This method of adversarial training continues till the discriminator gets trained and generates synthetic data that mimics the real data. The iteration of adversarial training is set in epochs which is set to 30000 and the batch size was set to 64. The loss function and optimizer used for reducing the losses from both networks are binary-cross entropy and Adam optimizer. On executing the model, displays are added that show and store the losses from both the generator and discriminator. The generated synthetic data is then transformed to the original scale using inverse standard scalar which is the final outcome of the model.

5.2 Recurrent Neural Network Model Implementation:

In this type of neural network, the output generated from the past process is fed as input to the current process and therefore the name recurrent is given to this neural network model also known as bi-directional artificial neural network. It consists of a hidden state that keeps memory of the sequence(memory state) and then it is utilized to predict the output as per the requirement. The RNN model used for this project is built using Keras. For sequence modelling, SimpleRNN layer is utilized which is also known as the building block of the architecture of RNN. The function of this layer is to capture dependencies that are temporal from sequential data, and for delivering the output, the dense layer is used. The model gets trained till completes the set epocs. While training, train_step function from tensorflow library is applied so that there is a continuous computation of gradients and weights carried out till the learning of the model. For increasing the efficiency of the training and preventing the gradient from exploding, adam optimizer and gradient clipping was used. Once the training is completed, by predicting the future value in an iteration pattern, the model generates the synthetic data. The prediction of these data depends on the seed sequence. To avoid the crash of the code while training the model, keras.backend.clear_session() function from tensorflow was used to release resources periodically. The generated output data is then transformed to the original scale by using inverse standardscaler since while training and generating the synthetic data, standardscalar() method is used to normalize train data to ensure that the data present in the training set are consistent which further helps the training process stable and efficient. Figure 9 [3]depicts the general model of RNN.

[3] https://towardsdatascience.com/implementation-of-rnn-lstm-and-gru-a4250bf6c090

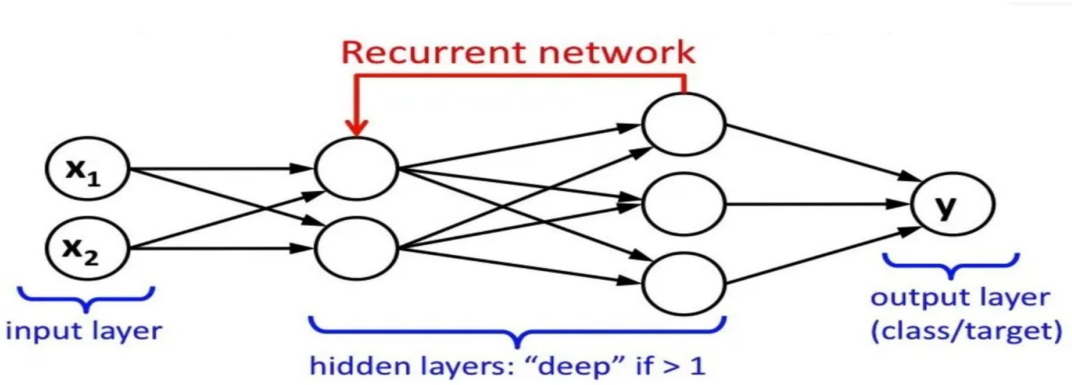

Figure 9 RNN Architecture

5.3 Ensembled VAE

Variational Autoencoders commonly known as VAE, describe an observation in latent space further in a probabilistic manner. It is different from conventional autoencoders since VAEs use a statistical approach to describe the samples from the dataset in the latent space. The architecture of the Variational Autoencoder consists of an encoder and a decoder. The encoder is fed the raw input and after getting computed the output generated through the bottleneck layer is in the form of a probability distribution. This output lets the VAE find a distribution associated with potential other representations. The decoder transforms the samples received from encoders back into data space and the desired reconstructed synthetic data is received. For the proposed paper, three variational encoders were used as a hybrid model named Ensembled VAE to achieve more accurate reconstructed output. Models are trained and optimized using mean squared error to keep track of losses and minimize them so that the efficiency of the model gets boosted during the training period. All three VAEs work independently to identify the underlying distribution and the result achieved from the respective models is concatenated and evaluated based on the statistical metrics. The purpose of assembling one or more VAE models is to achieve high-quality synthetic data and also increase the capability of data generation with minimal errors. Figure 10 describes the architecture of the Ensembled VAE.

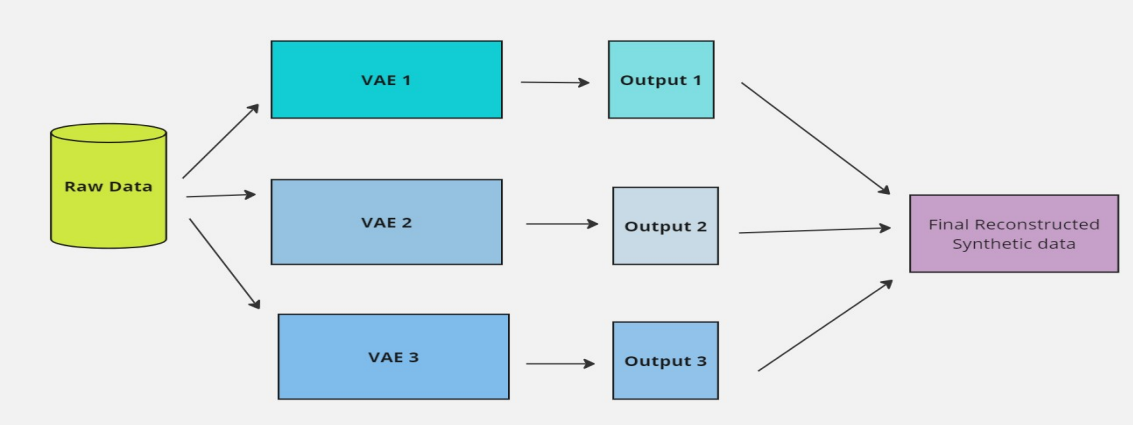

Figure 10 Architecture of Ensembled VAE

5.4 Hybrid Model made from GAN and VAE (VAEGAN).

Another approach for the automatic generation of synthetic was achieved by using a hybrid model that consists of the VAE and GAN deep learning models. The architecture of the VAE model used in this paper consists of an encoder and decoder. The raw data is fed to the encoder which compresses it into a latent space with a lower dimension than that of data space. This compressed data. The dimension of latent space is set to 2 and for maintaining stochasticity in the generated data an additional layer named the sampling layer is added to the model. The interpreted data received from the encoder is then fed to the decoder which reconstructs the data and transforms it back into data space, in its original form of representation. This VAE model is followed by GAN architecture, this model consists of a generator and discriminator of its own. The generator generates a synthetic sample when some random noise is given in it as input and the discriminator works on differentiating the training data (taken from the original dataset) and generated synthetic sample. For training the model, the adversarial method is used, and it is further compiled using the binary crossentropy loss function. An early callback function is used to stop the training of the models when there is no significant improvement in the validation loss. This is done to avoid overfitting and helps in retaining the best model weights. The data generated is a combined synthetic data generated from both models. The combined model consists best features from both models. The strong capability of VAE to identify the underlying distribution of data whereas GAN is known for generating realistic data. Moreover, both the loss functions used in the respective models that is mean squared error for VAE and binary crossentropy for GAN models were used to compile the combined hybrid model (VAEGAN) which provides more accuracy to the input that is used while training which leads to more accurate and realistic synthetic data generation. These models are trained independently, and the output received is integrated to achieve the final results. Figure 11 consists of the architecture of the proposed model.

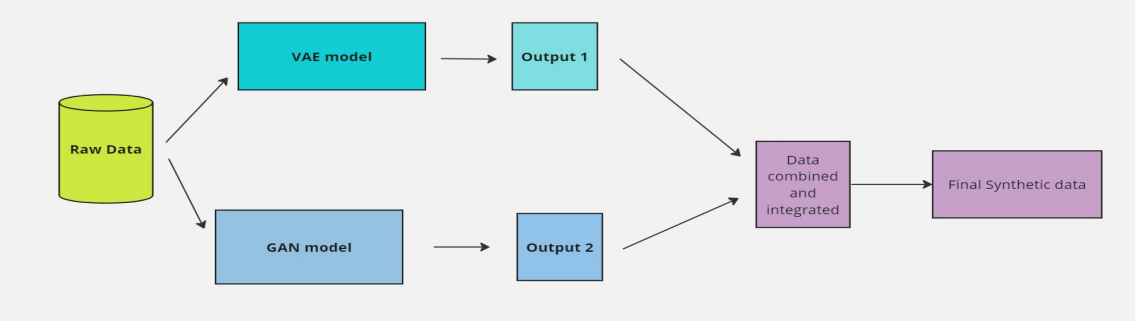

Figure 11 Architecture of Hybrid Model

6 Evaluation:

The mentioned section comprises the evaluation of the models implemented. The purpose of evaluating the models is to provide a descriptive analysis that has been carried out both statistically and academically. The evaluated results indicate the relevancy of the work done to achieve the purpose that was described and needed to be addressed. The performance of each model is evaluated by calculating statistical metrics such as mean squared error (MSE), mean absolute error (MAE), and root mean square error (RMSE). Visualization was done to compare the real data with the generated synthetic data.

6.1 Results Received from GAN Model.

The model was run for 30000 epochs and the number of synthetic samples generated after the training of the model was 5,00,00,000. Upon evaluation of the accuracy of the model was calculated statistically, the MSE, MAE, and RMSE value of the model was recorded as 1.073, 0.76, and RMSE as 1.036(shown in Figure12). The values of MSE and RMSE are slightly more than one which indicates an average performing model.

```
2/2 [==============================] - 0s 4ms/step
MSE: 1.0736464202263505, MAE: 0.7618403410393617, RMSE: 1.0361691079289859
```

Figure 12 Statistical Values for GAN

A histogram plot was plotted to show the distribution of the real data and the generated sample data to identify if synthetic data was able to mimic the real data. In Figure 13, the traces of synthetic data were visible overlapping the real data which shows that it was able to resemble a certain range of data since the number of samples generated is less than the data present in the dataset.

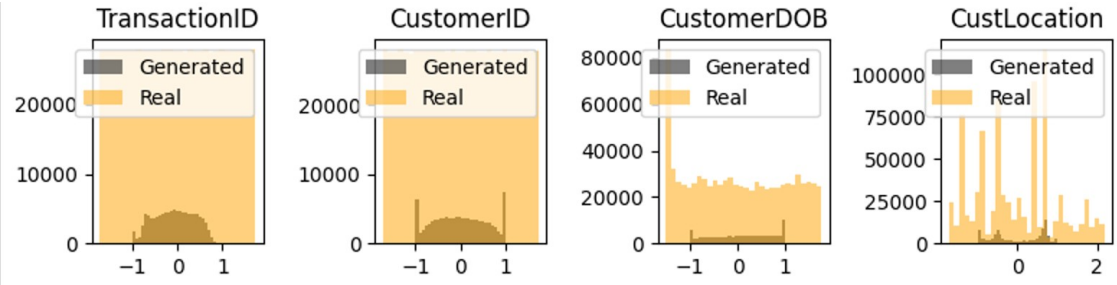

Figure 13 Real Data vs Synthetic Data (GAN)

6.2 Results evaluated for RNN Model.

The model was trained for only 50 epochs because of the high computational time. The number of sequences was set to 5 and the learning rate was set to 0.001. The performance of the model was evaluated using MSE and MAE which were 2.0034 (approx.) and 1.0087 (approx.) as shown in Figure 14. The statistical value indicates that the RNN model didn't perform as anticipated.

```
313/313 [==============================] - 1s 2ms/step
Mean Squared Error (MSE): 2.00039569640
Mean Absolute Error (MAE): 1.008746230943
```

Figure 14 Statistical value of RNN Model

6.3 Results Evaluated of Ensembled VAE

Each VAE model is trained for 100 epochs. The latent space dimension was set to 2 for each of the models. The learning rate was set to 0.0001. The computing results achieved from the model were MSE equal to 0.74 approx, MSE equal to 0.86 approx, and RMSE equal to 0.93 approx (Figure 15). Since all the statistical values are less than 1 it indicates that the model is a good fit for synthetic data generation.

```
Model Accuracy (Mean Squared Error): 0.8677
Model Accuracy (Mean Absolute Error): 0.7402
Model Accuracy (Root Mean Squared Error): 0.9315
```

Figure 15 Statistical value for Ensembled VAE Model

For visual representation histogram is used to depict the distribution of the synthetic data generated using the model, it can be seen that the data is close to normal but have some spikes on them which can be achieved while preprocessing or training of the data.

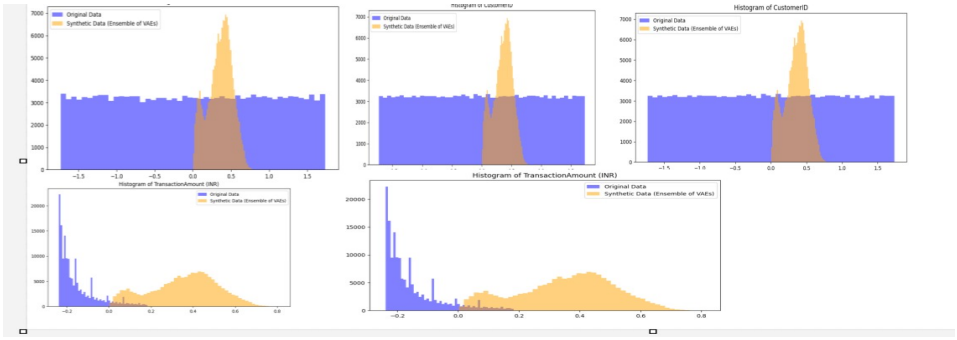

Figure 16 Synthetic data distribution

6.4 Result Evaluation of the VAEGAN:

This hybrid model was trained for 30 epochs. For variational Auto Encoder model was trained with latent space dimension set to 2. The model was compiled using a hybrid of two loss functions from each model named mean squared error and binary crossentropy loss function. The batch size was set to 64 and the validation split was set to 0.2. The generated data is further evaluated using histogram plots and the accuracy of the model was calculated using statistical values as follows- MSE received is 0.96 approximately, MAE is 0.6032 and RMSE is 0.977 approximately (Figure 16). The values of all the statistical measurements are less than 1 which depicts VAEGAN as a good fit for the synthetic data generation. The visualization for some of the features is carried out to understand the nature of the generated data as shown in Figure 17.

```
Model Accuracy (Mean Squared Error): 0.960348
Model Accuracy (Mean Absolute Error): 0.6032
Model Accuracy (Root Mean Squared Error): 0.979973
```

Figure 17 Statistical value of Hybrid Model (GAN + VAN)

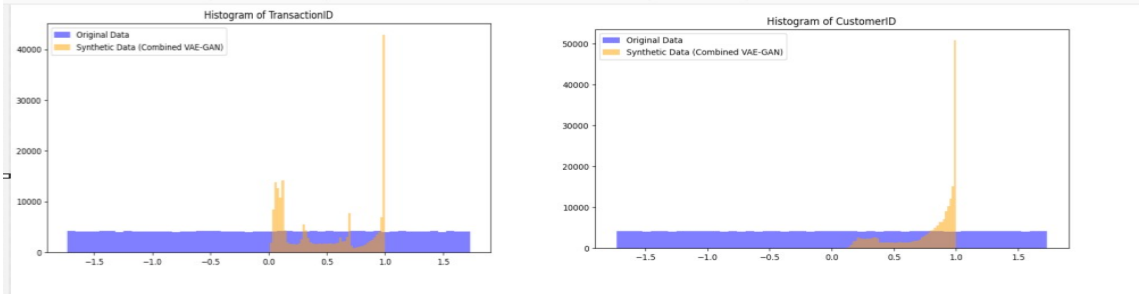

Figure 18 Real Data Vs Synthetic Data

6.5 Discussion

The conducted research aims to automatically generate synthetic data using various deep-learning models. For these four different models were created and the performance of these models is compared through different statistical values for evaluating the best-fit model which was the closest to meeting the objective of this research. The comparison of the values is mentioned in Table 2. The value of mean absolute error demonstrates the average absolute error that takes in between actual and predicted data, and the higher value of MAE depicts that the RNN model wasn't able to predict the data efficiently to generate synthetic data which would mimic the real data. The statistical observation of the Generative Adversarial network model shows that it is an average model for automatic synthetic data generation. The MAE value is 0.76 approx which is considered an average fit model. Since general models were lagging in providing the results as anticipated there can be many reasons for such performance (like hyperparameter tuning,etc.). Two new models were tried to be implemented to check their performance in delivering the results as required to meet the purpose of this project, they were Ensembled Variational Auto Encoders, and a VAEGAN (reference taken from research papers). On completion of the building of these models, the statistical measurements of both models were better than those received from the conventional deep learning models. The mean absolute error of both models is 0.74 and 0.60 and the value of MSE and RMSE of the former model is slightly less than the latter model .that is 0.86 and 0.93(approx. for Ensembled VAE) respectively which makes the latter model slightly better than the others. This signifies that the generated synthetic data was able to closely mimic the statistical measurements of the real transactional data.

Table: 2 Comparison of Accuracy of the Models

Models	MAE(approx.)	MSE(approx.)
GAN	0.7618	1.0736
RNN	1.008	2.003
Ensembled VAE	0.7402	0.86
Hybrid of VAE+GAN	0.6032	0.9603

7 Conclusion

To achieve efficient performance from banking applications maintaining data anonymity, security, and privacy needs to be efficiently tested. Testing of software is a necessary part of the industry and inadequate handling of data while testing can hinder the performance of the application/software which can lead to drastic measurements such as data breaches, frauds, and many more affecting both the organization as well as the clients/customers associated with it. Therefore, this project was carried out to create deep-learning models that will automate the synthesis of synthetic data which will help us to gain privacy, security, and a hassle-free testing environment. To achieve this purpose, four different deep-learning models

were implemented Generative Adversarial Network Model, Recurrent Neural Network, Ensembled variational autoencoder, and VAEGAN. Their performance was evaluated based on statistical measurements. The nature and quality of the data generated were evaluated using the visualization, histogram plots were plotted between real data (transaction data present in the dataset) and generated synthetic data. Through rigorous observations achieved while experimenting, the hybrid models that are ensemble VAE and a VAEGAN proved to be the best-performing model with MSE 0.86 and 0.96 respectively and MAE as 0.74 and 0.60 respectively (values are in approximation). Moreover, the computational time required from both models was comparatively less than that of GAN and RNN, and the epochs required for training the models were much less, 100 and 30 respectively which gave above-average statistical values that further notify the competency of the models. The computational time required was less than the conventional models. The observation chart makes the Generative Adversarial Network an average fit model since the epochs used for training this model was 30000 which took more than an hour to complete and still produced an average model. RNN is the most under-performed model in this case. The computational time taken by this model to run 5 epochs was around 3 hours and the results received were not satisfactory.

In conclusion, the new hybrid models Ensembled VAE and VAEGAN where the former is comprised of three VAE models and the latter is a combined model of GAN and VAE models, generated more efficient and robust synthetic data as compared to conventional GAN and RNN models. The hybrid data inherits the strengths of each of the models such as the ability of VAEs to capture intricate patterns and the ability of GANs to produce resembled synthetic data using adversarial training methods making these two models produce data of high efficiency and make it capable of producing data that resembles the statistical ability of the real data. Implementation of these models for test data generation in banking applications can be advantageous in many ways such as it will lead to increment of privacy and security protection. Data efficiency will be increased which will lead to efficient testing of the application leading to high and smooth performing applications.

8 Future Work.

There are areas of improvement that need to be addressed to get a more efficient model that will be able to exactly resemble the real data that has been provided to the model. The current model was implemented on a smaller scale of banking data (transaction), the model needs to be evaluated under larger and diverse datasets to evaluate the performance in more depth and achieve more efficient data. Also, more hyperparameter tuning can be done such as increasing the dense layers, manipulating the learning rate, and changing the dimension of latent space (in the case of VAE), increasing the number of epochs to understand the functionality of the models in more depth. These models should be implemented to predict data from other sectors of the Banking domain such as credit, loan, fraud detection, etc to evaluate its performance in these sectors. Implementation of more hybrid models using conventional deep learning and machine learning algorithms such as the Gausian Mixture

Model (machine learning model) and VAEs can be done to explore more such models with the best performance.

9 References

P. S. Patil and N. V. Dharwadkar, "Analysis of banking data using machine learning," 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 2017, pp. 876-881, doi: 10.1109/I-SMAC.2017.8058305.

P. V. R. Murthy and R. G. Shilpa, "Vulnerability Coverage Criteria for Security Testing of Web Applications," 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Bangalore, India, 2018, pp. 489-494, doi: 10.1109/ICACCI.2018.8554656.

Sen Chen, Lingling Fan, Guozhu Meng, Ting Su, Minhui Xue, Yinxing Xue, Yang Liu, and Lihua Xu. 2020. An empirical assessment of security risks of global Android banking apps. In Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering (ICSE '20). Association for Computing Machinery, New York, NY, USA, 1310–1322. https://doi.org/10.1145/3377811.3380417

C. Möckel and A. E. Abdallah, "Threat modeling approaches and tools for securing architectural designs of an e-banking application," *2010 Sixth International Conference on Information Assurance and Security*, Atlanta, GA, USA, 2010, pp. 149-154, doi: 10.1109/ISIAS.2010.5604049.

Akin, S. Sentürk and V. Garousi, "Transitioning from Manual to Automated Software Regression Testing: Experience from the Banking Domain," 2018 25th Asia-Pacific Software Engineering Conference (APSEC), Nara, Japan, 2018, pp. 591-597, doi: 10.1109/APSEC.2018.00074.

H. Ding, L. Cheng and Q. Li, "An Automatic Test Data Generation Method for Microservice Application," 2020 International Conference on Computer Engineering and Application (ICCEA), Guangzhou, China, 2020, pp. 188-191, doi: 10.1109/ICCEA50009.2020.00048

Y. Wang, Z. -D. Li and X. -Q. Xi, "Applications of Deep Learning in Unified Credit Management of Commercial Banks," 2020 12th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), Phuket, Thailand, 2020, pp. 703-707, doi: 10.1109/ICMTMA50254.2020.00155.

A. A. Ashlam, A. Badii and F. Stahl, "Multi-Phase Algorithmic Framework to Prevent SQL Injection Attacks using Improved Machine learning and Deep learning to Enhance Database security in Real-time," 2022 15th International Conference on Security of Information and Networks (SIN), Sousse, Tunisia, 2022, pp. 01-04, doi: 10.11

A. C. Pandey, M. G. (2019). Enhancing Text Mining Using Deep Learning Models. IEEE.

A. Sharma, M. P. (2023). Hybrid Framework for Generating Structured Data Synthetically whileMaintaining Referential Integrity. IEEE.

F. Romanelli and F. Martinelli, "Synthetic Sensor Data Generation Exploiting Deep Learning Techniques and Multimodal Information," in IEEE Sensors Letters, vol. 7, no. 7, pp. 1-4, July 2023, Art no. 7003404, doi: 10.1109/LSENS.2023.3290209.

M. A. Farooq and P. Corcoran, "Proof-of-Concept Techniques for Generating Synthetic Thermal Facial Data for Training of Deep Learning Models," 2021 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 2021, pp. 1-6, doi: 10.1109/ICCE50685.2021.9427690.

S. Shetty, A. V.S. and A. Mahale, "Data Augmentation vs. Synthetic Data Generation: An Empirical Evaluation for Enhancing Radiology Image Classification," 2023 IEEE 17th International Conference on Industrial and Information Systems (ICIIS), Peradeniya, Sri Lanka, 2023, pp. 1-6, doi: 10.1109/ICIIS58898.2023.10253599.

K. -H. Le Minh and K. -H. Le, "AirGen: GAN-based synthetic data generator for air monitoring in Smart City," 2021 IEEE 6th International Forum on Research and Technology for Society and Industry (RTSI), Naples, Italy, 2021, pp. 317-322, doi: 10.1109/RTSI50628.2021.9597364.

M. Cinquini, F. Giannotti and R. Guidotti, "Boosting Synthetic Data Generation with Effective Nonlinear Causal Discovery," 2021 IEEE Third International Conference on Cognitive Machine Intelligence (CogMI), Atlanta, GA, USA, 2021, pp. 54-63, doi: 10.1109/CogMI52975.2021.00016.

P. V. R. Murthy and R. G. Shilpa, "Vulnerability Coverage Criteria for Security Testing of Web Applications," 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Bangalore, India, 2018, pp. 489-494, doi: 10.1109/ICACCI.2018.8554656.

D. A. Rosa de Jesús, P. Mandal, T. Senjyu and S. Kamalasadan, "Unsupervised Hybrid Deep Generative Models for Photovoltaic Synthetic Data Generation," 2021 IEEE Power & Energy Society General Meeting (PESGM), Washington, DC, USA, 2021, pp. 1-5, doi: 10.1109/PESGM46819.2021.9637844.
R. S. Bhowmick, I. Ganguli and J. Sil, "Introduction and Correction of Bengali-Hindi Noise in Large Word Vocabulary using RNN," 2020 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 2020, pp. 277-281, doi: 10.1109/ICCSP48568.2020.9182244.

A. C. Pandey, M. Garg and S. Rajput, "Enhancing Text Mining Using Deep Learning Models," 2019 Twelfth International Conference on Contemporary Computing (IC3), Noida, India, 2019, pp. 1-5, doi: 10.1109/IC3.2019.8844895.

B. Hariharan, K. S, I. P. S, E. Nalina, W. B. N. R and P. N. Senthil Prakash, "Hybrid Deep Convolutional Generative Adversarial Networks (DCGANS) and Style Generative Adversarial Network (STYLEGANS) Algorithms to Improve Image Quality," 2022 3rd International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2022, pp. 1182-1186, doi: 10.1109/ICESC54411.2022.9885611.

M. A. Hossain, R. K. Chakrabortty, S. Elsawah and M. J. Ryan, "Hybrid Deep Learning Model for Ultra-Short-Term Wind Power Forecasting," 2020 IEEE International Conference on Applied Superconductivity and Electromagnetic Devices (ASEMD), Tianjin, China, 2020, pp. 1-2, doi: 10.1109/ASEMD49065.2020.9276090.

J. Chen and J. Zhao, "Synthetic Wind Speed Scenarios Generation using Artificial Neural Networks for Probabilistic Analysis of Hybrid Energy Systems," 2021 IEEE 30th International Symposium on Industrial Electronics (ISIE), Kyoto, Japan, 2021, pp. 1-6, doi: 10.1109/ISIE45552.2021.9576378.