PREDICTING RT-PCR TEST OUTCOMES WITH MACHINE LEARNING: GUIDED AUTOENCODER-BASED IMPUTATION AND CLINICAL UTILITY ASSESSMENT

PhD. Dissertation

Thierry MUGENZI

Eskişehir 2025

PREDICTING RT-PCR TEST OUTCOMES WITH MACHINE LEARNING: GUIDED AUTOENCODER-BASED IMPUTATION AND CLINICAL UTILITY ASSESSMENT

Thierry MUGENZI

PhD DISSERTATION

Department of Computer Engineering

Programme in Computer Science

Supervisor: Asst. Prof. Dr. Cahit PERKGÖZ

Eskişehir

Eskişehir Technical University

Institute of Graduate Programs

July 2025

FINAL APPROVAL FOR THESIS

This thesis titled PREDICTING RT-PCR TEST OUTCOMES WITH MACHINE LEARNING: GUIDED AUTOENCODER-BASED IMPUTATION AND CLINICAL UTILITY ASSESSMENT has been prepared and submitted by Thierry MUGENZI in partial fulfillment of the requirements in "Eskişehir Technical University Directive on Graduate Education and Examination" for the Degree of PhD in Computer Engineering Department has been examined and approved on 03/07/2025.

Committee Members	**Title, Name and Surname**	**Signature**
Member	Asst. Prof. Dr. Cahit PERKGÖZ	
Member	Prof. Dr. Kemal ÖZKAN	
Member	Prof. Dr. Ümmühan BAŞARAN FİLİK	
Member	Assoc. Prof. Dr. Ahmet ARSLAN	
Member	Assoc. Prof. Dr. Şahin IŞIK	

Prof. Dr. Harun BÖCÜK
Director of the Institute of Graduate Programs

SUPERVISOR APPROVAL

PhD student Thierry MUGENZI, whom I supervise, has completed this thesis titled PREDICTING RT-PCR TEST OUTCOMES WITH MACHINE LEARNING: GUIDED AUTOENCODER-BASED IMPUTATION AND CLINICAL UTILITY ASSESSMENT. According to my inspections, the work is scientifically and ethically appropriate for the student to the thesis defense exam.

Supervisor
Asst. Prof. Dr. Cahit PERKGÖZ

ABSTRACT

PREDICTING RT-PCR TEST OUTCOMES WITH MACHINE LEARNING: GUIDED AUTOENCODER-BASED IMPUTATION AND CLINICAL UTILITY ASSESSMENT

Thierry MUGENZI

Department of Computer Engineering

Programme in Computer Science

Eskişehir Technical University, Institute of Graduate Programs, July 2025

Supervisor: Asst. Prof. Dr. Cahit PERKGÖZ

Nowadays, machine learning (ML) is widely used in the healthcare industry for various purposes, including drug development, disease diagnosis and prediction, medical imaging analysis, and outbreak forecasting. Since its emergence in 2019, the COVID-19 pandemic has posed serious challenges to global health. The World Health Organization (WHO) reports that more than 6.9 million people have died from COVID-19, and there have been over 767 million confirmed cases worldwide. Additionally, 936 fatalities and more than 359,000 new cases were reported globally between May 12 and June 8, 2025. These figures underscore the burden on healthcare systems, especially in underdeveloped countries with limited funding. This thesis is organized into three main sections. In the first section, machine learning-based models were developed and evaluated using clinical data to predict the likelihood of active COVID-19 infection and recommend the optimal medical facility for treatment. Since the dataset contains missing data, several imputation techniques are employed. The second section of this study assesses multicollinearity and its impact on the effectiveness of imputation methods, ensuring the reliability of these predictive models. In the third section, an autoencoder-based model with noise-aware masked loss for imputing missing data in high-dimensional datasets is proposed. The proposed approach combines structured noise and a guided loss function to improve imputation accuracy and stability while preserving original data relationships. Collectively, these contributions aim to enhance predictive modeling and robust data completion in healthcare, supporting more trustworthy AI-driven decision-making for current and future public health challenges.

Keywords: One-class classification, Novelty detection, Missing data imputation, Multicolinearity, Autoencoders

ÖZET

MAKINE ÖĞRENMESI ILE RT-PCR TEST SONUÇLARININ TAHMINI: YÖNLENDIRILMIŞ OTOKODLAYICI TABANLI TAMAMLAMA VE KLINIK FAYDA DEĞERLENDIRMESI

Thierry MUGENZI

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Bilimleri Bilim Dalı

Eskişehir Teknik Üniversitesi, Lisansüstü Eğitim Enstitüsü, Temmuz 2025

Danışman: Dr. Öğr. Üyesi Cahit PERKGÖZ

Günümüzde makine öğrenmesi (ML), ilaç geliştirme, hastalık teşhis ve tahmini, tıbbi görüntü analizi ve salgın tahmini gibi çeşitli amaçlarla sağlık sektöründe yaygın olarak kullanılmaktadır. 2019 yılında ortaya çıkışından bu yana COVID-19 pandemisi, küresel sağlık açısından ciddi zorluklar yaratmıştır. Dünya Sağlık Örgütü (DSÖ) verilerine göre, COVID-19 nedeniyle 6,9 milyondan fazla kişi hayatını kaybetmiş, dünya genelinde 767 milyondan fazla doğrulanmış vaka rapor edilmiştir. Ayrıca, 12 Mayıs - 8 Haziran 2025 tarihleri arasında dünya genelinde 936 ölüm ve 359.000'den fazla yeni vaka bildirilmiştir. Bu tez üç ana bölümden oluşmaktadır. Birinci bölümde, klinik verileri kullanarak bir hastanın aktif COVID-19 enfeksiyonuna sahip olma olasılığını tahmin eden ve tedavi için en uygun sağlık kuruluşunu öneren makine öğrenmesi tabanlı modeller geliştirilmiş ve değerlendirilmiştir. Veri setinde eksik veriler bulunduğundan, ikinci bölümde çeşitli tamamlama (imputation) teknikleri uygulanmıştır. Üçüncü bölümde ise, yüksek boyutlu veri kümelerinde eksik verilerin tamamlanması için Gürültü Farkındalıklı Maskelenmiş Kayıp Fonksiyonuna sahip bir Otokodlayıcı tabanlı model önerilmektedir. Önerilen yaklaşım, yapılandırılmış gürültü ve yönlendirilmiş bir kayıp fonksiyonunu birleştirerek tamamlama doğruluğunu ve kararlılığını artırmakta, aynı zamanda orijinal veri ilişkilerini korumaktadır. Bu katkılar bir bütün olarak, sağlık alanında öngörü modellerini ve sağlam veri tamamlama süreçlerini geliştirmeyi, güncel ve gelecekteki halk sağlığı zorlukları için daha güvenilir yapay zekâ destekli karar verme mekanizmalarını desteklemeyi amaçlamaktadır.

Anahtar Sözcükler: Tek-sınıflı sınıflandırma, Yenilik tespiti, Eksik veri tamamlama, Çoklu doğrusal bağlantı, Otokodlayıcılar

ACKNOWLEDGEMENTS

I want to thank my thesis supervisor, Asst. Prof. Dr. Perkgöz Cahit, for his guidance and support. I'm also grateful to the faculty and staff at Eskisehir Technical University, my colleagues, classmates, and friends for their help and encouragement.

A special thanks to my family for their unwavering support and to my late mom, whose love and wisdom continue to inspire me. Lastly, I appreciate everyone who contributed to this thesis. Thank you all for being part of this journey.

Thierry MUGENZI

03/07/2025

STATEMENT OF COMPLIANCE WITH ETHICAL PRINCIPLES AND RULES

I hereby truthfully declare that this thesis is an original work prepared by me; that I have behaved in accordance with the scientific ethical principles and rules throughout the stages of preparation, data collection, analysis and presentation of my work; that I have cited the sources of all the data and information that could be obtained within the scope of this study, and included these sources in the references section; and that this study has been scanned for plagiarism with "scientific plagiarism detection program" used by Eskişehir Technical University, and that "it does not have any plagiarism" whatsoever. I also declare that, if a case contrary to my declaration is detected in my work at any time, I hereby express my consent to all the ethical and legal consequences that are involved.

Thierry MUGENZI

CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

GLOSSARY OF SYMBOLS AND ABBREVIATIONS

$\odot$	:	Element-wise (Hadamard) multiplication
1-SVM	:	One-Class Support Vector Machine
Adaboost	:	Adaptive Boosting
ADASYN	:	Adaptive Synthetic Sampling
AE	:	Autoencoder
AI	:	Artificial Intelligence
ANOVA	:	Analysis of Variance
CDC	:	Centers for Disease Control and Prevention
DAE	:	Denoising Autoencoder
DL	:	Deep Learning
DNN	:	Deep Neural Network
DT	:	Decision Tree
EM	:	Expectation-Maximization
EXT	:	Extra Trees
FN	:	False Negative
FP	:	False Positive
GAIN	:	Generative Adversarial Imputation Network
GB	:	Gradient Boosting
GNB	:	Gaussian Naive Bayes
GPC	:	Gaussian Process Classifier
HistGB	:	Histogram-Based Gradient Boosting
ICU	:	Intensive Care Unit
IF	:	Isolation Forest
KNN	:	K-Nearest Neighbors
LDA	:	Linear Discriminant Analysis
LGBM	:	Light Gradient Boosting Machine
LOF	:	Local Outlier Factor
MAR	:	Missing at Random
MCAR	:	Missing Completely at Random
MCD	:	Minimum Covariance Determinant
MI	:	Mutual Information

MICE	:	Multivariate Imputation by Chained Equations
MI-MFA	:	Multiple Imputation with Matrix Factorization Approach
ML	:	Machine Learning
MLP	:	Multi-Layer Perceptron
MNAR	:	Missing Not at Random
MNAR_QUANT	:	MNAR with Quantile Censorship
NAAT	:	Nucleic Acid Amplification Test
OBV	:	Observation
PCA	:	Principal Component Analysis
RF	:	Random Forest
RMSE	:	Root Mean Square Error
ROC	:	Receiver Operating Characteristic
RT-PCR	:	Reverse Transcription Polymerase Chain Reaction
COVID-19		Coronavirus Disease 2019
SARS-COV-2	:	Severe Acute Respiratory Syndrome Coronavirus 2
S-ICU	:	Semi-Intensive Care Unit
SOM	:	Self-Organizing Maps
SVM	:	Support Vector Machine
SVMSGD	:	Support Vector Machine with Stochastic Gradient Descent
SVR	:	Support Vector Regression
TN	:	True Negative
TP	:	True Positive
VAE	:	Variational Autoencoder
VIF	:	Variance Inflation Factor
WHO	:	World Health Organization
CDC		Centers for Disease Control and Prevention
XGBoost	:	Extreme Gradient Boosting

1. PREDICTING COVID-19 OUTCOMES AND OPTIMAL HEALTHCARE UNITS USING CLINICAL DATA AND MACHINE LEARNING

1.1. Introduction

The severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) is the cause of COVID-19 disease (World Health Organization, 2020). Multiple approaches have been developed by scientists to detect and diagnose the presence of the virus in the human body. The most common types of testing are diagnostic and molecular assays. The diagnostic test indicates that a patient has an active COVID-19 infection, whereas the molecular test detects the virus's genetic material. Molecular testing (NAAT) includes the reverse transcription polymerase chain reaction (RT-PCR) and nucleic acid amplification assays (CDC, 2021). The antigen test is a diagnostic technique that identifies specific viral proteins. The antibody or serology test detects antibodies produced by the immune system in response to a threat, such as a specific virus. However, it is not used to diagnose active infections (FDA, 2021). To test more people for COVID-19 in a short period, particularly in places where the majority of samples are predicted to be negative, laboratories use pooled sample testing, which entails pooling samples from multiple individuals into a single sample and evaluating them together (WHO, 2020). According to the World Health Organization (WHO), more than 767 million people have been infected with SARS-CoV-2 globally, and over 6.9 million deaths have been reported as of 2025 (WHO, 2025). This circumstance demonstrates that health care structures have limited capability, particularly in underdeveloped nations where the epidemic continues to spread. Coronavirus testing is expensive, especially for developing nations with limited finances (Lalmuanawma, Hussain, and Chhakchhuak, 2020). To assist in resolving this issue, nonclinical techniques, such as artificial intelligence (AI), should be encouraged and utilized. The use of machine learning (ML) and deep learning (DL) can improve the diagnosis and prognosis of COVID-19 disease. It is evident that this fact also applies to all other epidemic diseases. Several researchers worked on the application of machine learning to the diagnosis of COVID-19 and the use of readily accessible clinical and laboratory data to estimate mortality risk and severity. As Norah Alballa and Isra Al-Turaiki note, existing applications are hampered by the use of unbalanced data sets that are subject to bias (Alballa and Al-Turaiki, 2021). This study proposes the development and evaluation of machine learning-based models that integrate imbalanced class

handling and novelty detection techniques to predict, from clinical data, whether a patient has an active COVID-19 infection and to recommend the most appropriate healthcare unit for COVID-19-positive cases. The adopted strategy involves the development of supervised learning-based binary classification models and multiclass classification models.

1.2. Related Works

This section contains items published between 2020 and 2025. Norah Alballa and Isra Al-Turaiki evaluate current publications on machine learning techniques used to diagnose and predict COVID-19 mortality risk. Documenting and analyzing the 645 records published between January 2020 and October 2022. The authors acknowledged that the majority of the reported results were constrained by the current use of unbalanced class datasets, which could result in some bias. Li et al.(2020) created a classification model using the XGBoost algorithm to diagnose influenza and COVID-19 based on patient symptoms and routine test findings. A dataset including the information of 413 patients was utilized. The model achieved 92.5 percent sensitivity and 99 percent specificity. Significant criteria discovered include age, CT scan results, body temperature, lymphocytes, fever, and coughing. Kukar et al.(2021) used radio frequency (RF), deep neural networks (DNN), and XGBoost to construct models for predicting COVID-19 diagnosis. Using a dataset containing 5333 records, the mean corpuscular hemoglobin concentration (MCHC), eosinophil count, albumin, international normalized ratio (INR), and prothrombin activity percentage were determined to be significant characteristics. The model attained an area under the ROC curve (AUC) of 97%, a sensitivity of 81.9%, and a specificity of 97%. Levy et al.(2020) utilized the LASSO regression approach to build a 7-day survival calculator for COVID-19-positive patients. The model attained an AUC of 86%, while BUN, age, absolute neutrophil count, RDW, SpO2, and serum sodium were utilized as significant characteristics. Goodman-Meza et al.(2020) constructed an ensemble of seven standard machine learning methods for the diagnosis of COVID-19. Researchers applied Random Forest, Logistic Regression, SVM, multilayer perceptron, stochastic gradient descent, XGBoost, and AdaBoost algorithms to a dataset including 1455 records, 182 of which are COVID-19 positive. The ensemble learning-based model attained an AUC of 91%, a sensitivity of 93%, and a specificity of 64%. Important characteristics included inflammatory markers, particularly LDH, CRP,

and the combination of CRP, LDH, and ferritin. Zoabi et al.(2021) were able to obtain an AUC of 0.90 on eight clinical features, such as symptoms and age. Yan et al. (2020) proposed an interpretable XGBoost model that employed SHAP values to assess feature importance, and the obtained result showed that lactate dehydrogenase (LDH), lymphocyte count, and high-sensitivity C-reactive protein (hs-CRP) were the most significant predictors of patient survival. An AUC of 0.94 was obtained by researchers from their hybrid deep learning model that combined clinical metadata and CT imaging (Wang et al., 2021). A VGG19 + attention model for chest X-rays was more recently proposed by Basisi, Fadia, and Abdelbaki, achieved 98% accuracy in their quest to predict between COVID-19, pneumonia, and normal lungs (Basisi, Fadia, and Abdelbaki, 2024). In 2025, researchers at IIT Indore employed multi-omics machine learning to find severity biomarkers specific to variants in more than 3,000 patients. Fatima, Al-Amran, and Yousif (2023) used machine learning (ML) to identify at-risk individuals with high precision by detecting persistent inflammation in post-COVID patients. The aforementioned studies show an ongoing shift toward multimodal and explainable AI systems capable of leveraging imaging, clinical, and even omics data for robust COVID-19 screening and prognosis. Nevertheless, challenges such as data imbalance, overfitting, and lack of generalizability remain significant hurdles that must be addressed to ensure real-world adoption of these models.

1.3. Literature Review

1.3.1. Ensemble learning

Ensemble learning is the process of combining multiple machine learning-based models or weak learners' models into a single robust predictive model to make more accurate predictions. In other words, ensemble learning techniques mix multiple models to improve the model's stability and predictive capabilities. In many instances, individual base models perform poorly due to significant bias and or excessive variation, which results in overfitting or underfitting issues. The primary goal of ensemble learning approaches is to reduce bias and/or variation by integrating multiple weak learners into a single robust learner. This method improves predictive performance by reducing variation with bagging, bias with boosting, and improving predictions with stacking. Some models do well when attempting to model one component of the data, while others perform well when attempting to model another. The combination of multiple algorithms yields a

composite prediction with more accuracy than any of the individual models. An ensemble model can be created by combining many weak learners or by combining multiple strong, well-trained models (Zhang et al., 2021).

1.3.1.1. Voting

Voting is a strategy for ensemble learning that integrates prediction results from multiple models to improve the performance of the model. Regression and classification tasks are amenable to voting. For classification problems, voting involves taking into account the predictions of the majority of the used models, but for regression tasks, the predictions are the mean of the employed models. Soft voting and hard voting could be utilized within these ensemble learning methodologies. Soft voting is used to anticipate the likelihood of class label assignment. The following formula illustrates how probabilities associated with class labels are calculated (Zheng et al., 2025; Chen and Luc, 2023).

$$\hat{y} = argmax_{\mathrm{i}} \sum_{j=1}^{m} W_{ij} p_{ij} \tag{1.1}$$

Where; $\hat{y}$ is the final predicted class label, $argmax_{\mathrm{i}}$ means selecting the class i that maximizes the total weighted score across all classifiers,*m* is the total number of classifiers in the ensemble,W_{ij} is the weight reflecting the confidence or importance of classifier *j* in predicting class *i*,p_{ij} is the probability assigned by classifier *j* to class *i* for a given input and the sum $\sum_{j=1}^{m} W_{ij} p_{ij}$ represents the total weighted support that all classifiers give to class *i*.

The class label $\hat{y}$ is predicted via majority voting of each model mode.

$$\hat{\mathrm{y}} = mode\{C_1(x), C_2(x), \ldots, C_{\mathrm{m}}(x)\} \tag{1.2}$$

Where; $C_1(x), C_2(x), \ldots, C_{\mathrm{m}}(x)$ are the class predictions made by *m* different classifiers on input *x*; *mode{}* denotes that the mode function returns the most frequent class label among the predictions and $\hat{\mathrm{y}}$ represents the final predicted class label for input, Weighted majority voting could be computed as follows:

$$\hat{\mathrm{y}} = argmax_{\mathrm{i}} \sum_{\mathrm{j}}^{=1\mathrm{m}} w_{\mathrm{j}} \cdot \chi_{\mathrm{a}}(C_{\mathrm{j}}(x) = i) \tag{1.3}$$

Where; ŷ represents the final predicted class label, $argmax_{\mathrm{i}}$ denotes the class label i that receives the highest total weighted vote, $\sum_{\mathrm{j}}^{=1\mathrm{m}}$ denotes the summation over all classifiers $j = 1$ to m; w_{j} denotes the weight assigned to the *j-th* classifier (reflects its importance or accuracy); $C_{\mathrm{j}}(x)$ represents the prediction of the j-th classifier for input x and $\chi_{\mathrm{a}}(C_{\mathrm{j}}(x) = i)$ denotes the Indicator function that returns *1* if classifier *j* predicts class i, and *0* otherwise.

1.3.1.2. Bagging and bootstrap aggregation

Bagging is an ensemble learning technique that consists of using various homogeneous weak learners, training them independently in parallel, and combining them using a deterministic averaging process. Bootstrapping is a statistical technique consisting of generating samples of a certain size from an initial dataset of a certain size by randomly drawing replacement observations. For these techniques, the original dataset should be large enough to prevent sample correlation (Soloff, Barber, and Willett, 2024).

1.3.1.3. Boosting

The boosting approach consists of fitting several weak learners sequentially in a very adaptive manner, where each model in the sequence focuses more on the observations in the dataset that have been wrongly handled by the earlier models in the sequence. This approach aims to reduce bias. The boosting method works well for models with low variance but high bias, since weak learners are in a situation where they do not fit the model well enough (Chen and Guestrin, 2016).

1.3.1.4. Stacking

The main idea behind stacking techniques is to use predictions of multiple diverse algorithms as features for training a high-level model known as a meta-classifier. Stacking typically involves heterogeneous base learners, in contrast to bagging and boosting, which generally rely on homogeneous base learners. Multi-level stacking is an extension of this approach where multiple hierarchical layers of stacking are utilized. Multi-level stacking refers to more than one layer of stacking. Certain meta-models are trained using the outputs of meta-models from lower layers, and this hierarchical training process proceeds through successive layers.

1.3.2. Imbalanced class handling

Imbalanced data refers to datasets where the target class has an uneven distribution of observations, with one class label having a significantly higher number of observations and the other having a significantly lower number of observations. The main issue with imbalanced datasets is predicting both the majority and minority classes accurately. In this situation, the classifier may become biased towards the prediction. Various techniques to handle imbalanced class problems do exist. In this study, oversampling and a combination of oversampling and undersampling approaches will be adopted. Oversampling consists of duplicating samples from the minority class, and undersampling consists of deleting samples from the majority class in order to fix and prevent problems related to skewed class distribution (Carvalho, Pinho and Brás, 2025).

1.3.2.1. Synthetic minority oversampling techniques (SMOTE)

The synthetic minority oversampling technique is an oversampling technique that involves choosing samples that are close to one another in feature space. A randomly selected example from the minority class is first picked. Then, for that example, k of the closest neighbors is selected (by default, K = 5). A synthetic example is generated in feature space at a random position between two instances and their randomly selected neighbors. Using only samples from the minority class that were incorrectly classified, borderline SMOTE is an extension of SMOTE. The identification of the samples that were incorrectly classified is done using the K-nearest neighbor technique. The SVM method is used in a different variation of Borderline SMOTE. (Sakho, Scornet and Malherbe, 2024).

1.3.2.2. Adaptive synthetic sampling (ADASYN)

Generating a suitable number of synthetic alternatives for each observation belonging to the minority class is the core pillar of ADASYN. The definition of an "acceptable number" in this context is dependent on how complex it is to learn the initial observation. An observation from the minority class is considered "hard to learn" if there are several examples from the majority class that share the observation's characteristics (He et al., 2008).

1.3.3. Principal components analysis (PCA)

Principal component analysis is a dimensionality reduction technique that involves linearly transforming data to a lower-dimensional space to maximize the variance of the data there. In actuality, the data's covariance (and occasionally correlation) matrix is built, and the matrix's eigenvectors are calculated. Using the principal components, which are the eigenvectors that go with the largest eigenvalues, it is possible to recreate a big part of the original data's variation (Jolliffe and Cadima, 2016).

1.3.4. One-class classification algorithm

Rare instances that do not fit in with the rest of the data are known as outliers or anomalies. While novelty detection is a semi-supervised anomaly detection algorithm, outlier detection is an unsupervised anomaly detection algorithm. Anomalies in a particular dataset can be discovered using either of the two methods. These algorithms use unsupervised learning to categorize new inputs as normal or abnormal by modeling "normal" samples. One-class classification can be used successfully in the case of binary classification with unbalanced classes. The aforementioned strategies can be tested on a holdout test dataset after being fitted to input instances from the majority class in the training dataset (Ruff, 2021).

1.3.4.1. One-class support vector machine (1-SVM)

One-class SVM is an unsupervised learning method for developing the capacity to distinguish test samples from one class from test samples from other classes. One of the easiest ways to handle one-class classification (OCC) problem statements, including anomaly detection (AD), is with 1-SVM. The fundamental principle behind 1-SVM is to minimize the hypersphere of the single class of examples in the training data and to consider all other samples outside of the hypersphere to be outliers or outside the distribution of the training data. One-Class SVM with Stochastic Gradient Descent (1SVM SGD) is a version of One-Class SVM. Whereas One-Class SVM employs a Gaussian kernel by default, this version employs stochastic gradient descent with a kernel approximation approach (Schölkopf et al., 2001).

1.3.4.2. Local outlier factor(LOF)

By calculating the local density deviation of a given data point in relation to the data points nearby, the local outlier factor algorithm generates an anomaly score that indicates data points that are outliers in the data collection. By calculating the distances between nearby data points, local density is calculated (k-nearest neighbors). Thus, local density may be computed for each data point. By contrasting these, it is possible to determine which data points have densities that are comparable to their neighbors and which have lower densities. Outliers are those having densities that are below average. To identify the k-nearest neighbors of each point, k-distances, or distances between points, are first determined. The second nearest neighbor to the point is referred to as the 2nd closest point (Zhang et al., 2024).

1.3.4.3. Isolation forest(IF)

Isolation Forests is an unsupervised model, and it was developed using decision trees. Isolation Forests were created with the assumption that anomalies are represented by data points that are "few and unusual." In Isolation Forest, data is subsampled at random and processed using randomly selected features in a tree structure. Anomalies are less likely to be present in samples that move deeper into the tree since they need more splitting to separate. Similar to how shorter branches highlight anomalies, shorter branches indicate samples that were simpler for the tree to identify from other observations (Hariri, Kind and Brunner, 2021).

1.3.4.4. Minimum covariance determinant (MCD)

To determine which data points in a dataset have the fewest determinants in their covariance matrix, the Minimum Covariance Determinant method is used. Data points' normality or outlier status is determined by the covariance matrix, a distribution parameter. This technique can be useful as a rapid and efficient way to discover anomalies when the data points are distributed using a Gaussian distribution. This method can be generalized by constructing an ellipsoid (hypersphere) that contains the typical data; data that deviates from this shape is viewed as an outlier (Raymaekers and Rousseeuw, 2022).

1.3.5. Feature scaling

In machine learning, feature scaling is one of the most important procedures in data preparation prior to developing a machine learning model. Scaling can be the deciding factor between a weak machine learning model and a stronger one. Normalization and standardization are the most prevalent strategies for scaling features. In case data needs to be rescaled to a uniform range, commonly between [0,1] or [-1,1], normalization techniques are typically applied.

1.4. Exploratory Data Analysis

Exploring data is crucial to machine learning. To establish a modeling strategy, exploratory data analysis aims to comprehend the data through patterns and anomalies, test hypotheses, and validate assumptions using statistics and graphical representations (Da Poian et al., 2023). The dataset used in this study consists of 111 columns and 5644 samples with 71 continuous and 39 categorical nominal variables. The dataset presents a significant number of missing values, with more than 50% of the features having over 90% missing values. Two main groups of variables are present: variables related to blood tests and variables related to respiratory viruses. The SARS-CoV-2 exam result was selected as the target variable to address the binary classification problem stated above, and "hospital_admission" for the aforementioned multiclass classification issue. The dataset and class distribution are illustrated in Figure 1.1 below.

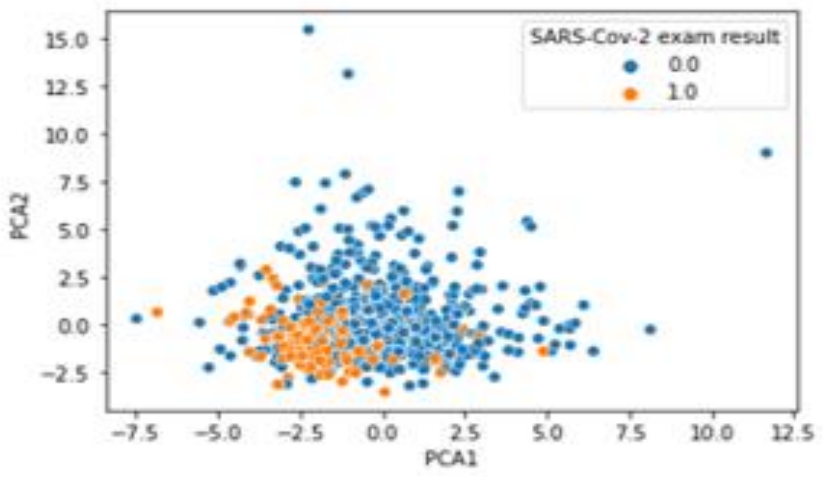


Figure 1.1 *Data visualization-target variable*

As a dimensionality reduction technique, principal component analysis has been used. Two features, PCA1 and PCA2, were generated, allowing us to represent the data. There are two classes observed, with the class label 0 representing patients with COVID-19 negative results and the class label 1 representing patients with COVID-19 positive results. The class distribution is highly imbalanced, as shown in the figure above, with 5086 patients being COVID-19 negative and only 558 being COVID-19 positive. In other words, 90% of patients are COVID-19 negative, while nearly 10% are COVID-19 positive.

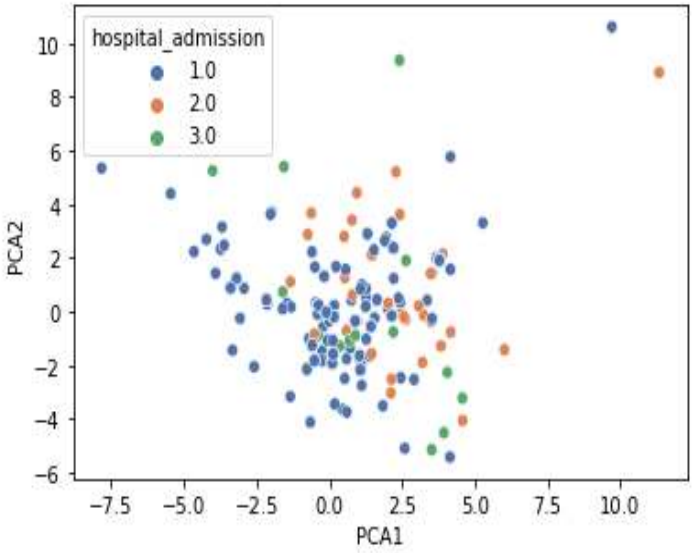


Figure 1.2 *Data visualization- hospital admission*

Figure 1.2 above depicts the class distribution of COVID-19 positive patients when only COVID-19 positive patients are taken into account. The 558 patients are divided among three hospital units: the surveillance unit (36 patients) and the intensive care unit (16 patients). The majority of the patients (506) are in unidentified departments. As seen in the figure above, three class labels are present: 1 for the unknown unit department, 2 for the surveillance unit, and 3 for the intensive care unit. Principal component analysis was used to generate two variables, PCA1 and PCA2, to represent the data. The class distribution is highly skewed, as shown in Figure 1.3 below.

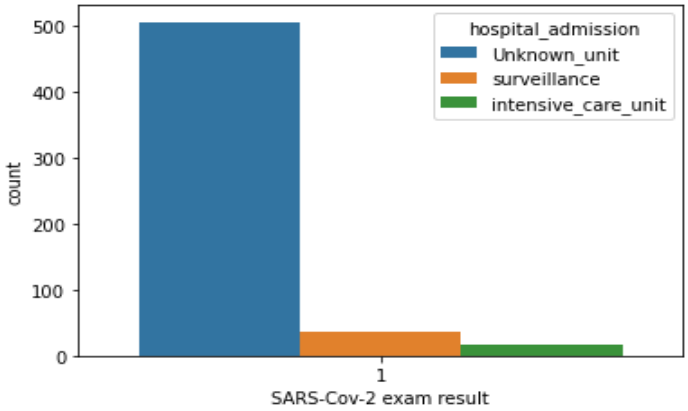


Figure 1.3 *Hospital admission class visualization*

After analysis, it was discovered that the characteristics of "Monocytes, Platelets, Leukocytes, Basophils, and Eosinophils" could be linked to COVID-19 because the rates of COVID-19-positive and negative patients differ. It has also been observed that young patients are less likely to be hospitalized due to COVID-19. The relationship between the age of the patients and the target variables revealed that all patients, regardless of age, can be infected by COVID-19, but that young patients are less hospitalized due to COVID-19, as shown below in Figure 1.4.

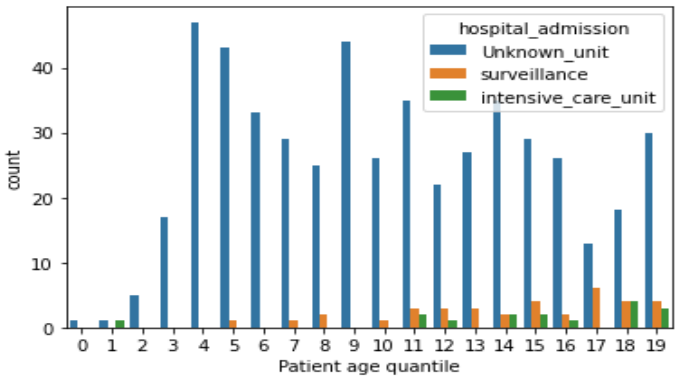


Figure 1.4: *Patient age - hospital admission visualization*

The correlation analysis revealed that some blood-related characteristics are highly correlated. The analysis results show that there is a low correlation between the patients' age and the features of their blood, as well as a low correlation between virus features. This exploratory data analysis led us to formulate and test two hypotheses. First hypothesis: H0=Blood Rate Mean does not differ significantly between COVID-19 positive and negative patients. Second hypothesis: H0=Blood Rate Mean does not differ

considerably between hospitalized and non-hospitalized patients. A T-test has been used to test each hypothesis. The first hypothesis testing results revealed a difference in "Hemoglobin, Platelets, Mean Platelet Volume, Leukocytes, Monocytes, and Eosinophils" rates among COVID-19 positive and negative patients. The second hypothesis testing results showed that the rate of lymphocytes was significantly different among COVID-19 hospitalized patients in different units. The T-Test outcome is shown below in Figure 1.5.

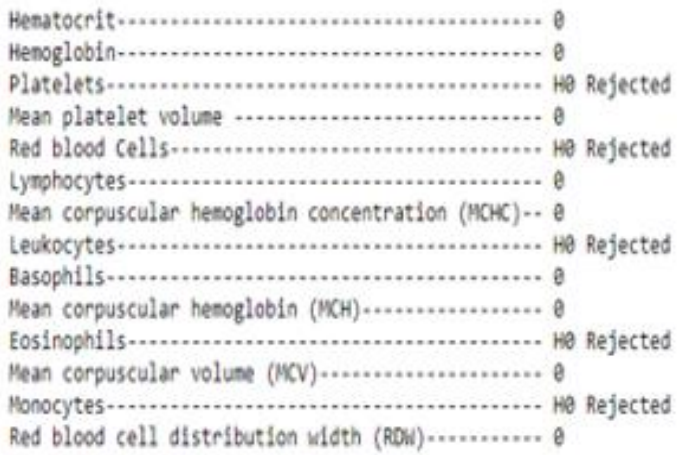


Figure 1.5 *Result of the T-test*

1.5. Predicting COVID-19 Based On Clinical Data

This phase consists of four steps, which are pre-processing, feature selection, modelling, and model evaluation.

1.5.1. Preprocessing (1)

The process of converting raw data into a structured and usable format is known as data preprocessing. Before using machine learning or data mining algorithms, the data quality must be checked. The goal of this step is to clean and transform the dataset before applying machine learning algorithms. This phase includes tasks for feature engineering, feature selection, data cleaning, and feature scaling. In the feature engineering task, categorical variables were encoded and transformed into numerical values, and the task of handling missing values was performed. Features with more than 90% missing values were removed, leaving 39 features. k-Nearest Neighbors with 5 neighbors was used to impute missing values. To create the second target variable, three features were combined and transformed into one. The features "patient admitted to regular ward (1=yes, 0=no),

"patient admitted to semi-intensive unit (1=yes, 0=no)", and "patient admitted to intensive care unit (1=yes, 0=no)" have been transformed into "hospital admission". During this stage, nominal variables were effectively encoded, and continuous features were normalized. The implementation of the label/ordinary encoding strategy resulted in the encoding of the class labels of the two aforementioned target variables. The implementation of the label/ordinary encoding strategy resulted in the encoding of the class labels of the two aforementioned target variables. For the first target variable, "SARS-CoV-2 exam result," positive has been encoded as 1 and negative as 0. "Hospital admission" is encoded with three class labels: 1 for the unknown unit, 2 for the surveillance unit, and 3 for the intensive care unit. Even though target variables were being encoded, other nominal variables were also being encoded. "detected" was encoded as 1, and "not detected" was encoded the same way. Data standardization was preferred over normalization to minimize the risk of inaccuracy of the model and maintain any pertinent information related to outliers. The StandardScaler methods were used rather than the MinMax scaler approaches.

1.5.2. Feature selection (1)

Feature selection minimizes the number of input variables, lowering processing costs and improving model performance. The variance threshold and mutual information features were chosen in this study. The variance threshold filters out features with low variance. Mutual information (MI) quantifies the link between two random variables. It is 0 when two random variables are independent; higher values indicate dependency. The SelectKbest algorithm chose features that had the greatest mutual information score between the target and the rest. During this step, the patient's age quantile, platelets, leukocytes, eosinophils, and monocytes were chosen.

1.5.3. Modeling (1)

During the modeling phases, several binary classification methods for COVID-19 diagnosis are investigated. These algorithms used ordinary clinical data as training and testing data. Precision, recall, F-1 score, and accuracy have been used as evaluation metrics due to the fact that the target classes present an imbalanced class distribution. The following formulas illustrate the evaluation metrics described above.

$$Precision = \frac{TP}{TP + FN} \tag{1.4}$$

$$Recall = \frac{TP}{TP + FP} \tag{1.5}$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{1.6}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1.7}$$

Where TP = True positive; FP = False positive; TN = True negative, and FN = False negative.

To begin, single models based on clinical data have been used to predict whether a patient is COVID-19 positive or negative. The algorithms used in this study are Random Forest, Adaptive Boosting (Adaboost), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Histogram Gradient Boosting, Gradient Boosting, Logistic Regression, Decision Tree, Extreme Gradient Boosting (Xgboost), and a voting classifier technique. The Extreme Gradient Boosting algorithm surpassed other algorithms with scores of 74% for precision, 73% for recall, 74% for F1 score, and 88% for accuracy. It has been noticed that single models perform badly due to the highly uneven target class. The resulting outcome is displayed below in Table 1.1 and Figure 1.6.

Table 1.1 *Results-single models experiment*

Models	Precision	Recall	F1-score	Accuracy
Random Forest	0.69	0.67	0.68	0.85
AdaBoost	0.7	0.62	0.65	0.86
SVM	0.69	0.67	0.68	0.85
KNN	0.66	0.59	0.61	0.85
HistGB	0.74	0.73	0.74	0.88
LR	0.6	0.54	0.54	0.84
GB	0.65	0.65	0.65	0.83
DT	0.66	0.68	0.67	0.83
XGBoost	0.69	0.67	0.68	0.85
Voting	0.7	0.65	0.67	0.86

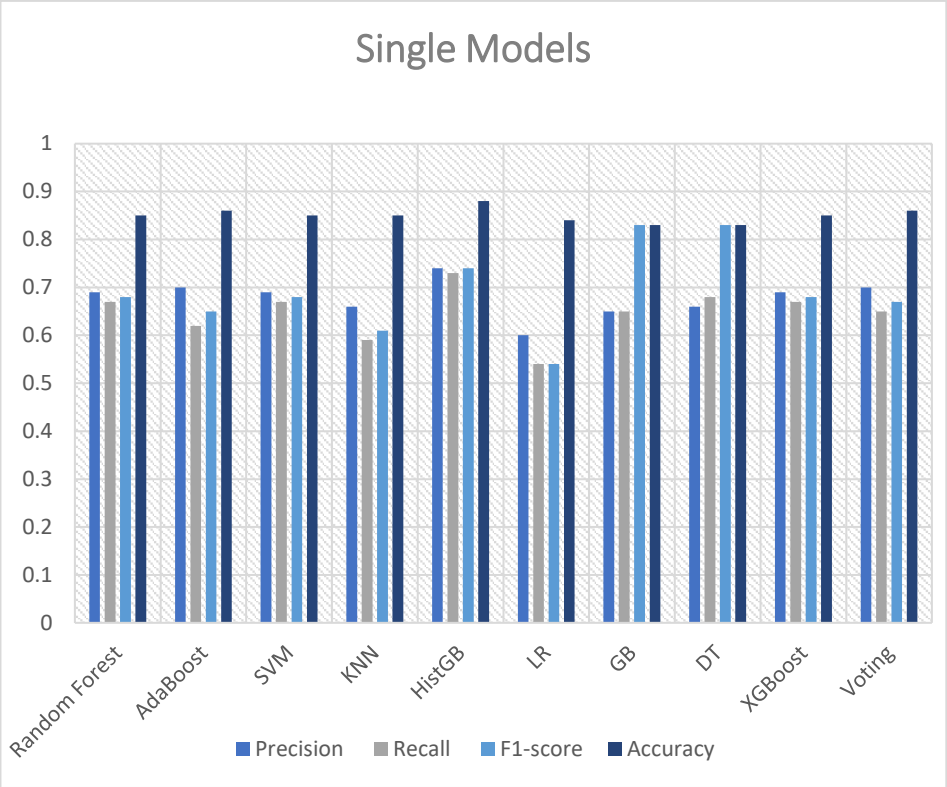


Figure 1.6 *Results single models experiment*

Several oversampling strategies were utilized to improve the performance of the models. Experiments on SMOTE, ADASYN, BorderlineSMOTE, and Random Oversampler have been conducted. As seen in the tables below, the acquired results indicate that the combination of oversampling approaches with single models led to improved results. The recall and F1 scores both improved significantly. With SMOTE oversampling, the Random Forest method outperformed the other algorithms with scores of 70% for precision, 83% for recall, 73% for F1, and 83% for accuracy. The Random Forest and the voting algorithm surpassed the other algorithms when using the ADASYN oversampler, achieving a precision score of 70%, a recall score of 83%, an F1 score of 73%, and an accuracy score of 83%. BorderlineSMOTE oversampling, Random Forest, Histogram Gradient Boosting, and the voting algorithm outperformed other algorithms when utilizing BorderlineSMOTE oversampling. Random Forest, Adaboost, and the voting algorithm outperformed other algorithms when random oversampling was employed. The obtained results are displayed in Tables 1.2-1.6 and in Figures 1.7-1.10.

Table 1.2 *Results SMOTE oversampler*

Models	Precision	Recall	F1-score	Accuracy
Random Forest	0.7	0.83	0.73	0.83
AdaBoost	0.69	0.8	0.71	0.82
SVM	0.65	0.77	0.67	0.77
KNN	0.67	0.78	0.69	0.79
HistGB	0.7	0.79	0.73	0.84
LR	0.68	0.81	0.7	0.79
GB	0.68	0.77	0.7	0.82
DT	0.67	0.76	0.69	0.8
XGB	0.66	0.72	0.68	0.82
Voting	0.71	0.81	0.74	0.84

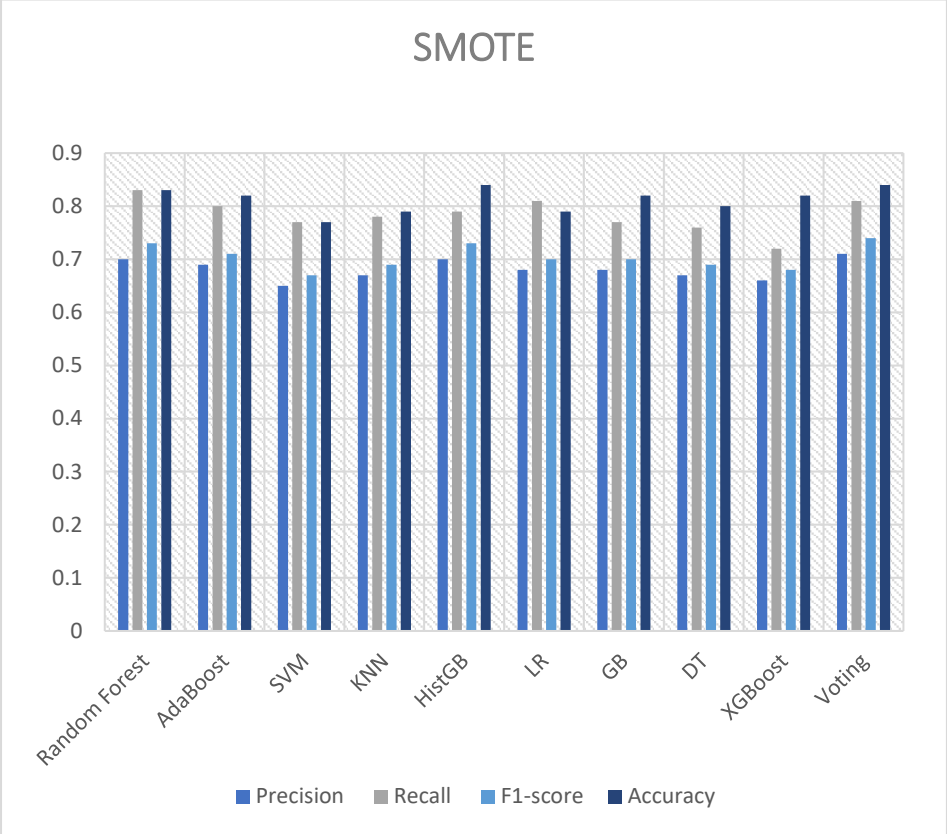


Figure 1.7 *Results SMOTE oversampling Experiment*

Table 1.3 *Results ADASYN oversampler*

Models	Precision	Recall	F1-score	Accuracy
Random Forest	0.7	0.83	0.73	0.83
AdaBoost	0.65	0.75	0.66	0.78
SVM	0.66	0.79	0.67	0.76
KNN	0.66	0.77	0.67	0.78
HistGB	0.67	0.75	0.69	0.82
LR	0.66	0.79	0.67	0.76
GB	0.63	0.71	0.64	0.76
DT	0.63	0.71	0.64	0.76
XGBoost	0.72	0.82	0.75	0.85
Voting	0.7	0.83	0.73	0.83

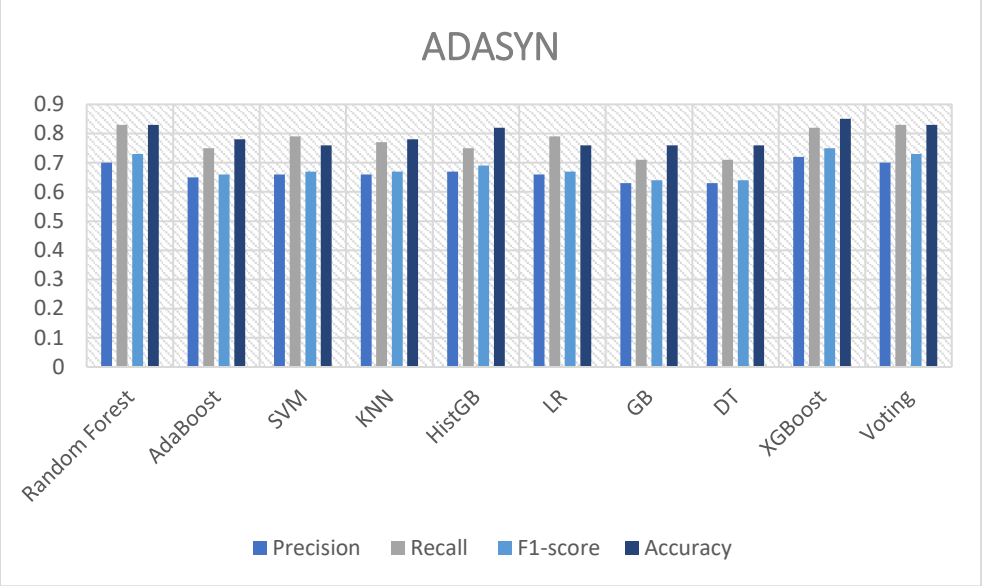


Figure 1.8 *Results ADASYN oversampler experiment*

Table 1.4 *Results borderline SMOTE oversampler*

Models	Precision	Recall	F1-score	Accuracy
Random Forest	0.7	0.83	0.73	0.83
AdaBoost	0.69	0.8	0.71	0.82
SVM	0.65	0.77	0.67	0.77
KNN	0.67	0.78	0.69	0.79
HistGB	0.7	0.79	0.73	0.84
LR	0.68	0.81	0.7	0.79
GB	0.68	0.77	0.7	0.82
DT	0.67	0.76	0.69	0.8
XGBoost	0.66	0.72	0.68	0.82
Voting	0.71	0.81	0.74	0.84

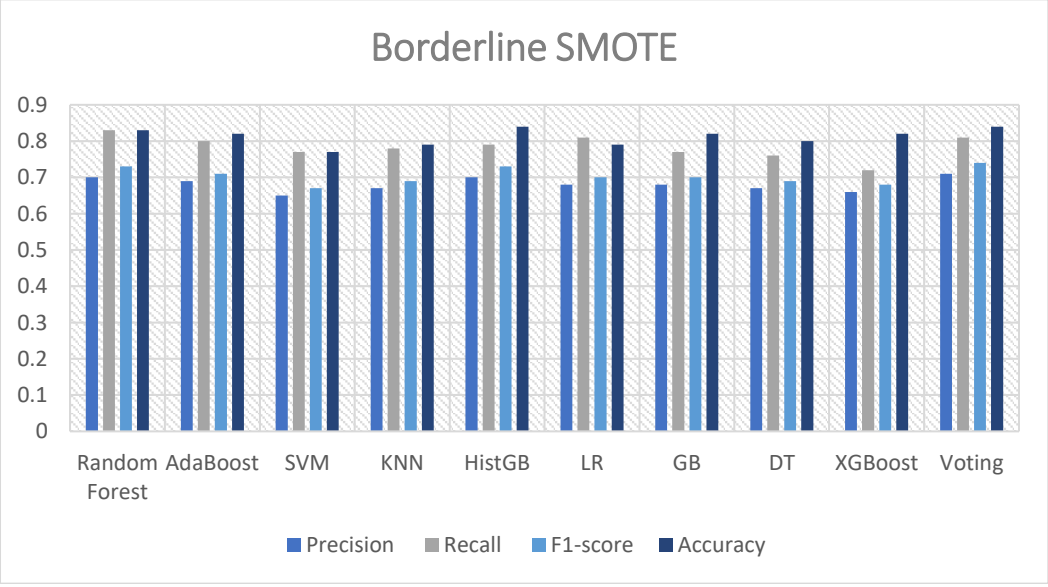


Figure 1.9 *Results borderline SMOTE oversampler*

Table 1.5 *Results random oversampler*

Models	Precision	Recall	F1-score	Accuracy
Random Forest	0.70	0.83	0.73	0.83
AdaBoost	0.69	0.8	0.73	0.82
SVM	0.65	0.77	0.67	0.77
KNN	0.67	0.78	0.68	0.79
HistGB	0.70	0.79	0.73	0.84
LR	0.68	0.81	0.70	0.79
GB	0.68	0.77	0.70	0.82
DT	0.65	0.73	0.67	0.79
XGBoost	0.66	0.72	0.68	0.82
Voting	0.71	0.81	0.74	0.84

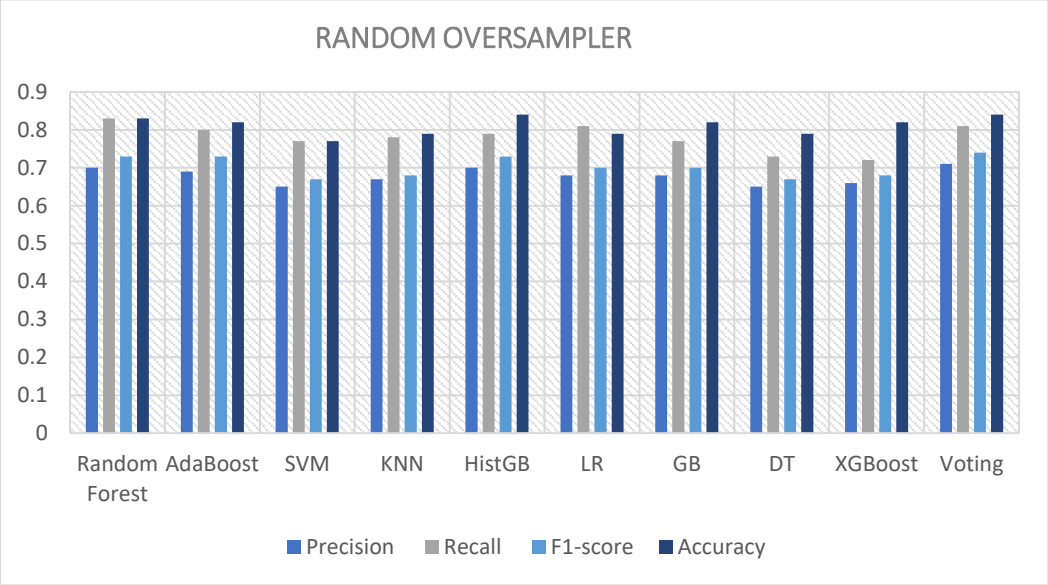


Figure 1.10 *Results random oversampler*

Techniques involving bagging have also been tested experimentally. A balanced bagging algorithm has been applied to various simple estimators. A better outcome was achieved using this method, as evidenced by an increase in the recall score. The results are depicted in Table 1.6 and Figure 1.11 below.

Table 1.6 *Results obtained from balanced Bagging modeling experiment*

Models	Precision	Recall	F1-score	Accuracy
Random Forest	0.72	0.85	0.75	0.83
AdaBoost	0.7	0.85	0.73	0.82
SVM	0.67	0.82	0.68	0.77
KNN	0.66	0.81	0.67	0.75
HistGB	0.72	0.85	0.75	0.83
LR	0.68	0.82	0.69	0.78
GB	0.72	0.85	0.75	0.83
DT	0.72	0.86	0.76	0.84
XGBoost	0.71	0.85	0.76	0.83

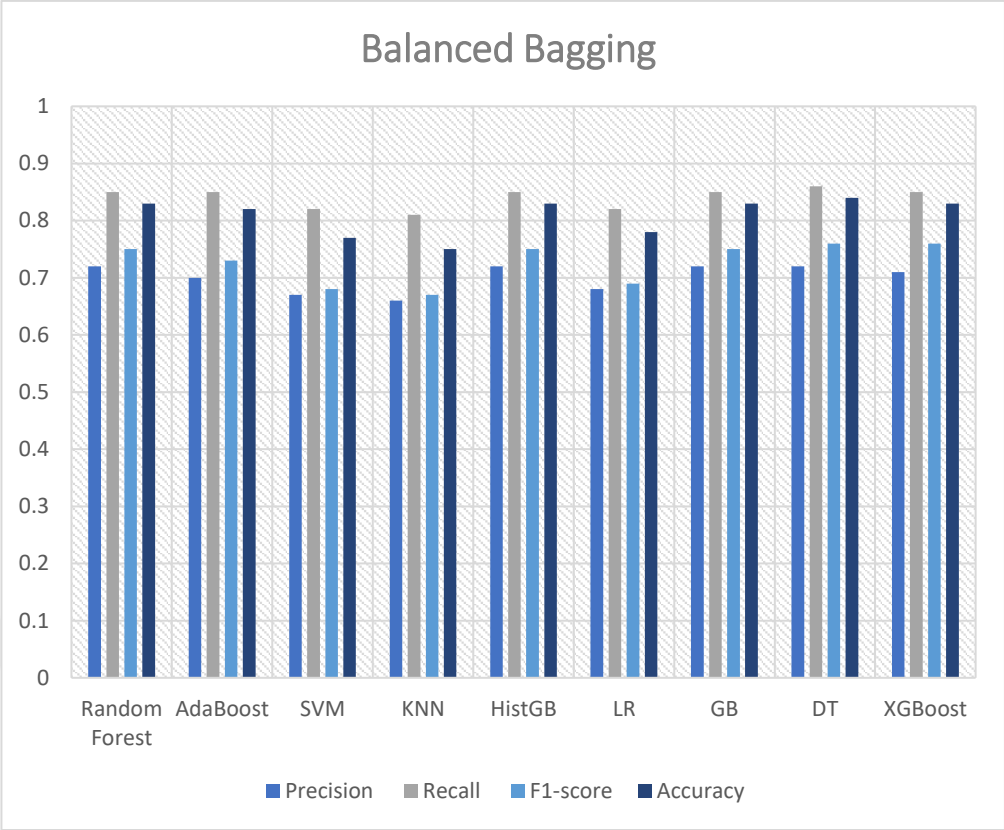


Figure 1.11 *Results balanced bagging modeling*

Stacking techniques have also been utilized to stack several models. Level 1, Level 2, and Level 3 stacking experiments have been conducted. Other stacking models were outperformed by the model built with Random Forest, Adaboost, and XGboost as the base learner and SVM as the meta-classifier algorithm. In contrast to Level-1 stacking, models performed poorly. The obtained results are displayed below in Table 1.7-1.9. The result is also shown in Figure 1.12-1.14.

Table 1.7 *Results level 1 stacking modeling experiment*

Models	Precision	Recall	F1-score	Accuracy
model_1	0.61	0.65	0.63	0.79
model_2	0.75	0.66	0.69	0.88
model_3	0.69	0.55	0.56	0.86
model_4	0.74	0.71	0.72	0.88
model_5	0.57	0.57	0.57	0.78

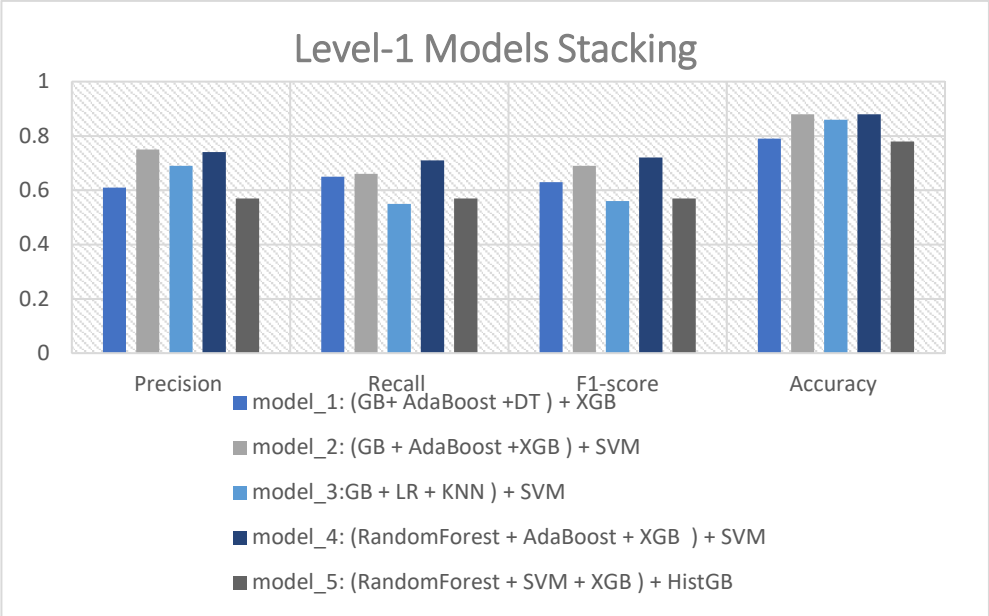


Figure 1.12 *Results level 1 stacking modeling experiment*

Table 1.8 *Results level 2 stacking modeling experiment*

Models	Precision	Recall	F1-score	Accuracy
stacking	0.69	0.57	0.59	0.86
stacking1	0.58	0.53	0.54	0.83
stacking2	0.59	0.55	0.56	0.83

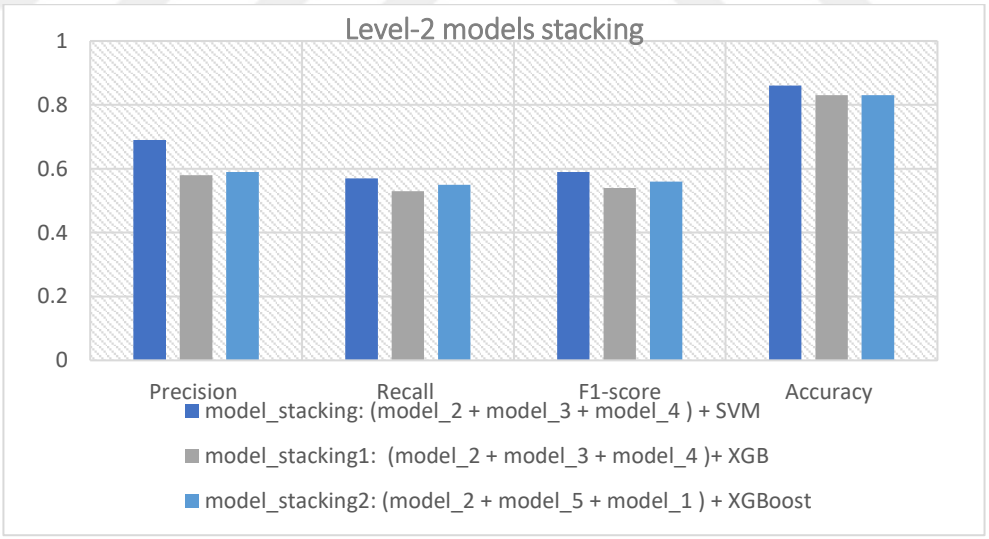


Figure 1.13 *Results level 2 stacking modeling experiment*

Table 1.9 *Results from the level 3 stacking modeling experiment*

Models	Precision	Recall	F1-score	Accuracy
stacking	0.74	0.58	0.6	0.87
stacking1	0.78	0.61	0.64	0.88

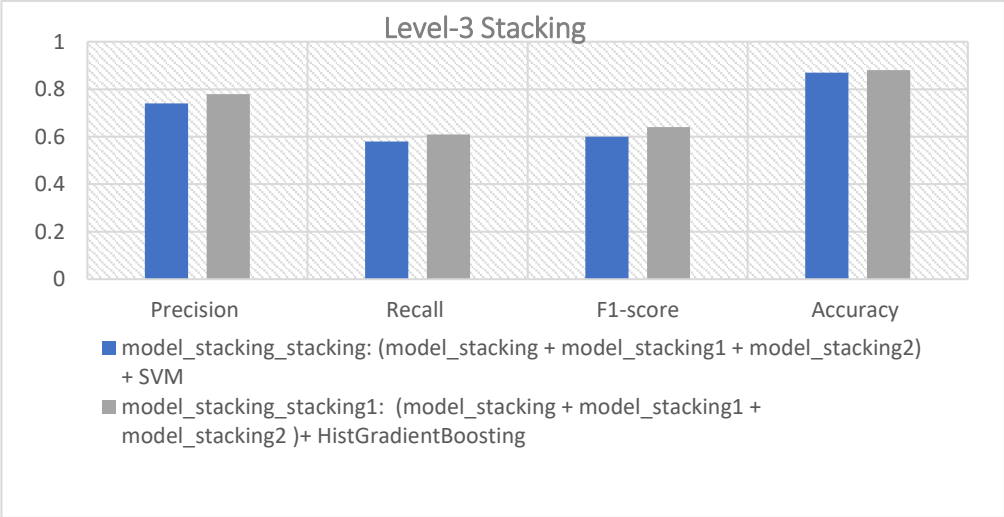


Figure 1.14 *Results level 3 stacking modeling experiment*

Three techniques, principal component analysis (PCA), kernel PCA (a variation of PCA), and linear discriminant analysis (LDA), were applied to test the effectiveness of dimension reduction techniques. Using principal component analysis (PCA) and kernel PCA with n = 2, two variables were generated whose features were combined with the target variables to form two new datasets used to train and test the models. LDA resulted in a variable that was joined with the target variable to build a new dataset. Additionally, the SMOTE oversampling method has been utilized to address the issue of unbalanced class labeling. The result achieved is displayed below in Tables 1.10-1.12 and in Figures 1.15-1.17.

Table 1.10 *Results PCA and SMOTE*

Models	Precision	Recall	F1-score	Accuracy
Random Forest	0.68	0.84	0.69	0.77
AdaBoost	0.69	0.82	0.71	0.81
SVM	0.68	0.83	0.7	0.79
KNN	0.64	0.73	0.66	0.79
HistGB	0.63	0.73	0.64	0.74
LR	0.67	0.8	0.68	0.78
GB	0.68	0.82	0.69	0.78
DT	0.66	0.79	0.67	0.77
XGBoost	0.65	0.78	0.65	0.74
Voting	0.68	0.79	0.71	0.81

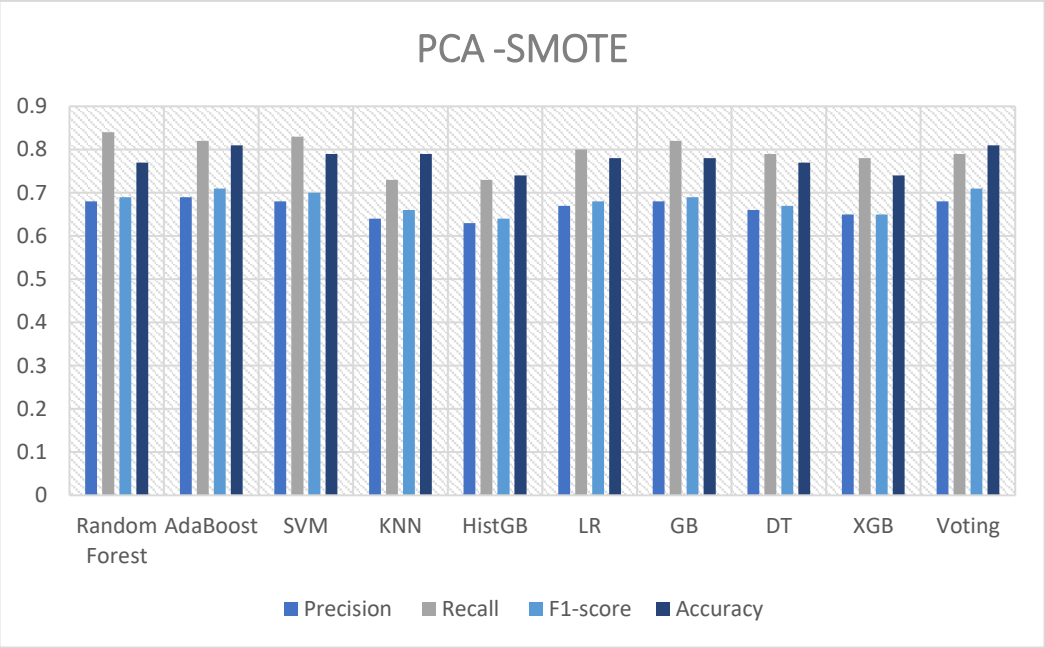


Figure 1.15 *Results PCA and SMOTE experiment*

Table 1.11 *Results-Kernel-PCA and SMOTE*

Models	Precision	Recall	F1-score	Accuracy
Random Forest	0.68	0.84	0.69	0.77
AdaBoost	0.69	0.82	0.71	0.81
SVM	0.68	0.83	0.7	0.79
KNN	0.64	0.73	0.66	0.79
HistGB	0.63	0.73	0.64	0.74
LR	0.67	0.8	0.68	0.78
GB	0.68	0.82	0.69	0.78
DT	0.66	0.79	0.67	0.77
XGBoost	0.65	0.78	0.65	0.74
Voting	0.68	0.79	0.71	0.81

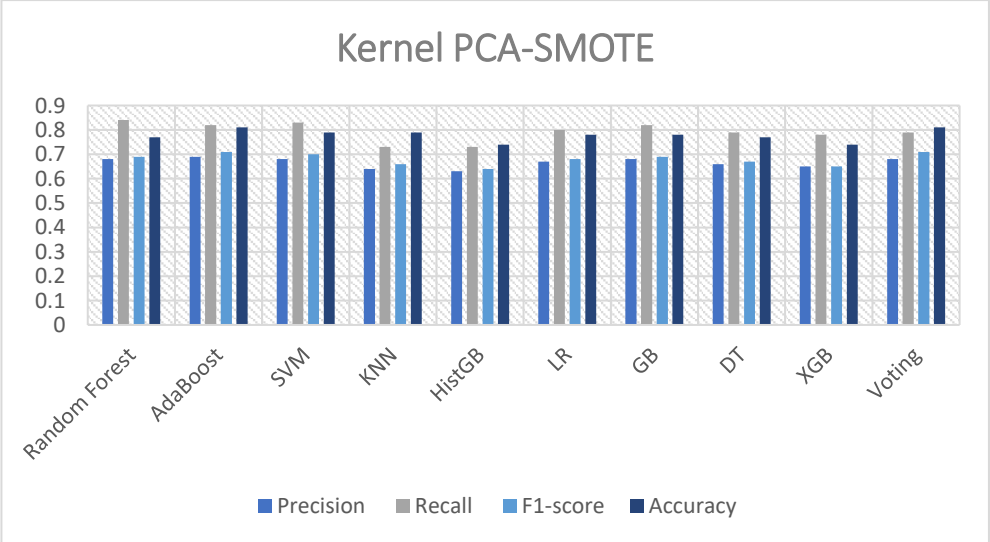


Figure 1.16 *Results kernel PCA-SMOTE experiment*

Table 1.12 *Results-LDA and SMOTE*

Models	Precision	Recall	F1-score	Accuracy
Random Forest	0.68	0.84	0.69	0.77
AdaBoost	0.69	0.82	0.71	0.81
SVM	0.68	0.83	0.7	0.79
KNN	0.64	0.73	0.66	0.79
HistGB	0.63	0.73	0.64	0.74
LR	0.67	0.8	0.68	0.78
GB	0.68	0.82	0.69	0.78
DT	0.65	0.8	0.66	0.75
XGBoost	0.65	0.78	0.65	0.74
Voting	0.68	0.81	0.71	0.8

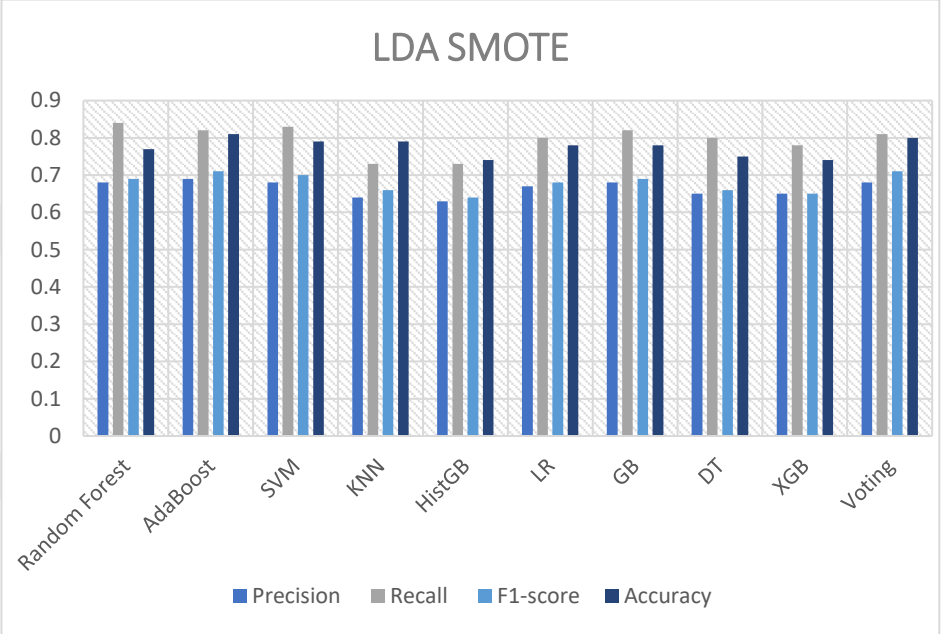


Figure 1.17 *Results LDA and SMOTE experiment*

One-class classification approaches can be utilized effectively for binary unbalanced classification tasks in which the majority is regarded as normal and the minority is an outlier or anomaly, despite being designed for anomaly identification and novelty detection. Five algorithms were utilized in this investigation: one-class support vector machines (1-SVM), one-class SVM with stochastic gradient descent (1-SGDSVM), isolation forest (IF), minimum covariant determinant (MCD), and local outlier factor (LOF). As demonstrated in Figure 1.18 and Table 1.13, the gathered data indicated that the isolation forest, one-class SVM with stochastic gradient descent, and one-class SVM algorithms performed better than other models. The performance of the minimum covariance determinant and the local outlier factor was unsatisfactory.

Table 1.13 *Results one-class classification experiment*

Model	Precision	Recall	F1-score	Accuracy
1-SVMSGD	0.7	0.84	0.74	0.88
1-SVM	0.72	0.77	0.8	0.74
MCD	0.52	0.53	0.52	0.8
IF	0.74	0.77	0.75	0.91
LOF	0.55	0.56	0.55	0.83

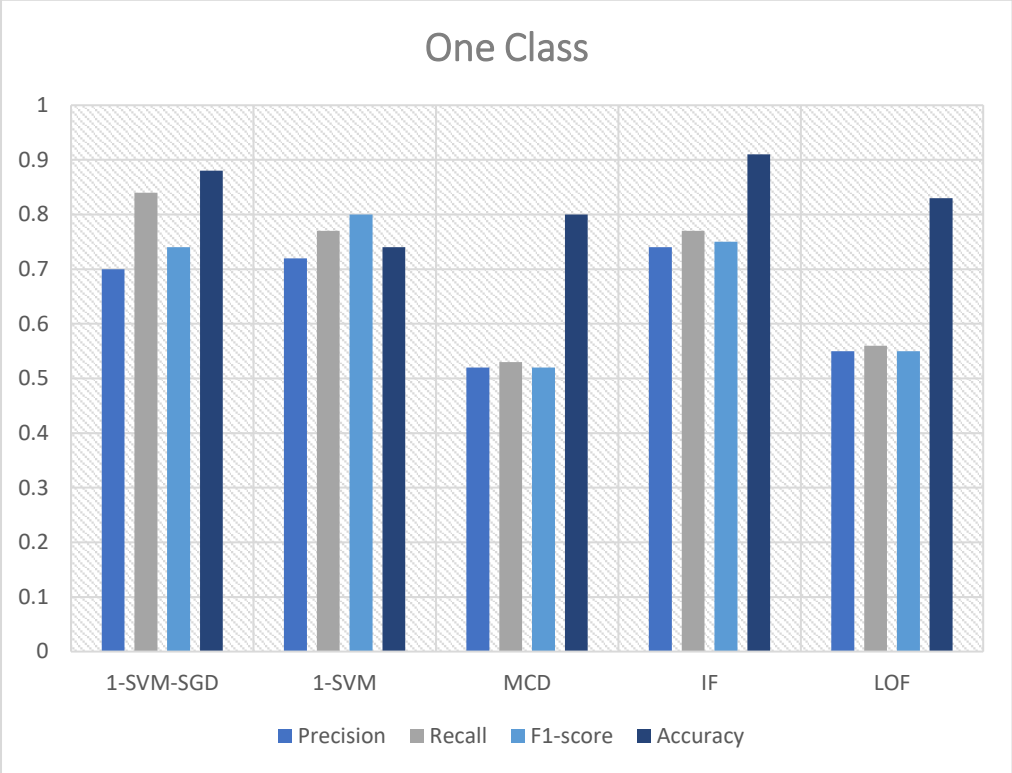


Figure 1.18: *Results one-class classification modeling experiment*

Isolation Forest and the One-class SVM technique performed better than other models in a one-class classification experiment using a dataset to which PCA with n=2 was applied. One-class SVM using stochastic gradient descent performed quite poorly, but the Minimum Covariance Determinant and Local Outlier Factor performed substantially better. The acquired result is depicted in Table 1.14 and Figure 1.19.

Table 1.14 *Results PCA one-class classification experiment*

Model	Precision	Recall	F1-score	Accuracy
1-SVMSGD	0.53	0.55	0.54	0.8
1-SVM	0.75	0.82	0.78	0.91
MCD	0.61	0.61	0.61	0.86
IF	0.71	0.78	0.74	0.89
LOF	0.61	0.64	0.55	0.85

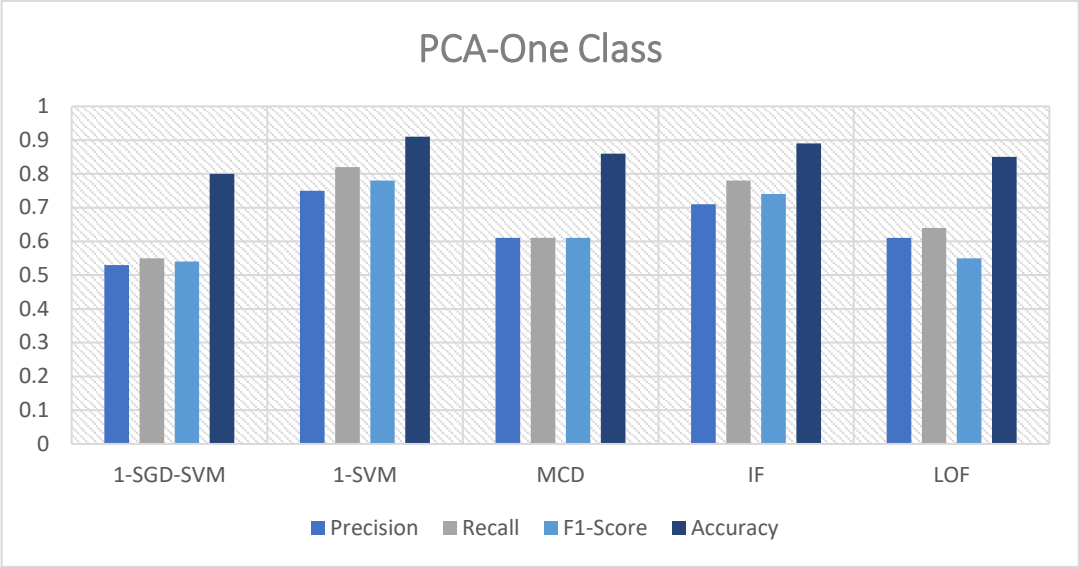


Figure 1.19 *Results PCA one class classification experiment*

1.6. Predicting A Suitable Healthcare Unit For COVID-19 Patients

1.6.1. Preprocessing (2)

The data pre-processing procedures for the multiclass classification task closely followed those described earlier for the binary classification problem. These included standard operations such as data cleaning, categorical encoding, and feature scaling. To enhance the handling of missing data, additional strategies were implemented. In addition to the previously applied k-Nearest Neighbors (KNN) imputation with k = 5, further configurations with k = 2 and k = 10 were explored to examine the impact of neighborhood size. Moreover, two implementations of Multiple Imputation by Chained Equations (MICE) were applied, one utilizing the Random Forest algorithm and the other employing Light Gradient Boosting (LGBM). These approaches resulted in five distinct imputed datasets, which were subsequently evaluated. The encoding schema for the multiclass target variable hospital admission was maintained, with class labels 1, 2, and 3 representing admission to an unknown unit, a surveillance unit, and an intensive care unit, respectively. As in the earlier task, data standardization using the StandardScaler was favored over normalization to preserve outlier-related information and maintain consistency across datasets.

1.6.2. Feature selection (2)

Feature selection entails reducing the number of input variables to reduce computational costs and improve model performance. Two techniques were combined

and used in this study: the variance threshold and the mutual information feature selection technique. The variance threshold technique considers a specific threshold and eliminates all features with low variance. Mutual information (MI) between two random variables is a non-negative value that measures the variables' dependence. It is equal to zero when two random variables are independent, and higher values indicate greater dependency. Formally, the mutual information between two random variables, X and Y, is defined as follows.

$$H(X) - H(X \mid Y) = I(X ; Y) \tag{1.8}$$

Where *I(X; Y)* represents the mutual information for *X* and *Y*, *H(X)* represents the entropy for *X*, and $H(X \mid Y)$ represents the conditional entropy for *X* given *Y*.

The SelectKbest algorithm was used to choose features with the highest mutual information score betweenthe target feature and the rest of the features. This step resulted in the selection of five characteristics: patient age quantile, platelets, leukocytes, eosinophils, and monocytes.

1.6.3. Modeling (2)

On each of the five produced datasets, a variety of machine learning models were trained and evaluatedduring this phase. Multiple classification techniques have been studied. Techniques such as one-vs-one and one-vs-rest, as well as other multiclass algorithms with built-in support for multiclass classification,have been implemented. Inherently multiclass algorithms include the Gaussian Naive Bayes (GNB) algorithm, the decision tree classifier, the k-nearest neighbors algorithm, the multi-layer perceptron classifier, the random forest classifier, and the extremely randomized tree classifier. Due to the skewed distribution of the target classes, specificity, sensitivity, and accuracy will be used as evaluation measures. Below are the equations for the aforementioned evaluation metrics. Firstly, due to the data's uneven class distributions, multiclass inherited models were explored and employed several oversampling strategies in an attempt to improve the performance of these models. The one-vs-one and one-vs-rest approaches were studied in the follow-up phase. Oversamplingmethods and pipelines have been used in conjunction with these two procedures to seal any potential data leaks. Oversampling strategies were implemented via a pipeline and have been applied only to the train set. The tables below depict the outcome of investigating the dataset created using the Missforestapproach to impute data. The tables that follow summarize the

acquired score result, where "OBV" stands for unknown unit class, "S-ICU" stands for surveillance unit, and "ICU" stands for intensive care unit.Table 1.15 shows the findings obtained by using single models without using any oversampling techniques,whereas Tables 1.16-1.19 show the results obtained by utilizing oversampling techniques. The SMOTE, ADASYN, Borderline SMOTE SVM, and Borderline SMOTE oversampling algorithms were used.

Table 1.15 *KNN =5 dataset- single models*

Models	**sensitivity**			**specificity**			**F-1 Score**
	OBV	*S-ICU*	*ICU*	*OBV*	*S-ICU*	*ICU*	
GNB	0.53	0.88	0.97	0.87	0.44	0.25	0.5
KNN	0.58	0.96	0.99	0.96	0.77	0.12	0.61
DT	0.41	0.96	1	0.98	0.44	0.1	0.47
RF	0.97	0.42	0.25	0.29	0.98	0.98	0.53
MLP	0.23	0.97	0.98	0.96	0.11	0.25	0.47
EXT	0.41	0.95	1	0.97	0.55	0.1	0.48

Table 1.16 *KNN = 5 dataset - SMOTE*

Models	**sensitivity**			**specificity**			**F-1 Score**
	OBV	*S-ICU*	*ICU*	*OBV*	*S-ICU*	*ICU*	
GNB	0.7	0.9	0.96	0.88	0.55	0.5	0.59
KNN	0.2	0.93	0.97	0.94	0.66	0.5	0.6
DT	0.8	0.93	0.96	0.91	0.55	0.52	0.55
RF	0.5	0.66	0.55	0.52	0.95	0.99	0.57
MLP	0.2	0.92	0.98	0.95	0.66	0.52	0.54
EXT	0.94	0.93	0.98	0.94	0.88	0.5	0.72

Table 1.17 *KNN = 5 dataset –ADASYN*

Models	**sensitivity**			**specificity**			**F-1 Score**
	OBV	*S-ICU*	*ICU*	*OBV*	*S-ICU*	*ICU*	
GNB	0.88	0.82	0.96	0.8	0.78	0.5	0.57
KNN	0.88	0.92	0.97	0.92	0.77	0.5	0.68
DT	0.76	0.92	0.98	0.93	0.67	0.25	0.58
RF	0.88	0.66	0.55	0.64	0.87	0.99	0.5
MLP	0.7	0.94	0.99	0.94	0.67	0.5	0.7
EXT	0.76	0.93	0.98	0.95	0.77	0.5	0.69

Table 1.18 *KNN = 5 dataset - SVM SMOTE*

Models	**Sensitivity**			**Specificity**			**F-1 Score**
	OBV	*S-ICU*	*ICU*	*OBV*	*S-ICU*	*ICU*	
GNB	0.76	0.86	0.98	0.88	0.67	0.52	0.48
KNN	0.76	0.93	0.99	0.95	0.78	0.25	0.63
DT	0.7	0.93	0.97	0.95	0.55	0.56	0.46
RF	0.92	0.66	0.25	0.58	0.93	0.98	0.59
MLP	0.76	0.9	0.98	0.93	0.78	0.57	0.48
EXT	0.76	0.93	0.98	0.95	0.77	0.5	0.63

Table 1.19 *KNN = 5 dataset - borderline SMOTE*

Models	Sensitivity			Specificity			F-1 Score
	OBV	*S-ICU*	*ICU*	*OBV*	*S-ICU*	*ICU*	
GNB	0.87	0.55	0.55	0.52	0.87	0.99	0.48
KNN	0.76	0.88	0.99	0.9	0.78	0.55	0.58
DT	0.7	0.93	0.98	0.95	0.55	0.56	0.59
RF	0.76	0.9	0.97	0.92	0.77	0.61	0.61
MLP	0.76	0.9	0.97	0.92	0.77	0.71	0.61
EXT	0.82	0.93	0.98	0.95	0.88	0.55	0.55

Table 1.20 *KNN = 5 dataset - ONE-vs-Rest –GBT*

Models	Sensitivity			Specificity			F-1 Score
	OBV	*S-ICU*	*ICU*	*OBV*	*S-ICU*	*ICU*	
GBT	0.95	0.66	0.12	0.58	0.93	0.99	0.56
SMOTE	0.9	0.44	0.37	0.47	0.91	0.98	0.57
ADASYN	0.89	0.33	0.38	0.52	0.9	0.96	0.52
B-SVM	0.94	0.33	0.12	0.35	0.93	0.99	0.4
BORDERLINE	0.94	0.44	0.25	0.52	0.92	0.99	0.56

Table 1.21 *KNN = 5 dataset - ONE-vs- Rest -GPC*

Models	sensitivity			specificity			F-1 Score
	OBV	*S-ICU*	*ICU*	*OBV*	*S-ICU*	*ICU*	
GPC	0.96	0.33	0.51	0.29	0.95	0.99	0.42
SMOTE	0.95	0.55	0.37	0.52	0.95	0.99	0.65
ADASYN	0.89	0.33	0.37	0.52	0.9	0.96	0.66
B-SVM	0.95	0.55	0.52	0.41	0.95	0.99	0.55
BORDERLINE	0.95	0.55	0.47	0.45	0.9	0.97	0.57

The outcomes of the One-vs-Rest multiclass classification approaches are presented in Tables 1.20 and 1.21, respectively.

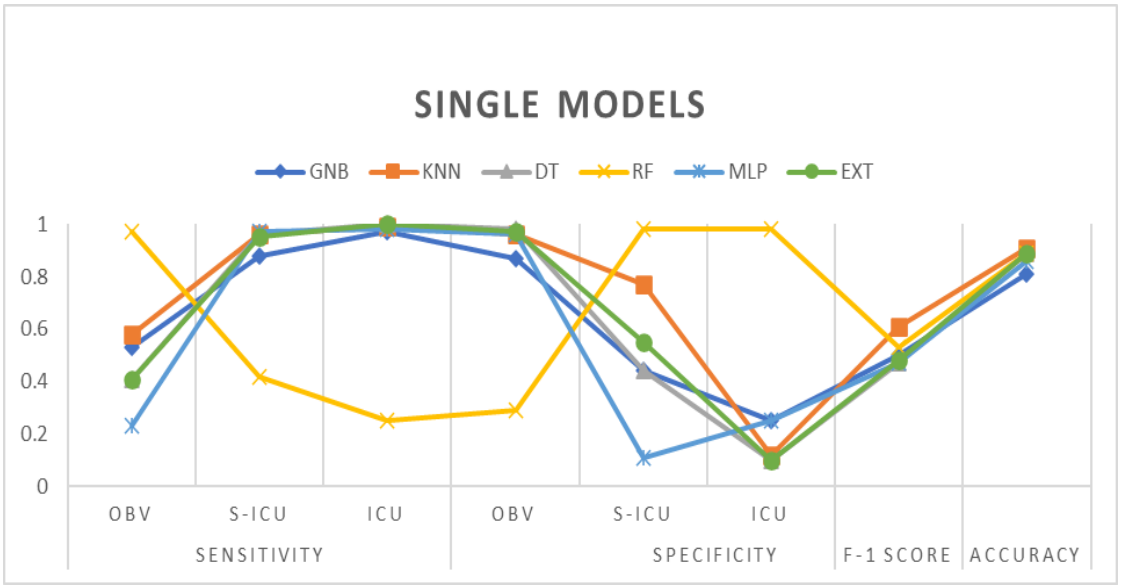


Figure 1.20 *KNN=5 dataset: single models*

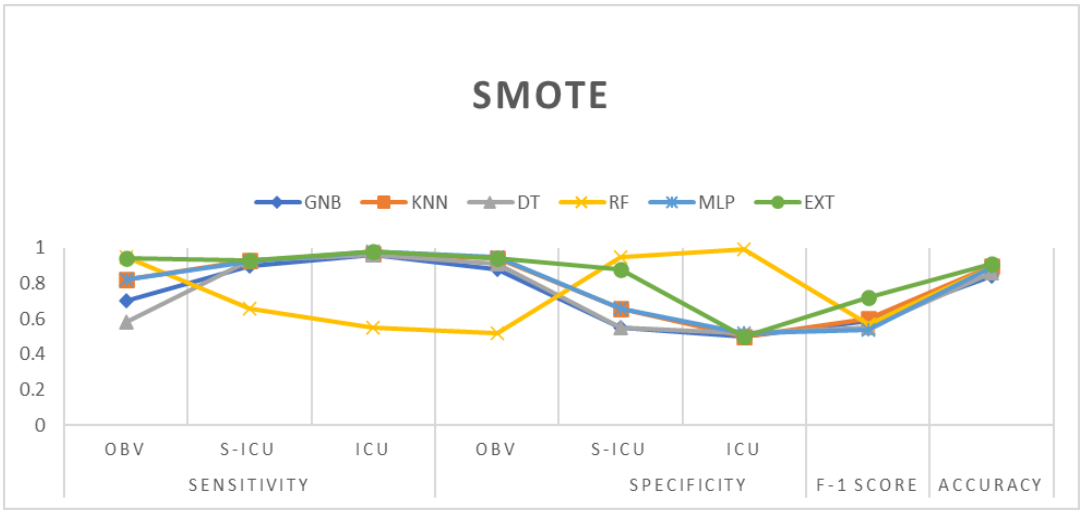


Figure 1.21 *KNN=5 dataset: SMOTE*

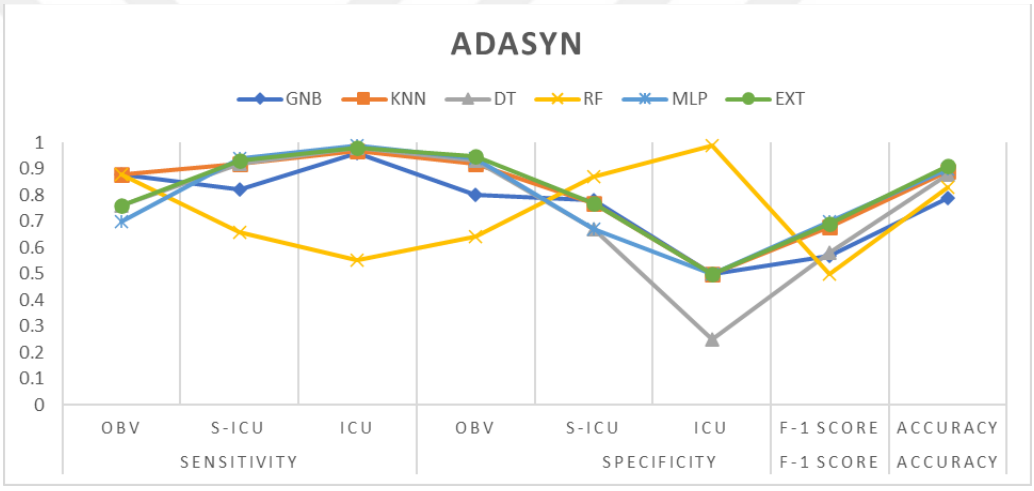


Figure 1.22 *KNN=5 dataset: ADASYN*

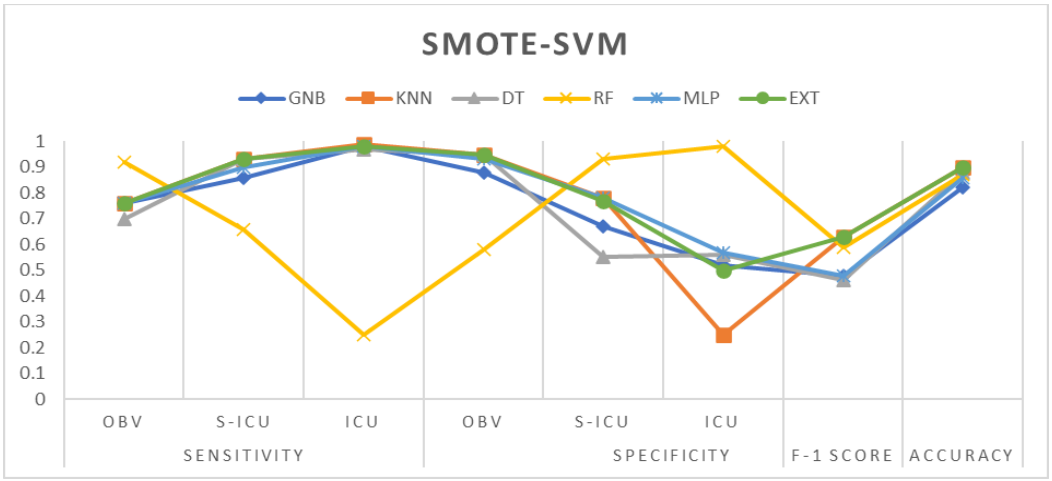


Figure 1.23 *KNN=5 dataset: SVM SMOTE*

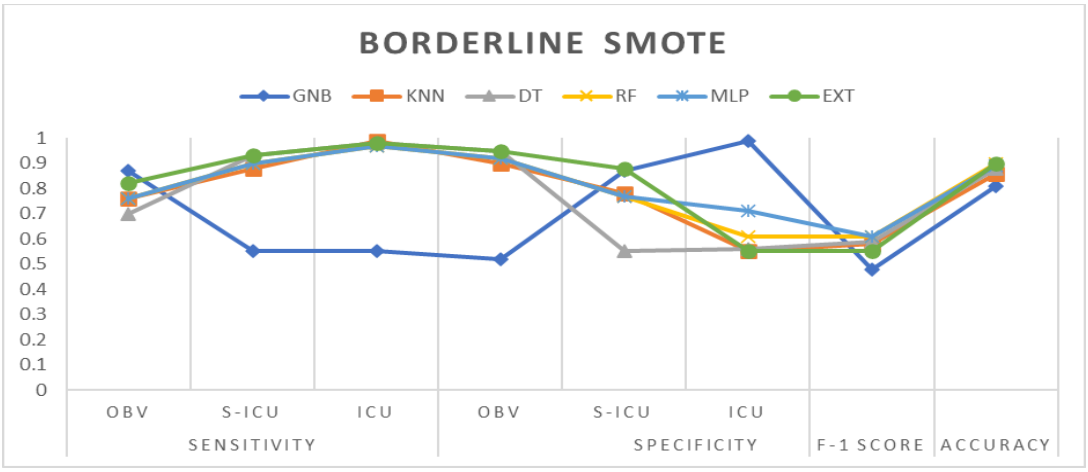


Figure 1.24 *KNN=5 dataset borderline SMOTE*

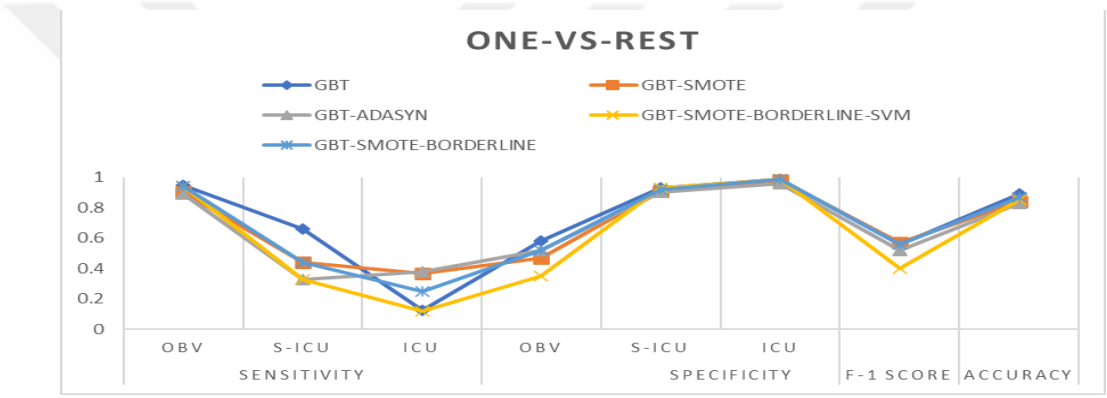


Figure 1.25 *KNN=5 dataset: ONE-vs-Rest GBT*

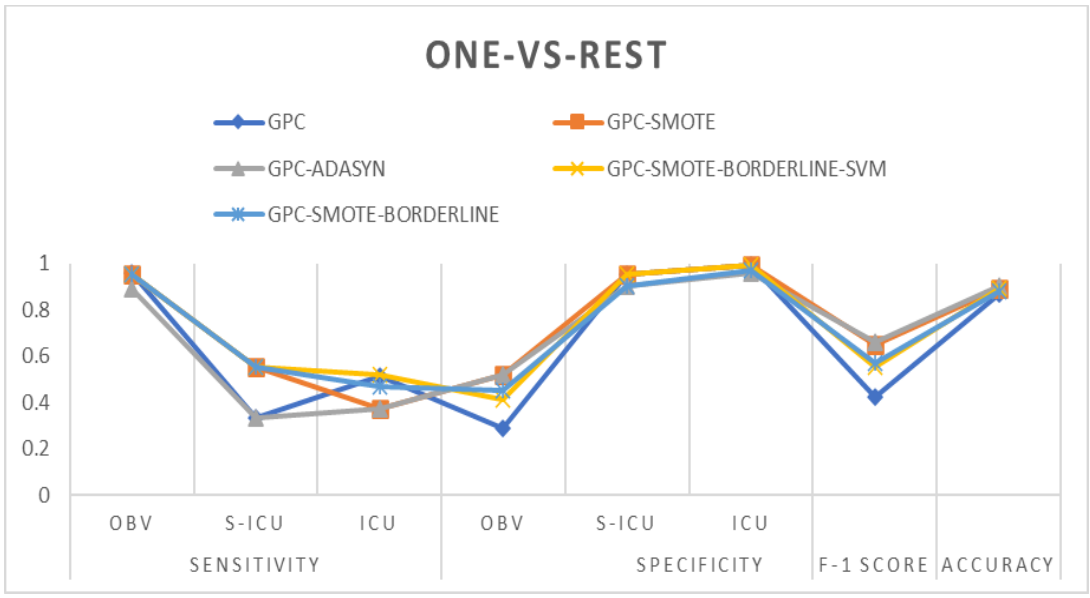


Figure 1.26 *KNN=5 dataset: ONE-vs-Rest GPC*

The results obtained from using a dataset in which missing data have been imputed using the KNN algorithm with five neighbors are shown in Figures 1.20-1.25. Figure 1.20 shows the outcomes from utilizing a single model, which involves using an algorithm without using oversampling techniques. Figures 1.21 and 1.22 show the results from SVM SMOTE and Borderline SMOTE, respectively, while Figures 1.23 and 1.24 show the outcome from SMOTE. The outcomes of the ONE-vs-Rest method are shown in Figures 1.25 and 1.26. The main focus of this study is the prediction of the minority classes, "S-ICU" (which represents the special surveillance unit) and "ICU" (which stands for the intensive care unit). It has been demonstrated that using oversampling techniques improves outcomes significantly. The outcomes of reviewing the dataset obtained using the KNN algorithm and 10-neighbor strategy as a data imputation technique are shown in the tables below. The scores are summarized in the tables below, where "OBV" implies an unknown unit type, "S-ICU" suggests a surveillance unit, and "ICU" refers to an intensive care unit. Table 1.22 displays the results of employing single models without using any oversampling approaches, whereas Tables 1.23-1.26 display the results of using oversampling techniques. The algorithms used were SMOTE, ADASYN, SMOTE SVM, and Borderline SMOTE. Tables 1.27 and 1.28 contain results obtained by using the ONE-vs-Rest approach.

Table 1.22 *KNN = 10 dataset: Single Models*

Models	**Sensitivity**			**Specificity**			**F-1 Score**
	OBV	*S-ICU*	*ICU*	*OBV*	*S-ICU*	*ICU*	
GNB	0.7	0.91	0.96	0.91	0.55	0.25	0.53
KNN	0.41	0.96	0.99	0.96	0.44	0.12	0.53
DT	0.64	0.95	0.98	0.96	0.78	0.1	0.53
RF	0.35	0.94	1	0.95	0.33	0.1	0.41
MLP	0.35	0.96	0.98	0.96	0.22	0.25	0.51
EXT	0.35	0.94	1	0.97	0.33	0.1	0.42

Table 1.23 *KNN = 10 dataset: SMOTE*

Models	**Sensitivity**			**Specificity**			**F-1 Score**
	OBV	*S-ICU*	*ICU*	*OBV*	*S-ICU*	*ICU*	
GNB	0.82	0.82	0.94	0.79	0.44	0.5	0.5
KNN	0.7	0.95	0.98	0.95	0.55	0.5	0.66
DT	0.7	0.93	0.98	0.95	0.55	0.25	0.57
RF	0.82	0.91	0.98	0.93	0.78	0.25	0.57
MLP	0.7	0.94	0.97	0.95	0.44	0.37	0.6
EXT	0.88	0.93	0.98	0.94	0.77	0.5	0.7

Table 1.24 *KNN = 10 dataset: ADASYN*

Models	Sensitivity			Specificity			F-1 Score
	OBV	*S-ICU*	*ICU*	*OBV*	*S-ICU*	*ICU*	
GNB	0.82	0.8	0.93	0.77	0.44	0.5	0.49
KNN	0.7	0.94	0.98	0.93	0.55	0.5	0.64
DT	0.76	0.93	0.98	0.95	0.55	0.25	0.58
RF	0.76	0.93	0.95	0.92	0.67	0.25	0.57
MLP	0.7	0.93	0.98	0.95	0.44	0.37	0.59
EXT	0.88	0.93	0.98	0.94	0.78	0.5	0.7

Table 1.25 *KNN = 10 dataset: SVM SMOTE*

Models	Sensitivity			Specificity			F-1 Score
	OBV	*S-ICU*	*ICU*	*OBV*	*S-ICU*	*ICU*	
GNB	0.82	0.82	0.98	0.84	0.67	0.12	0.46
KNN	0.64	0.94	0.99	0.95	0.66	0.25	0.62
DT	0.7	0.93	0.96	0.93	0.66	0.12	0.54
RF	0.82	0.93	0.96	0.94	0.66	0.12	0.53
MLP	0.76	0.78	0.98	0.79	0.66	0.25	0.49
EXT	0.76	0.93	0.98	0.94	0.78	0.25	0.62

Table 1.26 *KNN = 10 dataset: BORDERLINE SMOTE*

Models	Sensitivity			Specificity			F-1 Score
	OBV	*S-ICU*	*ICU*	*OBV*	*S-ICU*	*ICU*	
GNB	0.65	0.95	0.99	0.96	0.67	0.25	0.62
KNN	0.91	0.33	0.12	0.29	0.92	0.99	0.46
DT	0.7	0.93	0.98	0.95	0.55	0.1	0.54
RF	0.82	0.93	0.96	0.94	0.66	0.12	0.6
MLP	0.82	0.92	0.97	0.94	0.55	0.37	0.6
EXT	0.76	0.98	0.98	0.94	0.77	0.25	0.62

Table 1.27 *KNN = 10 dataset: ONE-vs- Rest -GBT*

Models	Sensitivity			Specificity			F-1 Score
	OBV	*S-ICU*	*ICU*	*OBV*	*S-ICU*	*ICU*	
GBT	0.95	0.66	0.52	0.58	0.93	0.98	0.56
SMOTE	0.9	0.44	0.57	0.67	0.91	0.97	0.58
ADASYN	0.9	0.43	0.58	0.52	0.9	0.93	0.52
BSVM	0.93	0.53	0.42	0.55	0.93	0.99	0.4
BORDERLINE	0.92	0.44	0.5	0.52	0.92	0.97	0.56

Table 1.28 *KNN = 10 dataset: ONE-vs- Rest - GPC*

Models	Sensitivity			Specificity			F-1 Score
	OBV	*S-ICU*	*ICU*	*OBV*	*S-ICU*	*ICU*	
GPC	0.86	0.53	0.5	0.6	0.95	0.99	0.52
SMOTE	0.89	0.65	0.57	0.67	0.95	0.99	0.65
ADASYN	0.89	0.63	0.67	0.66	0.9	0.96	0.66
B-SVM	0.94	0.67	0.52	0.64	0.95	0.99	0.55
BORDERLINE	0.97	0.65	0.62	0.69	0.9	0.97	0.57

Figures 1.27-1.33 demonstrate the outcomes of utilizing a dataset with missing data imputed using the KNN algorithm with 10 neighbors. Figure 1.27 depicts the results of using a single model, which entails using an algorithm without using oversampling

approaches. Figures 1.28 and 1.29 show the SVM-SMOTE and Borderline SMOTE results, respectively, while Figures 1.30 and 1.31 display the SMOTE and ADASYN results. Figures 1.32 and 1.33 depict the results of the ONE-vs-Rest approach.

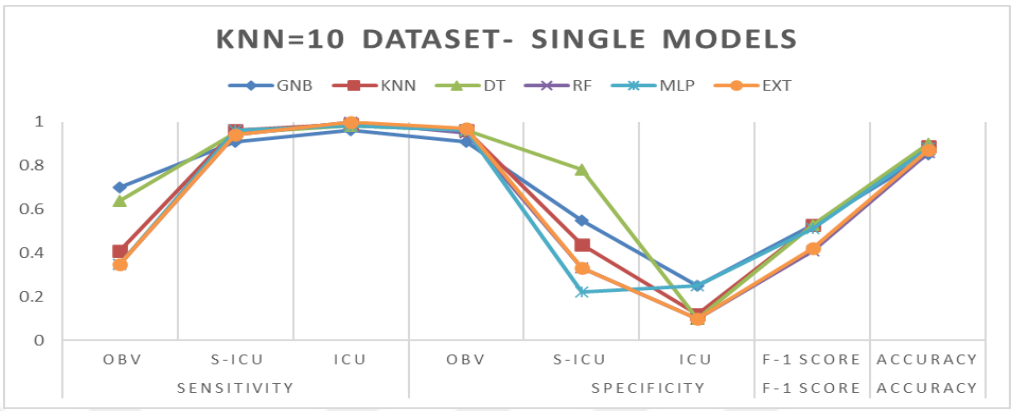


Figure 1.27 *KNN = 10 dataset : Single Models*

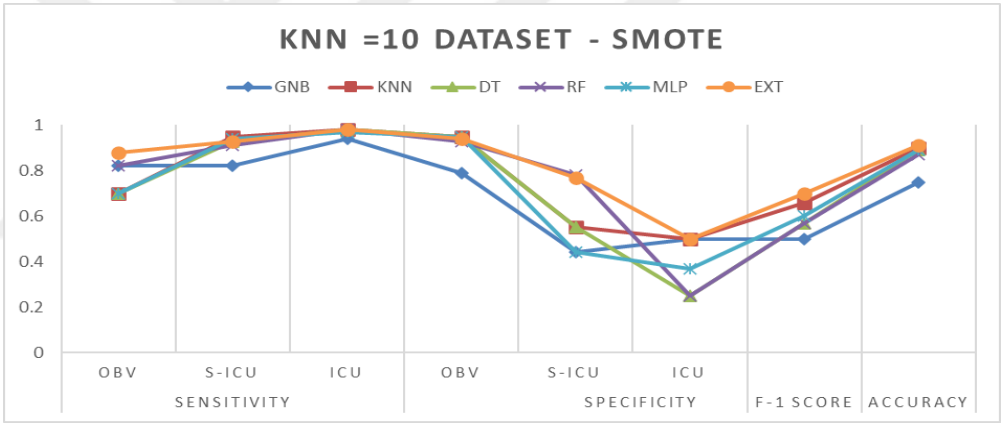


Figure 1.28 *KNN = 10 dataset : SMOTE*

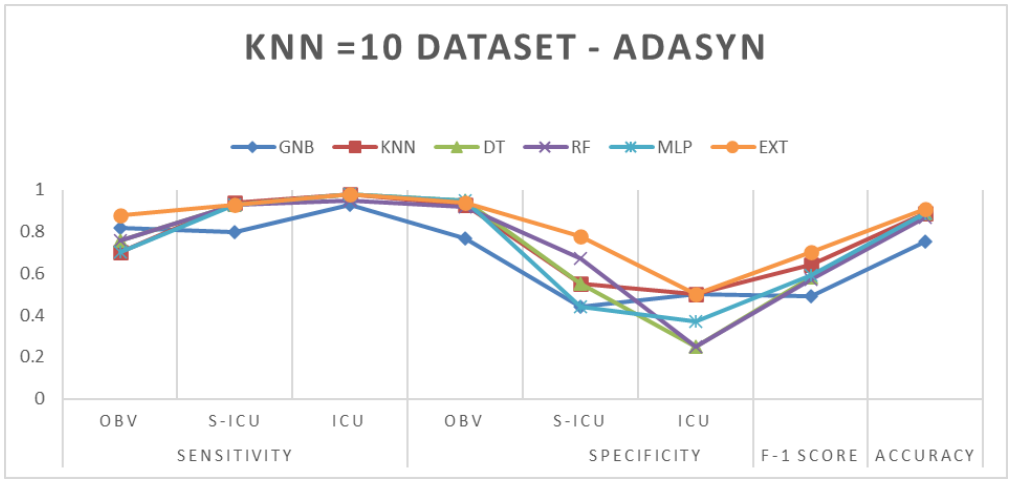


Figure 1.29 *KNN = 10 dataset : ADASYN*

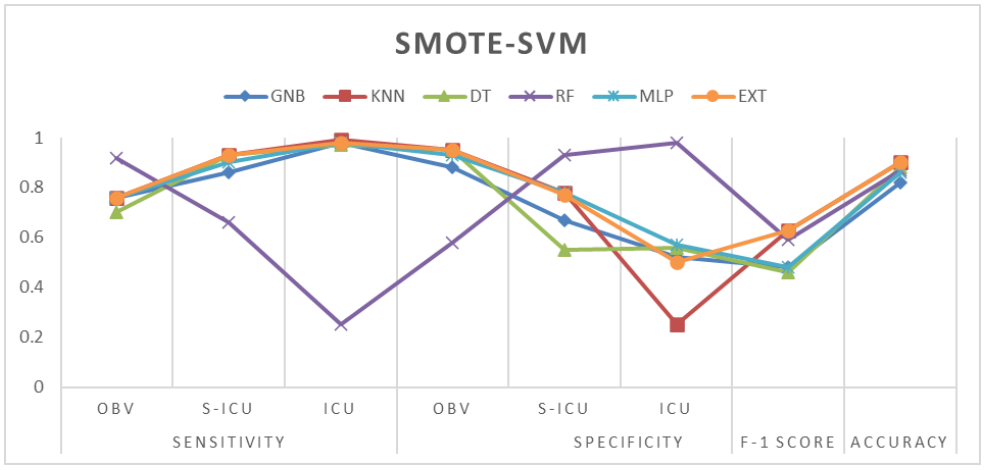


Figure 1.30 *KNN = 10 dataset - SVM SMOTE*

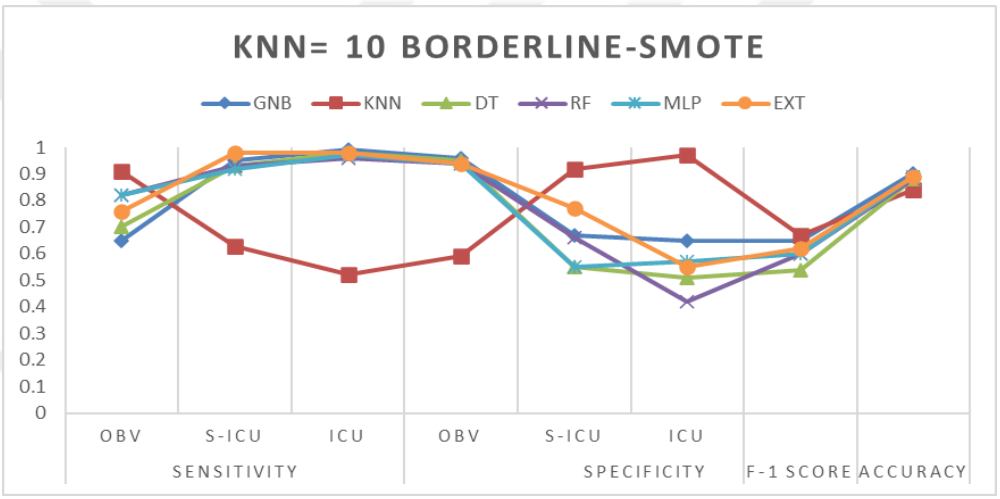


Figure 1.31 *KNN = 10 dataset - Borderline SMOTE*

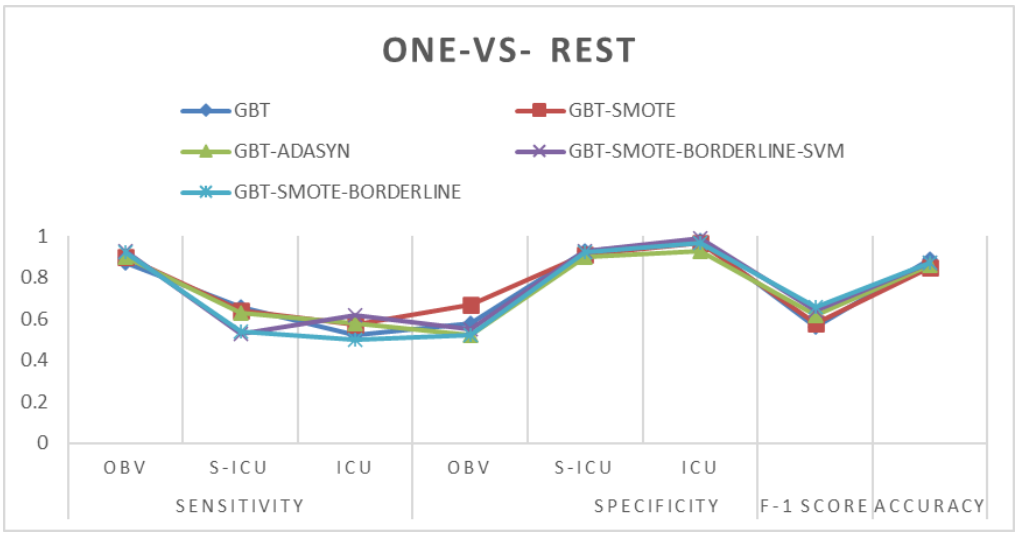


Figure 1.32 *KNN = 10 dataset - ONE-vs-Rest*

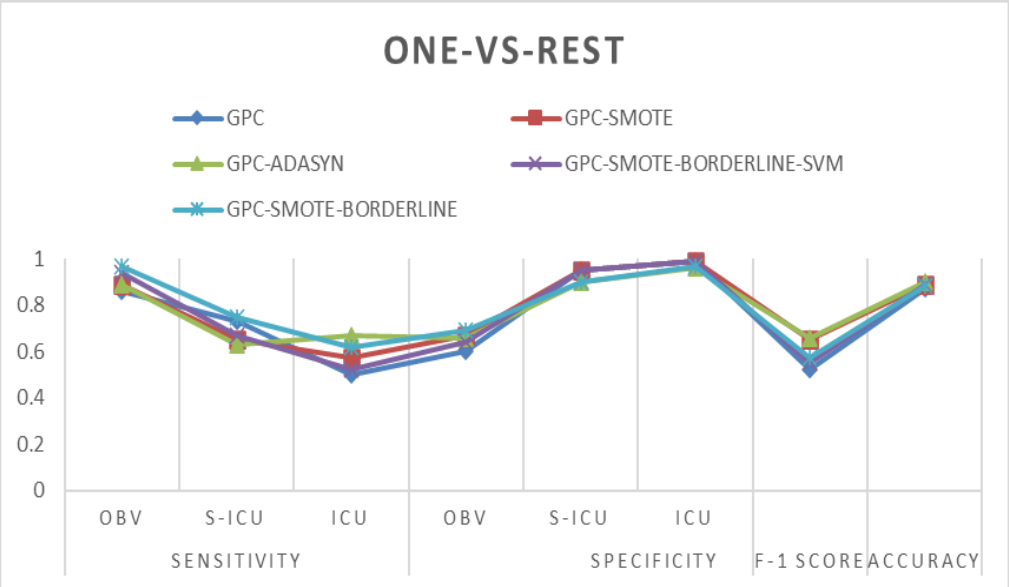


Figure 1.33 *KNN = 10 dataset - ONE-vs-Rest –GPC*

The KNN algorithm and two neighbors as a data imputation method were also evaluated, and the results are displayed in the tables below. Table 1.29 shows the results of utilizing single models without oversampling, whereas Tables 1.30-1.33 show the results of using oversampling techniques. SMOTE, ADASYN, SMOTE SVM, and Borderline SMOTE were the methods employed. Tables 1.34 and 1.35 show the results of the ONE-vs-Rest method.

Table 1.29 *KNN = 2 dataset - single models*

Models			**Sensitivity**			**Specificity**	**F-1 Score**
	OBV	*S-ICU*	*ICU*	*OBV*	*S-ICU*	*ICU*	
GNB	0.29	0.92	0.99	0.92	0.33	0.12	0.55
KNN	0.7	0.92	0.99	0.95	0.67	0.12	0.47
DT	0.41	0.98	1	0.98	0.67	0.1	0.54
RF	0.35	0.97	0.99	0.96	0.94	0.88	0.12
MLP	0.64	0.96	0.98	0.97	0.66	0.25	0.63
EXT	0.59	0.95	1	0.98	0.66	0.1	0.51

Table 1.30 *KNN = 2 dataset – SMOTE*

Models			**Sensitivity**			**Specificity**	**F-1 Score**
	OBV	*S-ICU*	*ICU*	*OBV*	*S-ICU*	*ICU*	
GNB	0.82	0.77	0.98	0.78	0.78	0.25	0.52
KNN	0.88	0.93	0.97	0.94	0.77	0.37	0.65
DT	0.88	0.92	0.98	0.94	0.78	0.25	0.6
RF	0.76	0.93	0.96	0.94	0.67	0.25	0.59
MLP	0.76	0.95	0.97	0.96	0.67	0.37	0.65
EXT	0.94	0.92	0.98	0.93	0.88	0.5	0.71

Table 1.31 *KNN = 2 dataset - ADASYN*

Models	Sensitivity			Specificity			F-1 Score
	OBV	*S-ICU*	*ICU*	*OBV*	*S-ICU*	*ICU*	
GNB	0.88	0.78	0.97	0.78	0.66	0.5	0.56
KNN	0.88	0.93	0.97	0.94	0.78	0.5	0.69
DT	0.76	0.91	0.99	0.94	0.78	0.12	0.56
RF	0.88	0.93	0.96	0.94	0.88	0.12	0.58
MLP	0.82	0.9	0.97	0.91	0.66	0.37	0.6
EXT	0.88	0.92	0.98	0.95	0.77	0.37	0.67

Table 1.32 *KNN = 2 dataset - SVM SMOTE*

Models	Sensitivity			Specificity			F-1 Score
	OBV	*S-ICU*	*ICU*	*OBV*	*S-ICU*	*ICU*	
GNB	0.76	0.88	0.98	0.9	0.67	0.12	0.5
KNN	0.82	0.93	0.98	0.95	0.89	0.25	0.65
DT	0.7	0.94	0.97	0.94	0.67	0.12	0.54
RF	0.76	0.91	0.98	0.93	0.78	0.12	0.55
MLP	0.76	0.95	0.98	0.97	0.67	0.25	0.62
EXT	0.82	0.93	0.98	0.95	0.77	0.37	0.67

Table 1.33 *KNN = 2 dataset - borderline SMOTE*

Models	Sensitivity			Specificity			F-1 Score
	OBV	*S-ICU*	*ICU*	*OBV*	*S-ICU*	*ICU*	
GNB	0.82	0.75	0.99	0.78	0.78	0.12	0.45
KNN	0.95	0.82	0.99	0.95	0.89	0.25	0.65
DT	0.7	0.93	0.96	0.94	0.67	0.12	0.54
RF	0.76	0.91	0.98	0.93	0.77	0.12	0.57
MLP	0.76	0.95	0.98	0.96	0.67	0.25	0.67
EXT	0.82	0.94	0.98	0.95	0.78	0.37	0.67

Table 1.34 *KNN = 2 dataset - ONE-vs-Rest -GPC*

Models	Sensitivity			Specificity			F-1 Score
	OBV	*S-ICU*	*ICU*	*OBV*	*S-ICU*	*ICU*	
GPC	0.96	0.63	0.41	0.49	0.85	0.88	0.52
SMOTE	0.95	0.75	0.67	0.52	0.93	0.89	0.65
ADASYN	0.89	0.53	0.67	0.52	0.9	0.93	0.66
B-SVM	0.95	0.65	0.61	0.41	0.95	0.95	0.55
BORDERLINE	0.95	0.67	0.68	0.45	0.9	0.97	0.57

Table 1.35 *KNN = 2 dataset - ONE-vs-Rest -GBT*

Models	Sensitivity			Specificity			F-1 Score
	OBV	*S-ICU*	*ICU*	*OBV*	*S-ICU*	*ICU*	
GBT	0.82	0.56	0.55	0.58	0.93	0.85	0.56
SMOTE	0.92	0.64	0.57	0.57	0.91	0.78	0.57
ADASYN	0.89	0.63	0.48	0.52	0.9	0.76	0.62
B-SVM	0.94	0.63	0.52	0.55	0.93	0.83	0.64
BORDERLINE	0.93	0.72	0.55	0.52	0.92	0.79	0.66

Figures 1.34-1.40 show the results of using a dataset with missing data that was imputed using the KNN technique with two neighbors. Figure 1.34 displays the outcomes of utilizing a single model, which implies applying an algorithm without the use of

oversampling methods. Figures 1.35 and 1.36 show the SVM, SMOTE, and Borderline SMOTE results, while Figures 1.37 and 1.38 show the SMOTE and ADASYN results, respectively. Figures 1.39 and 1.40 show the outcomes of the ONE-vs-Rest strategy.

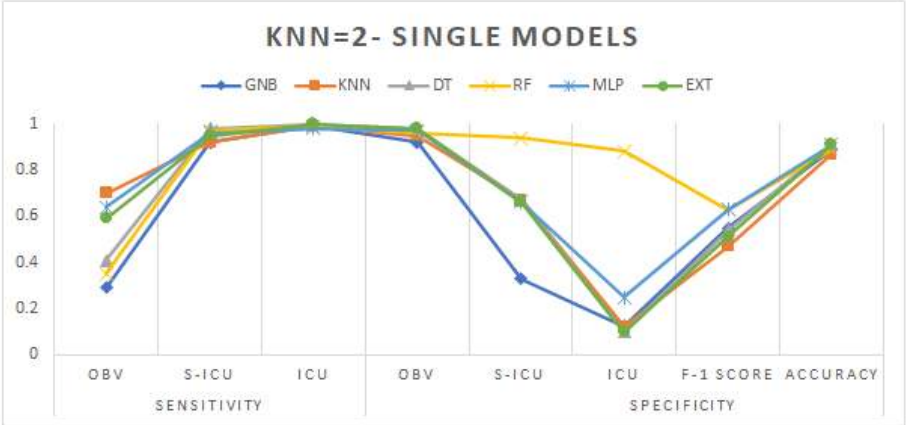


Figure 1.34 *KNN= 2 dataset - single models*

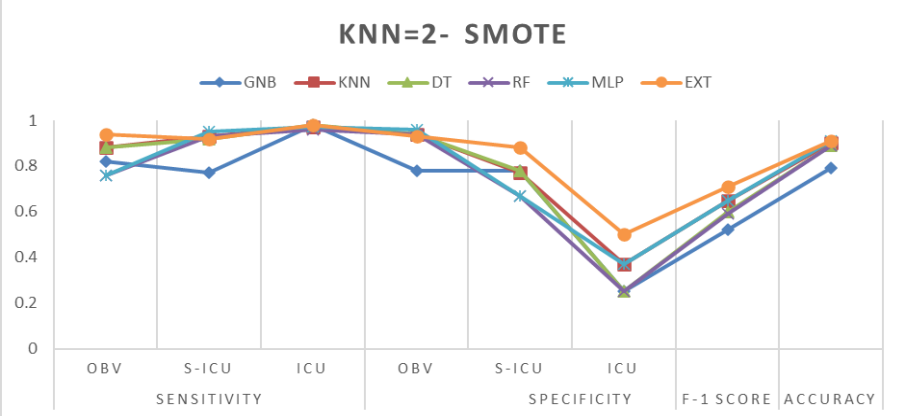


Figure 1.35 *KNN= 2 dataset – SMOTE*

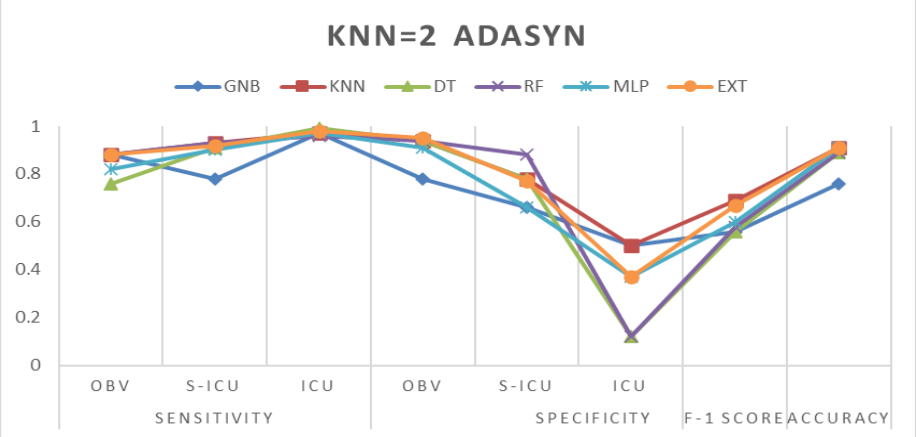


Figure 1.36 *KNN= 2 dataset - ADASYN*

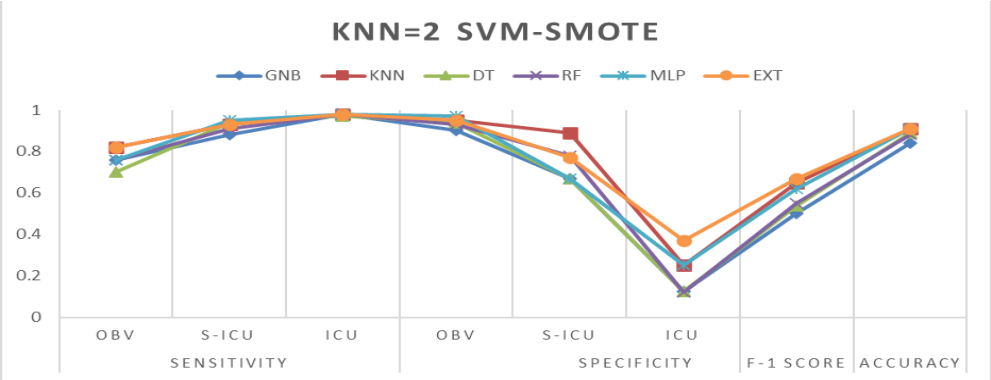


Figure 1.37 *KNN= 2 dataset - SVM SMOTE*

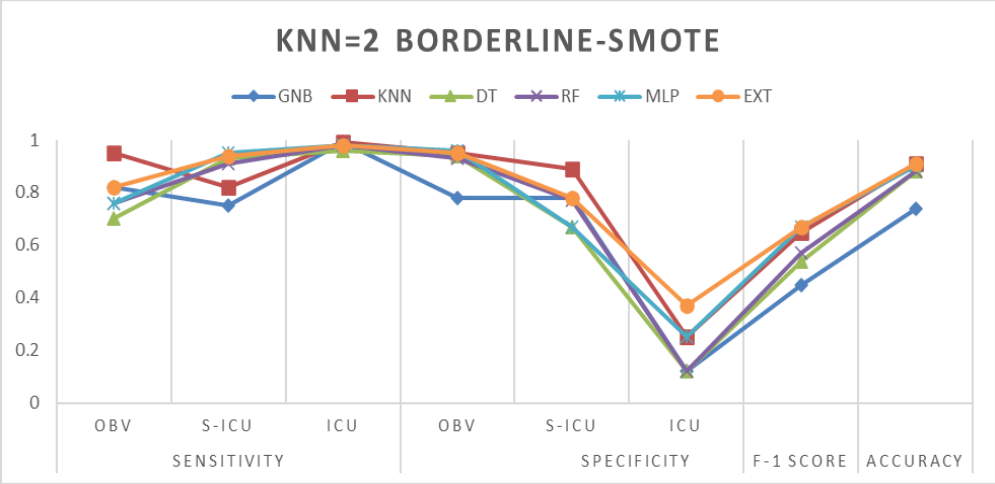


Figure 1.38 *Borderline SMOTE*

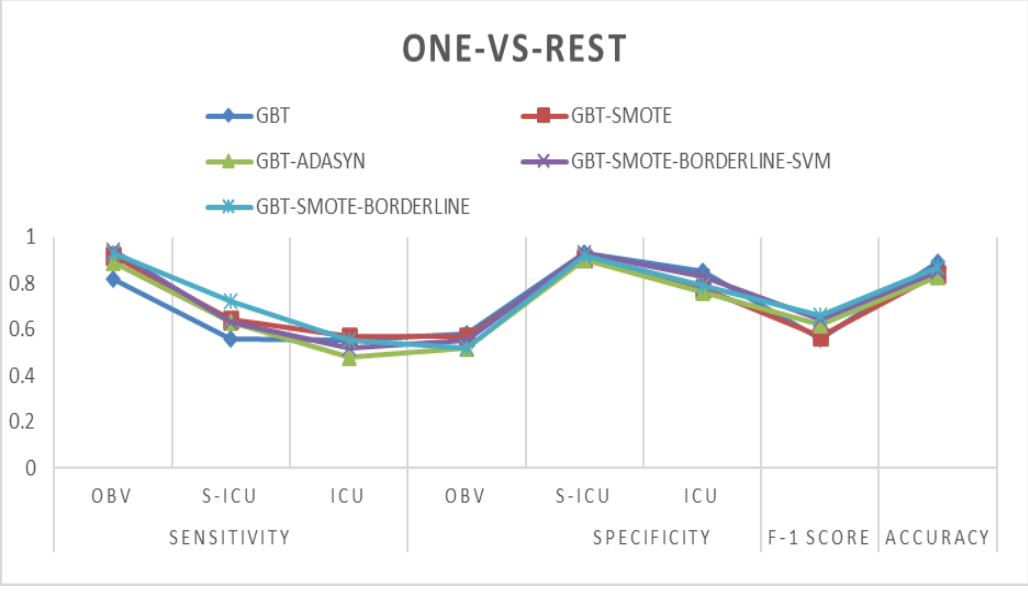


Figure 1.39 *ONE-vs-Rest -GBT*

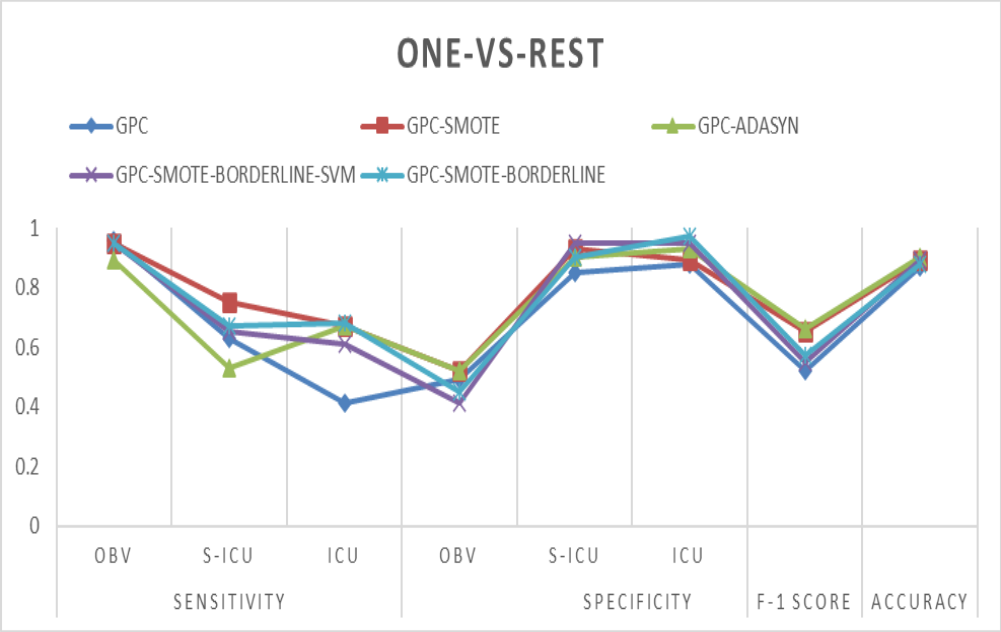


Figure 1.40 *KNN = 2 dataset - ONE-vs-Rest GPC*

The results of the evaluation of the Multivariate imputation by Chained Equation (MICE) algorithm as a data imputation method are displayed in the tables below. Tables 1.37-1.40 illustrate the results of using oversampling techniques, while Table 1.36 displays the results of using single models without oversampling. The techniques used were SMOTE, ADASYN, SMOTE SVM, and Borderline SMOTE. The findings of the ONE-vs-Rest approach are presented in Tables 1.41 and 1.42.

Table 1.36 *MICE dataset: single models*

Models	**sensitivity**			**specificity**			**F-1 Score**
	OBV	*S-ICU*	*ICU*	*OBV*	*S-ICU*	*ICU*	
GNB	0.75	0.33	0.25	0.35	0.33	0.29	0.54
KNN	0.76	0.22	0.125	0.43	0.56	0.49	0.47
DT	0.67	0.22	0.25	0.53	0.57	0.46	0.54
RF	0.71	0.22	0.25	0.43	0.58	0.54	0.53
MLP	0.73	0.44	0.13	0.39	0.47	0.59	0.48
EXT	0.74	0.22	0.12	0.31	0.59	0.59	0.42

Table 1.37 *MICE dataset – SMOTE*

Models	**sensitivity**			**specificity**			**F-1 Score**
	OBV	*S-ICU*	*ICU*	*OBV*	*S-ICU*	*ICU*	
GNB	0.82	0.55	0.37	0.7	0.83	0.96	0.52
KNN	0.89	0.44	0.37	0.47	0.9	0.98	0.56
DT	0.85	0.44	0.125	0.41	0.89	0.45	0.44
RF	0.95	0.66	0.125	0.52	0.95	0.99	0.57
MLP	0.93	0.44	0.37	0.52	0.93	0.99	0.6
EXT	0.95	0.44	0.125	0.35	0.96	0.99	0.53

Table 1.38 *MICE dataset - ADASYN*

Models	sensitivity			specificity			F-1 Score
	OBV	*S-ICU*	*ICU*	*OBV*	*S-ICU*	*ICU*	
GNB	0.79	0.55	0.5	0.7	0.8	0.97	0.55
KNN	0.91	0.44	0.37	0.52	0.9	0.99	0.58
DT	0.84	0.55	0.125	0.52	0.9	0.93	0.46
RF	0.88	0.66	0.125	0.64	0.87	0.99	0.5
MLP	0.91	0.55	0.37	0.58	0.91	0.99	0.61
EXT	0.95	0.44	0.125	0.35	0.96	0.99	0.53

Table 1.39 *MICE dataset - SVM SMOTE*

Models	sensitivity			specificity			F-1 Score
	OBV	*S-ICU*	*ICU*	*OBV*	*S-ICU*	*ICU*	
GNB	0.93	0.55	0.62	0.47	0.93	0.99	0.52
KNN	0.91	0.33	0.25	0.35	0.92	0.99	0.52
DT	0.92	0.55	0.12	0.47	0.96	0.99	0.52
RF	0.92	0.66	0.25	0.58	0.93	0.98	0.59
MLP	0.91	0.55	0.37	0.58	0.91	0.99	0.61
EXT	0.98	0.33	0.12	0.29	0.96	0.99	0.44

Table 1.40 *MICE dataset - borderline SMOTE*

Models	sensitivity			specificity			F-1 Score
	OBV	*S-ICU*	*ICU*	*OBV*	*S-ICU*	*ICU*	
GNB	0.87	0.55	0.12	0.52	0.87	0.99	0.48
KNN	0.91	0.33	0.12	0.29	0.92	0.99	0.46
DT	0.94	0.66	0.1	0.52	0.92	0.99	0.47
RF	0.86	0.33	0.12	0.41	0.88	0.96	0.42
MLP	0.95	0.55	0.37	0.58	0.93	0.99	0.65
EXT	0.98	0.33	0.12	0.23	0.97	0.99	0.45

Table 1.41 *MICE dataset - ONE-vs-Rest- GBT*

Models	sensitivity			specificity			F-1 Score
	OBV	*S-ICU*	*ICU*	*OBV*	*S-ICU*	*ICU*	
GBT	0.95	0.66	0.45	0.58	0.93	0.99	0.56
SMOTE	0.9	0.54	0.67	0.47	0.91	0.98	0.54
ADASYN	0.89	0.63	0.68	0.52	0.9	0.96	0.61
B-SVM	0.94	0.53	0.62	0.35	0.93	0.99	0.54
BORDERLINE	0.94	0.64	0.65	0.52	0.92	0.99	0.62

Table 1.42 *MICE dataset - ONE-vs-Rest- GPC*

Models	sensitivity			specificity			F-1 Score
	OBV	*S-ICU*	*ICU*	*OBV*	*S-ICU*	*ICU*	
GPC	0.86	0.43	0.41	0.59	0.95	0.99	0.46
SMOTE	0.85	0.55	0.57	0.52	0.95	0.99	0.65
ADASYN	0.89	0.33	0.57	0.52	0.9	0.96	0.66
B-SVM	0.94	0.55	0.52	0.51	0.95	0.99	0.55
BORDERLINE	0.9	0.55	0.62	0.55	0.9	0.97	0.57

Figures 1.41-1.47 show the results of using a dataset with missing data that was imputed using the Multivariate imputation by Chained Equation (MICE) algorithm. Figure 1.41 displays the results of utilizing a single model, which implies applying an algorithm without the use of oversampling methods. SVM-SMOTE and Borderline SMOTE findings are shown in Figures 1.44 and 1.45, respectively, whereas SMOTE and ADASYN results are shown in Figures 1.42 and 1.43. Figures 1.46 and 1.47 show the outcomes of the ONE-vs-Rest strategy.

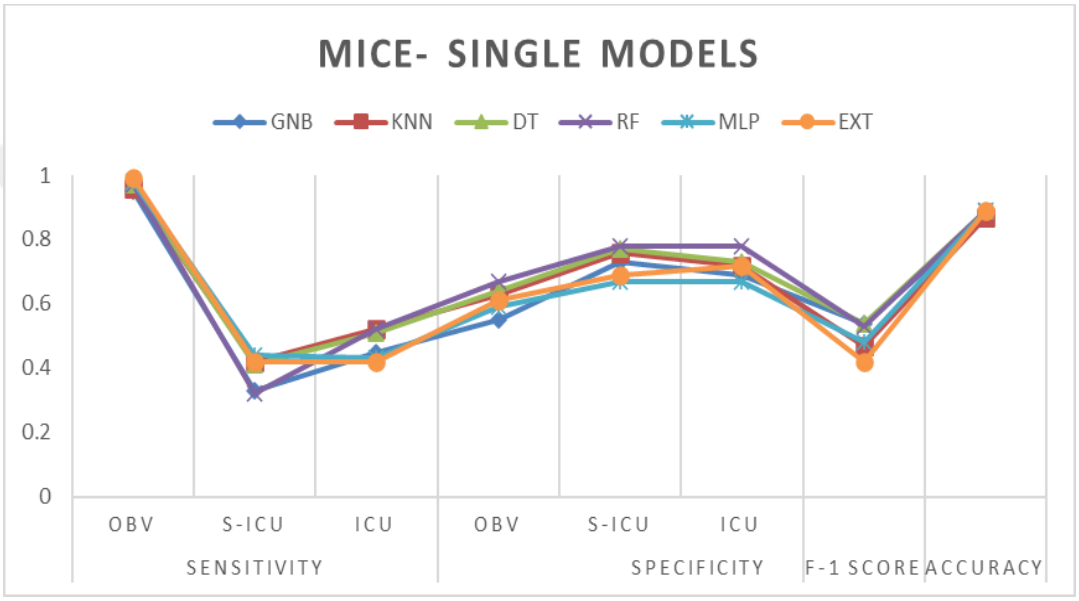


Figure 1.41 *MICE dataset - single models*

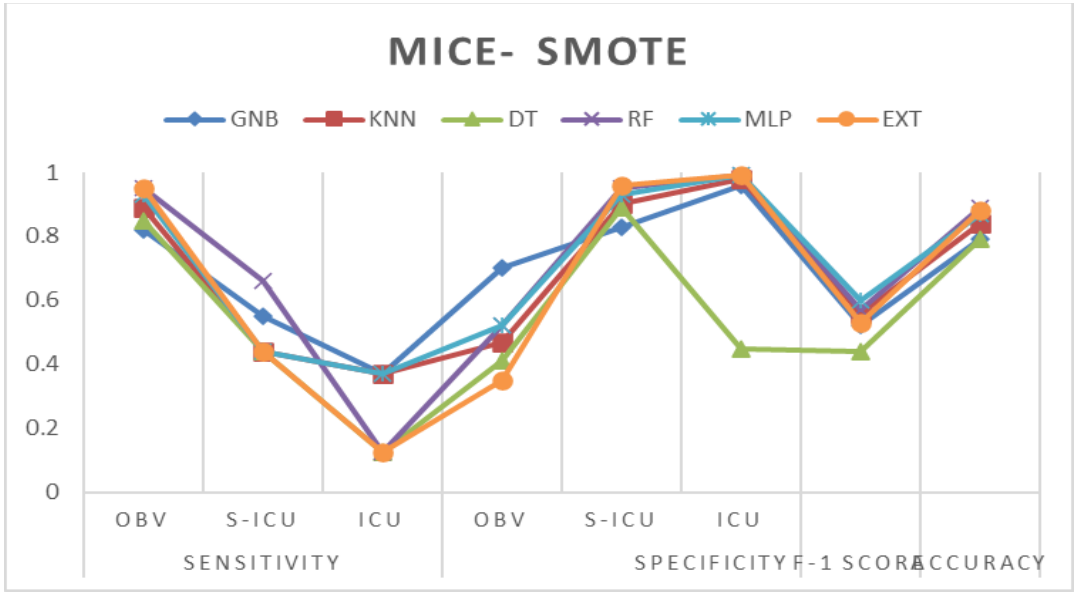


Figure 1.42 *MICE dataset- SMOTE*

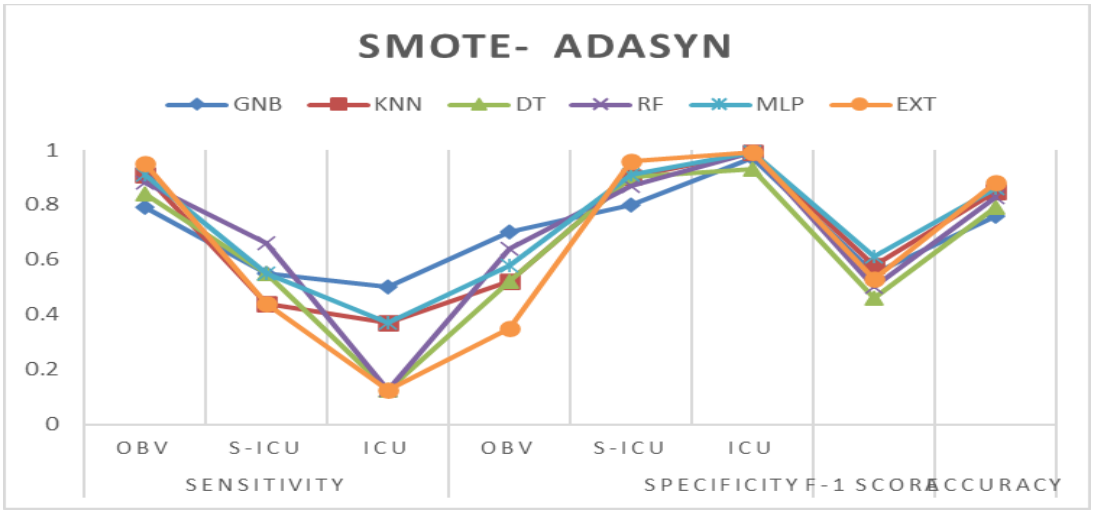


Figure 1.43 *MICE dataset- ADASYN*

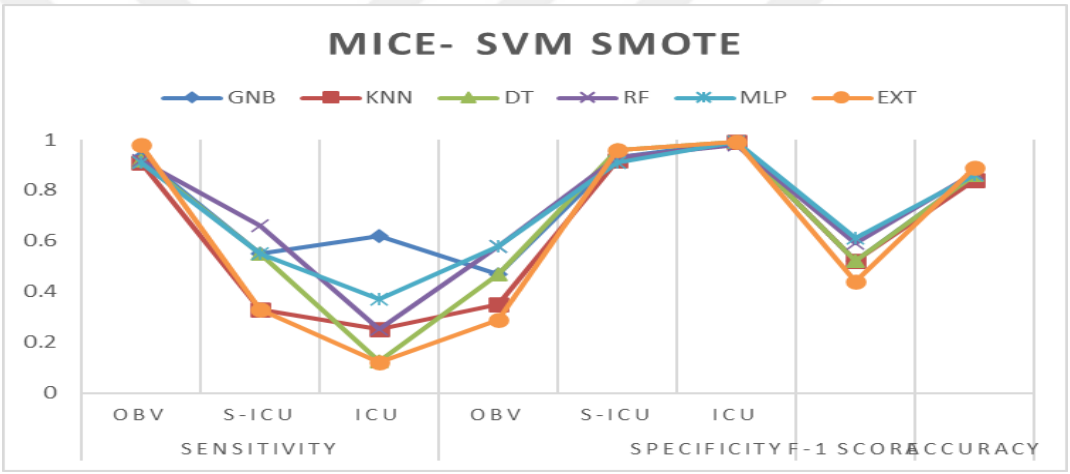


Figure 1.44 *MICE dataset - SVM SMOTE*

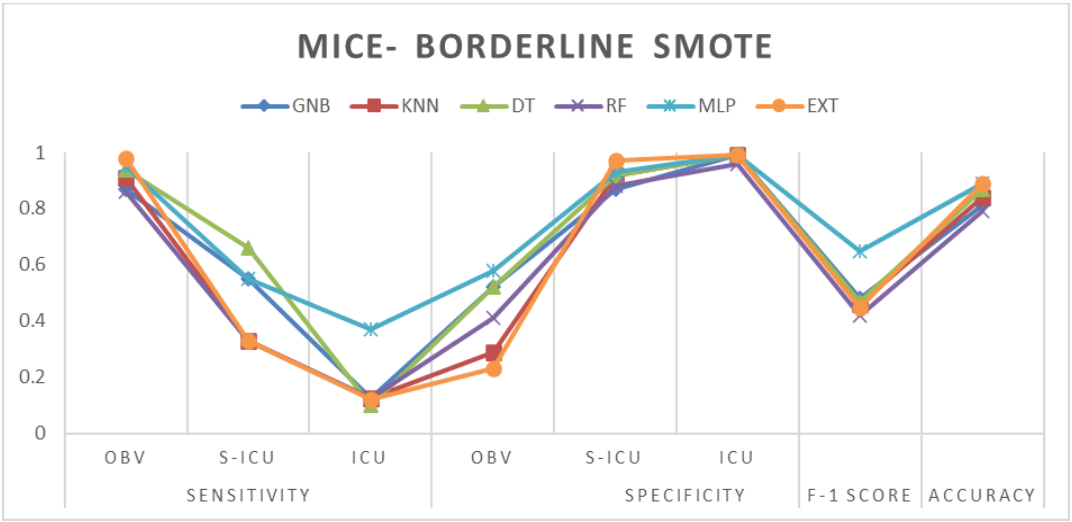


Figure 1.45 *MICE dataset - borderline SMOTE*

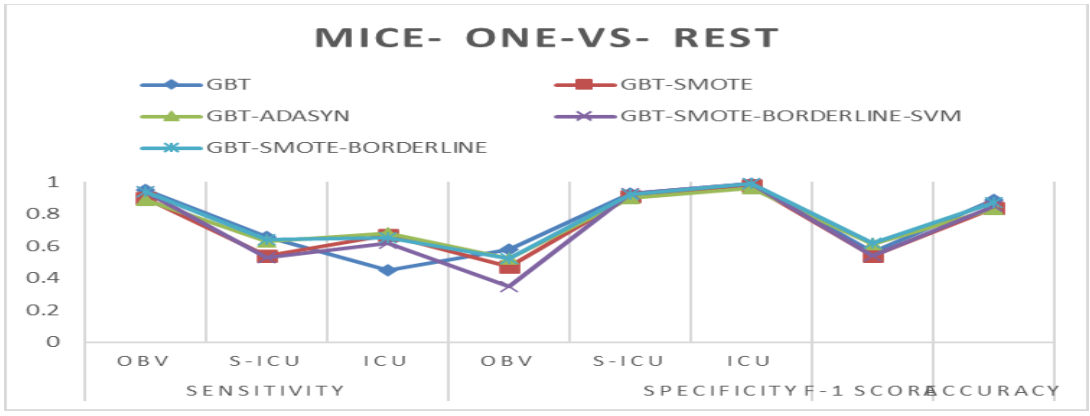


Figure 1.46 *MICE dataset - ONE-vs-Rest -GBT*

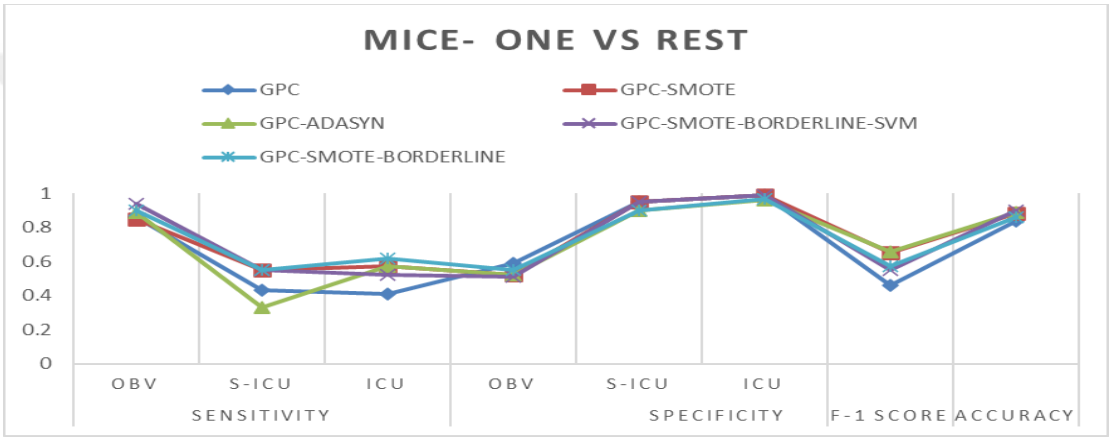


Figure 1.47 *MICE dataset - ONE-vs-Rest -GPC*

As a data imputation method, the Multivariate Imputation by Chained Equation (MICE) algorithm was combined with the LightGBM algorithm. The acquired results are shown in the tables below. Tables 1.44-1.47 show the outcomes of utilizing oversampling techniques, whereas Table 1.43 shows the results of using single models without oversampling. SMOTE, ADASYN, SMOTE SVM, and Borderline SMOTE were the approaches employed. Tables 1.48 and 1.49 show the results of the ONE-vs-Rest method.

Table 1.43 *Missforest dataset - single models*

Models	**sensitivity**			**specificity**			**F-1 Score**
	OBV	*S-ICU*	*ICU*	*OBV*	*S-ICU*	*ICU*	
GNB	0.82	0.55	0.37	0.7	0.83	0.96	0.54
KNN	0.29	0.92	0.99	0.91	0.33	0.12	0.54
DT	0.23	0.97	1	0.97	0.22	0.25	0.44
RF	0.29	0.98	0.98	0.97	0.22	0.25	0.53
MLP	0.29	0.97	1	0.98	0.44	0.1	0.48
EXT	0.11	0.99	1	0.99	0.22	0.1	0.42

Table 1.44 *Missforest dataset – SMOTE*

Models	sensitivity			specificity			F-1 Score
	OBV	*S-ICU*	*ICU*	*OBV*	*S-ICU*	*ICU*	
GNB	0.7	0.83	0.96	0.82	0.55	0.37	0.52
KNN	0.47	0.9	0.98	0.89	0.44	0.37	0.56
DT	0.41	0.89	0.95	0.85	0.44	0.12	0.44
RF	0.52	0.95	0.99	0.88	0.66	0.12	0.5
MLP	0.52	0.93	0.99	0.93	0.44	0.37	0.6
EXT	0.35	0.96	0.99	0.95	0.44	0.12	0.53

Table 1.45 *Missforest dataset – ADASYN*

Models	sensitivity			specificity			F-1 Score
	OBV	*S-ICU*	*ICU*	*OBV*	*S-ICU*	*ICU*	
GNB	0.7	0.8	0.96	0.95	0.33	0.25	0.54
KNN	0.47	0.9	0.98	0.89	0.44	0.37	0.56
DT	0.35	0.96	0.99	0.95	0.44	0.12	0.53
RF	0.64	0.87	0.99	0.88	0.66	0.12	0.5
MLP	0.58	0.91	0.99	0.91	0.55	0.37	0.61
EXT	0.35	0.96	0.99	0.95	0.44	0.12	0.53

Table 1.46 *Missforest dataset - SVM SMOTE*

Models	Sensitivity			Specificity			F-1 Score
	OBV	*S-ICU*	*ICU*	*OBV*	*S-ICU*	*ICU*	
GNB	0.47	0.93	0.94	0.93	0.55	0.52	0.52
KNN	0.35	0.92	0.93	0.91	0.43	0.55	0.52
DT	0.47	0.91	0.92	0.92	0.55	0.51	0.52
RF	0.58	0.93	0.94	0.92	0.66	0.62	0.59
MLP	0.64	0.81	0.87	0.83	0.67	0.52	0.57
EXT	0.29	0.96	0.86	0.98	0.43	0.61	0.51

Table 1.47 *Missforest dataset - borderline SMOTE*

Models	Sensitivity			Specificity			F-1 Score
	OBV	*S-ICU*	*ICU*	*OBV*	*S-ICU*	*ICU*	
GNB	0.52	0.87	0.79	0.87	0.55	0.42	0.48
KNN	0.69	0.92	0.89	0.91	0.56	0.62	0.56
DT	0.77	0.91	0.91	0.92	0.55	0.52	0.52
RF	0.76	0.93	0.87	0.95	0.53	0.57	0.6
MLP	0.78	0.93	0.78	0.95	0.55	0.67	0.65
EXT	0.63	0.98	0.71	0.98	0.63	0.61	0.54

Table 1.48 *Missforest dataset - ONE-vs- Rest -GBT*

Models	Sensitivity			Specificity			F-1 Score
	OBV	*S-ICU*	*ICU*	*OBV*	*S-ICU*	*ICU*	
GBT	0.79	0.66	0.42	0.58	0.72	0.79	0.56
SMOTE	0.89	0.64	0.57	0.47	0.91	0.88	0.57
ADASYN	0.9	0.53	0.58	0.52	0.89	0.85	0.52
B-SVM	0.92	0.63	0.52	0.55	0.93	0.89	0.54
BORDERLINE	0.92	0.64	0.55	0.52	0.92	0.79	0.64

Table 1.49 *Missforest dataset - ONE-vs- Rest -GPC*

Models	Sensitivity			Specificity			F-1 Score
	OBV	*S-ICU*	*ICU*	*OBV*	*S-ICU*	*ICU*	
GPC	0.96	0.33	0.41	0.29	0.95	0.99	0.42
SMOTE	0.95	0.55	0.47	0.52	0.95	0.99	0.65
ADASYN	0.89	0.33	0.57	0.52	0.9	0.96	0.66
B-SVM	0.95	0.55	0.52	0.41	0.95	0.99	0.55
BORDERLINE	0.95	0.55	0.56	0.45	0.9	0.97	0.57

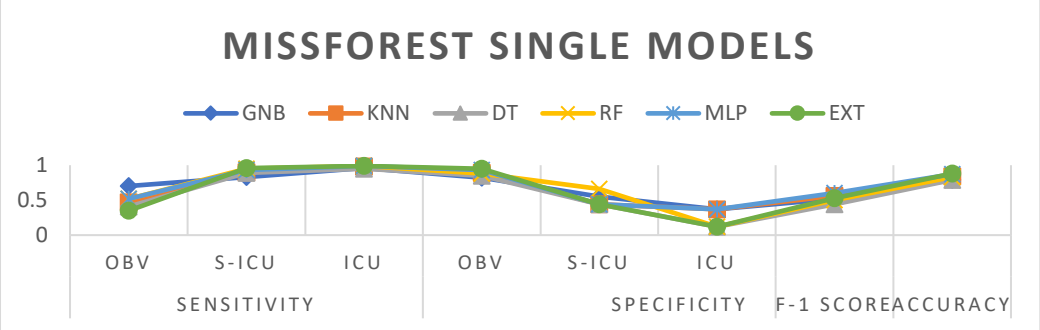


Figure 1.48 *Missforest - single models*

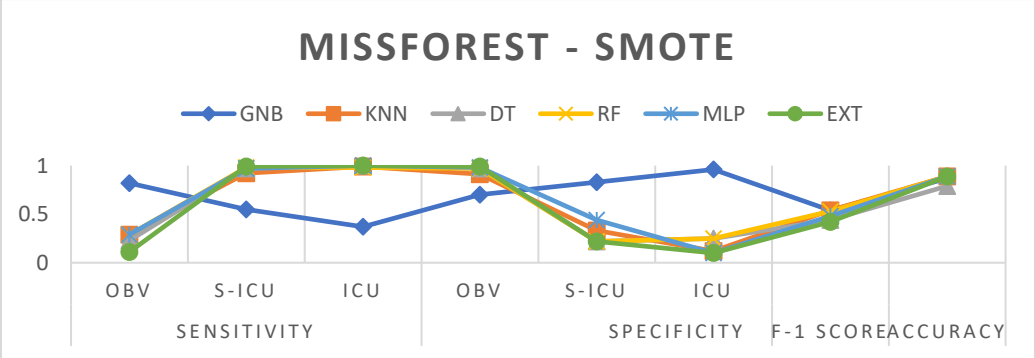


Figure 1.49*: Missforest – SMOTE*

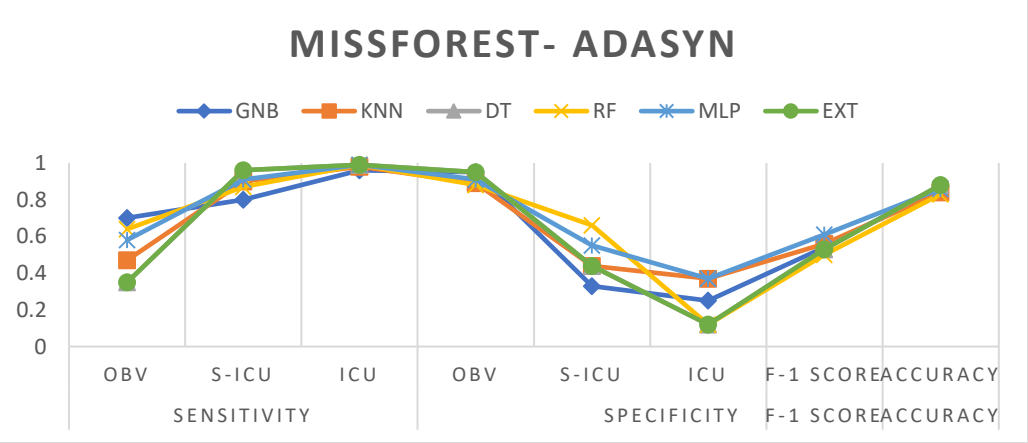


Figure 1.50 *Missforest - ADASYN*

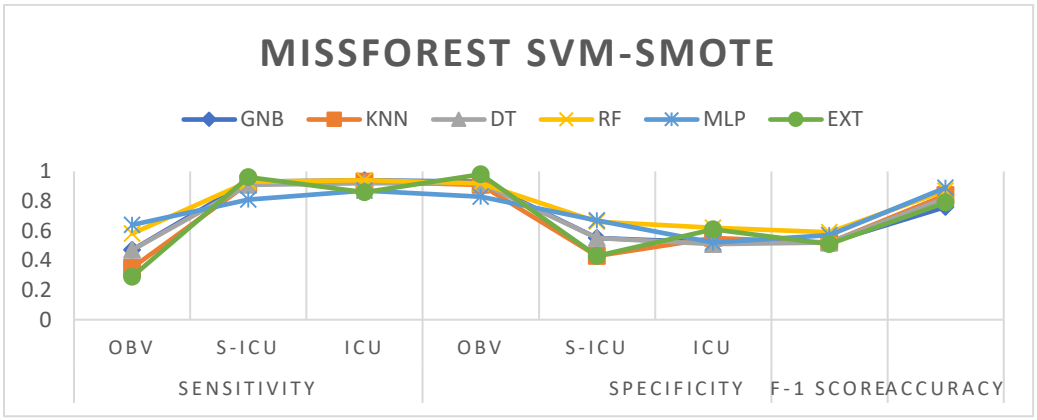


Figure 1.51 *Missforest - SVM SMOTE*

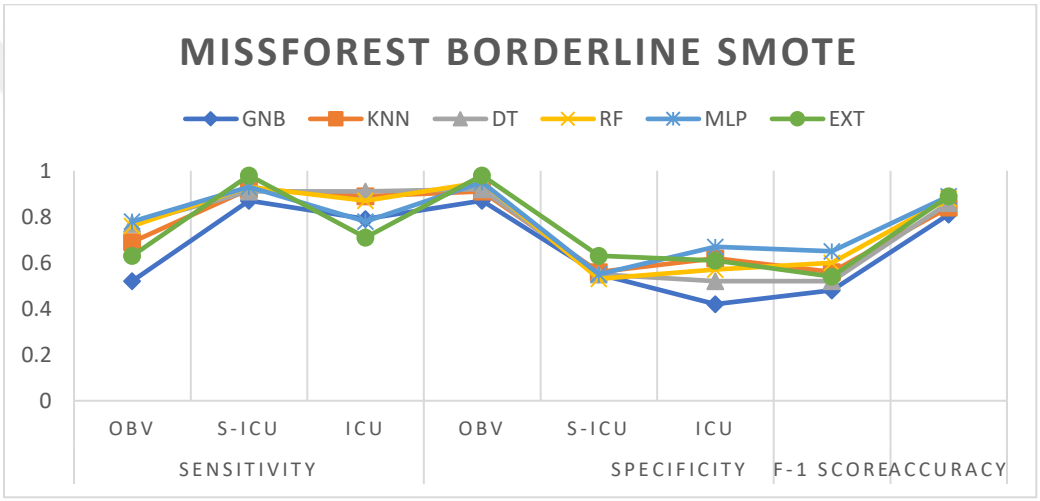


Figure 1.52 *Missforest - Borderline SMOTE*

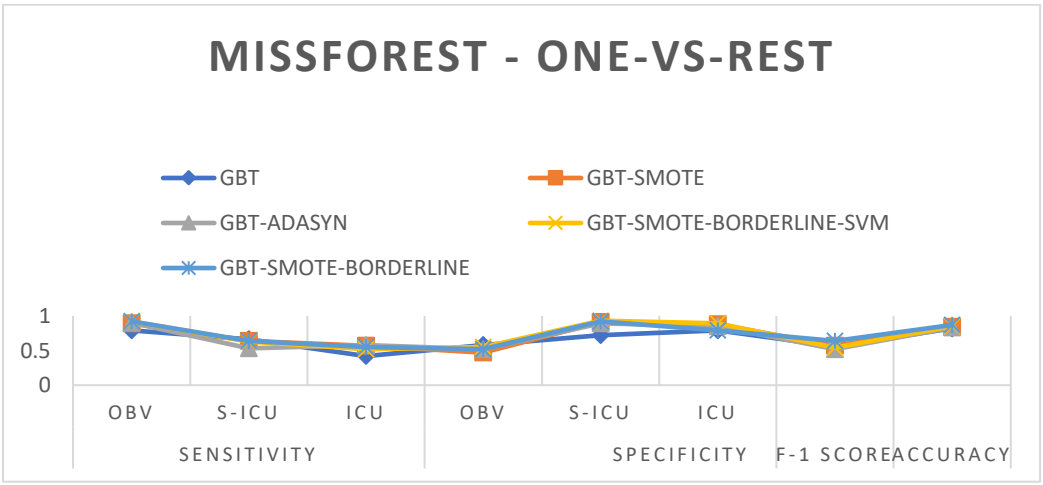


Figure 1.53 *Missforest - ONE-vs- Rest-GBT*

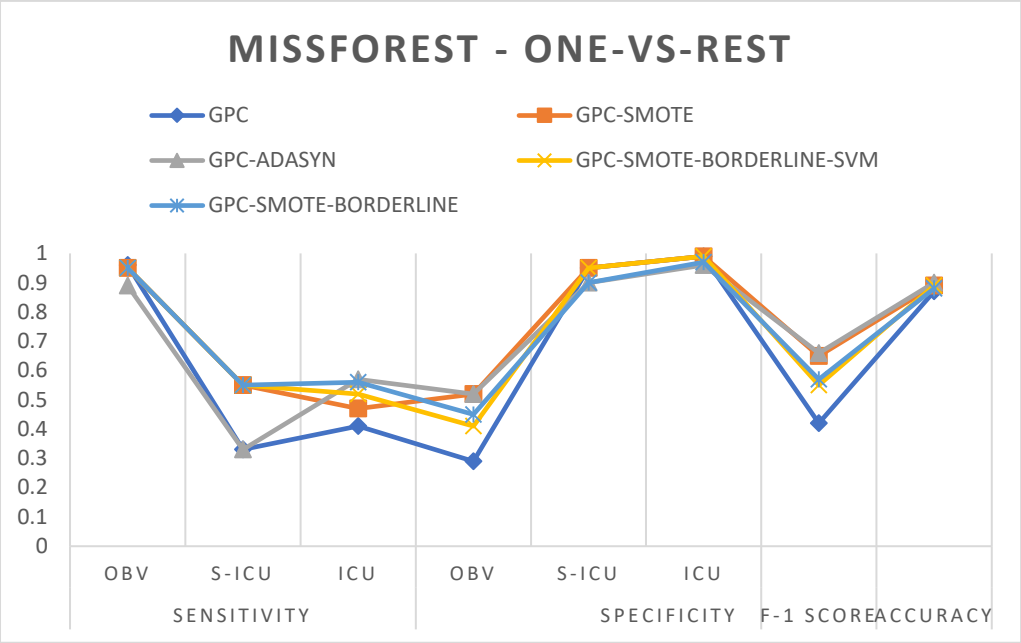


Figure 1.54 *Missforest - ONE-vs-Rest – GPC*

The outcomes of employing a dataset with missing data that was imputed using the Multivariate imputation by Chained Equation (MICE) in conjunction with the LightGBM algorithm are shown in Figures 1.48-1.54. The outcomes of using a single model, or running an algorithm without using oversampling techniques, are shown in Figure 1.48. Figures 1.51 and 1.52, which exhibit the results of SVM SMOTE and Borderline SMOTE, respectively, as well as Figures 1.49 and 1.50, which show the results of SMOTE and ADASYN. The outcome of the ONE-vs-Rest method is shown in Figures 1.53 and 1.54. In this study, various missing data imputation techniques were investigated, and multiple oversampling techniques were employed to enhance the performance of the models. SMOTE, ADASYN, Borderline SMOTE-SVM, and Borderline SMOTE have been the subject of experiments. As shown in the Figures above, the acquired findings indicate that the combination of oversampling methodologies with single models produced superior outcomes. Both the sensitivity and specificity ratings improved significantly. The created and used datasets were compared in order to check which dataset performed well. An extra tree algorithm combined with oversampling techniques was used as an evaluation. The obtained result is shown in the figures and tables below. Table 1.50-1.53 illustrates the results obtained after comparing the scores obtained while investigating the aforementioned created datasets.

Table 1.50. *Datasets comparison – SMOTE*

Models	sensitivity			specificity		
	OBV	*S-ICU*	*ICU*	*OBV*	*S-ICU*	*ICU*
KNN=5	0.93	0.98	0.94	0.94	0.88	0.5
KNN=2	0.94	0.92	0.98	0.93	0.88	0.5
KNN=10	0.88	0.93	0.98	0.94	0.77	0.5
MICE	0.95	0.44	0.12	0.35	0.96	0.99
Missforest	0.11	0.99	1	0.99	0.22	0.1

Table 1.51. *Datasets comparison - ADASYN*

Models	sensitivity			specificity		
	OBV	*S-ICU*	*ICU*	*OBV*	*S-ICU*	*ICU*
KNN=5	0.76	0.93	0.98	0.95	0.77	0.5
KNN=2	0.88	0.92	0.98	0.95	0.77	0.37
KNN=10	0.88	0.93	0.98	0.94	0.78	0.5
MICE	0.95	0.44	0.37	0.58	0.91	0.99
Missforest	0.35	0.96	0.99	0.95	0.44	0.12

Table 1.52. *Datasets comparison - SVM SMOTE*

Models	sensitivity			specificity		
	OBV	*S-ICU*	*ICU*	*OBV*	*S-ICU*	*ICU*
KNN=5	0.76	0.93	0.98	0.95	0.77	0.5
KNN=2	0.82	0.93	0.98	0.95	0.77	0.37
KNN=10	0.76	0.93	0.98	0.94	0.78	0.25
MICE	0.98	0.33	0.12	0.29	0.96	0.99
Missforest	0.29	0.96	0.86	0.98	0.43	0.61

Table 1.53. *Datasets comparison - borderline SMOTE*

Models	sensitivity			specificity		
	OBV	*S-ICU*	*ICU*	*OBV*	*S-ICU*	*ICU*
KNN=5	0.82	0.93	0.98	0.95	0.88	0.55
KNN=2	0.82	0.94	0.98	0.95	0.78	0.37
KNN=10	0.82	0.94	0.98	0.95	0.78	0.37
MICE	0.98	0.33	0.12	0.23	0.97	0.99
Missforest	0.63	0.98	0.71	0.98	0.63	0.61

Figure 1.55-1.58 displays the obtained results while comparing the scores in terms of sensitivity and specificity obtained on the created datasets. The results show that the datasets imputed using the KNN algorithm performed well in comparison with the datasets imputed by the multivariate imputation by chained equation (MICE) and MICE combined with the LightGBM algorithm.

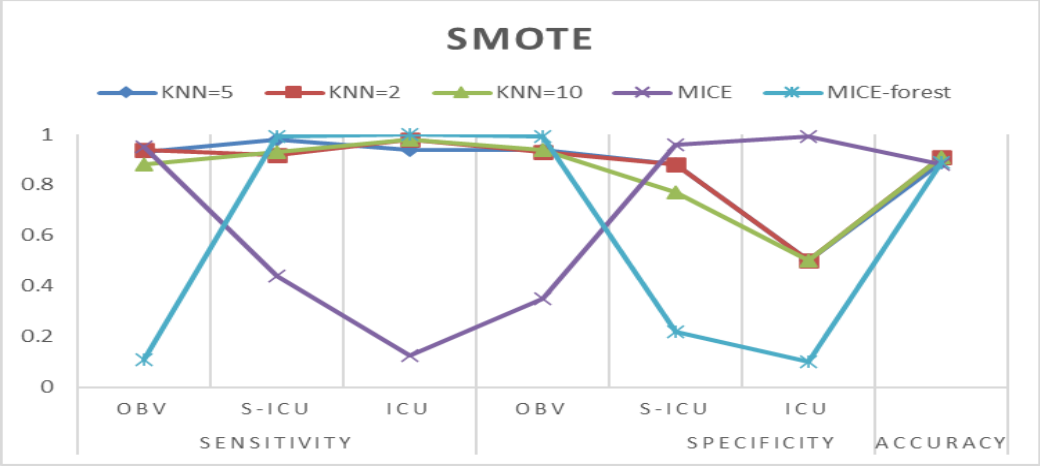


Figure 1.55 *Dataset comparison: EXTRA TREE SMOTE*

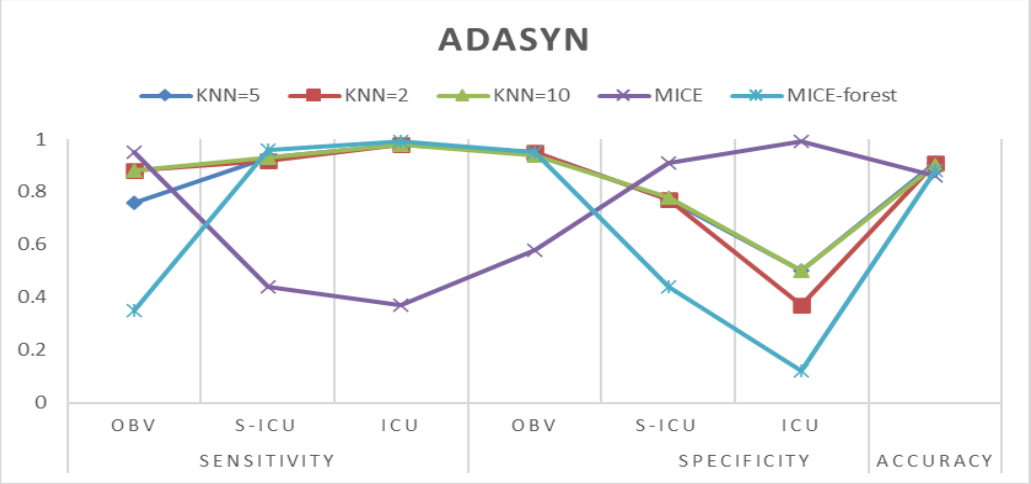


Figure 1.56 *Dataset comparison: EXTRA TREE ADASYN*

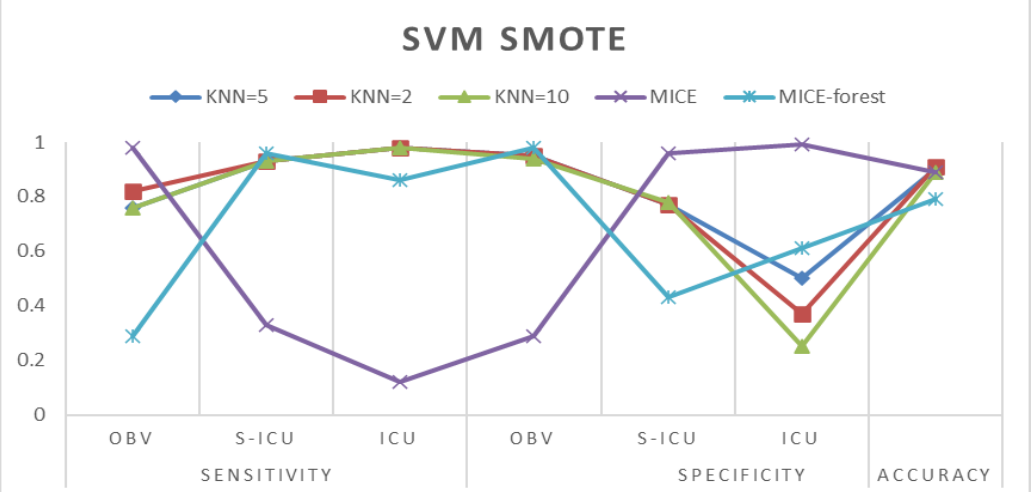


Figure 1.57 *Dataset comparison EXTRA TREE SVM SMOTE*

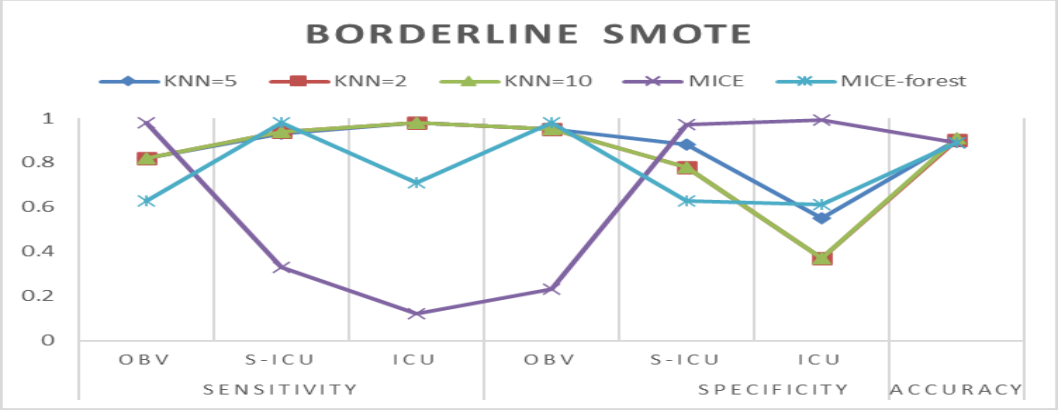


Figure 1.58 *Dataset comparison: EXTRA TREE BORDERLINE SMOTE*

As can be seen in the figures above, the acquired result demonstrated that machine learning techniques may be utilized to predict the ideal healthcare unit for a patient who tested positive for COVID-19. The acquired results also demonstrated that the methods used to impute the missing data have an impact on the results since datasets imputed using the KNN algorithm fared better than datasets imputed using multivariate imputation chained equations (MICE).

1.7. Discussion (Part 1)

This study examines whether machine learning can be used to predict COVID-19. The labeled dataset used contains 111 features such as blood tests, respiratory virus testing, and patient age. The study employed supervised learning. Since the target classes in the dataset are skewed, oversampling and machine learning techniques were employed. In the first phase, exploratory data analysis allowed us to understand the data used, generate a hypothesis, and test it with a T-test. The next steps were preprocessing, data cleaning, feature selection, and feature engineering. To achieve the set objectives, a large number of machine learning models are constructed. The purpose was to diagnose COVID-19 from routine clinical data using machine learning models. The oversampling algorithms SMOTE, SVMSMOTE, Random Oversampler, Borderline SMOTE, and ADASYN improved the model's performance. This study sought to increase the recall score to reduce false negatives. Dimensionality reduction techniques, such as PCA and LDA, enhanced the performance of models more than using merely selected features. In this initial phase, ensemble learning techniques such as bagging, boosting, and stacking generated improved results with higher recall scores. Level-1 stacking models performed better than Level-2 and Level-3 stacking models. This study also investigated novelty and

outlier detection techniques; COVID-19-positive patients were considered outliers since they belonged to the minority class. The recall scores for the one-SVM, one-SVMSGD, and local outlier factor algorithms were 84%, 83%, and 78%, respectively. These tests demonstrated that the small amount of data used to train the models hindered their performance. This study demonstrates that machine learning can assist physicians in decision-making. The lack of enough data to train the models and missing values hampered the performance of the models. This study utilized KNN and MICE imputation techniques; hence, further investigation into alternative data imputation approaches is recommended. As seen in the figures above, the acquired result demonstrated that machine learning may be utilized to predict the ideal healthcare department for a patient who tested positive for COVID-19. The acquired results also demonstrated that the methods used to impute the missing data have an impact on the results, since datasets imputed using the KNN algorithm fared better than datasets imputed using multivariate imputation chained equations (MICE).

2. MULTICOLLINEARITY AND MULTIVARIATE IMPUTATION: A CASE STUDY ON CONTINUOUS DATA WITH MICE ALGORITHM

The preceding part explored machine learning approaches for predicting and diagnosing COVID-19 based on clinical data, whereas the subsequent segment aims to determine the most adequate healthcare facility for treating COVID-19 patients. However, the used data contained a significant number of missing values, with several variables exhibiting a missingness ratio exceeding 70%. This issue had a significant impact on the result, necessitating the imputation of missing data. Consequently, the problem prompted a focused investigation into missing data imputation techniques to evaluate their effectiveness across different scenarios. A challenge arose due to the significant correlation among multiple variables, which had an impact on the outcome. The following work evaluates the impact of imputation mechanisms on multicollinearity. The MICE algorithm, along with several other imputation methods, was utilized and examined.

2.1. Background

Handling missing data is a common and crucial challenge in the fields of statistical analysis and machine learning. Inadequately addressing missing data can result in biased predictions, diminished statistical power, and deceptive conclusions (Zhang and Long, 2021). Several researchers proposed various approaches to tackle the challenges related to missing data. When it comes to predicting and filling in missing values, multivariate imputation methods are more reliable than others, like imputation using mean, mode, and median. This is because they take into account the correlations between many variables (Kim et al., 2022). Multiple Imputation by Chained Equations (MICE) is a well-known and highly efficient technique that involves repeatedly imputing missing variables through a series of regression models. MICE statistically models each variable with missing data based on the other variables in the dataset, allowing for the simultaneous analysis of correlations between multiple variables (Austin, Peter, and Buuren, 2023). This method iteratively continues until the imputations converge, resulting in the generation of several complete datasets. One major problem that can arise when using imputation methods is multicollinearity, which happens when the independent variables in a dataset are strongly correlated. Multicollinearity may considerably decrease the stability and predictability of regression models. It increases the variability of coefficient

estimates, making it more difficult to determine the contributions of different variables. Therefore, it is essential to deal with multicollinearity to ensure the reliability of imputed datasets. It can be evaluated using multiple metrics. The variance inflation factor (VIF) is a metric that quantifies the impact of multicollinearity on a regression coefficient's variance. The condition index, Eigenvalues, correlation matrix, and tolerance, which are the opposite of VIF and show the presence of multicollinearity, are some other measures. The study utilized VIF as a metric to assess multicollinearity. High VIF values indicate a substantial degree of multicollinearity, which may undermine the effectiveness of models such as regression models. To preserve the integrity of imputed data, it is critical to understand how various imputation algorithms deal with multicollinearity. This study investigates the impact of multivariate imputation methods on multicollinearity, with a specific emphasis on the MICE algorithm, in the context of continuous data. Multicollinearity in datasets can greatly affect the reliability and accuracy of imputation models. To tackle this issue, a publicly available dataset is used to simulate different types of missing values, employing mechanisms such as MCAR (Missing Completely At Random), MAR (Missing At Random), MNAR (Missing Not At Random), and MNAR with quantile censorship. Missing ratios of 20%, 40%, and 60% were applied to assess the robustness and effectiveness of various imputation methods under different levels of data sparsity. This study will employ multiple estimators, such as XGBRegressor, SVR, Random Forest, KNN, and Decision Tree, to impute missing data through MICE. The variance inflation factor (VIF) was calculated for each dataset after imputation to quantify the level of multicollinearity. VIF values will be compared to evaluate the effectiveness of various imputation methods in reducing multicollinearity. Additionally, the Kruskal-Wallis H Test will be employed to statistically assess differences in VIF values across various imputation techniques.

2.2. Related Work

Noora Shrestha's research, titled "Detecting Multicollinearity in Regression Analysis," explored the strong correlation between customer satisfaction and various important factors such as product quality, brand experience, product features, product attractiveness, and product price. The statistical significance of this association was confirmed with a p-value of less than 0.001. She argued that to guarantee the robustness and dependability of these findings, it is imperative to address the potential issue of

multicollinearity among the predictor variables. Multicollinearity can skew regression analysis, resulting in inaccurate estimations of the relationships between predictors and the dependent variable (Shrestha, 2020). Shrestha's study assessed the presence of multicollinearity using distinct methodologies: Correlation coefficients: Examining the correlation coefficients between pairs of variables to identify significant intercorrelations that suggest the existence of multicollinearity. The eigenvalue approach entails analyzing the eigenvalues of the correlation matrix to detect any linear associations across variables. Shrestha observed that there was no evidence of multicollinearity among the variables under investigation. Shrestha recommends employing advanced regression approaches to effectively detect and address multicollinearity. Principal components regression, weighted regression, and ridge regression are robust techniques for mitigating the influence of multicollinearity and generating more dependable results. In their paper, Benjamin D. Leiby and Darryl K. Ahner introduce a method called "Multicollinearity Applied Stepwise Stochastic Imputation: A Large Dataset Imputation Through Correlation-Based Regression." This method is designed to handle large datasets and address numerical problems that arise when using commercial imputation software. Their approach, based on a comprehensive dataset of conflicts in different countries, addresses the issue of missing values that surpass the commonly accepted threshold of 20% (Leiby and Ahner, 2021). They propose a new method called Multicollinearity Applied Stepwise Stochastic (MASS) imputation methodology. This technique exploits the relationships between variables and utilizes model residuals to approximate missing values, providing insights into the choice of linear or nonlinear modeling terms. The MASS-impute approach incorporates adjustable tolerances that employ residual information to accurately match each data element. The evaluation of this process involves analyzing the computing time, assessing the model fit, and comparing the generated values through imputation with the known values. The findings indicate that MASS-impute generates credible and justifiable imputations for the missing items in the dataset. The initial Large Dataset Imputation through a Correlation-based Regression approach emphasized the trade-off between time, computational capacity, and precision. Nevertheless, the presence of multicollinearity and extreme outliers raised considerable issues. The study by Leiby and Ahner tackles these difficulties by utilizing variable nomination process discounts and constraining imputation estimates within a variable range-based guard rail approach. These improvements boost the credibility and ability to defend the estimated outcomes,

strengthening the methodology's usefulness in managing extensive datasets with substantial missing information (Leiby and Ahner, 2021). Voillet et al. (2016) explore the challenge of missing rows in omics data integration in their work, "Handling Missing Rows in Multi-Omics Data Integration: Multiple Imputation in a Multiple Factor Analysis Framework." The study highlights the difficulty posed by missing values, noting that most statistical methods struggle to effectively handle incomplete datasets. To address this issue, the authors propose using a multiple imputation approach within a multivariate context. They particularly focus on leveraging multiple factor analysis to integrate and compare various layers of data. Multiple Imputation is a technique used to generate several finished datasets by replacing missing rows with plausible values. Each finished dataset undergoes MFA, resulting in M distinct configurations. These configurations are then pooled to generate a single consensus answer. The MI-MFA method provides a reliable approach to estimating the coordinates of persons on the first MFA components, even when there are missing rows. The MI-MFA configurations closely approximated the genuine configuration, even though some individuals were missing from multiple data tables. This approach takes into consideration the ambiguity caused by the absence of particular rows, thereby enabling the assessment of the dependability of the outcomes. In summary, MI-MFA offers a valuable and appealing approach to address missing data in omics investigations, improving the integration and analysis of complex information across multiple layers. Woods et al.(2024), in their study "Best Practices for Addressing Missing Data Through Multiple Imputation," offer practical recommendations for researchers, particularly those conducting developmental studies, to effectively handle missing data. They emphasize the importance of addressing the issue of missing data in developmental research to ensure the accuracy and reliability of findings. They stated that selecting an approach for managing missing data is critical for ensuring the accuracy and reliability of research outcomes and that dealing with missing data is equally important as choosing the appropriate analysis approach. Their paper provides guidelines for reporting multiple imputation methods, setting forth acceptable standards. They have also suggested that researchers should exercise informed and deliberate judgment when dealing with missing data, instead of depending on the default settings of software. Yoo et al.(2014) carried out a simulation study titled "A Study of Effects of Multicollinearity in Multivariable Analysis." The purpose of the study was to explore the effects of multicollinearity by manipulating the correlation structures among predictive and

explanatory variables. This study aims to establish recommendations to determine harmful degrees of collinearity. This study explores different scenarios: bivariate collinear structure, where two variables are associated, and multivariate collinear structure, where an explanatory variable is linked with two others. In a complex scenario, an independent variable can be represented by different functions of other variables, which allows for a more accurate representation of real-world data scenarios. The results for bivariate collinearity demonstrated that when determining significant correlations between two variables, the commonly accepted thresholds vary from 0.7 to 0.9, although some experts argue that a correlation of 0.35 is an indication of a strong link. Regarding multivariate collinearity, the researchers argued that when a variable is correlated with two others, each with correlation coefficients of 0.8 and 0.6, respectively, the regression coefficients can be overstated by a factor of up to 70. If the two correlation coefficients have opposite signs, the overestimation is reduced by a factor of around 10. They also stated that when the correlation coefficients in a correlation matrix have the same signs, it is important to take into account the impacts of multicollinearity.

2.3. Literature Review (Part 2)

2.3.1. Missing data types

Missing data in datasets is categorized into three main types, each with distinct characteristics that affect how they are handled in data analysis.

2.3.1.1. Missing completely at random

Missing completely at random (MCAR) is a scenario where the likelihood of a data point being missing is unrelated to both observed and unobserved data, implying that the probability of missing is consistent across all observations. The following formula illustrates the MCAR scenario.

$$P(R|X_{obs}, X_{mis}) = P(R) \tag{2.1}$$

Here R denotes the missing data indicator, X_{obs} are the observed values, and X_{mis} are the missing values.

2.3.1.2. Missing at random

Missing at random (MAR) is a scenario where the probability of a data point being missed depends on observed data and can be explained by existing variables. This allows

for systematic patterns in missing data, as long as these patterns can be explained by existing variables. The following mathematical formula illustrates the MCAR scenario.

$$P(R|X_{obs}, X_{mis}) = P(R|X_{obs}) \tag{2.2}$$

Here, the missingness is conditioned on the observed data.

2.3.1.3. Missing not at random

Missing not at random (MNAR) occurs when the probability of missing a data point depends on the missing data's values, introducing systematic biases due to non-response bias or measurement errors related to missing values, even after considering observed data. The following formula illustrates the MNAR scenario.

$$P(R|X_{obs}, X_{mis}) \tag{2.3}$$

2.3.2. Multiple imputation by chained equation

Multiple imputation by chained equation (MICE) is a sophisticated iterative imputation and widely used technique for estimating missing values in datasets, unlike simple imputation methods that rely on static summary statistics, iterative imputation models treat each variable with missing values as a function of other variables in the dataset, updating these estimates through a cyclic and iterative process. This approach captures complex dependencies among variables and often leads to more accurate imputations, particularly in datasets with multivariate relationships. The algorithm proceeds as follows:

- Initialization: Start with a dataset X containing missing values. Initialize the missing values with initial estimates (mean, median, or a simple imputation method)
- Iterative Process: Repeat the following steps for a predefined number of iterations or until convergence. For each variable X_j with missing values in the dataset:
- Define the Target and Predictors: Define X_j^{mis} as the missing values of X_j.
- Define X_{-j} as all other variables in the dataset except X_j.
- Formulate the Regression Problem: Use the observed part of (X_j^{obs}) as the dependent variable. Use the corresponding parts of X_{-j} that are observed (X_{-j}^{obs}) as the independent variables.

- Select an Estimator: Choose a regression model or estimator
- Fit the Model: Fit the chosen estimator to the observed data $(X_j^{obs}, X_{-j}^{obs})$
- Predict Missing Values: Use the fitted model to predict the missing values X_j^{mis} using $X_{-j}^{mis.}$
- Update the Dataset: Replace the missing values in X_j with the predicted values X_j^{mis}
- Repeat: Continue cycling through all variables and updating the missing values in the dataset until the imputation process converges or reaches the maximum number of iterations.
- Combine Results: After convergence, the process results in multiple completed datasets

2.3.3. Variance inflation factor

Multicollinearity in regression models can cause errors in coefficient calculations and compromise model interpretability. The Variance Inflation Factor (VIF) measures the increase or decrease in variability between variables. The following mathematical expression illustrates the VIF.

$$VIF(X_j) = \frac{1}{1 - Rj^2} \tag{2.4}$$

Here, R_j^2 is the R-squared value obtained from the regression of X_j on the other predictors. R_j^2 measures the proportion of variance in X_j that can be explained by the other predictors.

VIF can be interpreted as follows:

- VIF = 1: Indicates no correlation between the predictor X_j and the other predictors, implying no multicollinearity.
- 1 < VIF < 5: Suggests moderate correlation, but it is generally considered acceptable.
- VIF ≥ 5: Indicates high correlation, raising concerns about multicollinearity.
- VIF ≥ 10: Typically indicates severe multicollinearity, suggesting that the regression coefficients are poorly estimated and potentially unstable.

High VIF values imply that the regression coefficients are not stable because small changes in the data can lead to large changes in the estimates. Multicollinearity makes it

difficult to determine the individual effect of each predictor on the dependent variable, as the predictors are not independent of each other. Assessing VIF helps in diagnosing multicollinearity issues and deciding whether remedial measures, such as removing predictors, combining predictors, or using regularization techniques, are needed.

2.3.4. Kruskal-wallis test

The Kruskal-Wallis test, also known as the one-way ANOVA on ranks. The Kruskal-Wallis test is a statistical method used to determine if there are significant differences among the medians of three or more independent groups. Unlike traditional ANOVA, it does not assume that the data are normally distributed, making it suitable for ordinal data or continuous data that do not meet the assumptions of ANOVA. The Kruskal-Wallis H test procedure is outlined as follows. First, all data from the different groups are combined and ranked from the lowest to the highest value. If there are ties (identical values), each tied value is given the average rank. Next, the ranks are summed for each group, and the rank sum for group *i* is denoted as *Ri*. Finally, the test statistic is computed based on these rank sums to determine whether there are significant differences between the groups. The Kruskal-Wallis H statistic is given by the following mathematical formula:

$$H = \cos\frac{12}{N(N+1)}\sum_{n=1}^{K}\left(\frac{(R(i)^2}{\mathrm{n(i)}}\right) - 3(N+1) \qquad (2.5)$$

Where *N* is the total number of observations across all groups, *k* is the number of groups, $(i)^2$ is the sum of ranks for the *i-th* group, and (i) is the number of observations in the *i-th* group.

The distribution of the test statistic *H* approximates a chi-square distribution with $k-1$ degrees of freedom. Compare the computed *H* value with the critical value from the chi-square distribution table to determine the p-value. If the p-value is less than the chosen significance level (In this study, a p-value of 0.05 has been utilized), reject the null hypothesis that the medians of all groups are equal. Otherwise, do not reject the null hypothesis.

In this study, the Kruskal-Wallis H test has been utilized instead of the ANOVA test since it presents some advantages, such as the fact that it does not assume normality and it is more robust to outliers and heteroscedasticity compared to ANOVA.

2.4 Experiment And Result

This study utilizes the "SONAR MINE DATASET," a publicly available dataset on the Kaggle website. Various missing data mechanisms, including MCAR, MAR, MNAR, and MNAR with quantile censorship, were applied to the previously described dataset, resulting in 12 datasets with missing data. Different missing data ratios of 20%, 40%, and 60% were applied for each missing data mechanism. The following figures illustrate the datasets generated with missing values. Figures 2.1-2.3 show a summary of the datasets after the Missing Completely at Random (MCAR) mechanism was used to simulate data missingness at 20%, 40%, and 60% rates, respectively.

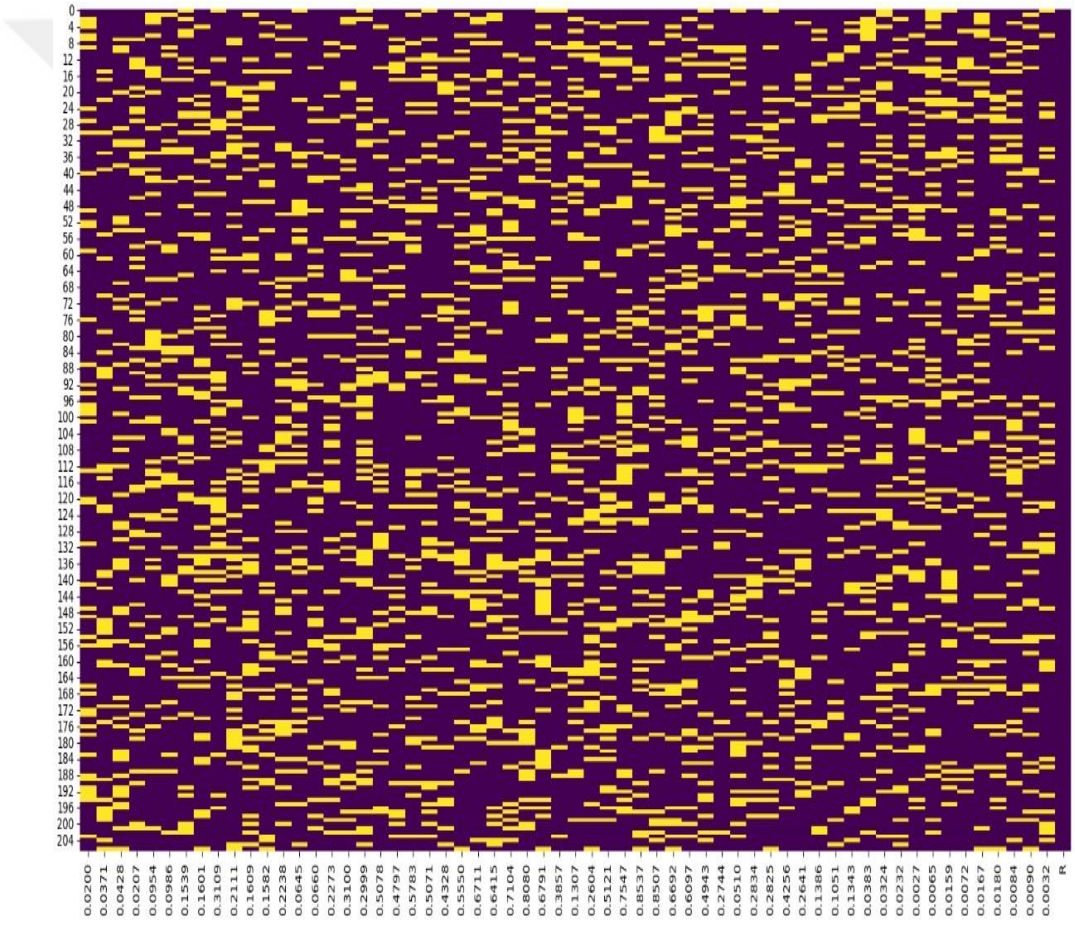


Figure 2.1 *MCAR 20% of missing values*

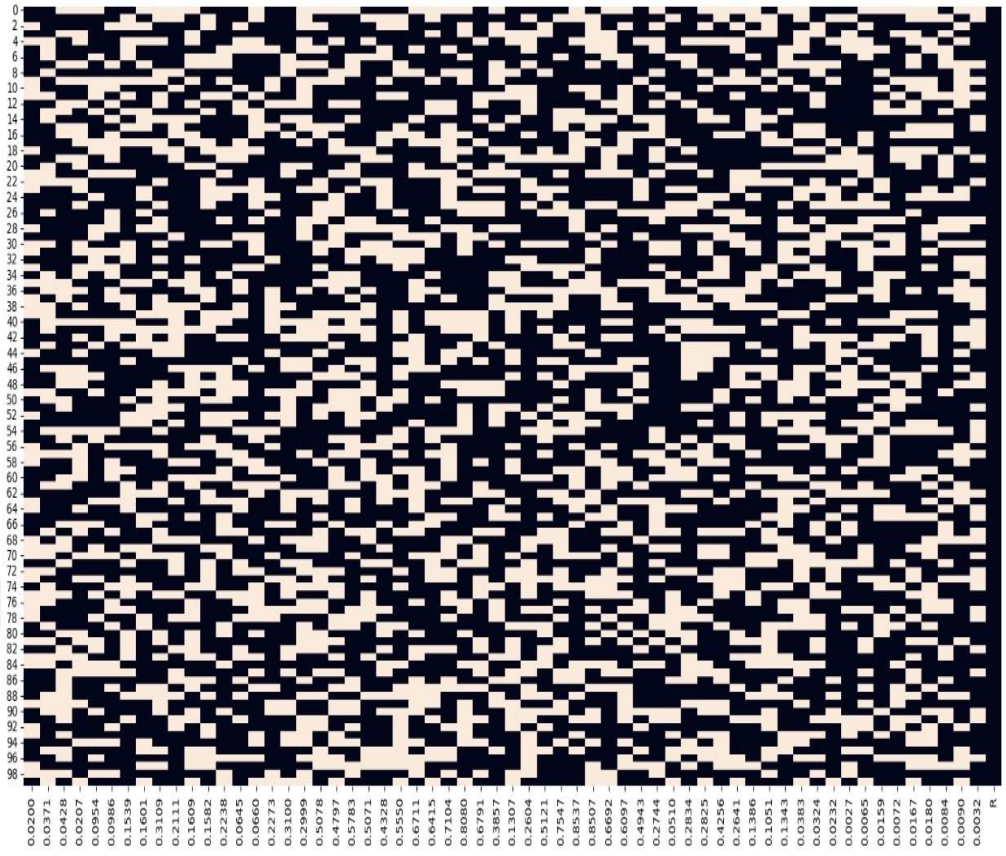


Figure 2.2 *MCAR 40% of missing values*

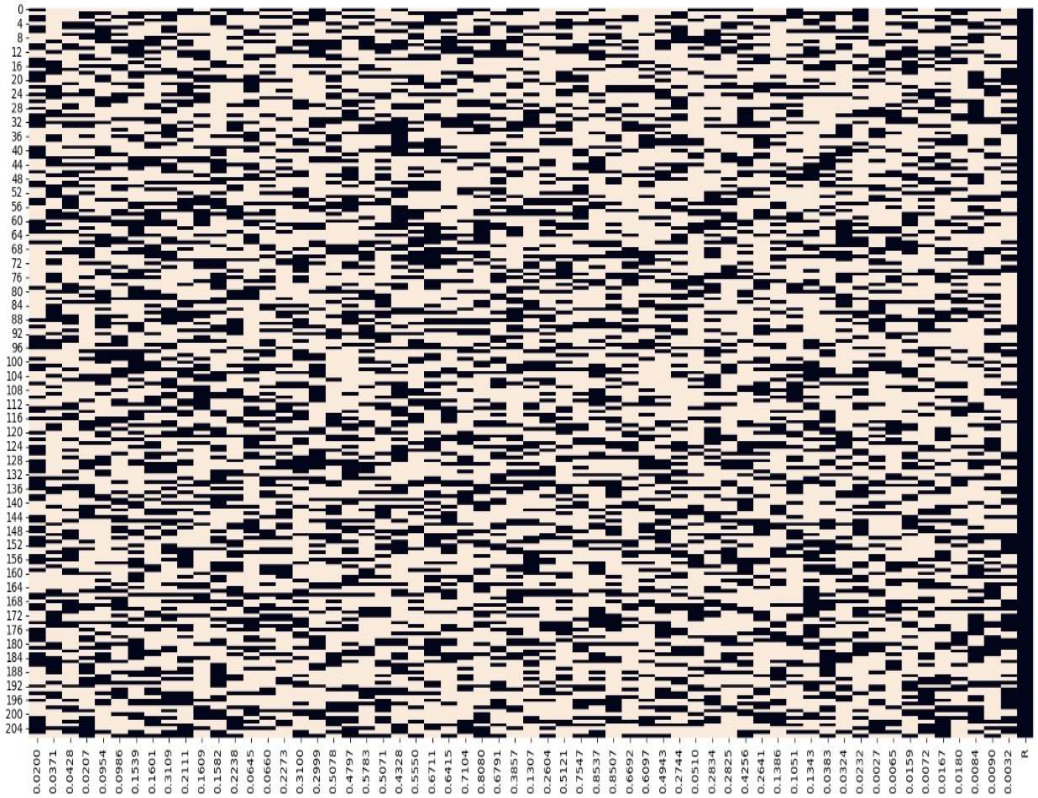


Figure 2.3 *MCAR 60% of missing values*

Figures 2.4-2.6 display the datasets following the implementation of the Missing at Random (MAR) mechanism, which simulated data missingness at rates of 20%, 40%, and 60%, respectively.

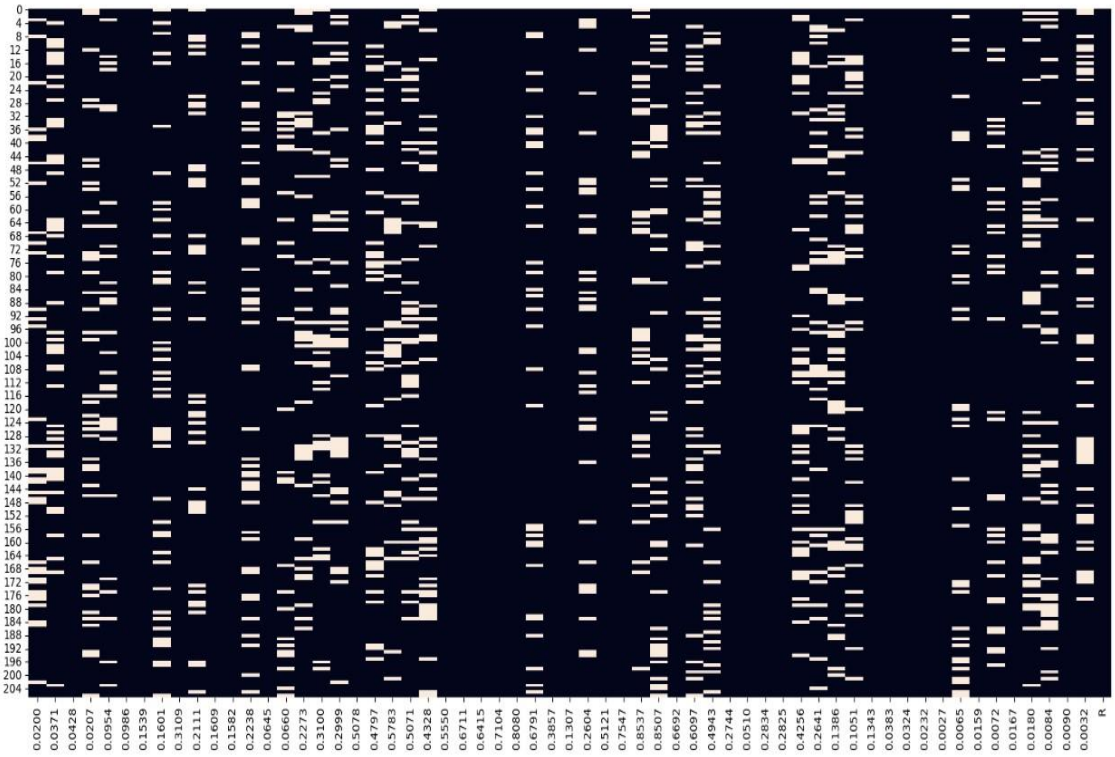


Figure 2.4 *MAR with 20% of missing values*

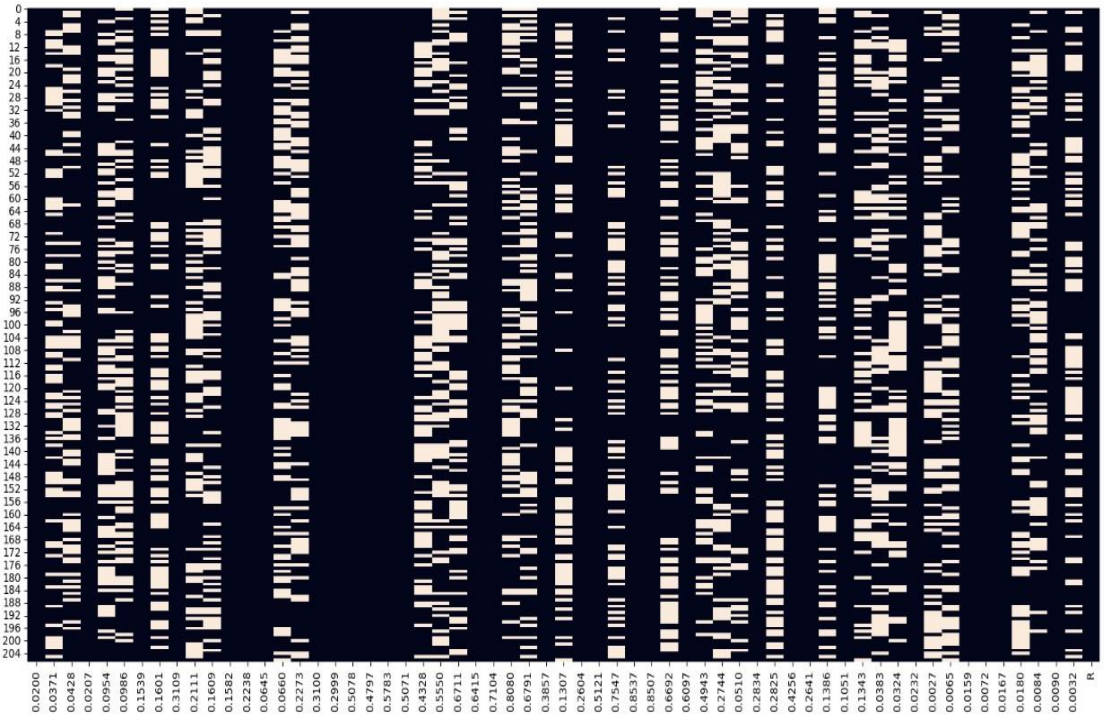


Figure 2.5 *MAR with 40% of missing values*

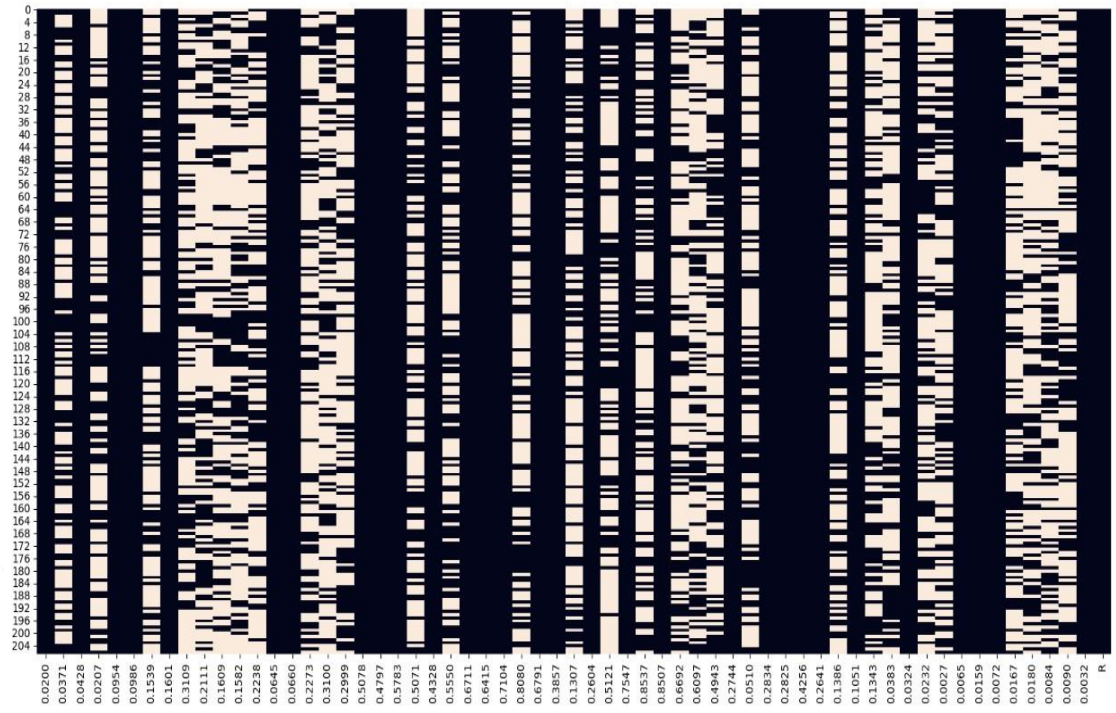


Figure 2.6 *MAR with 60% of missing values*

Figures 2.7-2.9 show a summary of the datasets after the Missing Not at Random (MNAR) mechanism was used to simulate data missingness at 20%, 40%, and 60% rates, respectively.

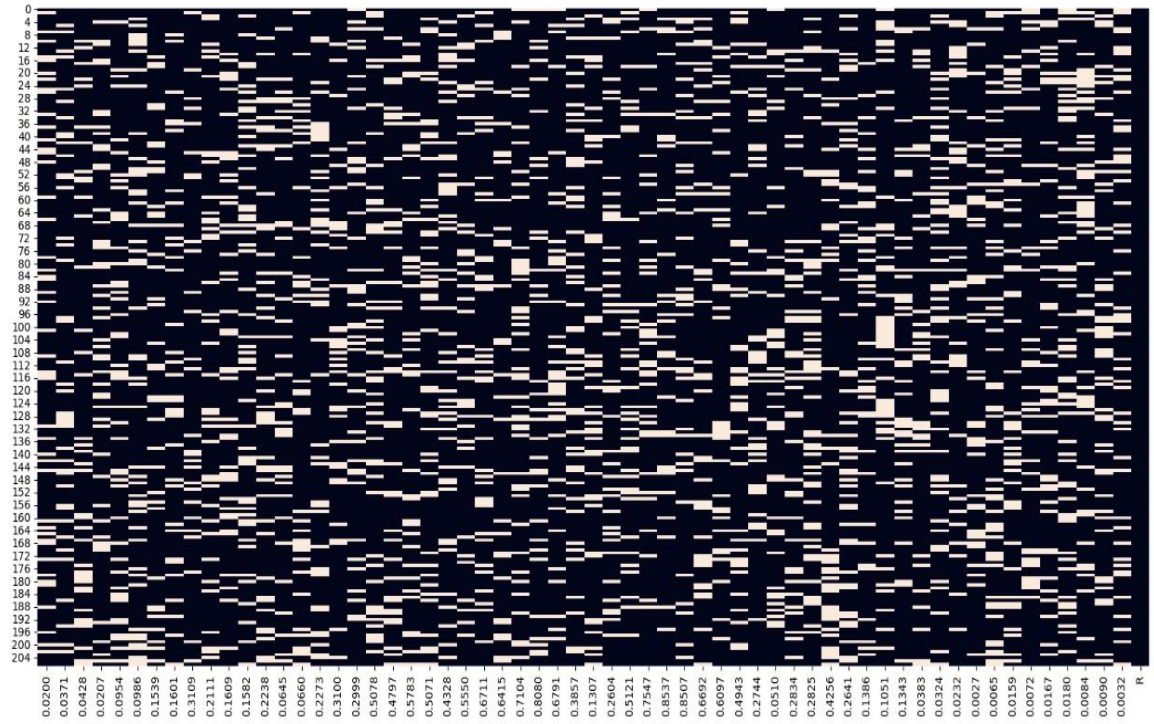


Figure 2.7 *MNAR with 20% of missing values*

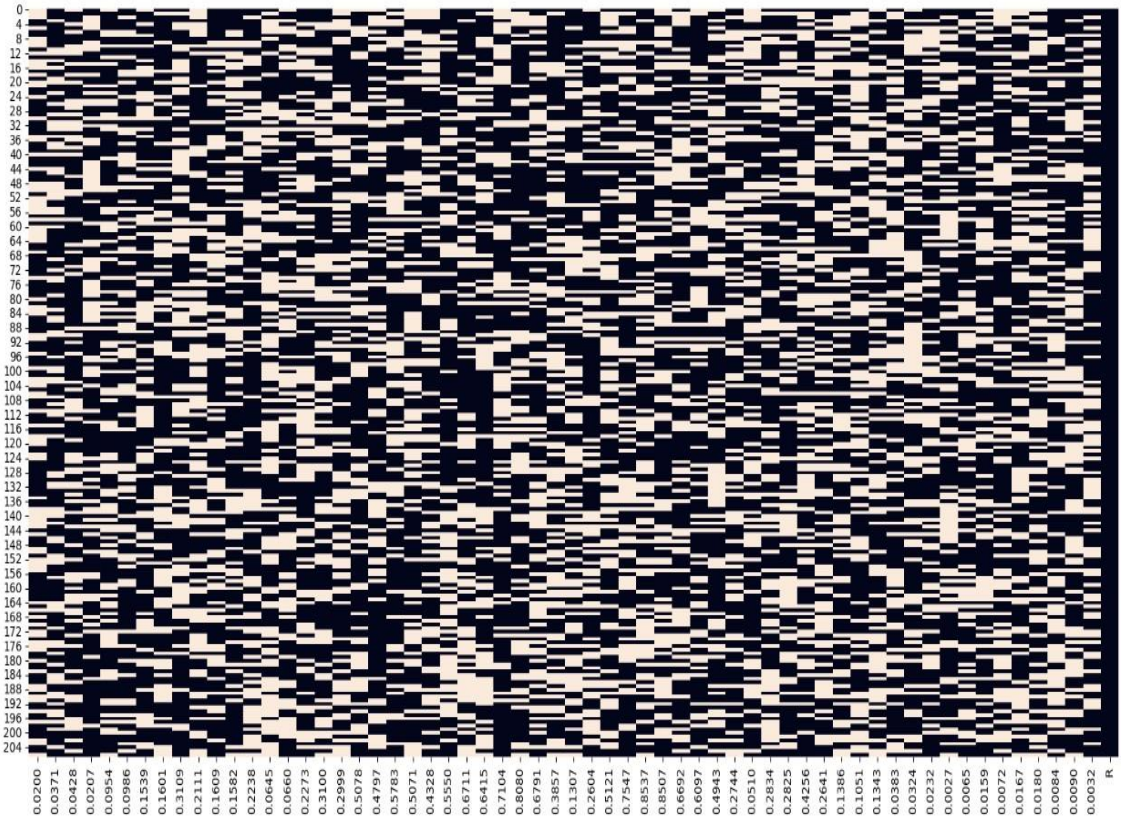


Figure 2.8 *MNAR with 40% of missing values*

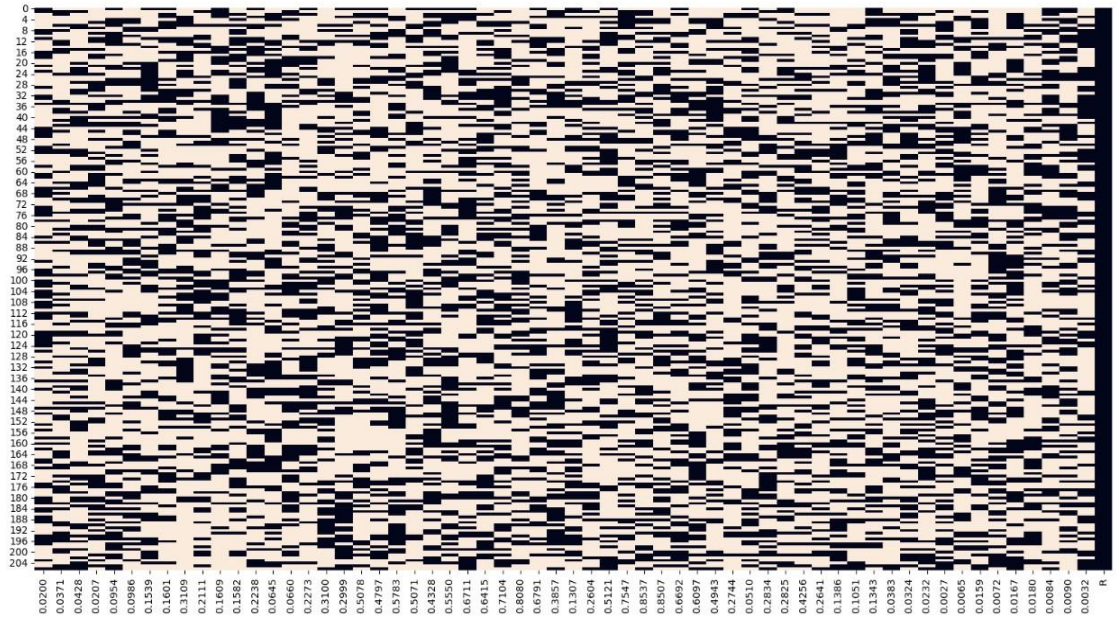


Figure 2.9 *MNAR with 60% of missing values*

Figures 2.10-2.12 show a summary of the datasets obtained after the simulation of the Missing Not at Random with quantile censorship (MNAR with quantile censorship) mechanism with data missingness at 20%, 40%, and 60% rates, respectively.

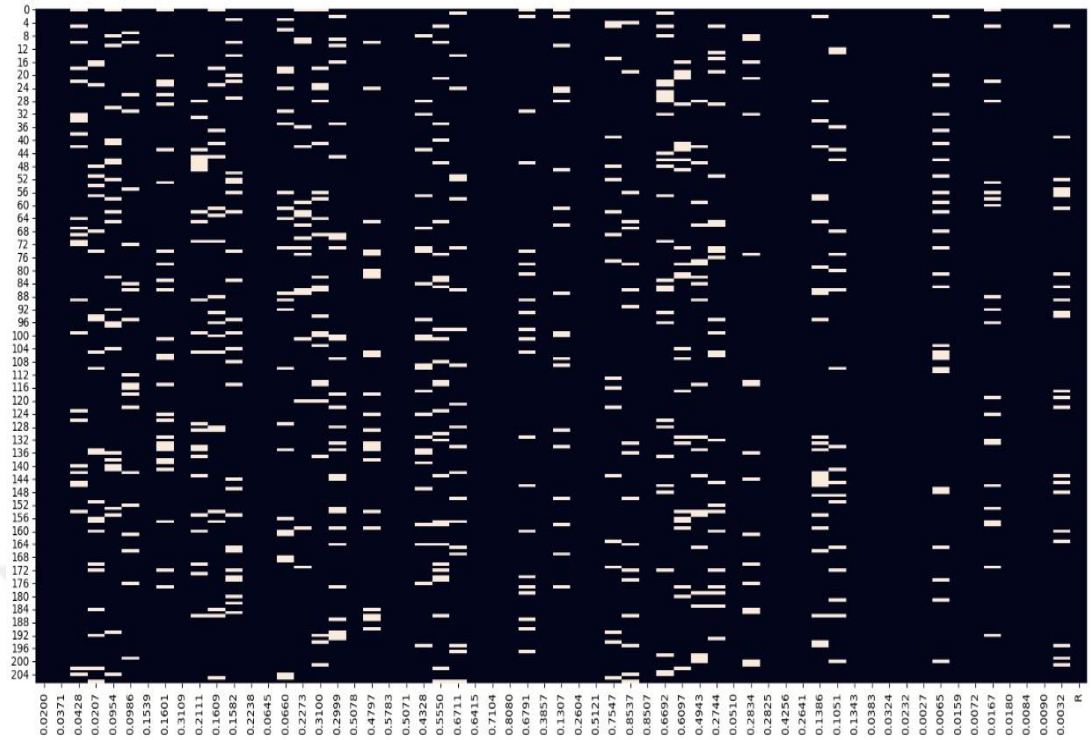


Figure 2.10 *MNAR with quantile censorship 20%*

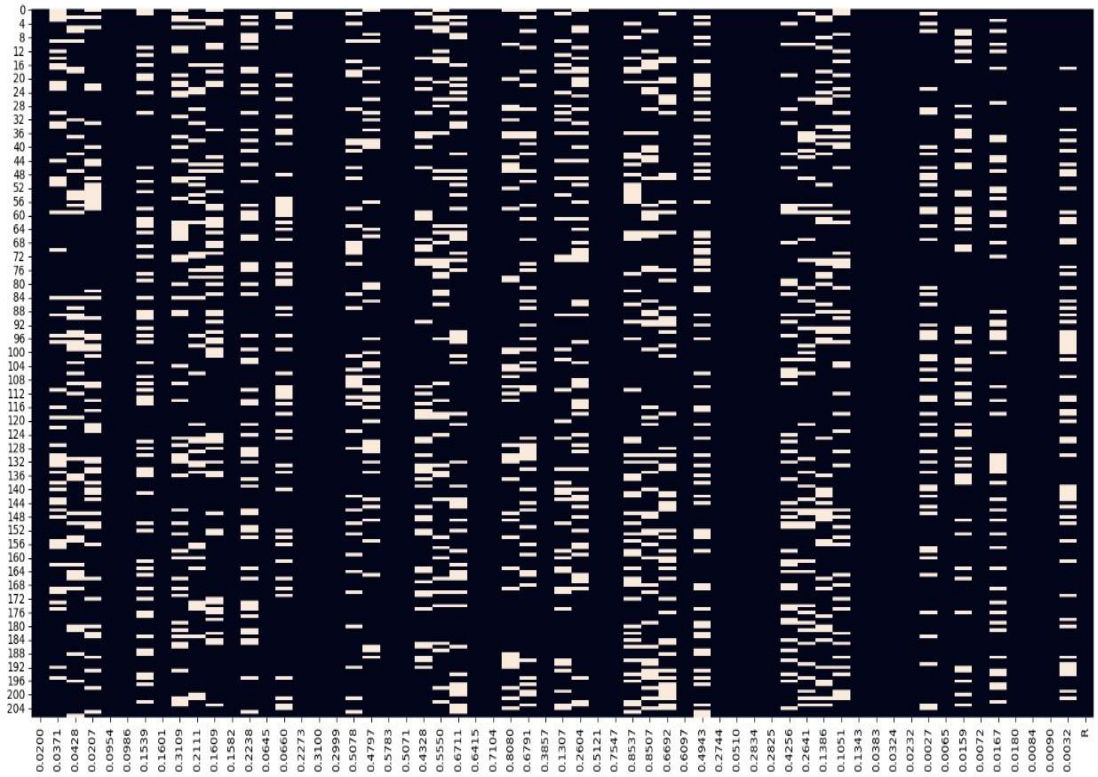


Figure 2.11 *MNAR with quantile censorship 40%*

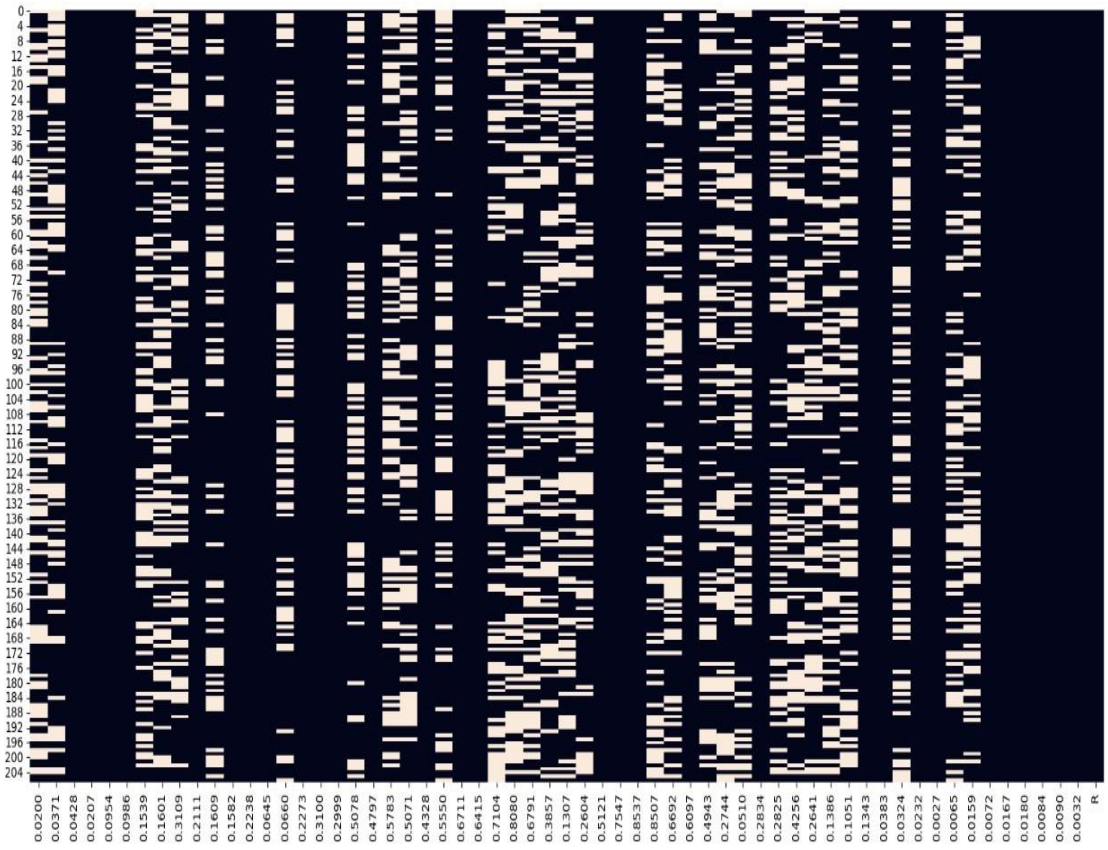


Figure 2.12 *MNAR with quantile censorship 60%*

Upon the completion of the missing data simulation stage, imputation was performed using the MICE algorithm. The MICE implementation has employed several algorithms as estimators, including XGBRegressor, Support Vector Regressor, Random Forest, KNN, and Decision Tree. Each dataset containing missing values was imputed using the MICE algorithm, utilizing the five aforementioned algorithms as estimators. As a result of this task, 60 imputed datasets were created. The subsequent stage involved calculating the Variance Inflation Factor on both the imputed datasets and the original datasets, followed by a comparison. The result is depicted in the figures below.

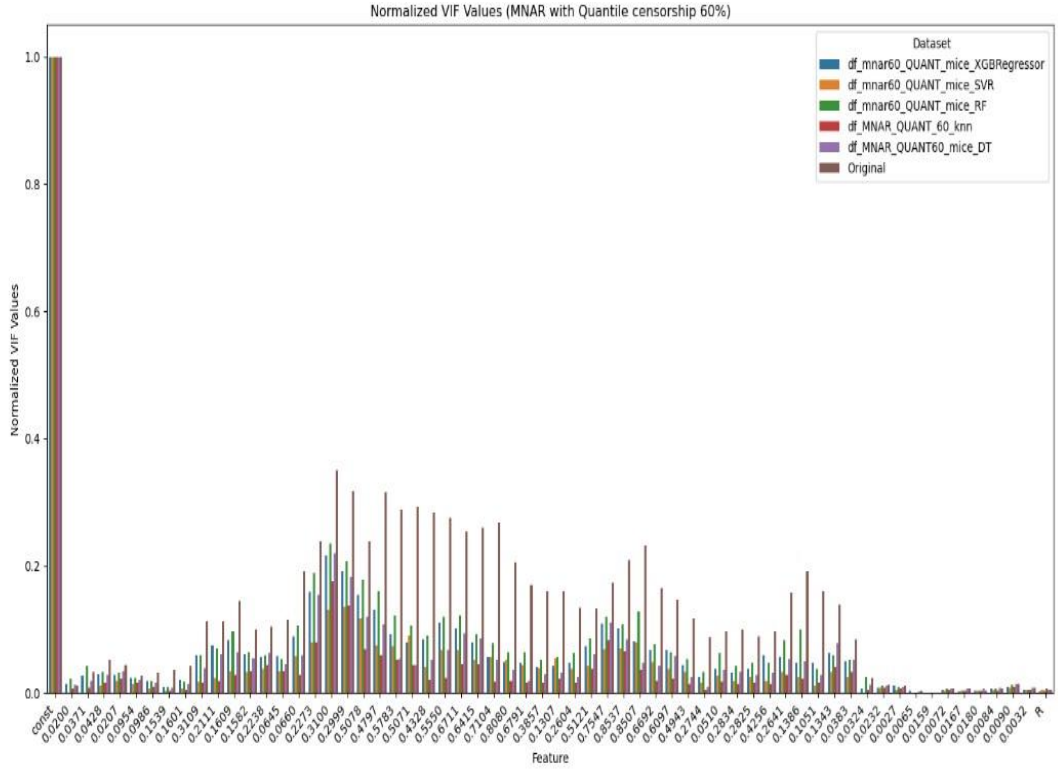


Figure 2.13 *Normalized VIF values (MNAR quantile censorship 60%)*

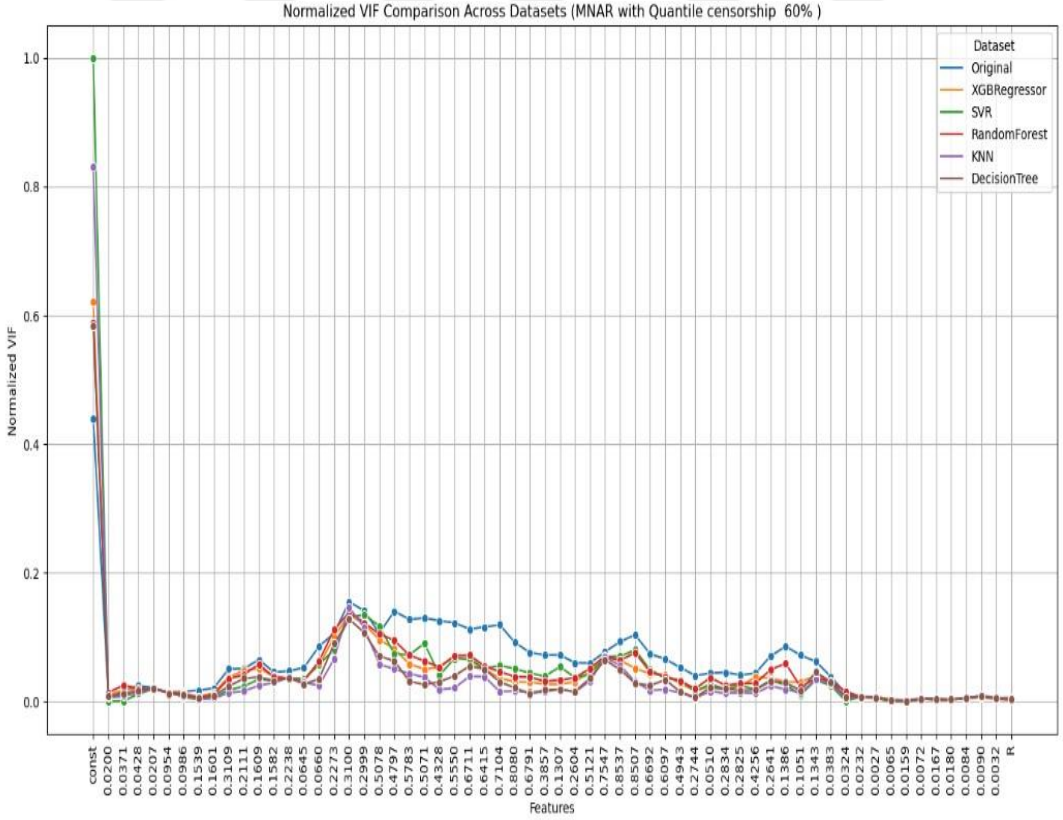


Figure 2.14 *Normalized VIF (MNAR quantile censorship 60%)*

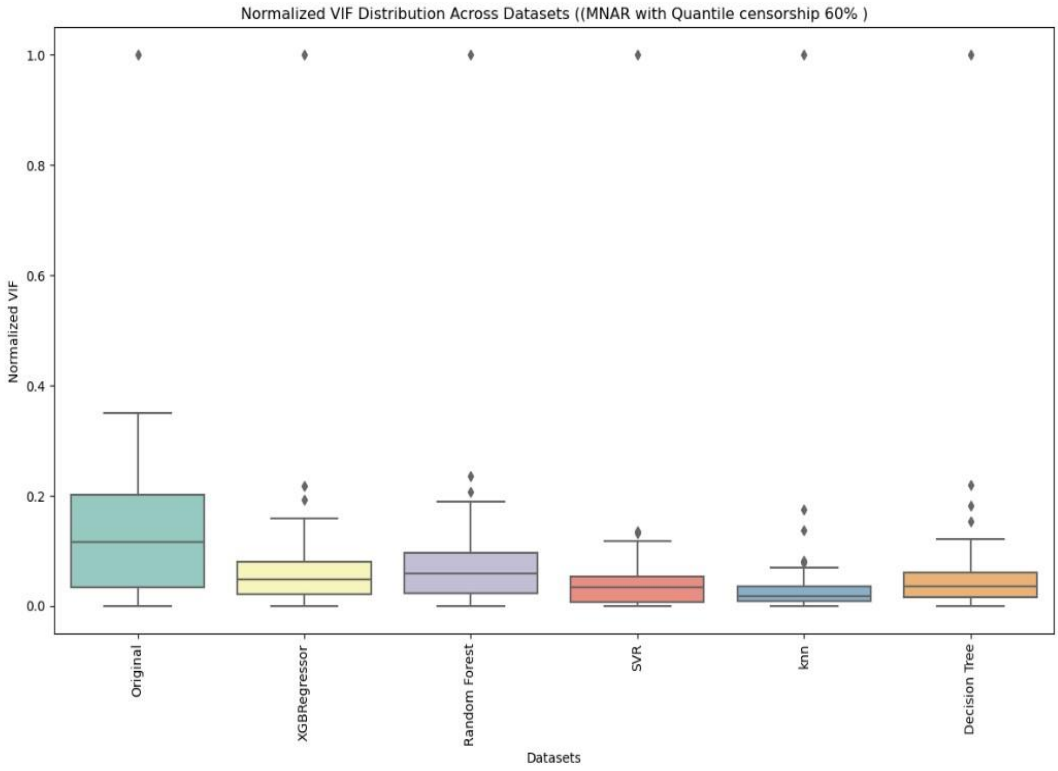


Figure 2.15 *Normalized VIF distribution (MNAR quantile censorship 60%)*

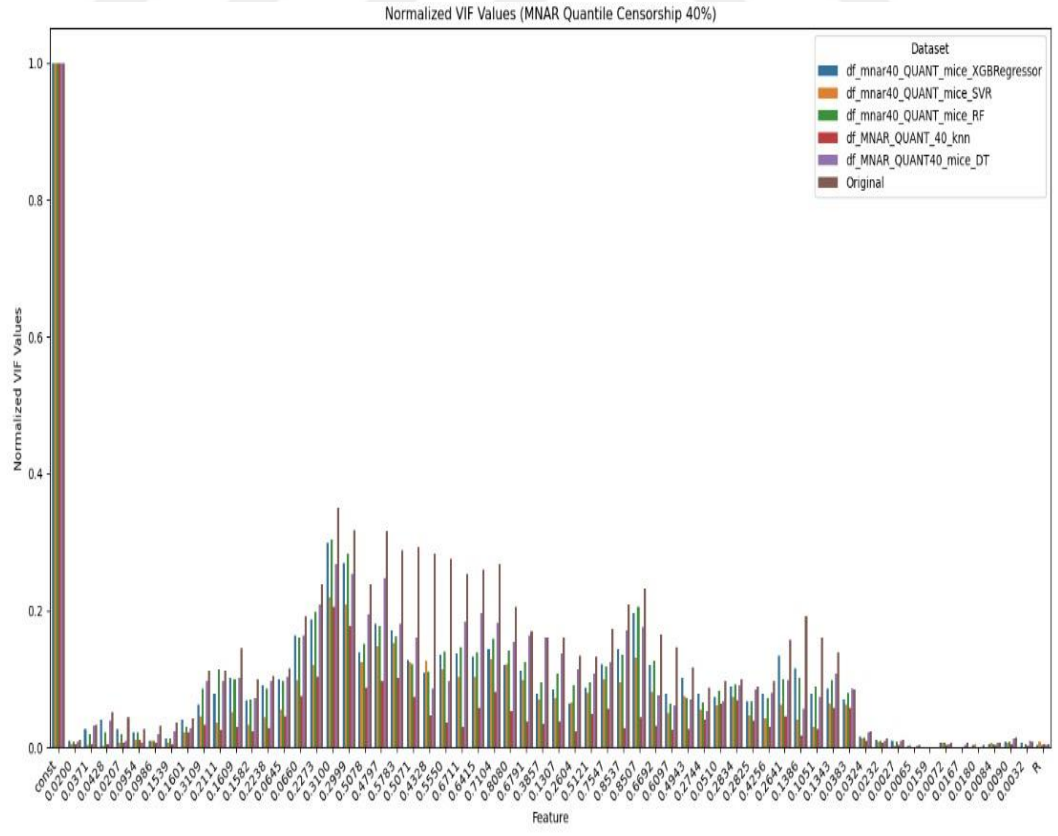


Figure 2.16 *Normalized VIF (MNAR quantile censorship 40%)*

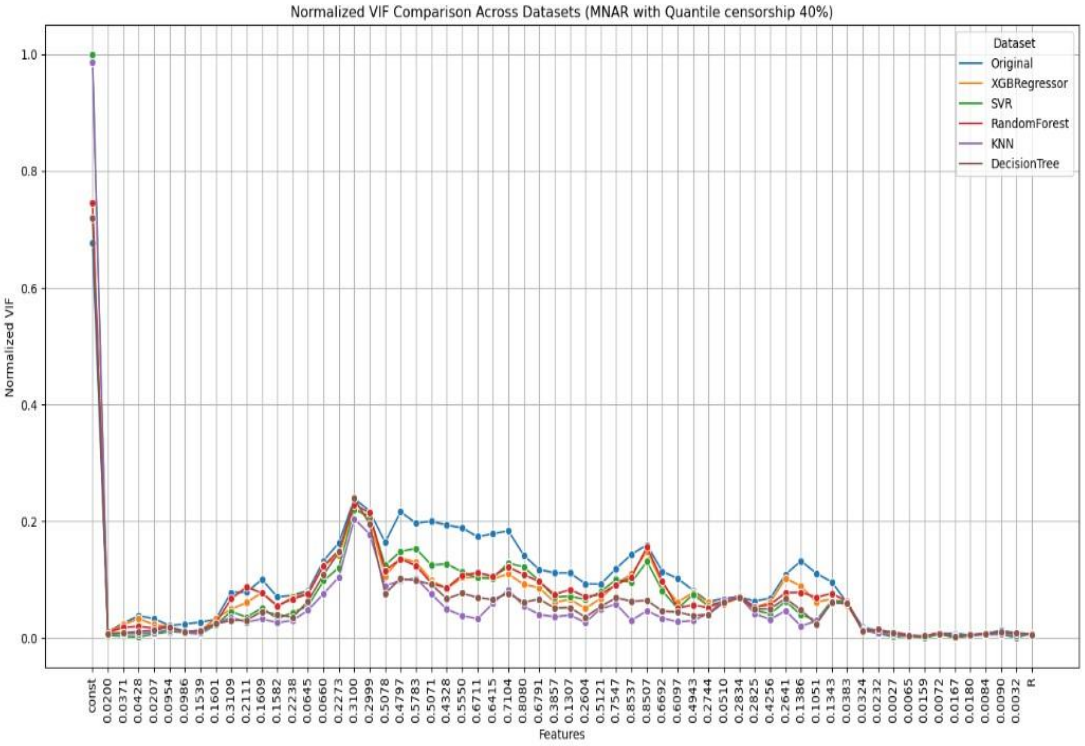


Figure 2.17 *Normalized VIF (MNAR quantile censorship 40%)*

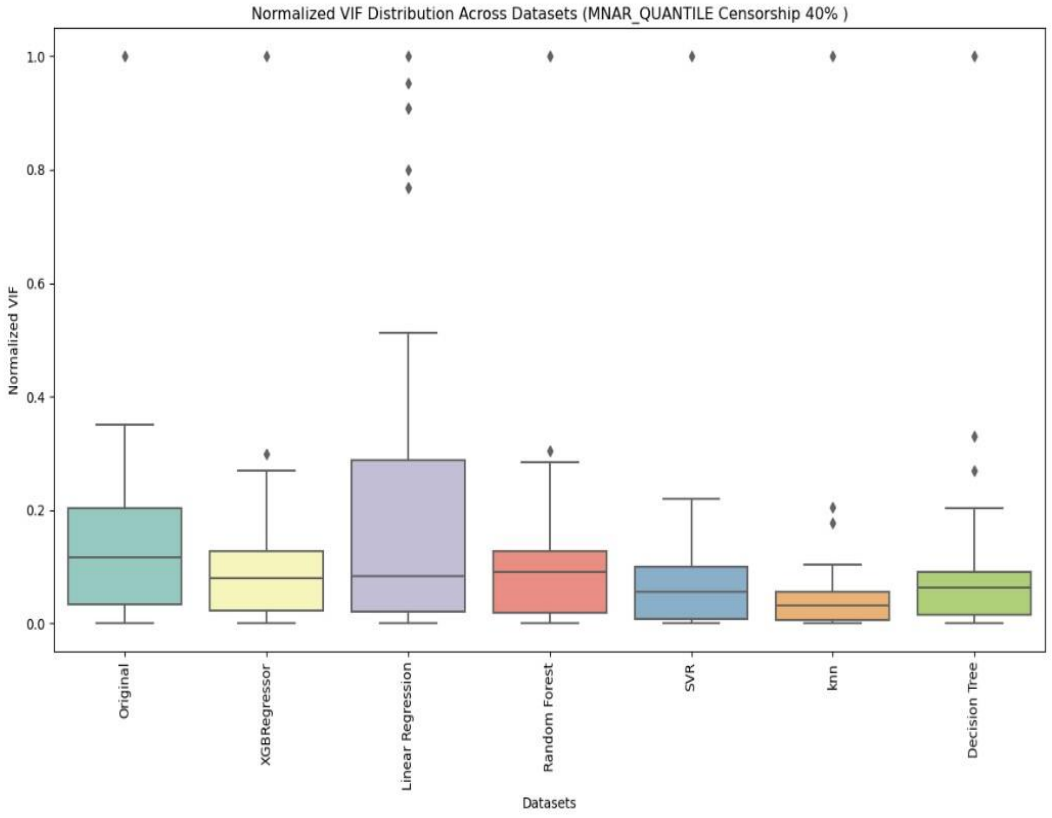


Figure 2.18 *Normalized VIF distribution (MNAR quantile censorship 40%)*

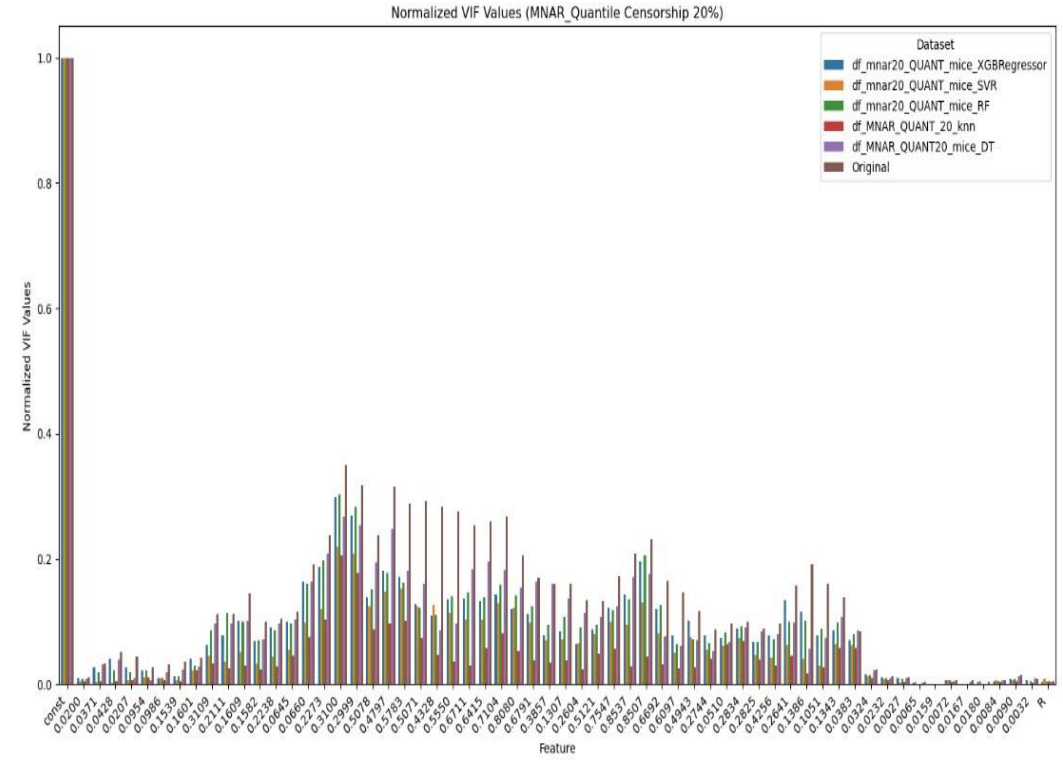


Figure 2.19 *Normalized VIF (MNAR quantile censorship 20%)*

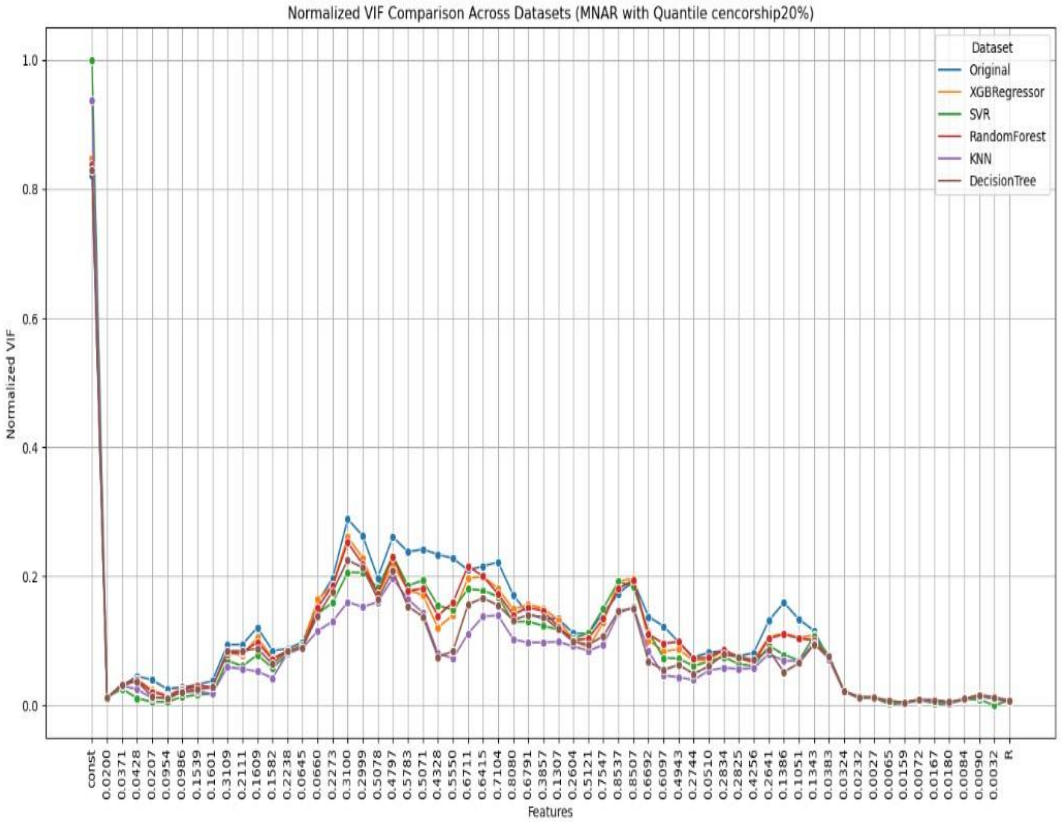


Figure 2.20 *Normalized VIF (MNAR quantile censorship 20%)*

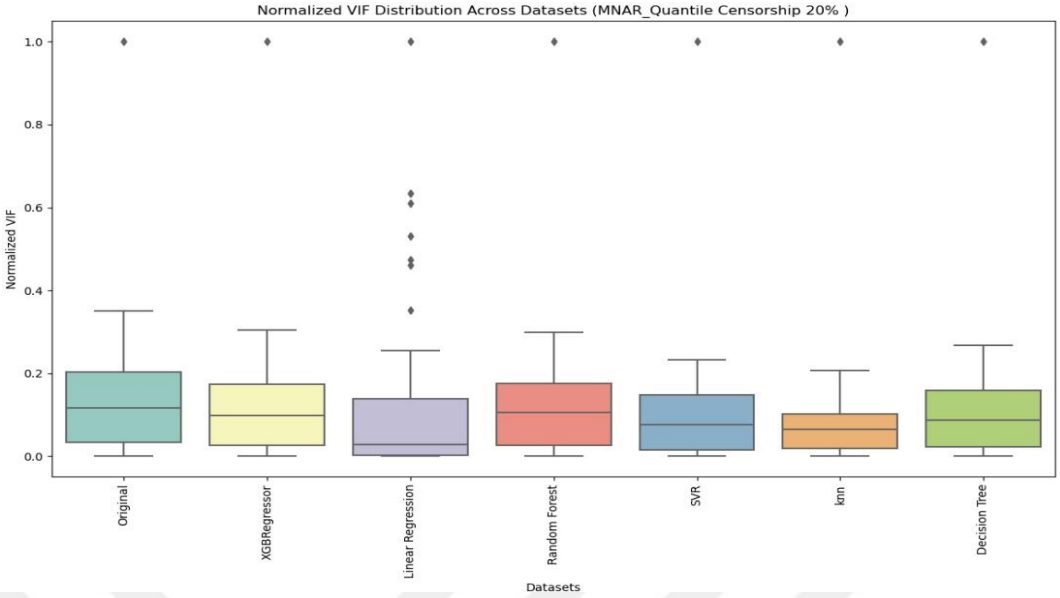


Figure 2.21 *Normalized VIF distribution (MNAR quantile censorship 20%)*

Figures 2.13-2.15 show the results obtained when considering datasets created through MNAR with a quantile censorship mechanism and a data missingness ratio of 60%. On the other hand, Figures 2.16-2.18 display the results obtained with a data missingness ratio of 40%. Finally, Figures 2.19-2.21 show the results obtained with a missingness ratio of 20%. The visualizations reveal disparities in the VIF values among the datasets.

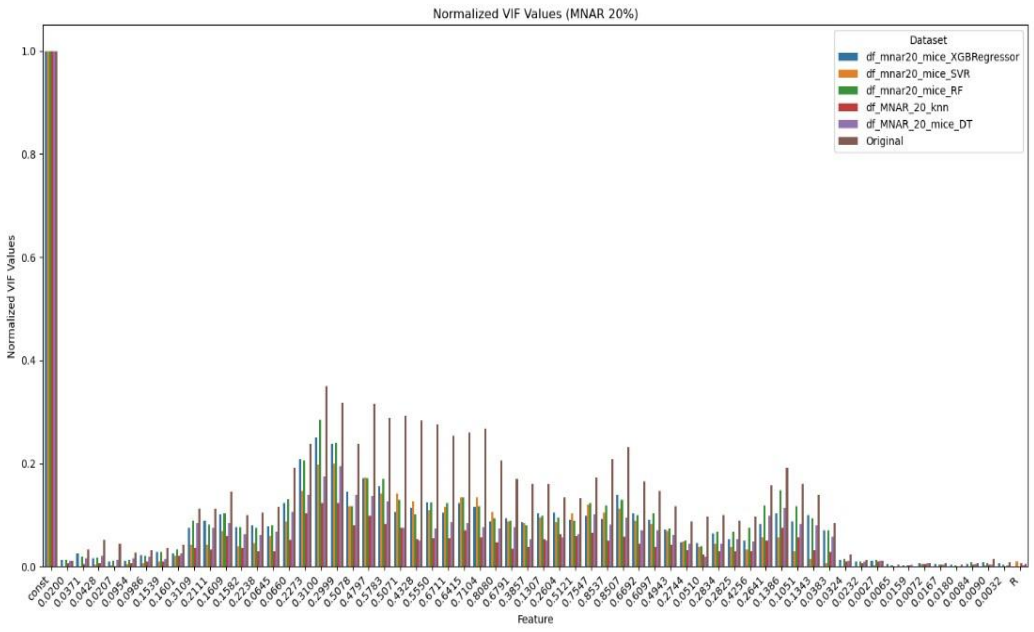


Figure 2.22 *Normalized VIF (MNAR 20%)*

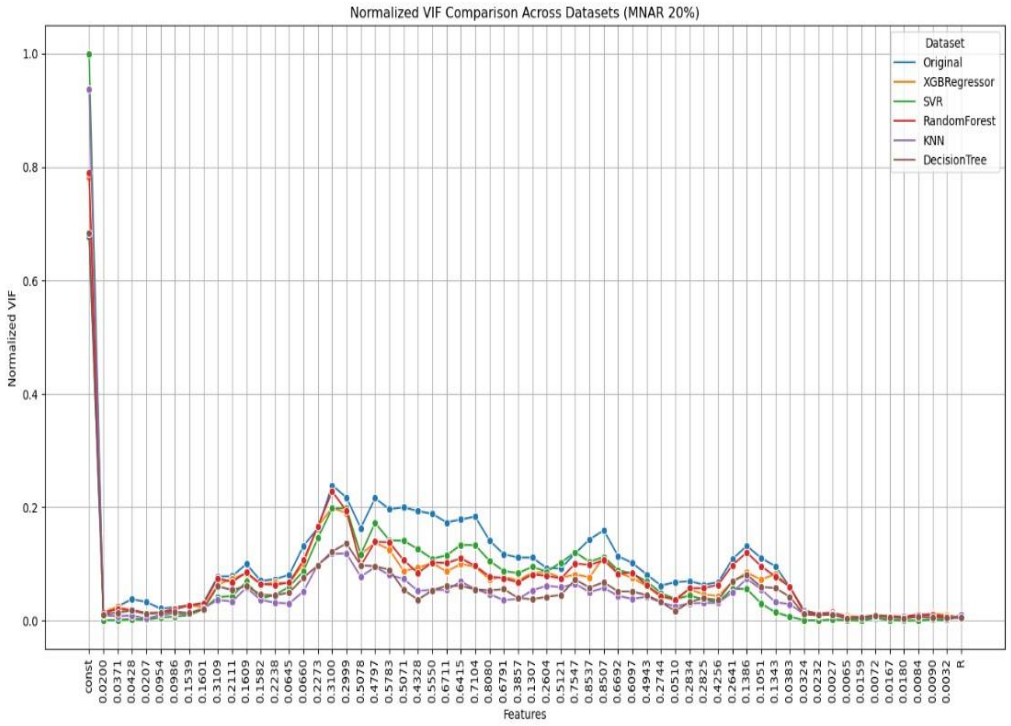


Figure 2.23 *Normalized VIF (MNAR 20%)*

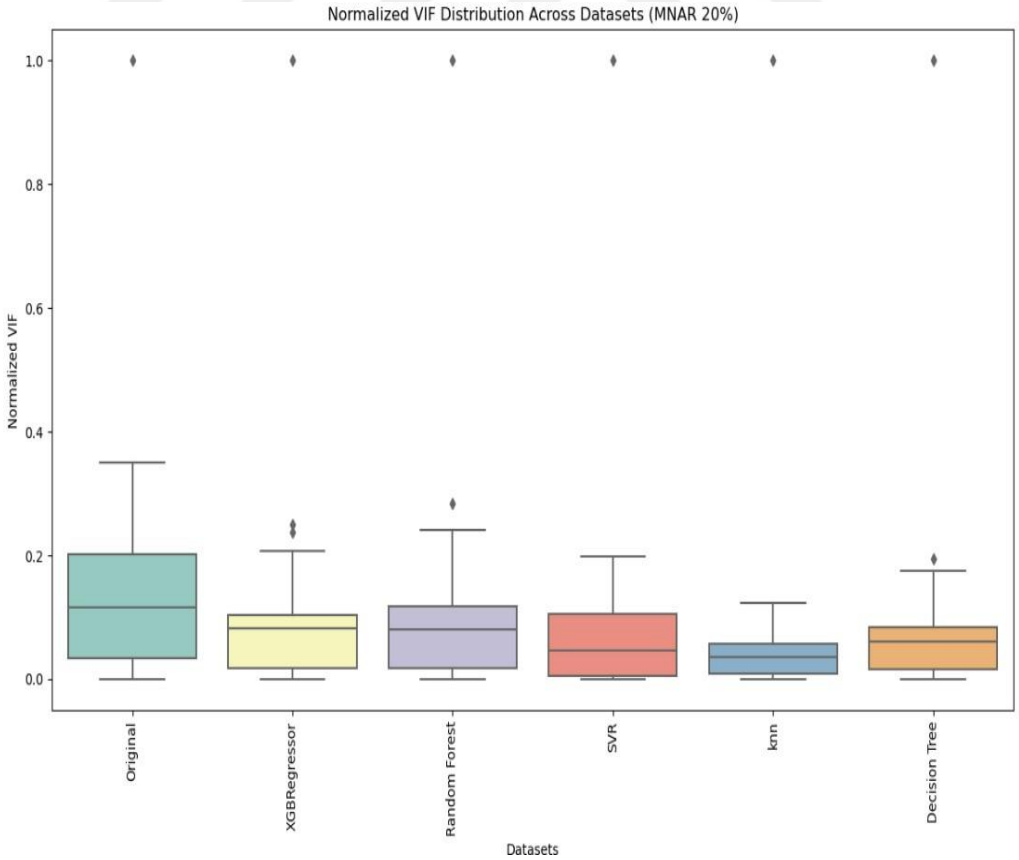


Figure 2.24 *Normalized VIF distribution (MNAR 20%)*

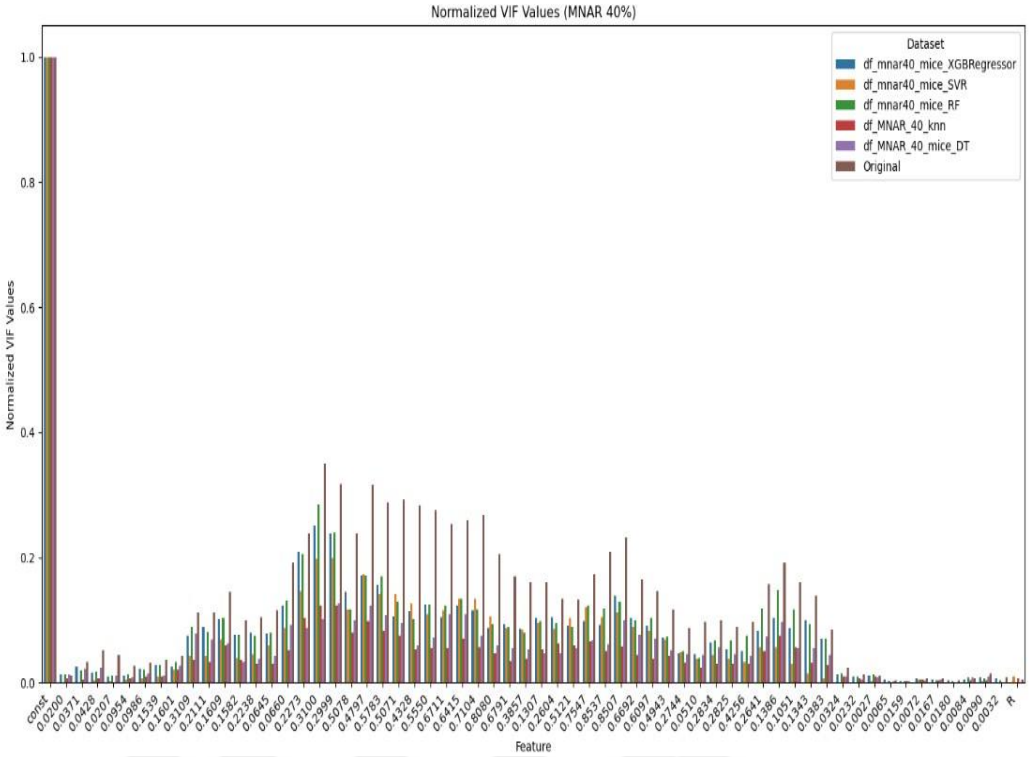


Figure 2.25 *Normalized VIF (MNAR 40%)*

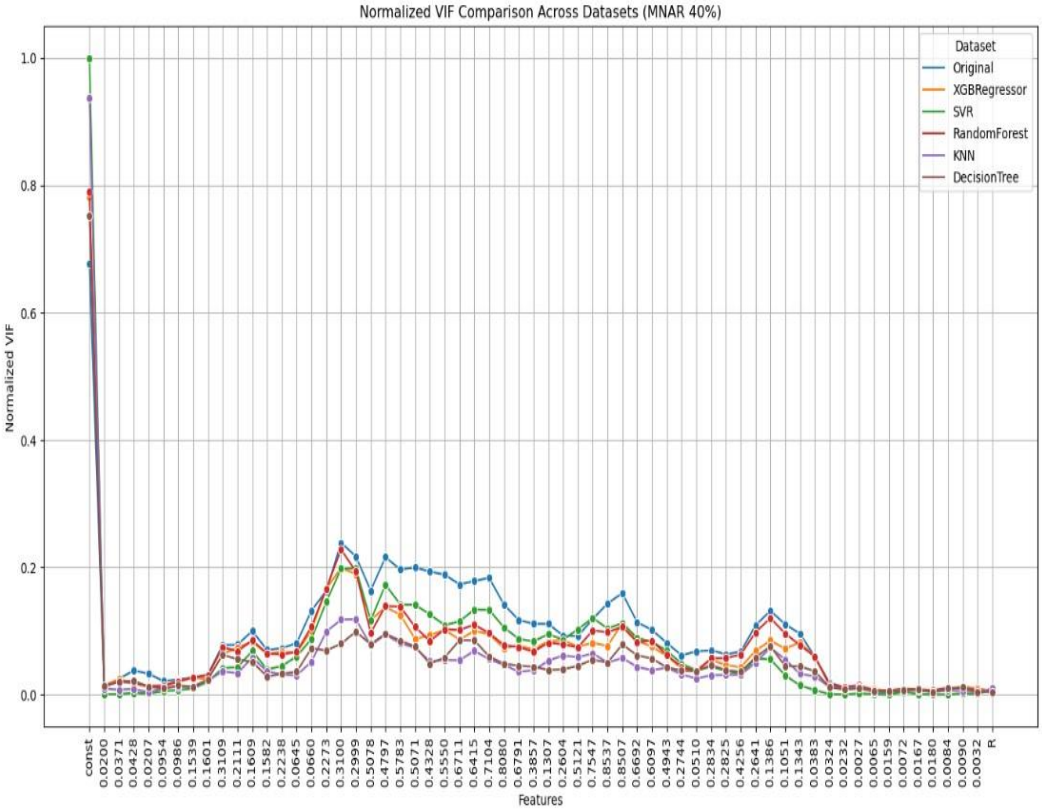


Figure 2.26 *Normalized VIF (MNAR 40%)*

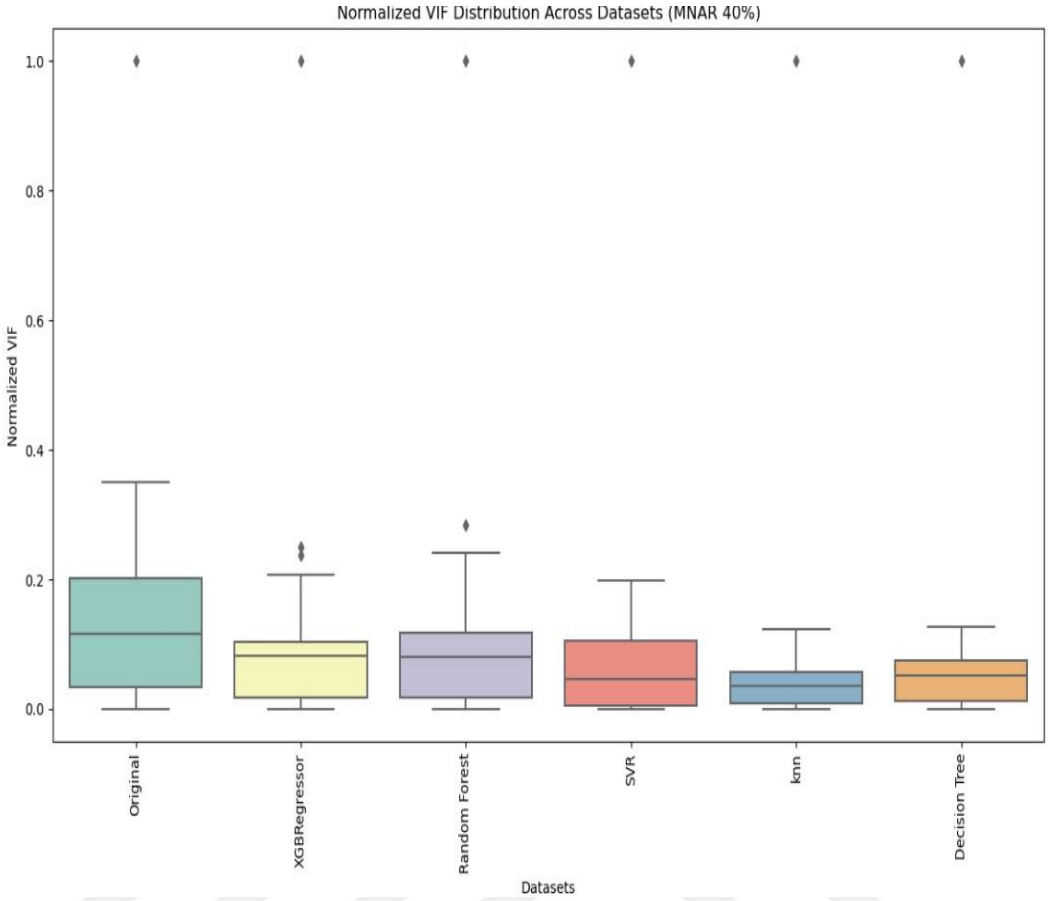


Figure 2.27 *Normalized VIF distribution (MNAR 40%)*

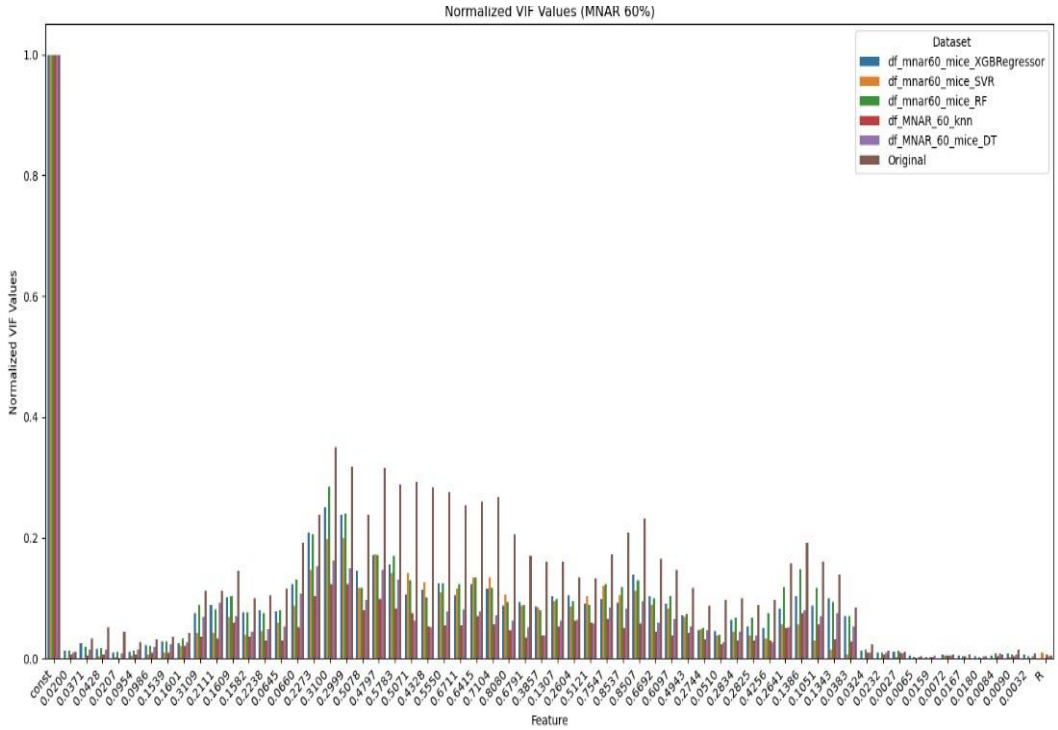


Figure 2.28 *Normalized VIF (MNAR 60%)*

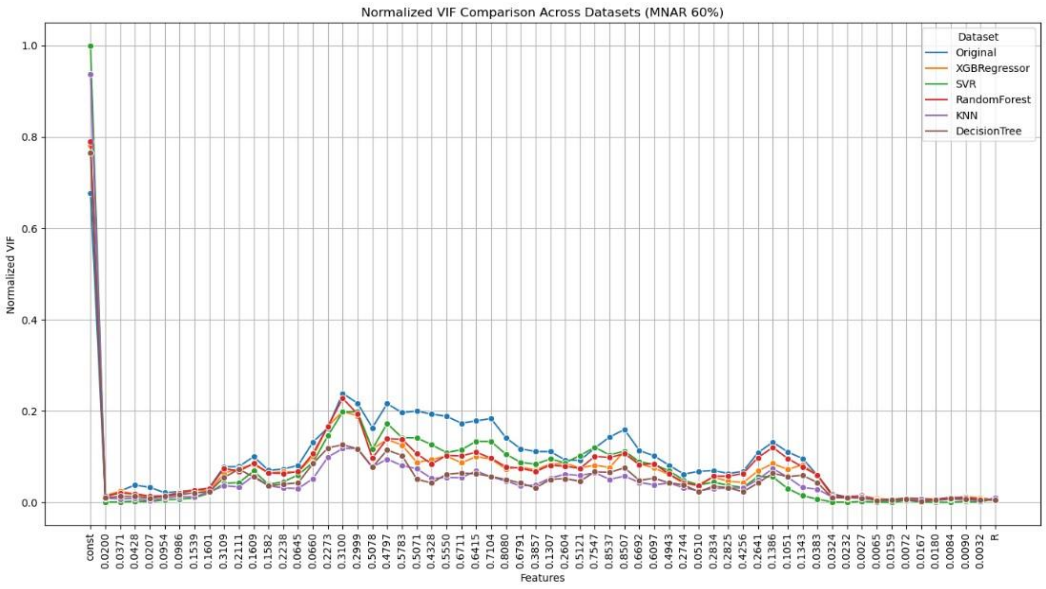


Figure 2.29 *Normalized VIF (MNAR 60%)*

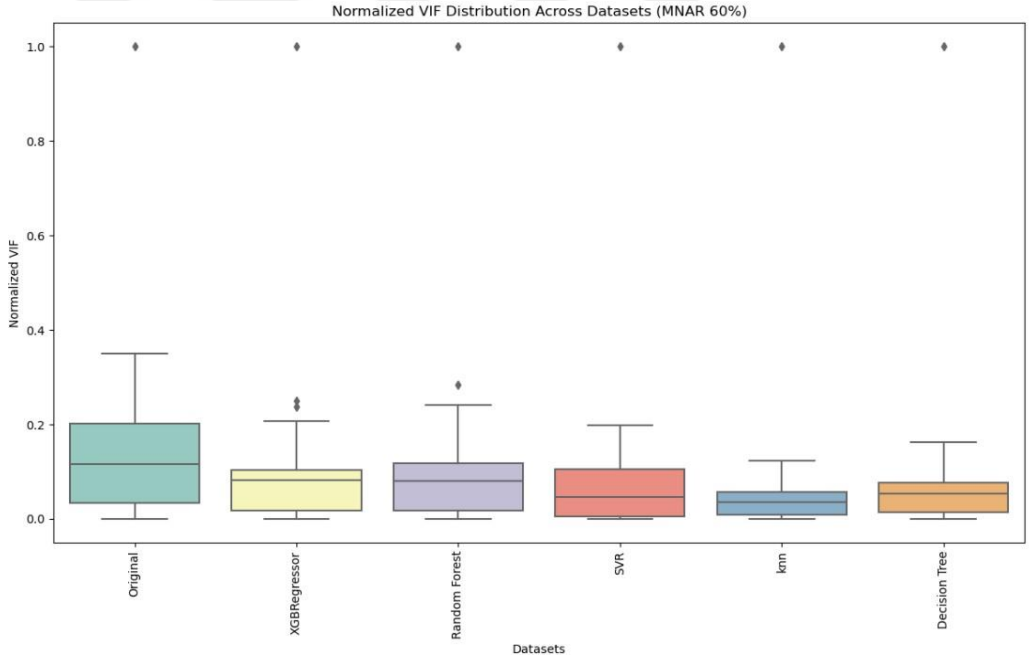


Figure 2.30 *Normalized VIF distribution (MNAR 60%)*

Figures 2.28-2.30 show the results obtained when considering datasets created through the MNAR mechanism and a data missingness ratio of 60%. On the other hand, Figures 2.25-2.27 display the results obtained with a data missingness ratio of 40%. Finally, Figures 2.22-2.24 show the results obtained with a missingness ratio of 20%. The visualizations reveal disparities in the VIF values among the datasets.

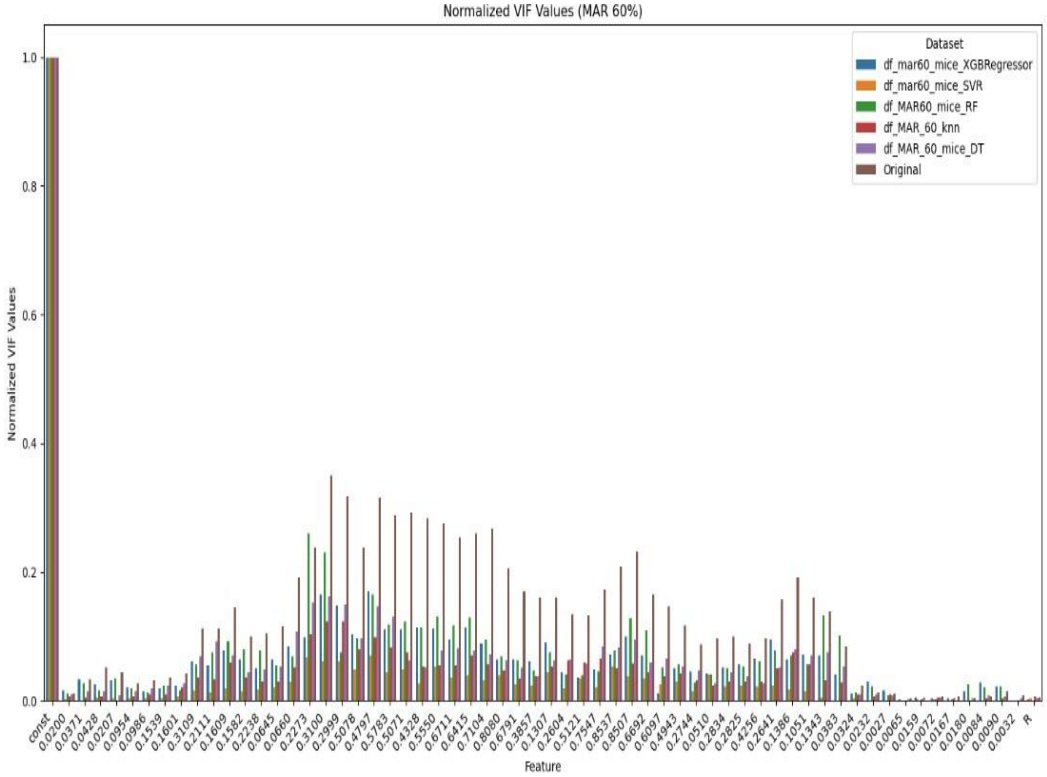


Figure 2.31 *Normalized VIF (MAR 60%)*

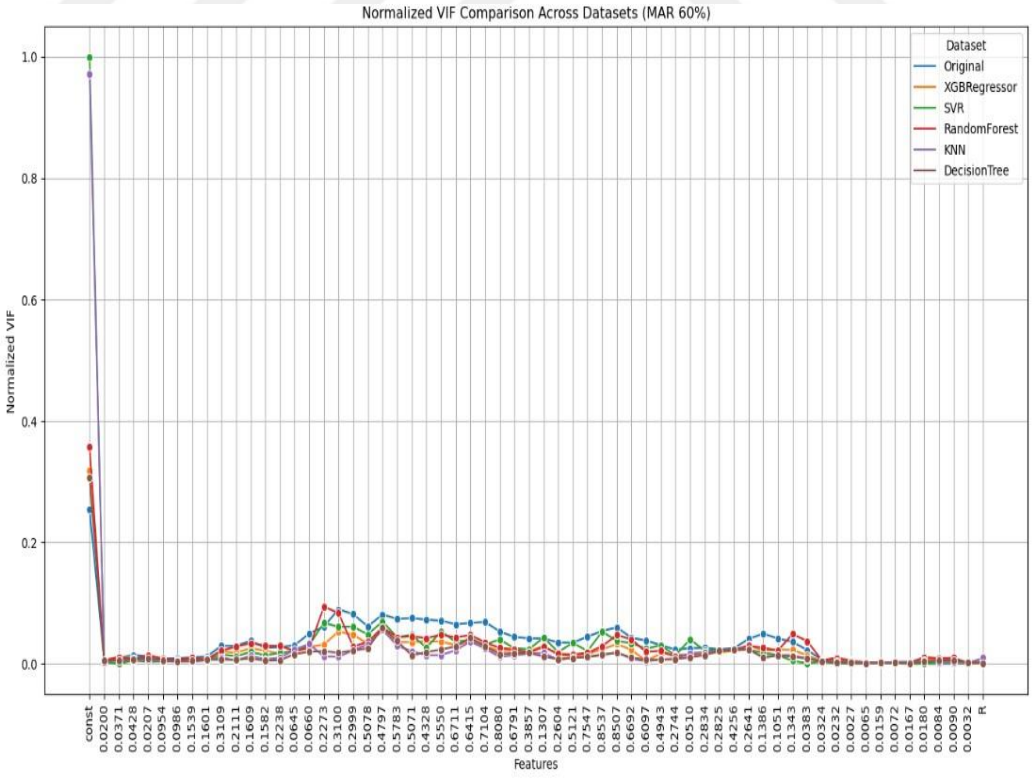


Figure 2.32 *Normalized VIF (MAR 60%)*

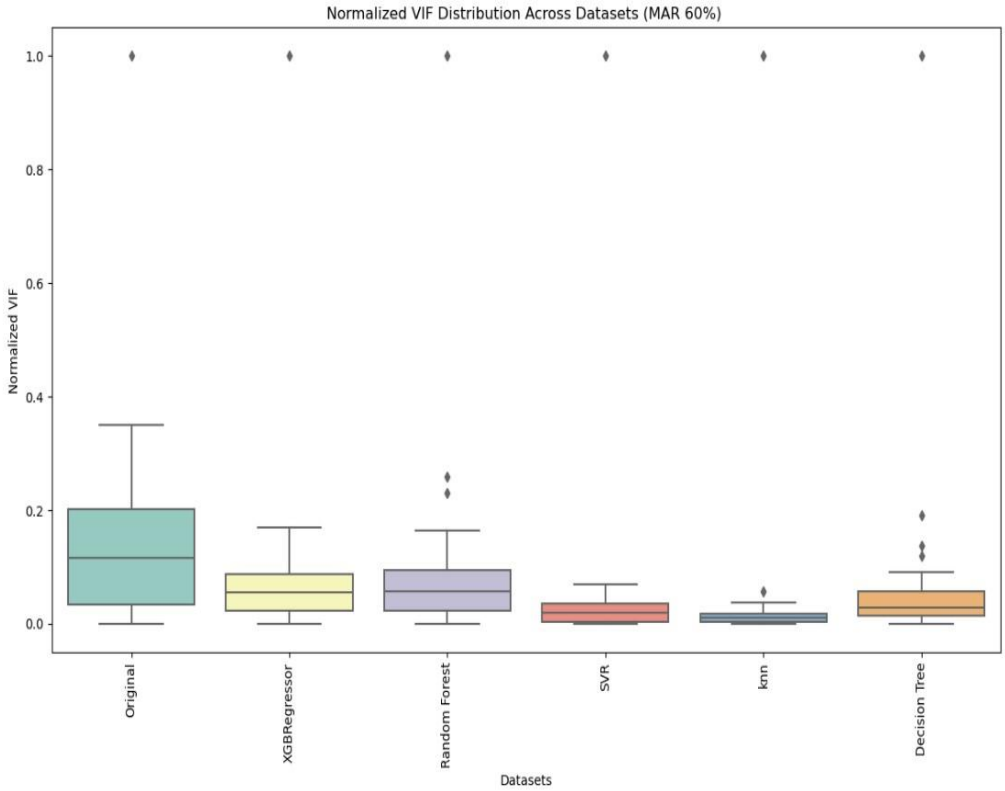


Figure 2.33 *Normalized VIF distribution (MAR 60%)*

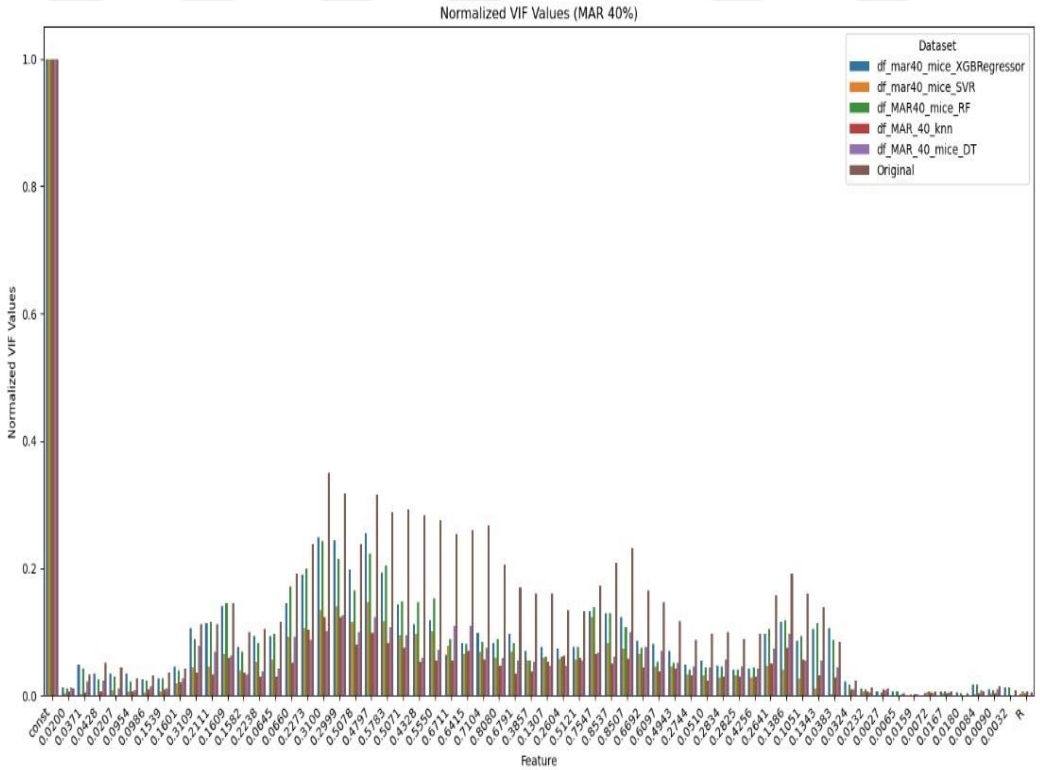


Figure 2.34 *Normalized VIF (MAR 40%)*

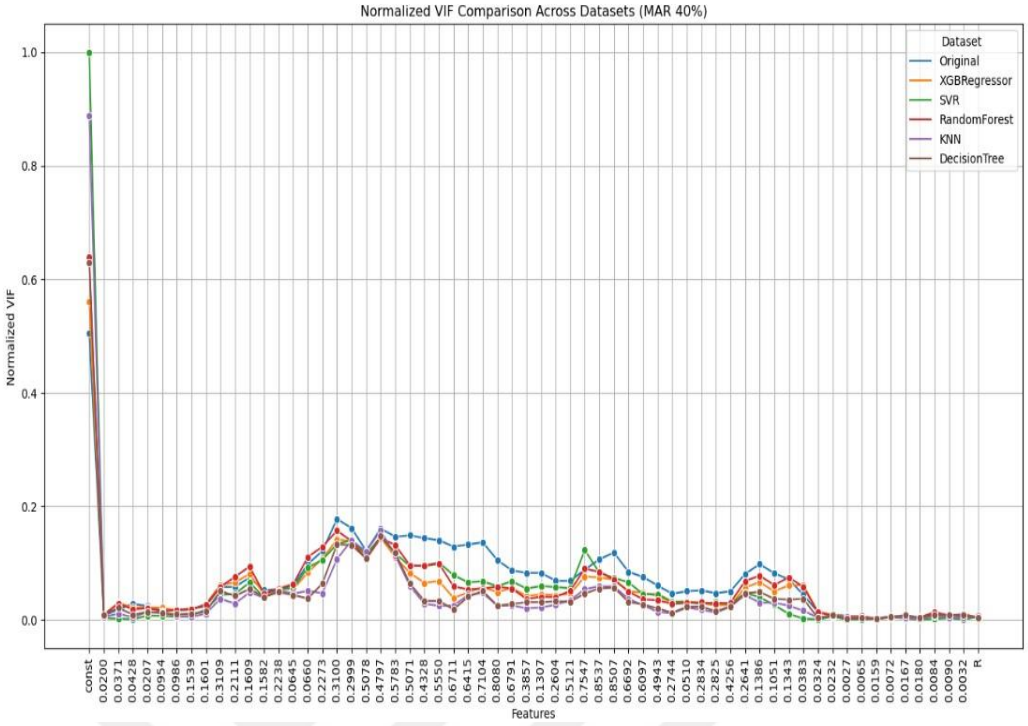


Figure 2.35 *Normalized VIF (MAR 40%)*

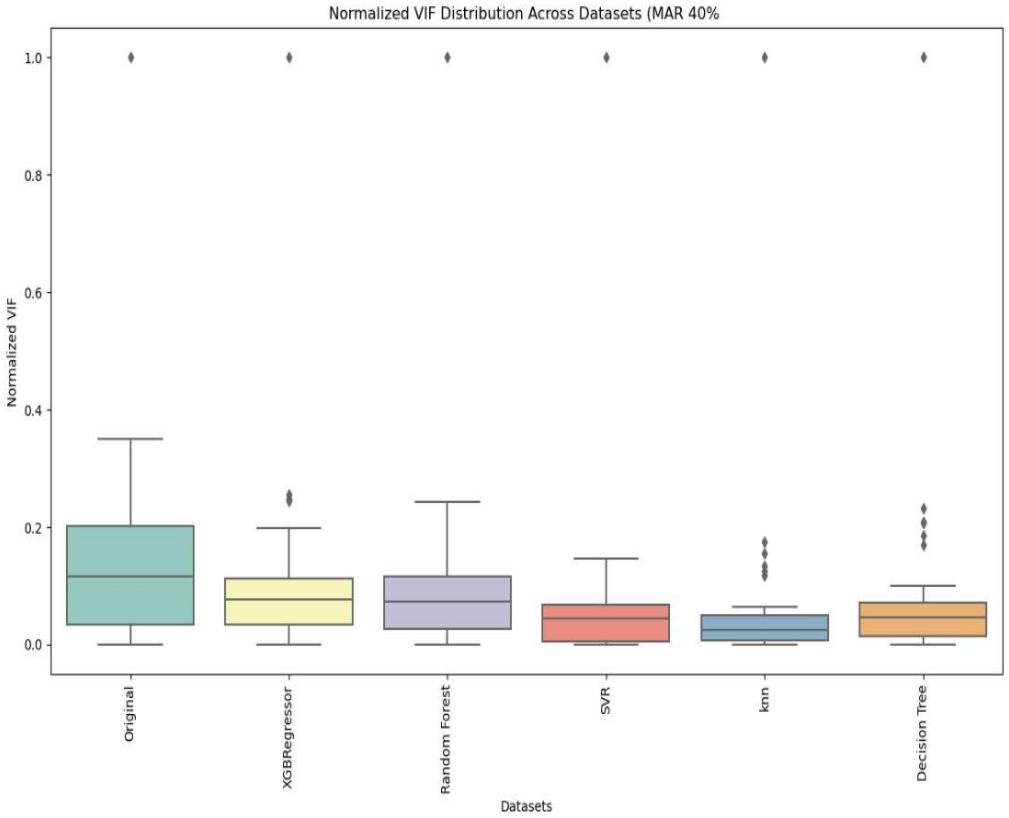


Figure 2.36 *Normalized VIF distribution (MAR 40%)*

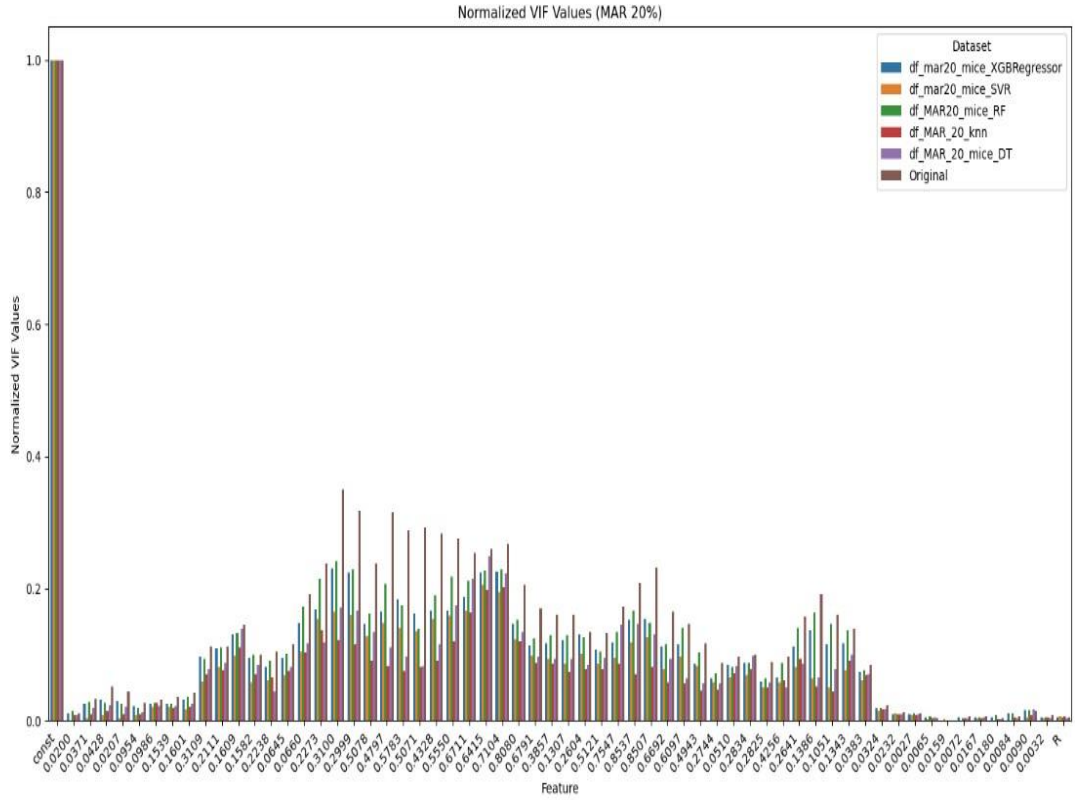


Figure 2.37 *Normalized VIF (MAR 20%)*

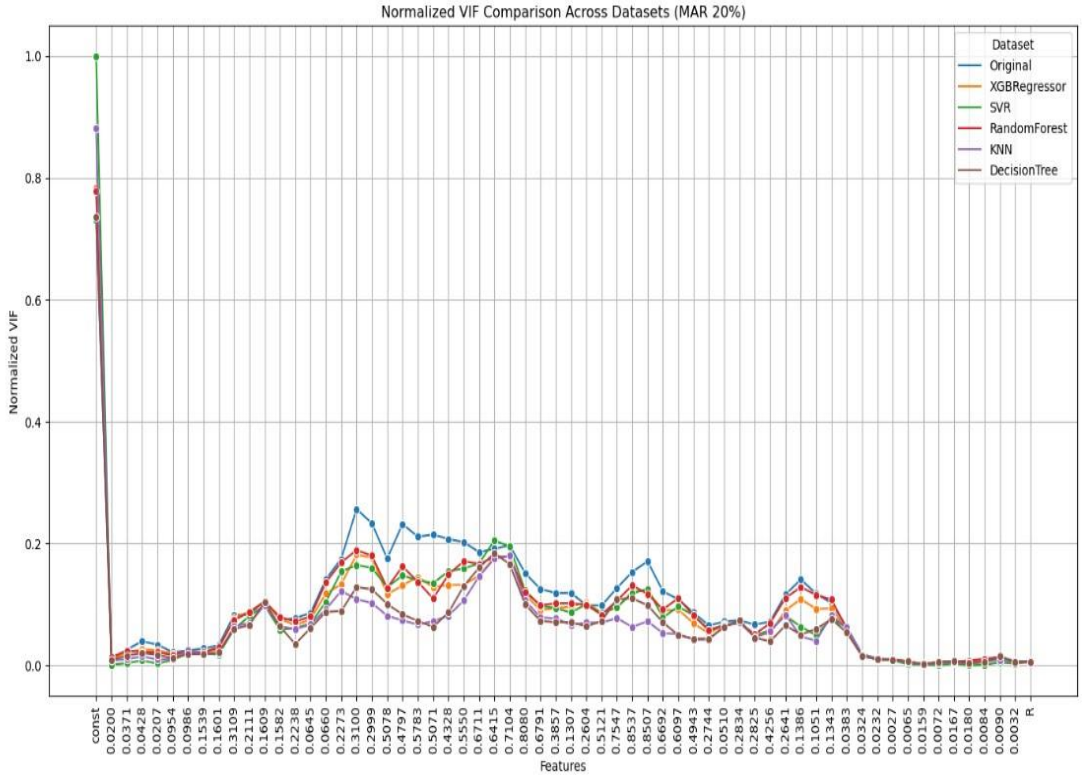


Figure 2.38 *Normalized VIF (MAR 20%)*

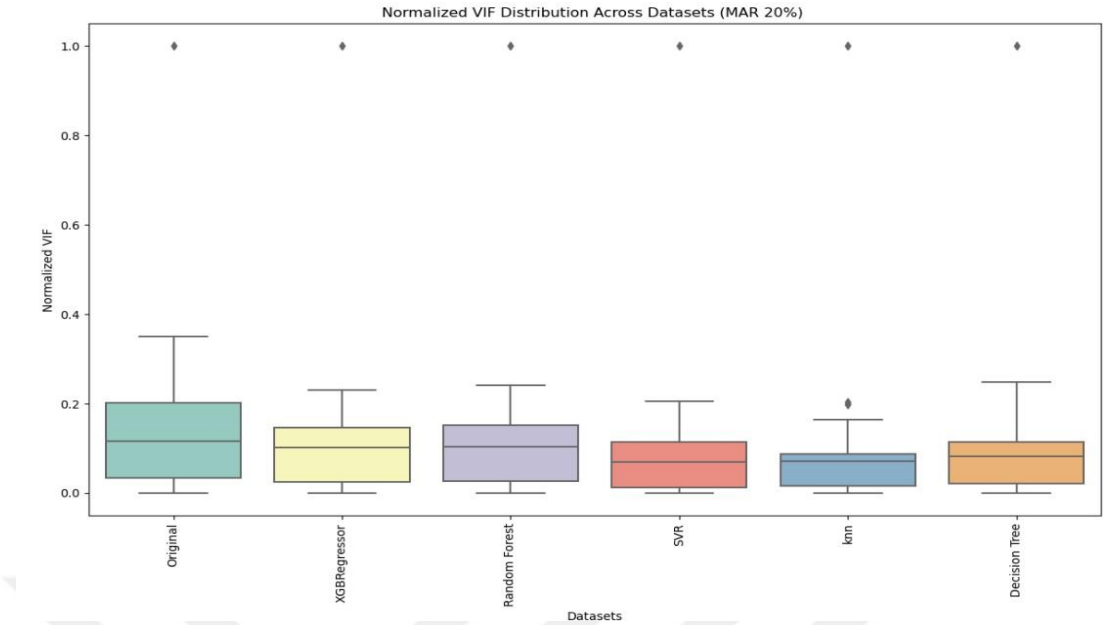


Figure 2.39 *Normalized VIF distribution (MAR 20%)*

The results obtained from analyzing datasets generated using the MAR technique and a data missingness percentage of 60% are presented in Figures 2.31-2.33. Figures 2.34-2.36 show the results obtained when there was a data missingness ratio of 40%. Finally, the results obtained with a missingness ratio of 20% are shown in Figures 2.37-2.39. Various datasets exhibit distinct Variance Inflation Factor (VIF) values, and the visual representations confirm the presence of these disparities.

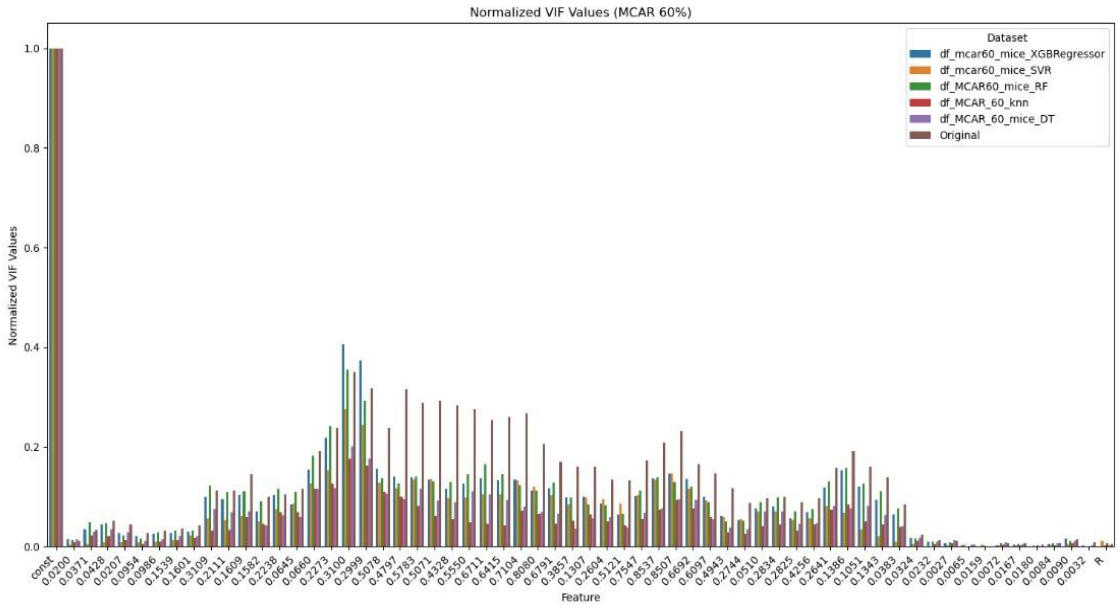


Figure 2.40 *Normalized VIF (MCAR 60%)*

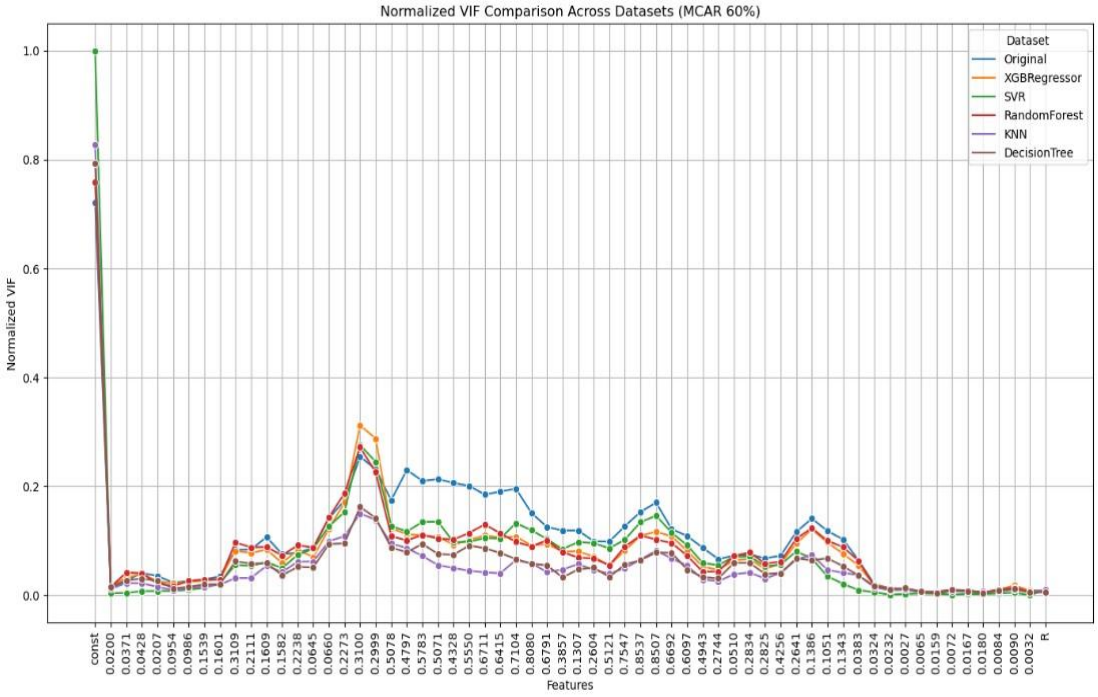


Figure 2.41 *Normalized VIF (MCAR 60%)*

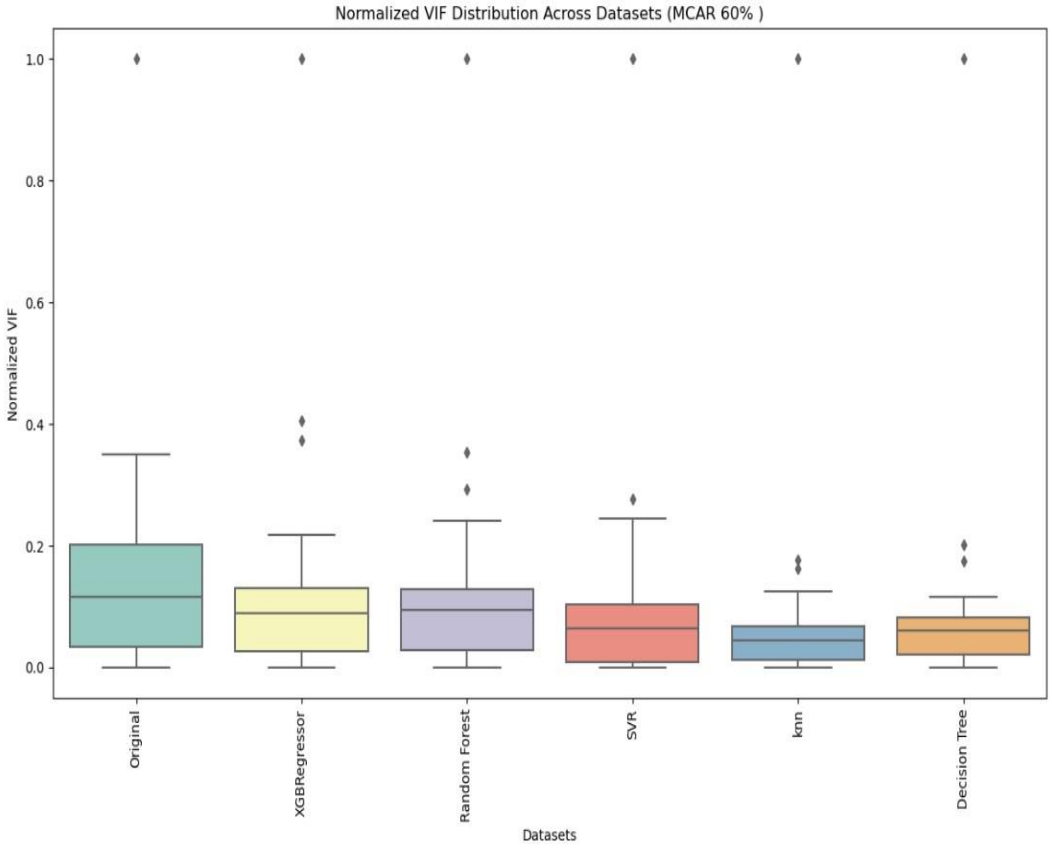


Figure 2.42 *Normalized VIF distribution (MCAR 60%)*

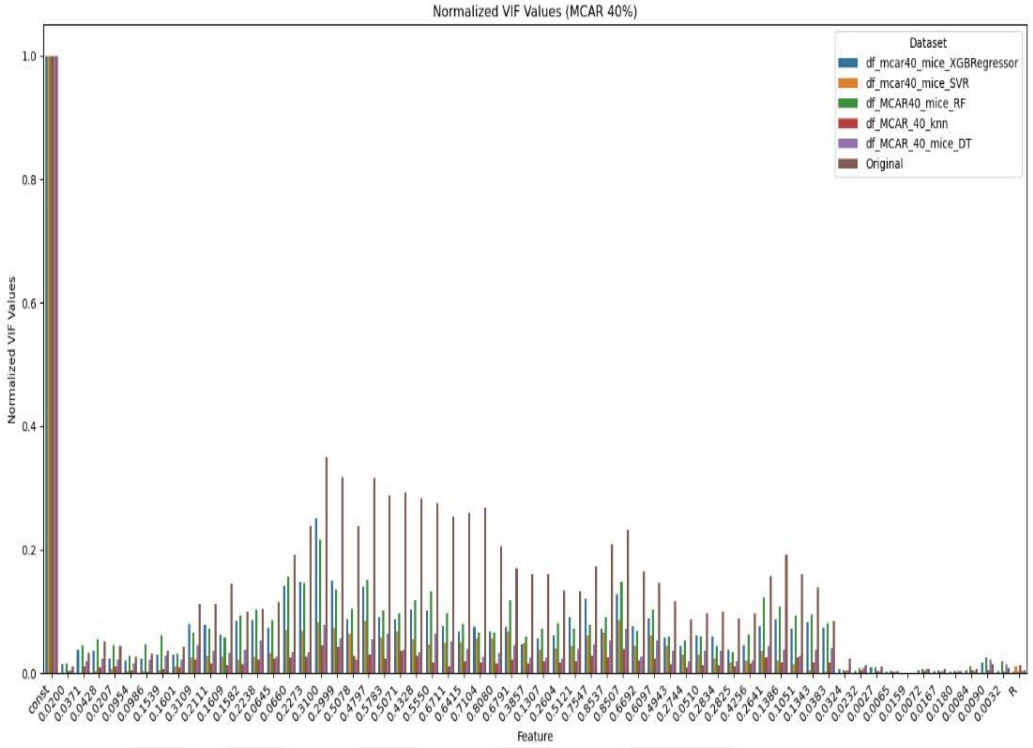


Figure 2.43 *Normalized VIF (MCAR 40%)*

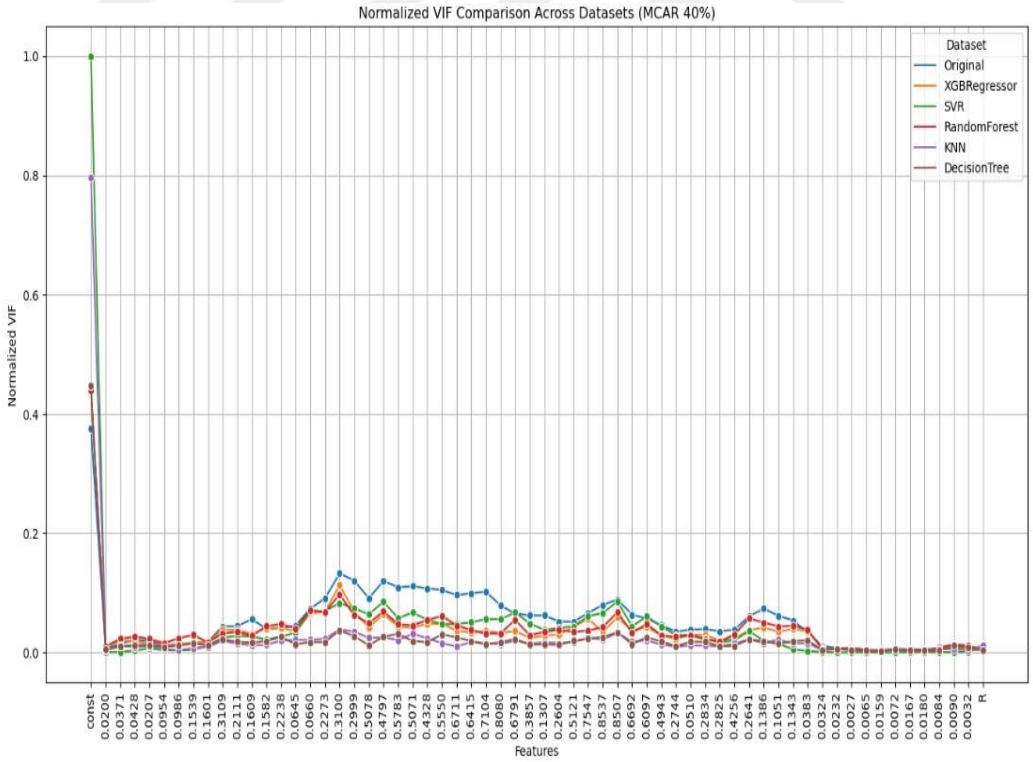


Figure 2.44 *Normalized VIF values (MCAR 40%)*

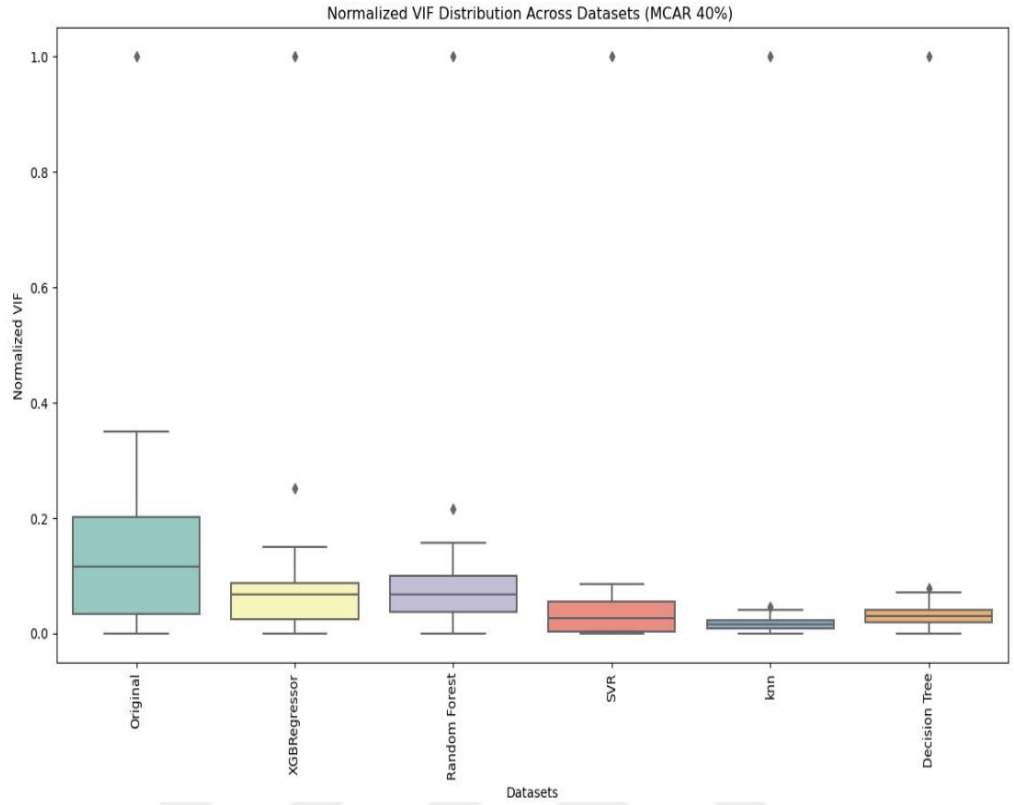


Figure 2.45 *Normalized VIF distribution (MCAR 40%)*

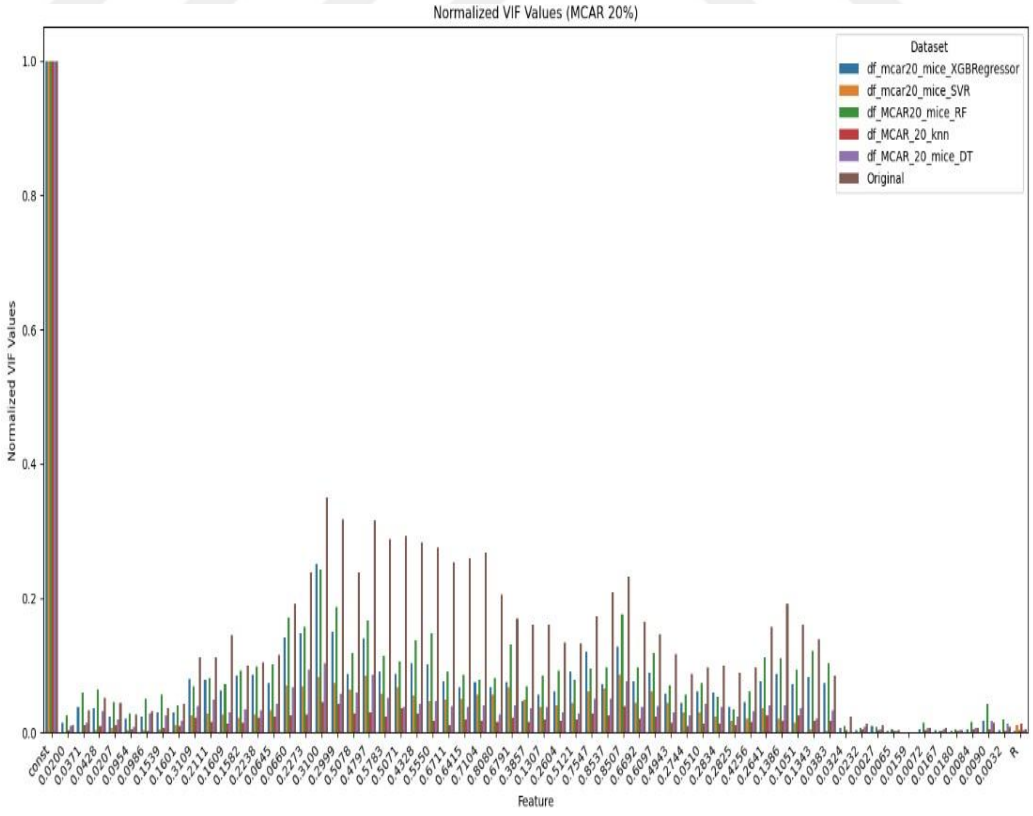


Figure 2.46 *Normalized VIF values (MCAR 20%)*

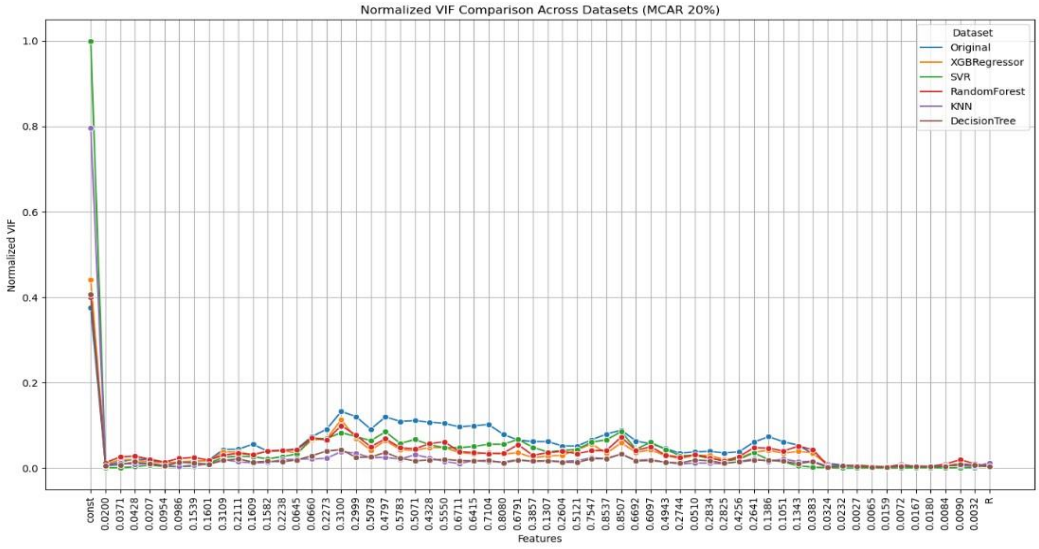


Figure 2.47 *Normalized VIF (MCAR 20%)*

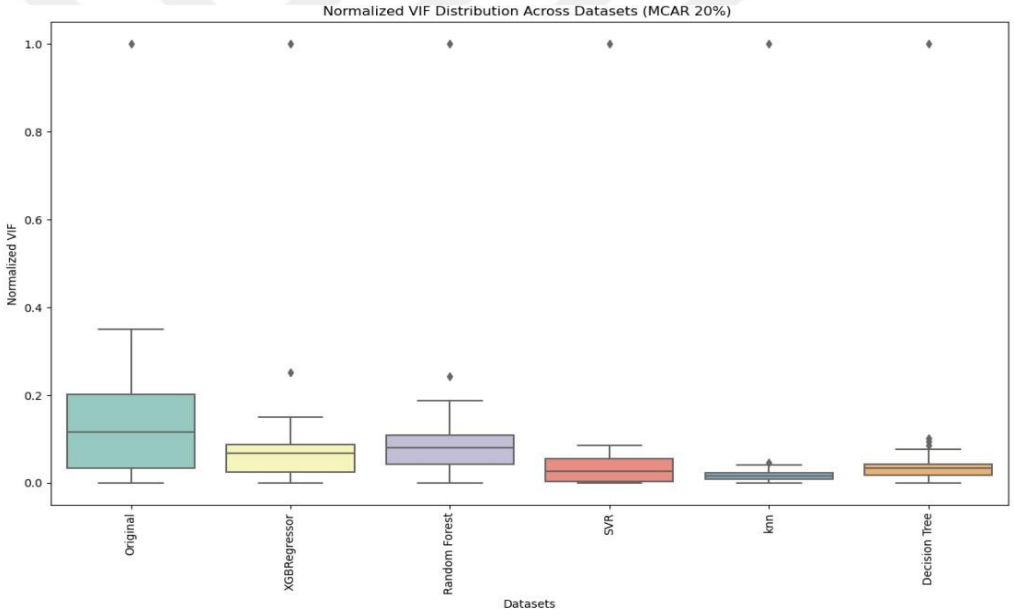


Figure 2.48 *Normalized VIF distribution (MCAR 20%)*

The results obtained from analyzing datasets generated using the MCAR mechanism and a data missingness percentage of 60% are presented in Figures 2.40-2.42. Figures 2.43-2.45 show the results obtained when there was a data missingness ratio of 40% . Finally, the results obtained with a missingness ratio of 20% are shown in Figures 2.46-2.48. Various datasets exhibit distinct Variance Inflation Factor (VIF) values, and the visualizations confirm the presence of these disparities. In the next stage, the Kruskal-Wallis H statistical test was employed to evaluate the differences in VIF

values across various imputation procedures. The Kruskal-Wallis H test is a statistical technique used to determine if several samples originate from the same distribution.

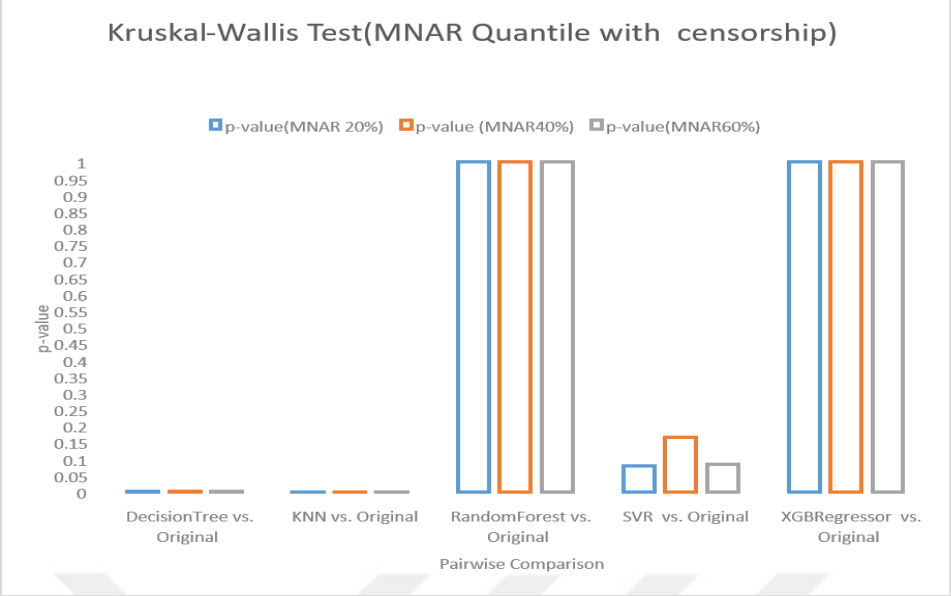
1. MNAR with quantile censorship datasets.

The datasets used in this section were generated under the Missing Not At Random (MNAR) mechanism combined with quantile censorship at missingness ratios of 20%, 40%, and 60%. The imputation performance was evaluated using multiple models, and the corresponding pairwise comparisons against the original data are presented in Table 2.1.

Table 2.1 *MNAR with quantile censorship pairwise comparison*

Pairwise Comparison	**p-value (MNAR-quant20%)**	**p-value (MNAR-quant 40%)**	**p-value (MNAR-quant 60%)**
DecisionTree vs. Original	0.007735	0.000838	0.002000
KNN vs. Original	0.000065	0.000017	0.001389
RandomForest vs. Original	1.000000	0.684994	1.000000
SVR vs. Original	0.157534	0.055162	1.000000
XGBRegressor vs. Original	1.000000	0.175405	1.000000

In addition, the Kruskal-Wallis test was applied to assess the overall variation in Variance Inflation Factor (VIF) values across different imputation models. The p-values obtained from this test were 0.012856 for 20% missing data, 0.000759 for 40%, and 0.00074 for 60%. All these values are well below the conventional significance threshold of 0.05, indicating that the null hypothesis that all groups have the same distribution can be confidently rejected. This outcome implies that significant differences exist in the VIF values among the tested models and datasets. However, despite these statistical differences, it can be inferred that the choice of imputation method does not substantially affect multicollinearity, as measured by VIF, in this specific MNAR with quantile censorship setting. The visual summary of the Kruskal–wallis test results is provided in Figure 2.49 below.



:

Figure 2.49 *Kruskal-wallis test (MNAR_quant)*

Figure 2.49 demonstrates that specific imputation techniques, such as MICE implemented using KNN and DecisionTree, have a notable impact on multicollinearity, as evidenced by their VIF values compared to other approaches.

2. MNAR datasets

Considering the datasets that have been created using the MNAR mechanism, the obtained results are displayed in Table 2.2 below.

Table 2.2 *MNAR pairwise comparison*

Pairwise Comparison	**p-value (MNAR 20%)**	**p-value (MNAR40%)**	**p-value (MNAR60%)**
DecisionTree vs. Original	0.001443	0.001159	0.001287
KNN vs. Original	0.000099	0.000104	0.000122
RandomForest vs. Original	1.000000	1.000000	1.000000
SVR vs. Original	0.079225	0.164872	0.084268
XGBRegressor vs. Original	1.000000	1.000000	1.000000

The findings of the Kruskal-Wallis test suggest that there are notable disparities in multicollinearity, as assessed by VIF, among the datasets. The p-values for missing not at random (MNAR) for 20%, 40%, and 60% of the imputed data are 0.0027, 0.0020, and 0.00278, respectively. The obtained result is shown in Figure 2.50 below.

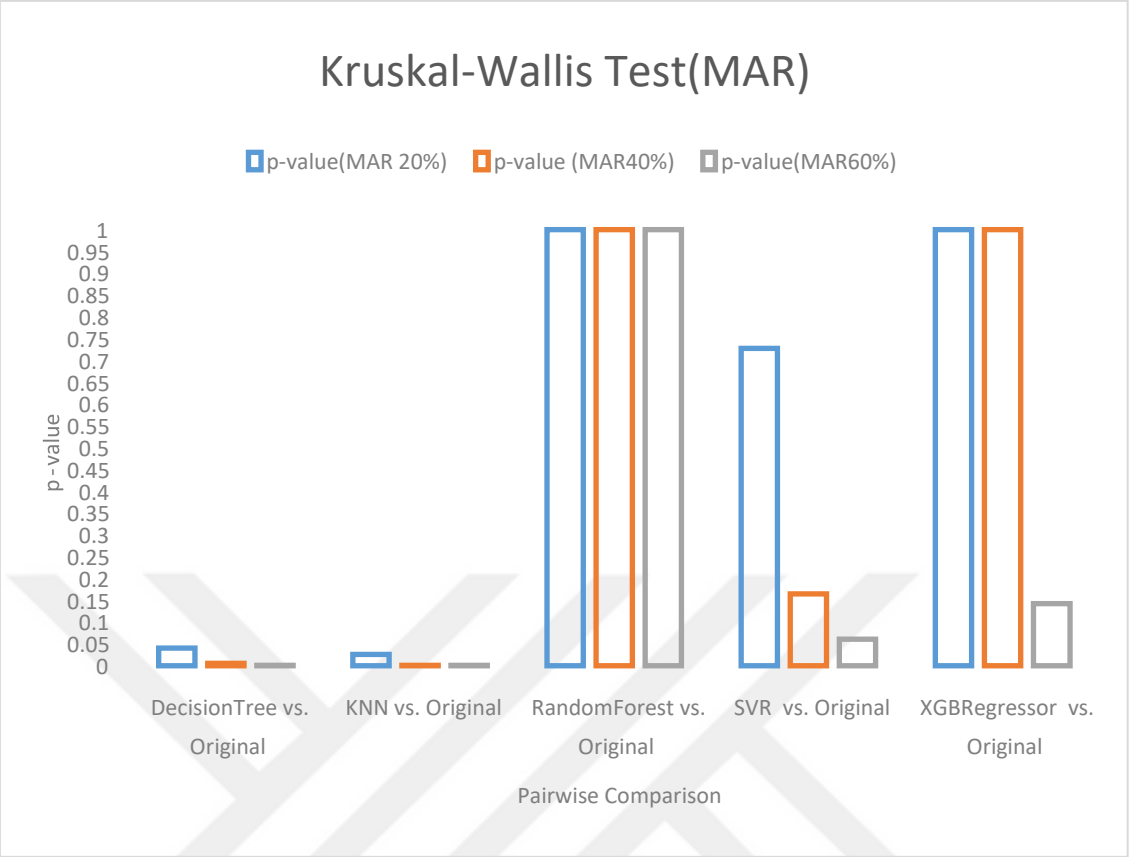


Figure 2.50 *Kruskal-wallis test (MNAR)*

The obtained p-values provide evidence to reject the null hypothesis, suggesting that there are substantial variations in multicollinearity across the datasets. This scenario shows that the selection of the imputation method, specifically DecisionTree and KNN, has a substantial impact on multicollinearity when compared to the original dataset as it is illustrated in the figures below.

3. MAR pairwise comparison

The results of the Kruskal-Wallis test reveal that there are statistically significant differences. When taking into consideration the datasets that have been produced by utilizing the MNAR mechanism, the results that were attained are presented in Table 2.3 below.

Table 2.3 *MAR pairwise comparison*

Pairwise Comparison	p-value (MAR 20%)	p-value (MAR40%)	p-value (MAR60%)
DecisionTree vs. Original	0.060377	0.005685	0.000012
KNN vs. Original	0.025552	0.000444	0.000013
RandomForest vs. Original	1.000000	1.000000	1.000000
SVR vs. Original	0.728158	0.164872	0.061019
XGBRegressor vs. Original	1.000000	1.000000	0.142336

The Kruskal-Wallis test resulted in p-values of 0.000005, 0.000213, and 0.008595 for MAR datasets with missingness ratios of 20%, 40%, and 60% respectively. These p-values are smaller than the significance level of 0.05, indicating that the null hypothesis is rejected. Therefore, it can be concluded that there are significant differences in multicollinearity, as measured by VIF, among the datasets. The abovementioned result can be seen in Figure 2.51 below.

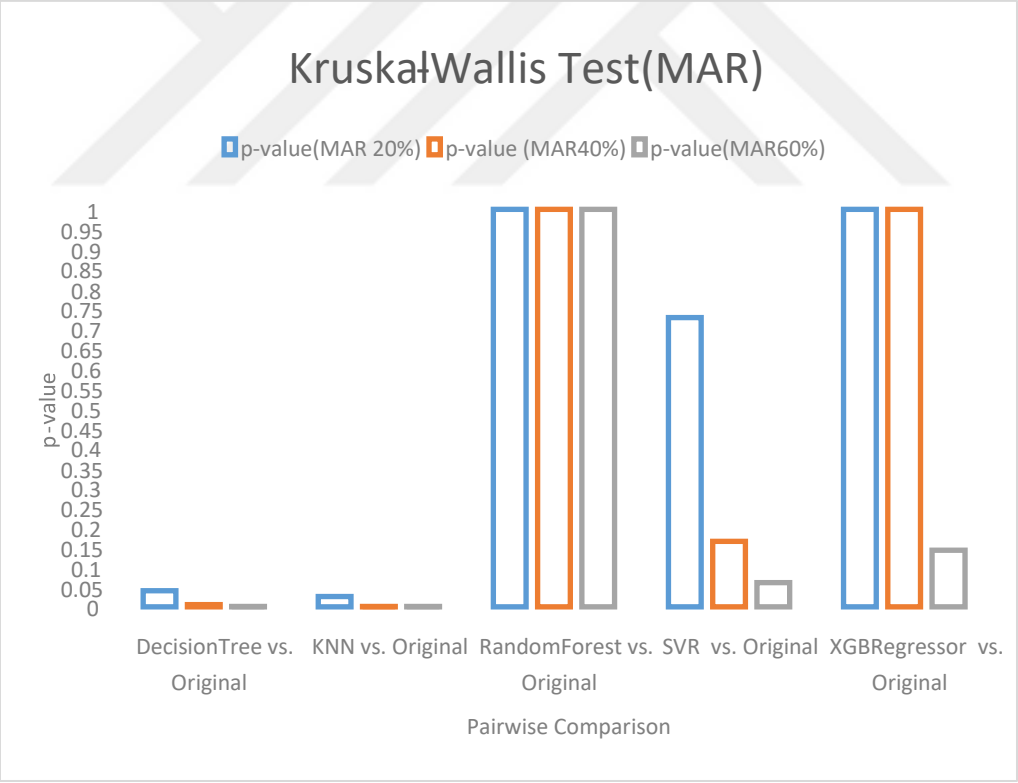


Figure 2.51 *Kruskal-wallis test(MAR)*

Figure 2.51 above demonstrates a notable pairwise distinction between the datasets imputed using MICE with KNN and Decision Tree as estimators, and the original dataset.

This indicates that the selection of the imputation method has a significant impact on multicollinearity in comparison to the original dataset.

4. MCAR pairwise comparison

While taking into account the datasets that have been produced through the application of the MCAR mechanism, the results that were achieved are presented in Table 2.4 below.

Table 2.4 *MCAR pairwise comparison*

Pairwise Comparison	**p-value (MCAR 20%)**	**p-value (MCAR40%)**	**p-value (MCAR60%)**
DecisionTree vs. Original	0.000087	0.000018	0.000012
KNN vs. Original	0.000038	0.000002	0.000013
RandomForest vs. Original	1.000000	1.000000	1.000000
SVR vs. Original	0.045322	0.042455	0.061019
XGBRegressor vs. Original	0.787208	0.813490	0.142336

The Kruskal-Wallis test yielded p-values of 0.000112, 0.000180, and 0.000526 for the datasets generated using the MCAR mechanism with missingness ratios of 20%, 40%, and 60% respectively. These results indicate significant differences in multicollinearity among the datasets. Given a substantial p-value obtained from the Kruskal-Wallis test, the null hypothesis can be confidently rejected, thereby confirming that Notable disparities are evident in several pairwise comparisons, particularly between the DecisionTree, KNN, and original datasets. This suggests that the selection of an imputation method has a substantial impact on the presence of multicollinearity in comparison to the original dataset. Nevertheless, there is a lack of noticeable disparities seen among the SVR, Random Forest, and XGBRegressor techniques. Figure 2.52 below illustrates the MCAR scenario.

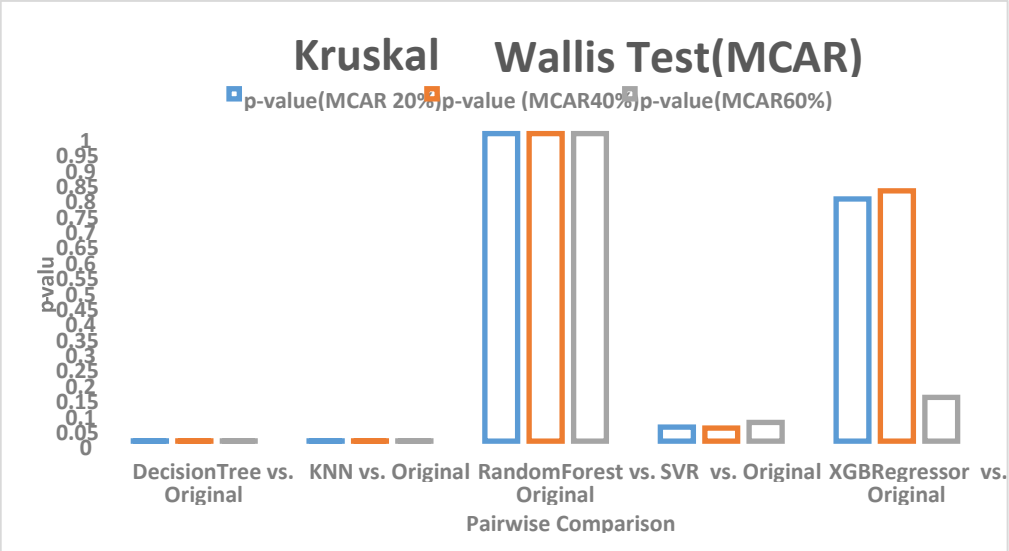


Figure 2.52 *Kruskal-wallis test (MCAR)*

2.5. Discussion (Part 2)

In this study, the impact of missing value imputation techniques on multicollinearity is assessed, with a particular focus on the use of the Multiple Imputation by Chained Equations (MICE) algorithm utilizing different estimators: XGBRegressor, SVR, Random Forest, KNN, and Decision Tree. The analysis employed visualizations and the Kruskal-Wallis H test to evaluate the disparities in datasets resulting from different imputation strategies. Several scenarios have been studied, including Missing Completely at Random (MCAR), Missing at Random (MAR), Missing Not at Random (MNAR), and MNAR with quantile censorship, to comprehensively understand how different missing data mechanisms affect multicollinearity. The findings indicate significant differences in the levels of multicollinearity depending on the imputation method used. Specifically, the implementation of MICE with KNN and Decision Tree as estimators showed a notable reduction in multicollinearity compared to other methods. This suggests that these two estimators are particularly effective in managing the correlations between variables introduced during the imputation process. The Kruskal-Wallis H test revealed statistically significant variations in multicollinearity levels across the different imputation techniques,reinforcing the importance of carefully selecting the appropriate estimator within the MICE framework to minimize multicollinearity.

3. WEIGHTED GUIDED AUTOENCODER WITH NOISE-AWARE MASKED LOSS FOR HIGH-DIMENSIONAL DATA IMPUTATION

3.1 Background

The primary objective of this study is to explore machine learning techniques for predicting and diagnosing COVID-19, as well as determining the most suitable healthcare department for COVID-19 patients. Several missing data imputation techniques have been used, like KNN with various numbers of neighbors and Multiple Imputation by Chained Equations (MICE); however, the obtained result was at an average level since there is a high ratio of missing data, which posed a significant challenge to these predictive tasks. Section 2 addresses the above-mentioned problem by examining how multicollinearity, a prevalent feature in clinical data, can negatively impact the quality of imputation when using regression-based techniques such as MICE, through a targeted case study on continuous clinical variables.The obtained result reveals significant limitations, notably the instability of imputed values and the distortion of inter-variable relationships, especially in high-dimensional dataset configurations. These findings encourage the development of stronger and more flexible imputation models that can handle complicated missing data patterns, multicollinearity, and non-linear relationships (like MNAR). This serves as the basis for Section 3, the Weighted Guided Autoencoder with Noise-Aware Masked Loss, a novel deep learning-based model is presented. Unlike conventional techniques, this method penalizes noisy reconstructions, maintains data variability, and focuses the learning process on missing entries by utilizing the representational power of neural networks and a custom loss function. It is especially made to perform better than current methods in high-dimensional, practical medical data imputation situations.

3.2. Introduction

In the contemporary data-driven era, information serves as a vital asset across diverse domains such as medicine and pharmacy, finance and economy, industry, agriculture, social sciences, and engineering. Despite its significance, real-world data frequently suffers from missing values, which pose substantial challenges for data analysis, statistical inference, and machine learning applications. As emphasized by Little and Rubin (2002), improper handling of missing data can lead to biased estimates and undermine the effectiveness of predictive models, particularly in tasks involving

regression and classification. Missing data mechanisms have been rigorously classified in both classical statistics and modern computational fields into three primary types: Missing Completely at Random (MCAR), Missing at Random (MAR), and Missing Not at Random (MNAR) (Jakobsen et al., 2017). To address the challenges associated with these mechanisms, numerous imputation strategies have been proposed. These methods range from traditional approaches such as mean, median, and mode substitution; Expectation-Maximization (EM); Multiple Imputation by Chained Equations (MICE); K-Nearest Neighbors (KNN); and regression-based techniques to more advanced models, including MissForest (Stekhoven and Bühlmann, 2012), matrix factorization (Mazumder et al., 2010), and deep learning frameworks. Some important deep learning methods include standard autoencoders, denoising autoencoders, generative adversarial imputation networks (GAIN), variational autoencoders, and transformer-based models, which have greatly improved the ability to impute missing data in complicated situations. Despite these advancements, several limitations persist. A key issue is the discrepancy between minimizing reconstruction loss and optimizing performance in downstream tasks such as classification or regression. Furthermore, many modern deep learning models demand large sample sizes for effective generalization, which is often impractical in real-world applications with limited data availability. In addition, certain models are prone to systematic biases due to their inadequate handling of the underlying missingness mechanism, particularly under MNAR conditions. Overfitting to noise, especially in high-missingness scenarios, further diminishes the robustness of these models (Mattei and Frellsen, 2019). To solve these problems, this study introduces a new imputation framework that effectively handles different types of missing data, especially MNAR and its version that involves quantile-based censorship (MNAR_QUANT). Taking inspiration from the guidance used in GAIN, the new method moves away from GAN designs and uses an autoencoder base, adding a special loss function to improve how well it fills in missing data. This composite loss includes (1) a Guided Masked Mean Squared Error (GM-MSE) component that emphasizes learning only on the missing entries, aligning with best practices in imputation-focused neural models (Gondara and Wang, 2018); (2) a Noise Loss component that discourages the reconstruction of randomly injected noise, following principles from denoising autoencoders (Vincent et al., 2008); and (3) a Variance Penalty that promotes diversity in imputed values and counteracts over-smoothing, an issue noted in overly constrained models (Mattei and Frellsen, 2019). The

resulting architecture, referred to as the Guided Autoencoder (Guided_AE), introduces a robust mechanism for handling missing data by integrating noise-aware masking and regularization techniques. The model was carefully tested under different situations of missing data—MCAR, MAR, MNAR, and MNAR_QUANT—and with different amounts of missing data (5%, 15%, 25%, 40%, and 60%). Experimental results demonstrated that the Guided_AE consistently outperforms conventional and deep learning-based imputation methods, highlighting its effectiveness and adaptability in high-dimensional and clinically relevant datasets.

3.3. Related Work

According to Little and Rubin (2002), missing data mechanisms can be categorized into Missing Completely at Random (MCAR), Missing at Random (MAR), and Missing Not at Random (MNAR)**.** In the case of MCAR, the probability of missingness is unrelated to either observed or unobserved data, implying that the missingness introduces no bias. In the context of MAR, the probability of missingness may depend on the observed data but not on the missing values themselves, and each mechanism demands a specific imputation strategy. However, for the MNAR scenario, the probability of missingness is related to the unobserved data, making valid inference significantly more challenging without explicitly modeling the missingness mechanisms. They also stated that the performance of imputation and inference methods is largely contingent on the missingness mechanism and that methods such as maximum likelihood and multiple imputation perform well under MAR and MCAR when the imputation model is correctly specified; their effectiveness deteriorates significantly under MNAR, often resulting in biased estimates unless the missingness mechanism itself is explicitly modeled. To improve the situation mentioned above, multivariate approaches such as Multiple Imputation by Chained Equations (MICE) have been proposed (Van Buuren and Groothuis-Oudshoorn, 2011). MICE models each variable conditionally using iterative regressions, which allow for uncertainty estimation by generating multiple imputations and combining results; however, it has been observed that the performance is limited in cases of non-linear or highly complex inter-variable relationships. Non-parametric methods have also been investigated, such as MissForest (Stekhoven and Bühlmann, 2012), which uses the Random Forest algorithm to iteratively impute missing values and has proven improved performance on mixed-type data, although it could be

computationally expensive on large datasets and is highly sensitive to hyperparameter tuning. Techniques such as probabilistic principal component analysis (PPCA) and Bayesian PCA (Ilin and Raiko, 2010) have been investigated by learning latent representations, and it has been observed that their effectiveness was linked to the linearity assumption. Matrix factorization (Mazumder, Hastie and Tibshirani, 2010) and spectral regularization (Hastie, Tibshirani and Wainwright, 2015) methods have been largely employed in the field of recommendation systems and bioinformatics for missing value imputations by exploiting low-rank structure. Several researchers have conducted comparative studies to assess the efficiency of imputation models. (Jerez et al., 2010). In their study, they conducted a comparison involving several statistically based traditional methods, such as mean and hot-deck imputation, with techniques like Multi-Layer Perceptron (MLP), Self-Organizing Maps (SOM), and k-Nearest Neighbors (KNN), and they stated that the machine learning-based method performed better than the statistical methods, especially in the case of complex datasets. In their study on the impact of imputation techniques on accuracy, Farhangfar, Greiner and Zinkevich (2008) argued that different imputation techniques could yield varying results, and some of them could influence the rates of classification error. The importance of selecting an appropriate imputation method is emphasized, as the chosen approach must align with the underlying missing data mechanism. Liew, Law and Yan (2011), in their study involving 19 imputation algorithms, concluded that among the aforementioned algorithms, no one outperformed the others across all scenarios. They also stated that the efficiency of an imputation strategy depends on the data context. In recent years, several researchers have shown a particular interest in deep learning-based imputation methods. However, existing studies often lack a comprehensive evaluation, with most approaches tested primarily under MCAR or MAR assumptions (Zhou, Arya and Bouadjenek, 2024). Furthermore, many fail to specify or justify the missingness mechanisms employed in their experimental setups. The more challenging Missing Not At Random (MNAR) remains significantly underexplored, despite its prevalence in real-world applications such as healthcare, finance, and social sciences (Khan et al., 2025). Moreover, very few works systematically examine how imputation models perform across a range of missingness ratios. They do not address more realistic forms of data loss, such as quantile-based MNAR mechanisms or data censoring (Schelter et al., 2018). Benchmark studies that do exist tend to focus on limited datasets or compare a narrow set of models, making it

difficult to draw generalizable conclusions about model robustness across different types of missingness. These gaps highlight the need for a comprehensive, multi-model, and multi-scenario evaluation framework that not only compares traditional and deep learning approaches but also does so under various controlled missingness patterns and proportions. Although several researchers express much interest in deep learning-based imputation, various studies remain limited in the scope of their evaluation. Yoon, Jordon and Schaar (2018), in their foundational work on GAIN, focused primarily on MAR settings without exploring MNAR or complex data censoring scenarios. Nazabal et al. (2020) proposed a variational autoencoder framework for heterogeneous data imputation, yet their experiments were restricted to synthetic MAR setups. Moreover, many benchmark studies use a limited number of datasets or compare only a small subset of models, making it difficult to generalize their findings.The literature also lacks rigorous evaluations across varying missingness ratios and realistic mechanisms such as quantile-censored MNAR, which are highly relevant to domains like healthcare and finance. These gaps underscore the need for a comprehensive, model-agnostic benchmarking framework that systematically compares both traditional and deep learning approaches across multiple missingness types and levels. In response to these limitations, this study proposes a comprehensive evaluation framework that benchmarks both traditional and modern imputation models under a wide range of missingness scenarios. In contrast to previous studies, this work explicitly simulates and evaluates four missingness mechanisms, MCAR, MAR, MNAR, and MNAR with quantile censorship, across five missingness levels (5%, 15%, 25%, 40%, and 60%). The proposed Guided Autoencoder (Guided_AE) is compared against a diverse set of baseline methods, including MICE, MissForest, EM, Matrix Factorization, standard Autoencoder, VAE, GAIN, Denoising Autoencoder, and Transformer-based models. This study is among the first works to systematically evaluate deep learning–based imputation across all major missingness types, highlighting strengths and limitations through both reconstruction error (RMSE) and downstream classification (ROC AUC).

3.4. Problem Formulation

Let $X \in \mathbb{R}^{n \times d}$ describe the full data matrix, in which n corresponds to the number of samples and d represents the feature dimensionality. In practical scenarios, missing values in $\boldsymbol{X}$ may be observed due to factors such as random omission, measurement errors,

or censoring. A binary mask matrix $M \in \{0,1\}^{n \times d}$, *where* $M_{ij} = 1$ *if* X_{ij} *is observed and* $M_{ij} = 0$ if it is missing, is defined. Accordingly, the observed input is expressed as:

$$X_{obs} = M \odot X \tag{3.1}$$

Here, $\odot$ denotes element-wise multiplication. The imputation problem aims to estimate the missing entries such that:

$$X_{ij} \approx X_{ij} \forall (i,j) \tag{3.1a}$$

where $M_{ij} = 0$

Formally, the objective is to learn a function

$$f\theta: \mathbb{R}n \times d \times \{0,1\}\{n \times d\} \rightarrow \mathbb{R}n \times d \tag{3.1b}$$

parameterized by θ, satisfying:

$$\hat{X} = f_\theta (X^{obs}, M) \tag{3.1c}$$

To support this learning, additional information is provided to the model in the form of a hint matrix $H \in \mathbb{R}\{0,1\}^{n \times d,}$ which is a partially observed version of the mask matrix *M*, as inspired by Yoon, Jordon and Schaar in 2018. Specifically, a matrix binary *H* is generated from a matrix *M* by randomly revealing a subset of known entries, introducing inductive bias. This encourages the model to distinguish observed and unobserved values during training.

$$H = Bernoulli\ (p) \odot M \tag{3.1d}$$

Here, $0 < p < 1$

To encourage variability and prevent overfitting, a random noise $N \in \mathbb{R}^{n \times d}$ is drawn from a Gaussian or uniform distribution and injected into the missing value during training. The aim is to balance reconstruction accuracy with regularization:

$$L_{total} = L_{masked} + \lambda_{noise} * l_{noise} + \lambda_{var} * L_{var} \tag{3.2}$$

Where L_{masked} corresponds to the reconstruction error over missing entries, λ_{noise} and $\lambda_{var,}$ $describes$ weighting hyperparameter. l_{noise} penalizes imputed values deviating toward noise and L_{var} enforces variance regularization to avoid collapsed solutions

This approach is designed to handle a range of missing data mechanisms, including MCAR, MAR, MNAR, and MNAR with quantile censoring, by generating diverse *M*

matrices within controlled experimental settings. Its objective extends beyond minimizing reconstruction error for missing values; it also emphasizes maintaining the predictive utility of the imputed data for downstream applications, such as disease classification.

3.4.1. Guided autoencoder imputation

The suggested method aims to use extra information from three sources: a mask matrix that shows which entries are observed, a random matrix that adds noise, and a partially informative hint matrix. These components collectively guide the decoder during the data reconstruction process.

Let:

- $X \in \mathbb{R}^{n \times d}$ describes the input data matrix with missing entries, where n denotes the sample size and d the number of features.
- $M \in \{0,1\}^{n \times d}$ denotes the mask matrix indicating observed entries such that M_k = 1 when the value at X_{ij} is present, and $M_{ij} = 0$ when missing.
- $R \in \mathbb{R}^{n \times d}$: Used to introduce stochasticity into the model, R denotes a random noise matrix, which is drawn from a uniform or normal distribution.
- $H \in \mathbb{R}^{n \times d}$: hint matrix derived from M by sampling partial information.

The model contains an encoder *f_enc* and a decoder *f_dec*. The input to the encoder concatenates three components:

- $X \odot M$: observed values
- $R \odot (1 - M)$: random noise replacing missing entries
- H: hint matrix
- The latent representation Z is computed as:

$$Z = f\_enc([X \odot M, R \odot (1 - M)]) \tag{3.3}$$

- The decoder reconstructs the data:

$$\hat{X} = f\_dec(Z,H) \tag{3.4}$$

- The total loss is defined as shown in Equation (3.2)

3.4.2 Masked reconstruction loss

The components of this loss function ensure that the model concentrates on imputing missing entries only, thereby avoiding alteration of already observed ones.

$$L_{masked} = \frac{1}{\sum(1-M)} + \sum_{i,i} (1 - M_{ij}) * (X_{ij} - \hat{X}_{ij})^2 \quad (3.5)$$

Here, X represents the original input matrix with missing values, $\hat{X}$ denotes the reconstructed output, and M describes the binary mask matrix

3.4.3 Noise-aware regularization

A regularization method is defined to prevent the model from overfitting to noise-driven patterns within the missing entries, thereby preventing excessive confidence during the reconstruction phase. The noise-aware regularization term is computed as shown in the mathematical expression below.

$$L_{noise} = \frac{1}{\sum(1-M)} + \sum_{i,j} (1 - M_{ij}) * (\hat{X}_{ij} - N_{ij})^2 \quad (3.6)$$

Here, N denotes a random noise matrix, where each element is independently drawn from a predefined distribution (normal or uniform).

3.4.4 Variance penalty

To avoid collapsed outputs and constant predictions, a penalty is added. The variance penalty is illustrated in the mathematical expression below.

$$L_{var} = -Var(\hat{X}) \quad (3.7)$$

The model is optimized by minimizing L_{total} using stochastic gradient descent. Once trained, $\hat{X}$ is used as the final imputed dataset. During training, λ_{noise} and λ_{var} are tuned to balance learning accuracy and generalization.

Empirically, this composite loss has shown strong performance under both standard (MCAR, MAR) and more difficult (MNAR, MNAR_QUANT) missingness conditions.

3.4.5 Algorithm of the proposed model and flowchart

- Step 1: Initialize model parameters θ
- Step 2: for epoch = 1 to MaxEpochs, do construct the input by concatenating: $X \odot M$ and $R \odot (1 - M)$
- Step 3:Pass the combined input through the decoder, as shown in Equation (3.3)

- Step 4: Decode the latent representation to reconstruct the full data as shown in Equation (3.4)
- Step 5: Compute the masked reconstruction loss using Equation (3.5)
- Step 6: Compute the noise penalty term as shown in Equation (3.6):
- Step 7: Compute the variance regularization term as shown in Equation (3.7)
- Step 8: Compute the total loss as shown in Equation (3.2)
- Step 9: Backpropagate the gradient $\nabla\theta L\_total$
- Step 10: Update the parameter of the model θ
- Step 11: end for
- Step 12:Return the final imputed matrix $\hat{X}$

Figure 3.11 presents a detailed overview of the architecture of the proposed guided autoencoder-based imputation model. The encoder component is designed to accept three key inputs: (1) the incomplete data matrix containing missing values, (2) a binary mask matrix that indicates the positions of the missing and observed entries, and (3) an optional noise matrix that introduces controlled stochasticity into the input to improve robustness. These inputs are processed through multiple hidden layers to extract meaningful representations that capture the underlying data structure.

During the decoding phase, a hint matrix is introduced to guide the reconstruction process, particularly focusing on the imputation of missing values. This matrix provides partial information about the original data, helping the decoder to distinguish between observed and unobserved entries, thereby enhancing reconstruction accuracy. The model is trained end-to-end using a custom-designed loss function that combines three essential components: (i) a masked reconstruction error that penalizes incorrect predictions only at missing entries, (ii) a noise penalty term that regularizes the effect of input noise, and (iii) a variance control term that discourages over-smoothed or trivial imputations by promoting output diversity.

The integration of these components within the loss function ensures that the model learns to impute missing values in a stable and generalizable manner. Upon convergence, the model produces a fully imputed dataset in which all originally missing entries are replaced with predicted values. This output can then be used in subsequent stages of data analysis, such as classification, clustering, or statistical inference.

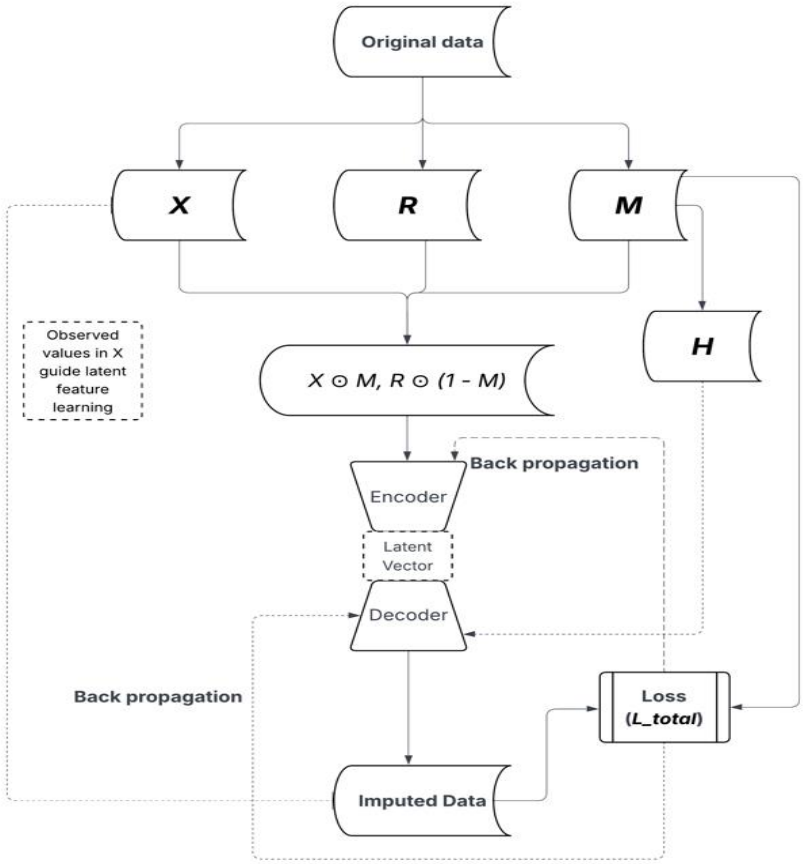


Figure 3.1 *Model architecture*

3.5. Experiment Setup (Part 3)

Each dataset contains a defined target variable, enabling the evaluation of the proposed imputation strategy in terms of its effectiveness and its impact on downstream classification tasks. Before imputation, standard preprocessing procedures were applied, including the encoding of categorical variables and the normalization of continuous features to ensure consistency and suitability for machine learning algorithms. A diverse set of widely used imputation techniques was selected to ensure a comprehensive comparison. These included Multiple Imputation by Chained Equations (MICE), which uses repeated regression modeling; MissForest, a method that is random forest–based, iterative and, non-parametric; Expectation-Maximization (EM), a statistical technique based on likelihood; and Matrix Factorization, which assumes that data can be simplified into lower-dimensional forms. Also, a number of deep learning methods were included, such as the regular Autoencoder (AE), Variational Autoencoder (VAE), Denoising Autoencoder (DEA), Generative Adversarial Imputation Network (GAIN), and models

based on Transformers. The experimental framework was organized into three phases. In the first phase, a qualitative analysis was conducted to explore the behavior and characteristics of the proposed model. The second phase included a numerical assessment of how well the model impute missing data using four publicly available datasets: the Framingham Heart Study dataset (4,240 samples, 15 features), the Pima Indians Diabetes dataset (768 samples, 8 features), the Stroke Prediction dataset (5,110 samples, 10 features), and the Cardiovascular Disease dataset (70,000 samples, 11 features). For each dataset, the proposed method was compared against state-of-the-art algorithms to assess its relative performance. In the third phase, the robustness of the model was examined under four commonly studied missingness mechanisms: Missing Completely at Random (MCAR), where missingness is unrelated to any variable; Missing at Random (MAR), where missingness depends only on observed variables; Missing Not at Random (MNAR), where missingness is related to unobserved values; and MNAR with Quantile Censoring (MNAR_QUANT), a specific MNAR scenario in which data is removed if it falls within particular quantiles of the distribution. To evaluate robustness, each missing data mechanism was tested across five missingness ratios: 5%, 15%, 25%, 40%, and 60%. The original, fully observed datasets were preserved as ground truth to assess reconstruction accuracy. To ensure robustness, each experiment was repeated five times with different random seeds, and five-fold cross-validation was applied in each run. Root Mean Squared Error (RMSE) was used to evaluate the accuracy of data reconstruction, while the Area Under the Receiver Operating Characteristic Curve (AUC) was employed to assess classification performance. Final results were averaged across repetitions and reported with standard deviations to capture both the stability and variability of each method.

3.6. Results

3.6.1. Effect of model components on imputation performance

The potential source of the proposed framework for missing data imputation originates from the used standard autoencoder-like architecture, the L_{masked} loss function, the noise penalty term L_{noise}, and the hint H.

The impact of each of the aforementioned components, namely, the guided reconstruction loss, the hint matrix, the noise-aware regularization term, and the variance penalty, on the overall performance of the proposed guided autoencoder model is

thoroughly evaluated. This analysis is conducted to capture the contribution of each individual element and understand how they collectively influence the imputation accuracy and generalization capability of the model.

The results presented in Tables 3.1-3.4 demonstrate a progressive improvement in performance as these components are incrementally incorporated into the model architecture and training objective. Specifically, models that utilize only the baseline reconstruction loss perform less effectively compared to those that integrate guidance and regularization terms. The most substantial performance gains are observed when all components are simultaneously employed, indicating that their combination yields a synergistic effect.

Figure 3.2 provides a visual summary of the performance metrics obtained under different model configurations, further corroborating the advantage of including all components in the proposed approach.

Table 3.1 *Components of Guided_AE(MCAR)*

Algorithm	Stroke	Framingham	Diabetes	Cardiovascular
AE	**0.1650**	**0.3055**	**0.8287**	**0.2912**
AE+Hint	0.1680	0.3485	0.8383	0.2734
	(-1.7%)	**(-14.1%)**	**(-1.2%)**	**(+6.1%)**
AE+Hint+Noise	0.1640	0.3076	0.7944	0.2756
	(+2.3%)	**(-0.7%)**	**(+4.1%)**	**(+5.4%)**
Guided_AE	0.1620	0.3023	0.7501	0.2665
	(+1.1%)	**(+1.1%)**	**(+9.5%)**	**(+8.5%)**

Table 3.2 *Components of Guided_AE(MAR)*

Algorithm	Stroke	Framingham	Diabetes	Cardiovascular
AE	**0.1550**	**0.5564**	**0.2434**	**0.5036**
AE+Hint	0.1600	0.6924	0.2667	0.4785
	(-3.6%)	**(-24.4%)**	**(-9.6%)**	**(+5.0%)**
AE+Hint+Noise	0.1550	0.5907	0.2620	0.4817
	(+3.4%)	**(-6.2%)**	**(-7.6%)**	**(+4.3%)**
Guided_AE	0.1480	0.5546	0.2243	0.4032
	(+4.6%)	**(+0.3%)**	**(+7.8%)**	**(+19.9%)**

Table 3.3 *Components of Guided_AE*(MNAR)

Algorithm	Stroke	Framingham	Diabetes	Cardiovascular
AE	**0.1810**	**0.5318**	**0.0786**	**0.5663**
AE+Hint	0.1800	0.5661	0.0760	0.5351
	(+0.6%)	**(-6.5%)**	**(+3.2%)**	**(+5.5%)**
AE+Hint+Noise	0.1760	0.5000	0.0626	0.5523
	(+2.2%)	**(+6.0%)**	**(+20.3%)**	**(+2.5%)**
Guided_AE	0.1750	0.4665	0.0693	0.5185
	(+0.2%)	**(+12.3%)**	**(+11.7%)**	**(+8.4%)**

Table 3.4 *Components of Guided_AE*(MNAR_Quant)

Algorithm	Stroke	Framingham	Diabetes	Cardiovascular
AE	**0.1880**	**0.3644**	**0.1380**	**0.3837**
AE+Hint	0.1480	0.4040	0.1356	0.3589
	(+20.8%)	**(-10.9%)**	**(+1.7%)**	**(+6.5%)**
AE+Hint+Noise	0.1480	0.4003	0.1321	0.3682
	(+0.2%)	**(-9.9%)**	**(+4.3%)**	**(+4.0%)**
Guided_AE	0.1400	0.3417	0.1258	0.3711
	(+5.8%)	**(+6.2%)**	**(+8.8%)**	**(+3.3%)**

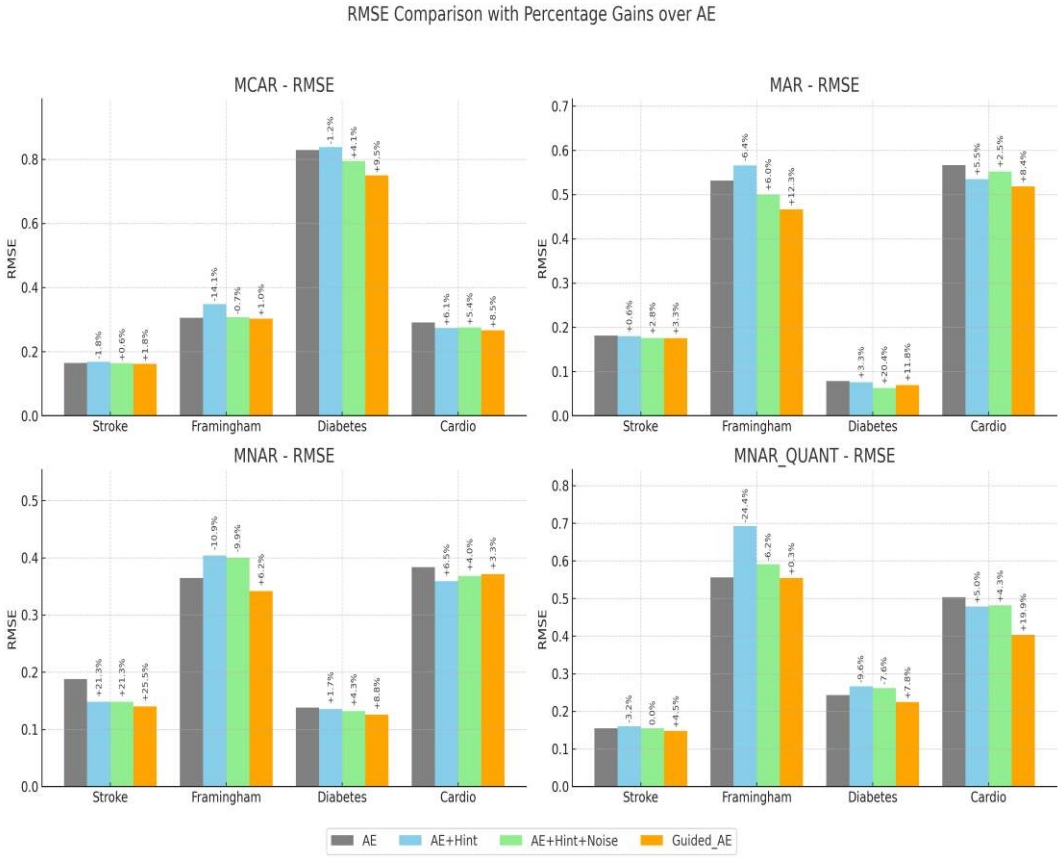


Figure 3.2 *Model component performance enhancement*

3.6.2. Model performance evaluation

This section presents the results from testing the proposed imputation model using four different types of missing data: MCAR, MAR, MNAR, and MNAR with censorship (MNAR_QUANT). The evaluation was conducted across four publicly available datasets: the Stroke Prediction dataset, the Framingham Heart Study dataset, the Cardiovascular Disease dataset, and the Pima Indians Diabetes dataset. For each dataset and missingness configuration, the proposed model was compared against a range of widely used imputation techniques. These included methods based on deep learning, like standard Autoencoder (AE), Variational Autoencoder (VAE), Denoising Autoencoder (DAE), Transformer-based models, and Generative Adversarial Imputation Network (GAIN), as well as traditional methods like MissForest, Multiple Imputation by Chained Equations (MICE), Expectation-Maximization (EM), and Matrix Factorization. To see how well the models performed, two measures were used: the Root Mean Squared Error (RMSE), which shows how closely the data matches the original, and the Area Under the

Receiver Operating Characteristic Curve (ROC AUC), which shows how effective the models are at sorting data into categories. Each experiment was repeated five times with different random seeds, and the results are reported as mean ± standard deviation. To make it easier to compare the results, visual aids such as bar charts, line plots, and heatmaps were incorporated to illustrate how performance varies across different missingness mechanisms. The detailed RMSE outcomes for each experimental setting and model configuration are summarized in Tables 3.5-3.8.

Table 3.5 *Result RMSE(MCAR)*

Algorithm	Stroke	Framingham	Diabetes	Cardiovascular
GUIDED_AE	**0.181 ±0.004**	**0.166±0.093**	**0.323 ± 0.058**	**0.266 ± 0.136**
AE	0.186 ± 0.007	0.400±0.147	0.424 ± 0.064	0.388 ± 0.123
DEA	0.212 ± 0.012	0.483±0.041	0.459 ± 0.072	0.284 ± 0.057
EM	0.225 ± 0.119	0.246±0.096	0.252 ± 0.051	0.606 ± 0.083
GAIN	0.239 ± 0.009	0.155±0.049	0.316 ± 0.056	0.264 ± 0.033
MATRIX	0.251 ± 0.012	0.472±0.189	0.317 ± 0.084	0.744 ± 0.073
MICE	0.229 ± 0.007	0.234±0.159	0.261 ± 0.049	0.839 ± 0.063
MISSFOREST	0.235 ± 0.012	0.526±0.108	0.521 ± 0.079	0.721 ± 0.115
TRANSFORMERS	0.219 ± 0.009	0.279±0.089	0.463 ± 0.068	0.325 ± 0.072
VAE	0.185 ± 0.001	0.385±0.113	0.462 ± 0.067	0.256 ± 0.095

Table 3.6*: Result RMSE(MAR)*

Algorithm	Stroke	Framingham	Diabetes	Cardiovascular
GUIDED_AE	**0.164 ± 0.003**	**0.248 ± 0.132**	**0.308 ± 0.343**	**0.411 ± 0.084**
AE	0.213 ± 0.012	0.570 ± 0.093	0.486 ± 0.226	0.737 ± 0.027
DEA	0.134 ± 0.011	0.411 ± 0.067	0.518 ± 0.250	0.533 ± 0.075
EM	0.197 ± 0.010	0.251 ± 0.134	0.281 ± 0.243	0.553 ± 0.135
GAIN	0.172 ± 0.011	0.138 ± 0.108	0.328 ± 0.233	0.374 ± 0.051
MATRIX	0.280 ± 0.111	0.742 ± 0.162	0.370 ± 0.294	0.770 ± 0.080
MICE	0.167 ± 0.009	0.240 ± 0.012	0.288 ± 0.252	0.583 ± 0.078
MISSFOREST	0.262 ± 0.010	0.515 ± 0.147	0.589 ± 0.276	0.915 ± 0.066
TRANSFORMERS	0.305 ± 0.007	0.271 ± 0.157	0.529 ± 0.243	0.413 ± 0.044
VAE	0.222 ± 0.016	0.229 ± 0.193	0.517 ± 0.241	0.549 ± 0.066

Table 3.7 *Result RMSE(MNAR)*

Algorithm	Stroke	Framingham	Diabetes	Cardiovascular
GUIDED_AE	**0.139±0.010**	**0.166 ± 0.088**	**0.322 ± 0.030**	**0.601 ± 0.095**
AE	0.275 ±0.010	0.615 ± 0.143	0.375 ± 0.074	0.692 ± 0.059
DEA	0.265 ± 0.013	0.383 ± 0.096	0.406 ± 0.064	0.735 ± 0.043
EM	0.207 ± 0.003	0.448 ± 0.120	0.190 ± 0.081	0.672 ± 0.066
GAIN	0.282 ± 0.009	0.090 ± 0.044	0.249 ± 0.056	0.829 ± 0.113
MATRIX	0.286 ± 0.011	0.551 ± 0.194	0.255 ± 0.088	0.766 ± 0.078
MICE	0.276 ± 0.009	0.341 ± 0.144	0.214 ± 0.067	0.692 ± 0.047
MISSFOREST	0.203 ± 0.005	0.430 ± 0.199	0.465 ± 0.055	0.895 ± 0.060
TRANSFORMERS	0.292 ± 0.005	0.276 ± 0.135	0.413 ± 0.067	0.714 ± 0.047
VAE	0.417 ± 0.008	0.476 ± 0.077	0.410 ± 0.066	0.706 ± 0.050

Table 3.8 *Result RMSE(MNAR_Quant)*

Algorithm	Stroke	Framingham	Diabetes	Cardiovascular
GUIDED_AE	**0.153±0.011**	**0.141 ±0.089**	**0.337 ± 0.253**	**0.077 ± 0.085**
AE	0.233 ± 0.007	0.300 ± 0.128	0.493 ± 0.075	0.167 ± 0.039
DEA	0.212 ± 0.013	0.287 ± 0.165	0.542 ± 0.083	0.164 ± 0.055
EM	0.232 ± 0.009	0.257 ± 0.037	0.303 ± 0.157	0.199 ± 0.064
GAIN	0.329 ± 0.006	0.157 ± 0.192	0.387 ± 0.155	0.122 ± 0.083
MATRIX	0.363 ± 0.014	0.442 ± 0.174	0.394 ± 0.153	0.142 ± 0.104
MICE	0.264 ± 0.011	0.241 ± 0.074	0.328 ± 0.157	0.217 ± 0.092
MISSFOREST	0.288 ± 0.008	0.464 ± 0.171	0.615 ± 0.090	0.218 ± 0.054
TRANSFORMERS	0.157 ± 0.008	0.253 ± 0.064	0.544 ± 0.078	0.107 ± 0.070
VAE	0.169 ± 0.009	0.339 ± 0.140	0.539 ± 0.082	0.217 ± 0.053

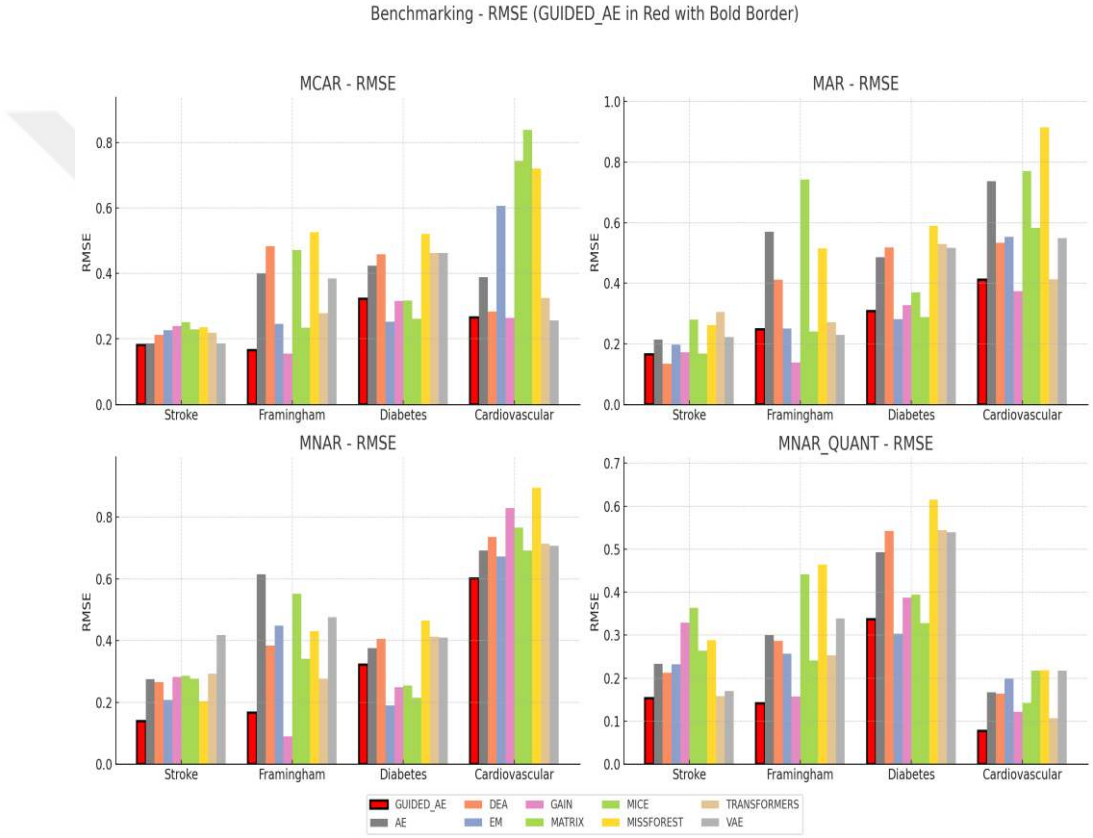


Figure 3.3 *RMSE result across datasets*

The proposed model achieved superior performance, establishing itself as an effective and adaptable solution for missing data imputation across different scenarios. Its architecture, featuring hint-guided decoding, noise-aware regularization, and masked reconstruction loss, proved robust in both simple missingness mechanisms (MCAR, MAR) and more complex ones (MNAR, MNAR_QUANT). While GAIN showed competitive performance and even outperformed the proposed model under certain conditions, particularly on the Framingham dataset with MAR and MNAR, this

underscores the value of adversarial learning with hint guidance. Traditional methods like MICE, EM, and matrix factorization performed relatively well on smaller, simpler datasets but experienced significant performance drops on larger or more complex ones, highlighting their limitations. Overall, the findings demonstrate the strong generalization ability, flexibility, and consistency of the proposed model , as visualized in Figures 3.3 and 3.4.

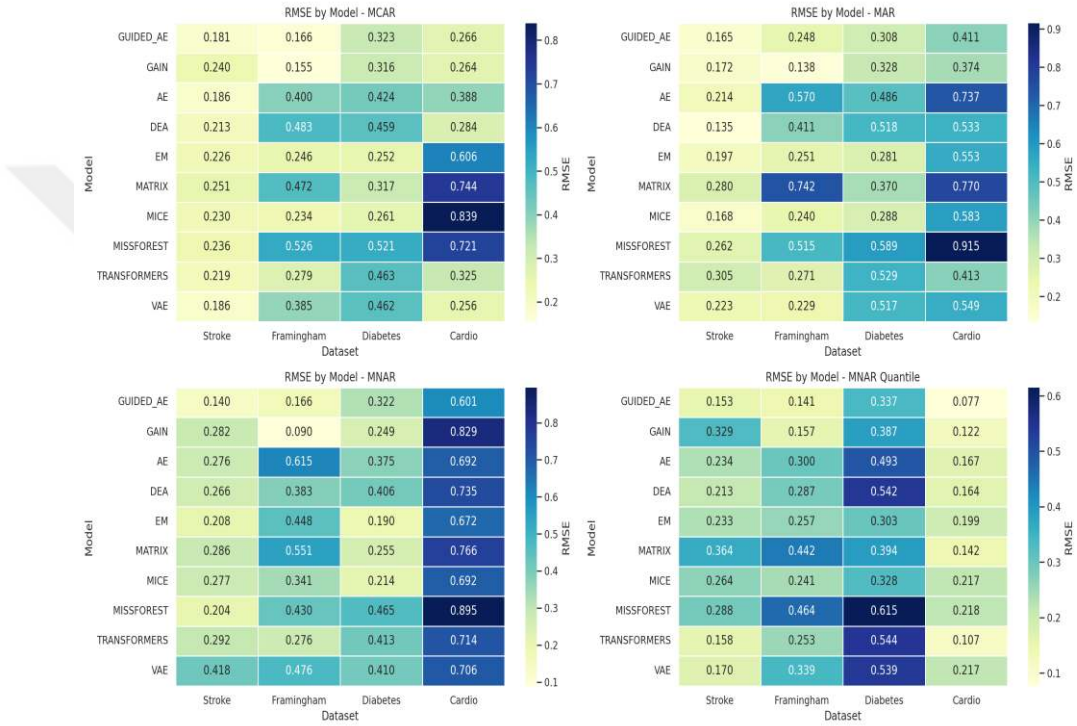


Figure 3.4 *Heatmap- results*

To evaluate classification performance following imputation, the Area Under the Receiver Operating Characteristic Curve (ROC AUC) was computed for all models across four real-world healthcare datasets: Stroke, Framingham, Diabetes, and Cardiovascular under four distinct missingness mechanisms: MCAR, MAR, MNAR, and MNAR with quantile censorship (MNAR_QUANT). The results are summarized in Tables 3.9–3.12 and visualized in Figure 3.5. The proposed model demonstrated strong performance, particularly on the Cardiovascular dataset, achieving ROC AUC scores of 0.944 and 0.934 under MCAR and MAR, respectively, indicating its robustness even in complex data environments. On the Framingham dataset, GAIN outperformed all other models under both MAR and MCAR conditions, attaining scores of 0.827 and 0.835,

respectively, underscoring the effectiveness of adversarial learning approaches enhanced by hint-based guidance. In contrast, traditional imputation techniques such as MissForest, MICE, and Matrix Factorization consistently yielded lower ROC AUC scores, particularly under MAR and MNAR_QUANT conditions in datasets characterized by higher feature interdependencies, such as the Cardiovascular dataset. These findings highlight the advantages of hybrid deep learning models in producing accurate imputations that preserve the predictive structure of the data, especially under challenging and non-random missingness scenarios (Jerez et al., 2010).

Table 3.9 *Result -ROC AUC (MCAR)*

Algorithm	Stroke	Framingham	Diabetes	Cardiovascular
GUIDED_AE	**0.879 ±**	**0.855 ± 0.123**	**0.897 ± 0.083**	**0.944 ± 0.069**
AE	0.833 ±	0.820 ± 0.116	0.862 ± 0.075	0.935 ± 0.086
DEA	0.863 ±	0.801 ± 0.099	0.863 ± 0.069	0.904 ± 0.090
EM	0.785 ±	0.694 ± 0.078	0.854 ± 0.077	0.895 ± 0.054
GAIN	0.883 ±	0.835 ± 0.121	0.853 ± 0.093	0.894 ± 0.074
MATRIX	0.836 ±	0.593 ± 0.033	0.741 ± 0.035	0.739 ± 0.015
MICE	0.855 ±	0.650 ± 0.035	0.767 ± 0.056	0.759 ± 0.015
MISSFOREST	0.858 ±	0.607 ± 0.038	0.727 ± 0.052	0.723 ± 0.030
TRANSFORMERS	0.879 ±	0.844 ± 0.112	0.889 ± 0.071	0.910 ± 0.083
VAE	0.846 ±	0.779 ± 0.122	0.874 ± 0.074	0.920 ± 0.087

Table 3.10 *Result -ROC AUC (MAR)*

Algorithm	Stroke	Framingham	Diabetes	Cardiovascular
GUIDED_AE	**0.904 ± 0.032**	**0.783 ± 0.140**	0.866 ± 0.076	**0.911 ± 0.086**
AE	0.821 ± 0.019	0.798 ± 0.129	0.832 ± 0.076	0.913 ± 0.078
DEA	0.863 ± 0.010	0.764 ± 0.110	0.829 ± 0.07	0.881 ± 0.092
EM	0.842 ± 0.034	0.650 ± 0.050	0.828 ± 0.061	0.860 ± 0.071
GAIN	0.858 ± 0.031	0.827 ± 0.150	0.824 ± 0.066	0.919 ± 0.080
MATRIX	0.743 ± 0.033	0.618 ± 0.014	0.767 ± 0.021	0.737 ± 0.027
MICE	0.826 ± 0.034	0.610 ± 0.008	0.772 ± 0.017	0.759 ± 0.011
MISSFOREST	0.805 ± 0.021	0.606 ± 0.016	0.751 ± 0.023	0.738 ± 0.025
TRANSFORMERS	0.858 ± 0.024	0.766 ± 0.122	0.839 ± 0.064	0.879 ± 0.080
VAE	0.835 ± 0.035	0.723 ± 0.095	0.822 ± 0.051	0.881 ± 0.086

Table 3.11 *Result -ROC AUC (MNAR)*

Model	Stroke	Framingham	Diabetes	Cardiovascular
GUIDED_AE	**0.922 ± 0.024**	**0.821 ± 0.149**	**0.892 ± 0.09**	**0.937 ± 0.072**
AE	0.802 ± 0.028	0.820 ± 0.123	0.866 ± 0.071	0.940 ± 0.083
DEA	0.827 ± 0.021	0.795 ± 0.112	0.865 ± 0.077	0.900 ± 0.094
EM	0.873 ± 0.023	0.702 ± 0.093	0.865 ± 0.082	0.880 ± 0.062
GAIN	0.851 ± 0.026	0.831 ± 0.116	0.852 ± 0.069	0.892 ± 0.077
MATRIX	0.820 ± 0.019	0.614 ± 0.040	0.747 ± 0.046	0.741 ± 0.014
MICE	0.862 ± 0.022	0.660 ± 0.031	0.759 ± 0.043	0.744 ± 0.016
MISSFOREST	0.882 ± 0.025	0.606 ± 0.022	0.747 ± 0.054	0.722 ± 0.026
TRANSFORMERS	0.820 ± 0.016	0.834 ± 0.131	0.894 ± 0.075	0.914 ± 0.087
VAE	0.853 ± 0.024	0.779 ± 0.116	0.875 ± 0.087	0.908 ± 0.092

Table 3.12 *Result -ROC AUC (MNAR_Quant)*

Model	Stroke	Framingham	Diabetes	Cardiovascular
GUIDED_AE	**0.884 ±0.028**	**0.753 ± 0.100**	**0.833 ± 0.061**	**0.882 ± 0.084**
AE	0.817 ± 0.019	0.697 ± 0.056	0.816 ± 0.062	0.859 ± 0.089
DEA	0.887 ± 0.033	0.696 ± 0.059	0.807 ± 0.033	0.847 ± 0.084
EM	0.838 ± 0.025	0.684 ± 0.074	0.786 ± 0.033	0.859 ± 0.077
GAIN	0.841 ± 0.036	0.819 ± 0.123	0.818 ± 0.072	0.905 ± 0.086
MATRIX	0.793 ± 0.040	0.622 ± 0.033	0.773 ± 0.019	0.738 ± 0.023
MICE	0.797 ± 0.036	0.632 ± 0.021	0.775 ± 0.016	0.754 ± 0.012
MISSFOREST	0.817 ± 0.029	0.618 ± 0.022	0.771 ± 0.018	0.740 ± 0.023
TRANSFORMERS	0.835 ± 0.021	0.729 ± 0.088	0.824 ± 0.044	0.856 ± 0.086
VAE	0.769 ± 0.020	0.669 ± 0.040	0.811 ± 0.035	0.851 ± 0.086

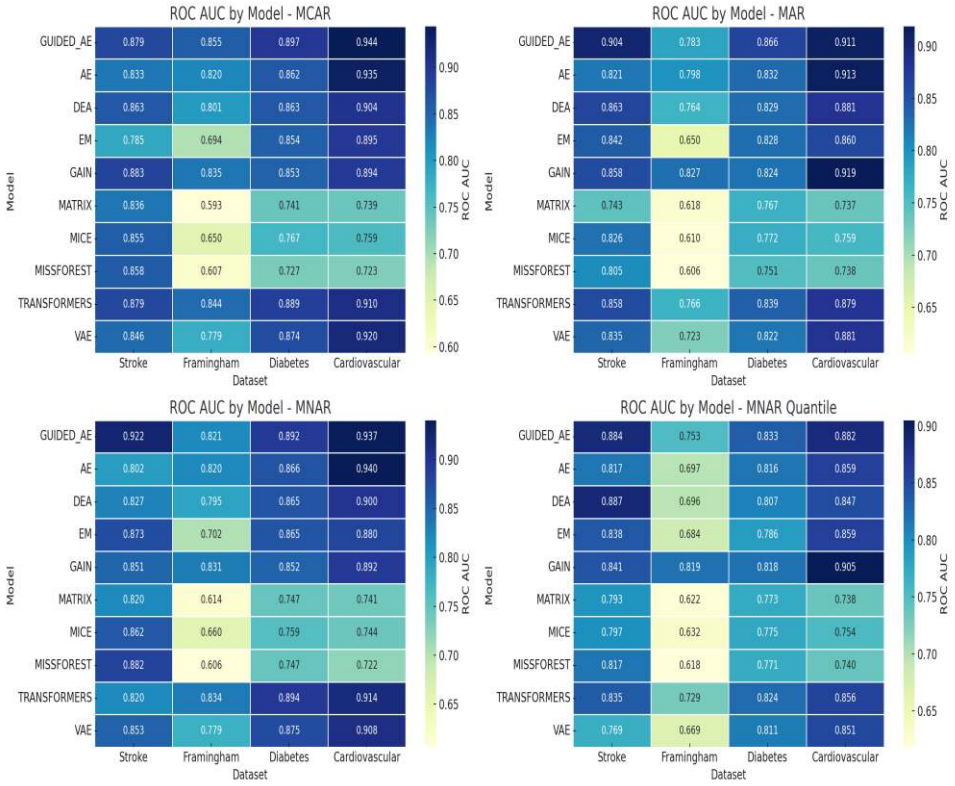


Figure 3.5 *Hetamap- ROC AUC*

The performance of the proposed model was evaluated across multiple levels of missingness. The results, obtained from the Stroke Prediction dataset with missingness ratios of 5%, 15%, 25%, 40%, and 60%, are presented in the figures below. Figure 3.6 reports the Root Mean Squared Error (RMSE), while Figure 3.7 displays the Area Under the ROC Curve (ROC AUC). The model consistently achieved low RMSE values, particularly under MNAR and MNAR with quantile censorship conditions, demonstrating its robustness in reconstructing data even in scenarios involving non-random and distributionally censored missingness.

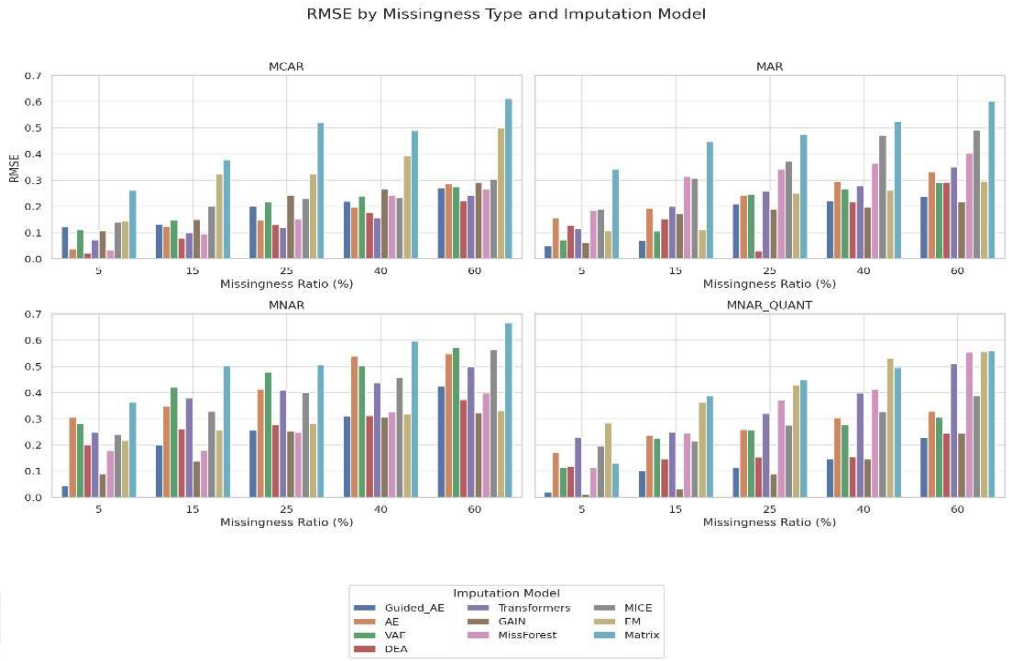


Figure 3.6 *RMSE result across missingness ratios*

Figure 3.7 *ROC AUC by missingness ratios*

Figures 3.8 and 3.9 below illustrate the results of several imputation algorithms across distinct types and ratios of data missingness. Each heatmap subplot illustrates a unique missingness mechanism, with performance assessed either RMSE (where lower values are preferred) or ROC AUC (where higher values are advantageous).

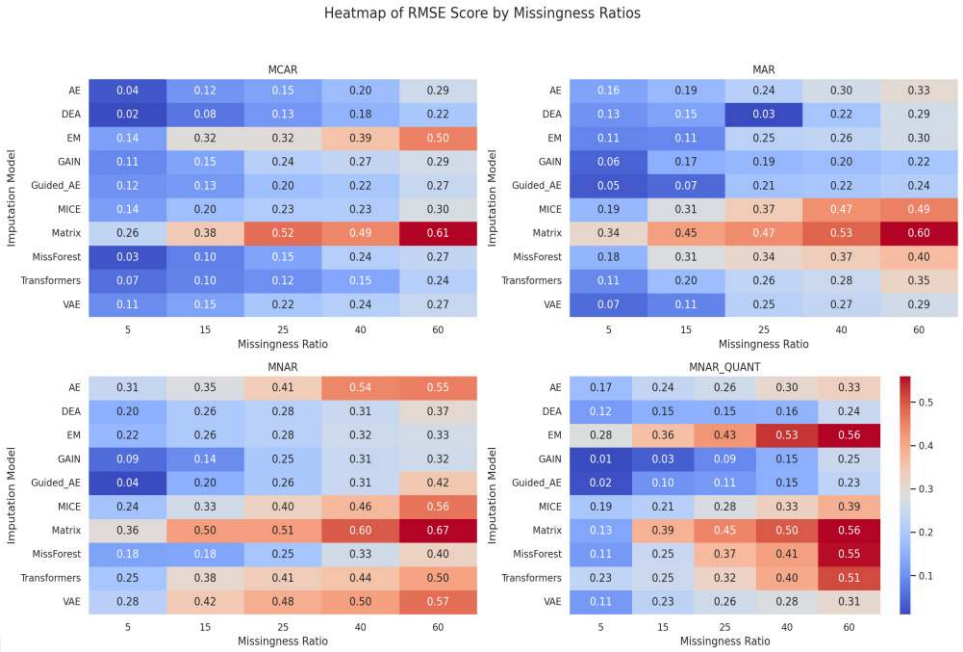


Figure 3.8 *Heatmap of RMSE by missingness ratios*

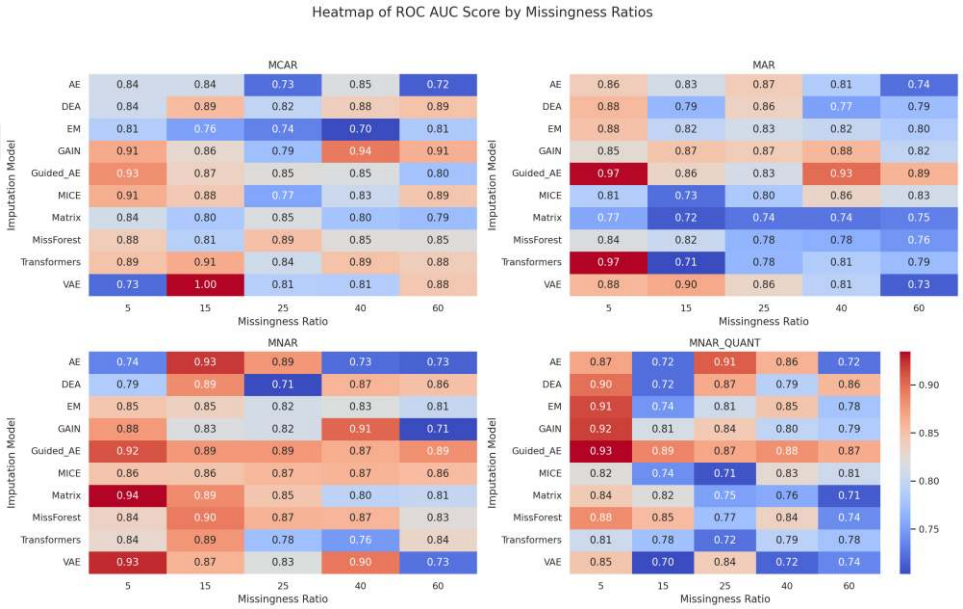


Figure 3.9 *Heatmap of ROC AUC score by missingness Ratios*

The obtained AUROC results reflect the ability of the model to maintain the quality of its imputations while preserving predictive structure, performing comparably to generative models like GAIN. The proposed model consistently achieves low RMSE and high AUROC across all scenarios, with particularly strong performance under MNAR and MNAR with quantile censorship, which are among the most challenging missingness

mechanisms. In contrast, traditional approaches such as EM and MICE, along with simpler neural models like AE, show substantial performance drops as missingness becomes structured or non-random. This evidence confirms that the proposed model not only reconstructs missing values effectively but also retains the underlying structure necessary for accurate downstream predictions. Figure 3.10 illustrates how RMSE evolves with increasing sample size for three imputation models: AE, GAIN, and Guided_AE. Sample sizes tested include 500, 1000, 2000, and 4000 instances. To maintain consistency, the evaluation was conducted using only the Stroke dataset. A general decrease in RMSE is observed as sample size increases, indicating improved imputation accuracy with more data. Under difficult conditions such as MNAR and MNAR_QUANT, Guided_AE consistently outperforms both AE and GAIN, achieving lower RMSE values. Performance differences become more apparent with larger sample sizes, highlighting the model's ability to leverage data richness. In contrast, under simpler mechanisms like MCAR and MAR, differences between models are minimal, suggesting that such scenarios can be adequately handled by conventional methods like AE. However, under MNAR and MNAR_QUANT, where missingness is dependent on unobserved or censored values, Guided_AE maintains a clear advantage, demonstrating robustness in complex imputation settings.

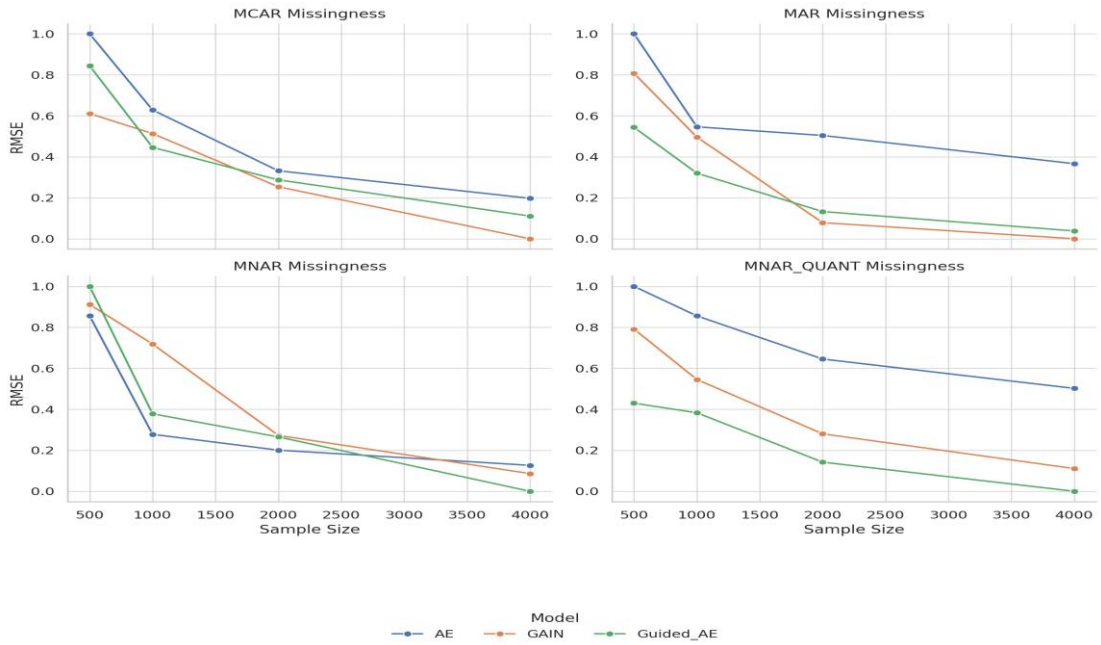


Figure 3.10*: Sample size analysis*

To assess the impact of input dimensionality on imputation performance, four feature subsets of increasing size (3, 5, 8, and 11 features) were systematically tested. This evaluation was conducted under four different missingness mechanisms: MCAR, MAR, MNAR, and MNAR with quantile censorship. Results indicate that as the number of input features increases, the model's performance consistently improves. Notably, under more complex missingness conditions such as MNAR and MNAR_QUANT, the proposed model continues to exhibit strong robustness and scalability. These findings highlight the model’s capacity to generalize effectively across varying missingness scenarios and its ability to exploit richer feature representations more efficiently than traditional methods. The corresponding results are presented in Figure 3.11, where darker cells reflect better performance. The proposed model consistently demonstrates superiorimputation accuracy, particularly under MNAR and MNAR with quantile censorship settings.

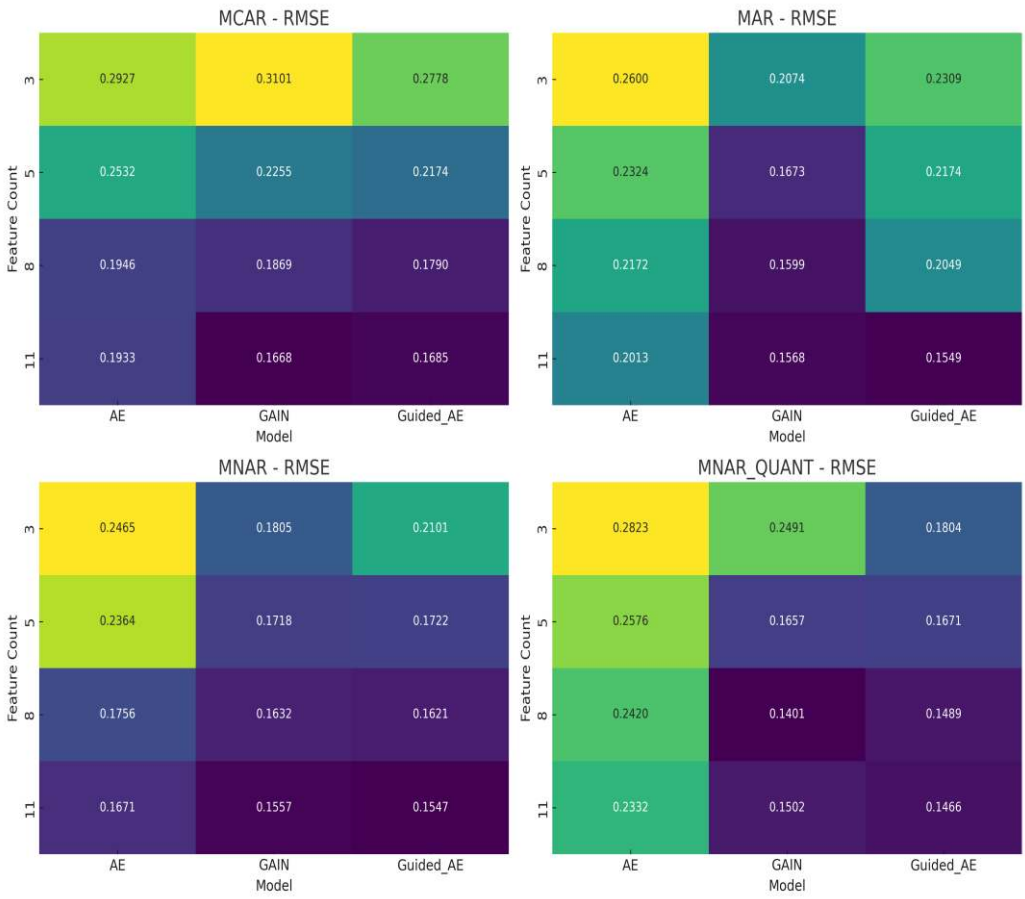


Figure 3.11 *RMSE - features count visualization*

3.7. Discussion (Part 3)

A novel deep learning-based framework for imputing missing values in high-dimensional healthcare datasets was presented in this study. The core of the proposed model is a meticulously crafted composite loss function that consists of (1) a noise-aware regularization term that promotes robustness by introducing stochastic noise into missing components during training; (2) a guided masked mean squared error (GM-MSE) loss that uses a binary mask to target only the missing entries; and (3) a variance penalty that encourages expressive reconstructions while suppressing trivial solutions. This multi-term loss improves generalization across a range of data conditions while allowing the model to precisely target imputation learning where it matters. Sample size, feature count, and missingness ratio (ranging from 5% to 60%) were systematically varied to evaluate the proposed model across several missing data mechanisms, including MCAR, MAR, MNAR, and MNAR with quantile censorship. Particularly in the MNAR and MNAR with quantile censorship scenarios, which are the most difficult because of non-random missingness patterns (Yoon, Jordon, and Schaar, 2018), the model continuously outperformed baseline autoencoders and achieved competitive or superior performance compared to GAIN. To assess the usefulness of the imputed data, a downstream classification task involving diabetes diagnosis, cardiovascular risk assessment, stroke prediction, and Framingham Heart Study datasets has been conducted. The capacity of the model to maintain class-discriminative structure, which is essential for clinical inference, was demonstrated by the consistently highest AUC-ROC scores obtained from the proposed model. Furthermore, the obtained result shows that the proposed model is resilient and scalable, showing scalability, and that RMSE performance improves with increasing data availability. Due to resource limitations, experiments were conducted on mid-sized datasets; however, the trends show that the model can be applied to larger-scale settings. In conclusion, the proposed model combines architectural innovation with a task-aware loss design to provide a scalable and principled approach to missing data imputation. To incorporate the proposed model into practical decision-support systems in the healthcare industry, it is recommended that future studies expand this framework to incorporate longitudinal time series and multimodal clinical records.

4. FINAL RESULTS

To achieve the main aim, which consists of using machine learning and clinical data to diagnose COVID-19 and to predict an appropriate healthcare unit for COVID-19 positive patients, the proposed missing data imputation model has been applied to the used dataset, "Diagnosis of COVID-19 and its clinical spectrum". The obtained results are displayed in Table 4.1 below.

Table 4.1 *ROC AUC COVID-19 diagnosis*

Imputation Method	ADASYN	Class Weight	No Oversampling	SMOTE
EM	0.7360	0.6930	0.6040	0.7270
GAIN	0.7610	0.7220	0.6580	0.7980
Guided AE	0.7380	0.7619	0.6410	0.7840
KNN	0.7270	0.7210	0.6120	0.7330
MICE	0.6960	0.7030	0.6440	0.6970
MissForest	0.7290	0.7190	0.6420	0.6770
Standard AE	0.7183	0.7510	0.6520	0.7210

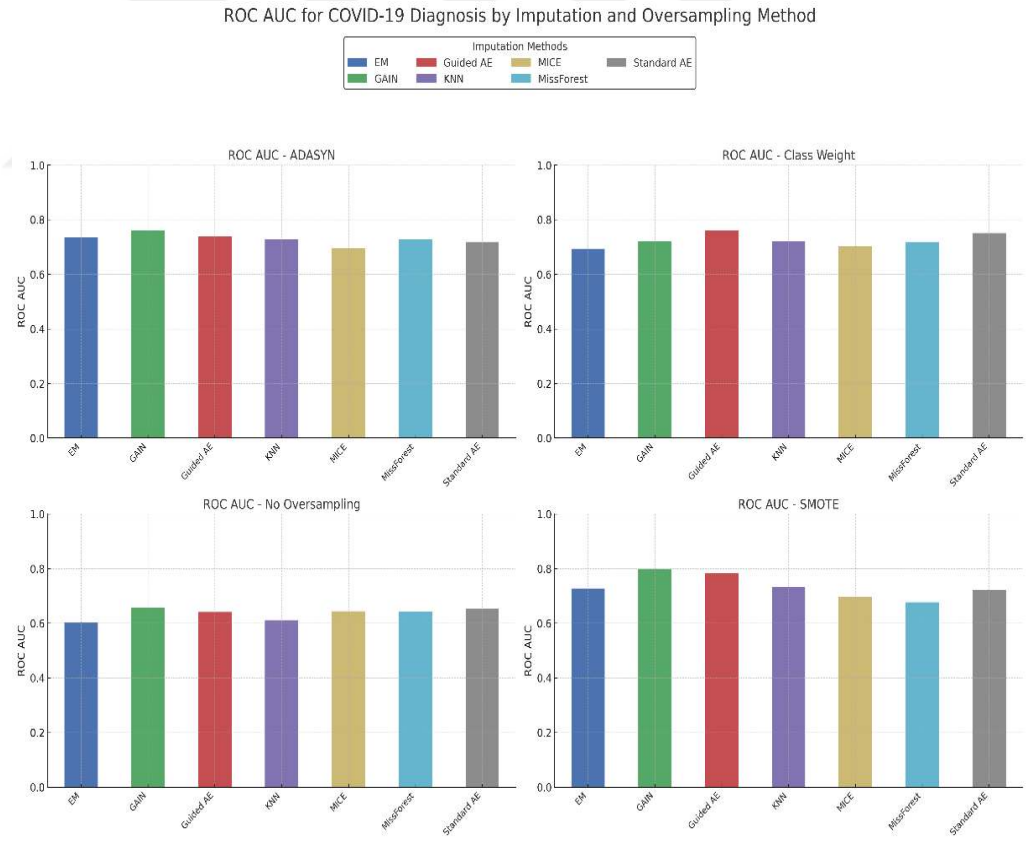


Figure 4.1 *Performance by imputation method and oversampling strategy*

Figure 4.1 shows how seven different imputation methods perform when using four oversampling strategies: no oversampling, SMOTE, ADASYN, and class weight. The result backs up the experiments in the thesis, showing how various mixes of imputation and oversampling influence the model's ROC AUC performance, which is important for making accurate predictions with missing data. The proposed model for handling missing data performed effectively, highlighting that deep learning-based models surpass other types of models.

Table 4.2 *Imputation benchmark results*

Imputer	Inherent	OvR	OvO
Guided AE	0.9930	0.9937	0.9713
Standard AE	0.9971	0.9963	0.9767
GAIN	0.9877	0.9846	0.9552
KNN	0.9291	0.9305	0.9230
MICE	0.9801	0.9833	0.9409
EM	0.9793	0.9840	0.9391
MissForest	0.9725	0.9737	0.9319

Table 4.2 and Figures 4.2 and 4.3 summarize the mean multiclass classification performance for each imputation method using the Balanced Random Forest algorithm. Inherent and OvR share the ROC AUC metric, while OvO uses Accuracy. The findings confirm that modern deep learning imputers (Guided AE, Standard AE, and GAIN) outperform traditional methods (KNN, MICE, EM, MissForest), and also the proposed model performed well.

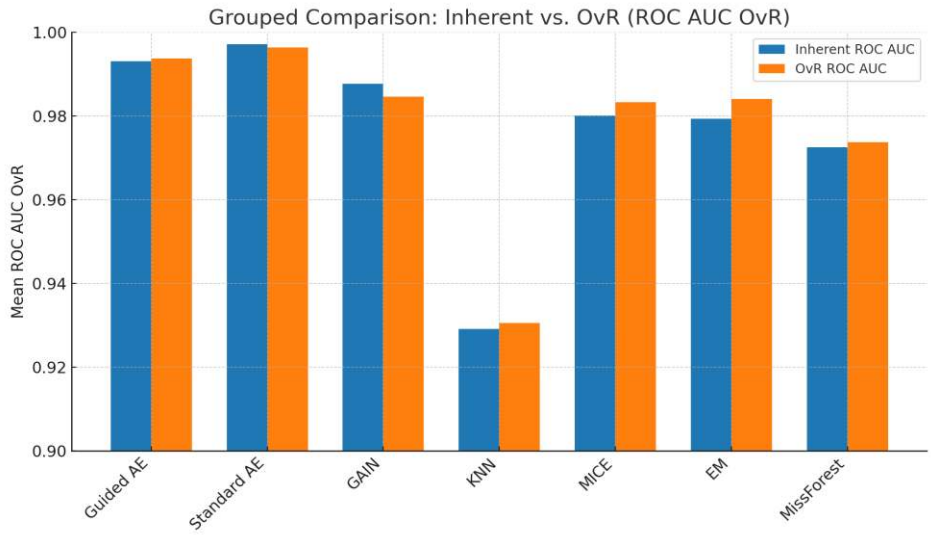


Figure 4.2 *Inherent vs. One-vs-Rest classification (ROC AUC OvR)*

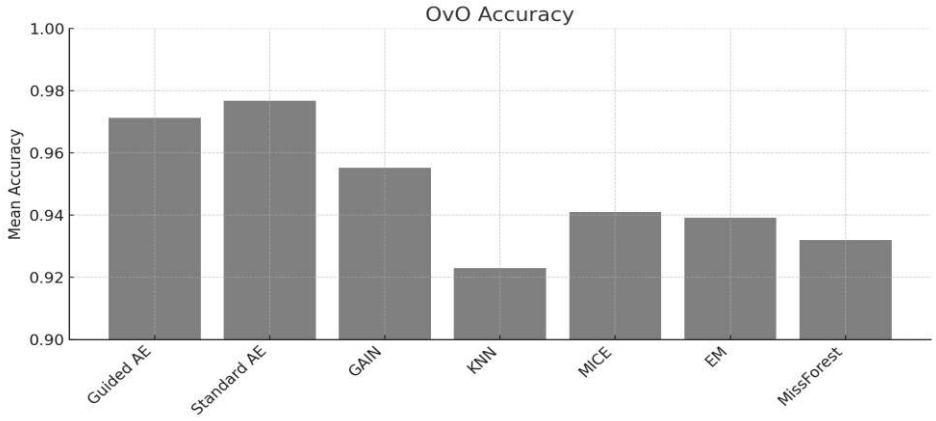


Figure 4.3 *One-vs-One classification mean accuracy*

4.1. Conclusion And Recommendation

This thesis aims to diagnose COVID-19 infection and predict the appropriate healthcare unit for COVID-19-positive patients using machine learning techniques and clinical data. The used data was both highly imbalanced and contained a substantial proportion of missing values. In the first phase, conventional imputation methods such as KNN and MICE were combined with various oversampling techniques, including SMOTE and ADASYN, to tackle imbalance. However, the obtained result was at an average level due to persistent data quality issues. Motivated by this, the second phase critically examined the effect of missing data imputation techniques, revealing important concerns such as potential multicollinearity when using MICE and other standard imputers. The obtained results from phase 1 and phase 2 motivated us to propose a novel Autoencoder-based imputation model, inspired by the Generative Adversarial framework. The proposed model was evaluated under various missingness mechanisms, such as MCAR, MAR, MNAR, and MNAR, with quantile censorship demonstrating consistent improvements over traditional methods and competitive performance alongside advanced models such as GAIN. In the last part, the proposed imputation model was validated by applying it to the original COVID-19 dataset, and the obtained result shows an improvement over the initial baseline results. The findings confirm that advanced, tailored imputation techniques can play a pivotal role in enhancing predictive performance when working with incomplete, imbalanced health datasets. Since most researchers focus more on MCAR and MAR missingness mechanisms, it is recommended that researchers focus more on missing not at random (MNAR) scenarios, which are

common yet challenging in real-world health datasets. Developing and validating imputation models that can handle MNAR mechanisms robustly, especially in combination with advanced methods like autoencoders or adversarial approaches and transformers, will help minimize bias and ensure more reliable and stable predictive modeling.

REFERENCES

Alballa, N., and Al-Turaiki, I. (2021). Machine learning approaches in COVID-19 diagnosis, mortality, and severity risk prediction: A review. *Informatics in Medicine Unlocked*, 24, 100564.

Austin, P. C., and van Buuren, S. (2023). Logistic regression vs. predictive mean matching for imputing binary covariates. *Statistical Methods in Medical Research*, 32(11), 2172–2183.Baissi, F., and Abdelbaki, E. A. (2024). *End-of-study project report*.

Carvalho, M., Pinho, A. J., and Brás, S. (2025). Resampling approaches to handle class imbalance: a review from a data perspective. *Journal of Big Data,* 12(1), 71.

Centers for Disease Control and Prevention. https://www.cdc.gov/(Access date 11.04.2025).

Chen, T., and Guestrin, C. (2016, August). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794).

Chen, W., Yang, K., Yu, Z., Shi, Y., and Chen, C. L. P. (2024). A survey on imbalanced learning: Latest research, applications, and future directions. *Artificial Intelligence Review*, *57*(6). https://doi.org/10.1007/s10462-024-10759-6(Access date 13.11.2024).

Chen, S., and Luc, N. M. (2022). *RRMSE Voting Regressor: A weighting function-based improvement to ensemble regression* (No. arXiv:2207.04837). arXiv. https://doi.org/10.48550/arXiv.2207.04837 (Access date 26.12.2024)

Chowdhury, M. E. H., Rahman, T., Khandakar, A., Al-Madeed, S., Zughaier, S. M., Doi, S. A. R., Hassen, H., and Islam, M. T. (2024). An Early Warning Tool for Predicting Mortality Risk of COVID-19 Patients Using Machine Learning. *Cognitive Computation*, *16*(4), 1778–1793.

Da Poian, V., Theiling, B., Clough, L., McKinney, B., Major, J., Chen, J., and Hörst, S. (2023). Exploratory data analysis (EDA) machine learning approaches for ocean world analog mass spectrometry. *Frontiers in Astronomy and Space Sciences*, *10*.

https://doi.org/10.3389/fspas.2023.1134141(Access date 03.02.2022)

Deecke, L., Ruff, L., Vandermeulen, R. A., and Bilen, H. (2021). Transfer-based semantic anomaly detection. *International Conference on Machine Learning*, 2546–2558.

Farhangfar, A., Greiner, R., and Zinkevich, M. (2008). A fast way to produce near-optimal fixed-depth decision trees. *Proceedings of the 10th International Symposium on Artificial Intelligence and Mathematics (ISAIM-2008).*

Fatima, G., Al-Amran, F. G., and Yousif, M. G. (2023). *Automated Detection of Persistent Inflammatory Biomarkers in Post-COVID-19 Patients Using Machine Learning Techniques* (No. arXiv:2309.15838).

Feng, W., Zong, W., Wang, F., and Ju, S. (2020). Severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2): A review. *Molecular Cancer*, *19*(1), 100.

Goodman-Meza, D., Rudas, A., Chiang, J. N., Adamson, P. C., Ebinger, J., Sun, N., and Manuel, V. (2020). A machine learning algorithm to increase COVID-19 inpatient diagnostic capacity. *PLOS ONE*, 15(9), e0239474.

Jolliffe, I. T., and Cadima, J. (2016). Principal component analysis: A review and recent developments. Philosophical Transactions of the Royal Society A: *Mathematical, Physical and Engineering Sciences,* 374(2065), 20150202.

Kim, E. H., Oh, S. K., Pedrycz, W., and Fu, Z. (2022). Design of reinforced fuzzy model driven to feature selection through univariable-based correlation and multivariable-based determination coefficient analysis. *IEEE Transactions on Fuzzy Systems*, 30(10), 4224–4238.

Lalmuanawma, S., Hussain, J., and Chhakchhuak, L. (2020). Applications of machine learning and artificial intelligence for Covid-19 (SARS-CoV-2) pandemic: A review. *Chaos, Solitons and Fractals*, 139, 110059

Gao, Y., Cai, G.-Y., Fang, W., Li, H.-Y., Wang, S.-Y., Chen, L., Yu, Y., Liu, D., Xu, S., Cui, P.-F., Zeng, S.-Q., Feng, X.-X., Yu, R.-D., Wang, Y., Yuan, Y., Jiao, X.-F., Chi, J.-H., Liu, J.-H., Li, R.-Y., … Gao, Q.-L. (2020). A machine learning based early warning system enables accurate mortality risk prediction for COVID-19. *Nature Communications*, *11*(1), 5033.

Hariri, S., Kind, M. C., and Brunner, R. J. (2021). Extended Isolation Forest. *IEEE Transactions on Knowledge and Data Engineering*, *33*(4), 1479–1489.

Hastie, T., Tibshirani, R., and Wainwright, M. (2015). Statistical learning with sparsity. *Monographs on Statistics and Applied Probability*, *143*(143), 8.

He, H., Bai, Y., Garcia, E. A., and Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, 1322–1328.

Ilin, A., and Raiko, T. (2010). Practical approaches to principal component analysis in the presence of missing values. *The Journal of Machine Learning Research*, *11*, 1957–2000.

Jafarigol, E., and Trafalis, T. (2023). *A Review of Machine Learning Techniques in Imbalanced Data and Future Trends* (No. arXiv:2310.07917).

Jakobsen, J. C., Gluud, C., Wetterslev, J., and Winkel, P. (2017). When and how should multiple imputation be used for handling missing data in randomised clinical trials – a practical guide with flowcharts. *BMC Medical Research Methodology*, *17*(1).

Jerez, J. M., Molina, I., García-Laencina, P. J., Alba, E., Ribelles, N., Martín, M., and Franco, L. (2010). Missing data imputation using statistical and machine learning methods in a real breast cancer problem. *Artificial Intelligence in Medicine*, *50*(2), 105–115.

Jolliffe, I. T., and Cadima, J. (2016). Principal component analysis: A review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *374*(2065), 20150202.

Khan, F. A., Herasymuk, D., Protsiv, N., and Stoyanovich, J. (2025). *Still More Shades of Null: An Evaluation Suite for Responsible Missing Value Imputation* (No. arXiv:2409.07510). arXiv.

Kukar, M., Gunčar, G., Vovko, T., Podnar, S., Černelč, P., Brvar, M., Zalaznik, M., Notar, M., Moškon, S., and Notar, M. (2021). COVID-19 diagnosis by routine blood tests using machine learning. *Scientific Reports*, *11*(1), 10738.

Kwak, S. K., and Kim, J. H. (2017). Statistical data preparation: Management of missing values and outliers. *Korean Journal of Anesthesiology*, *70*(4), 407.

Lalmuanawma, S., Hussain, J., and Chhakchhuak, L. (2020). Applications of machine learning and artificial intelligence for Covid-19 (SARS-CoV-2) pandemic: A review. *Chaos, Solitons and Fractals*, *139*, 110059.

Leiby, B. D., and Ahner, D. K. (2023a). Multicollinearity applied stepwise stochastic imputation: A large dataset imputation through correlation-based regression. *Journal of Big Data*, *10*(1).

Levy, T. J., Richardson, S., Coppa, K., Barnaby, D. P., McGinn, T., Becker, L. B., Davidson, K. W., Cohen, S. L., Hirsch, J. S., and Zanos, T. P. (2020). Development and validation of a survival calculator for hospitalized patients with COVID-19. *MedRxiv*, 2020–04.

Li, W. T., Ma, J., Shende, N., Castaneda, G., Chakladar, J., Tsai, J. C., Apostol, L., Honda, C. O., Xu, J., Wong, L. M., Zhang, T., Lee, A., Gnanasekar, A., Honda, T. K., Kuo, S. Z., Yu, M. A., Chang, E. Y., Rajasekaran, M. " R., and Ongkeko, W. M. (2020). Using machine learning of clinical data to diagnose COVID-19: A systematic review and meta-analysis. *BMC Medical Informatics and Decision Making*, *20.*

Liew, A. W.-C., Law, N.-F., and Yan, H. (2011). Missing value imputation for gene expression data: Computational techniques to recover missing data from available information. *Briefings in Bioinformatics*, *12*(5), 498–513.

Lin, Y., Chen, H., Xia, W., Lin, F., Wang, Z., and Liu, Y. (2024). *A Comprehensive Survey on Deep Learning Techniques in Educational Data Mining* (No. arXiv:2309.04761). arXiv.
Liu, F. T., Ting, K. M., and Zhou, Z.-H. (2008). Isolation forest. *2008 Eighth IEEE International Conference on Data Mining*, 413–422.

Mattei, P.-A., and Frellsen, J. (2019). MIWAE: Deep generative modelling and imputation of incomplete data sets. *International Conference on Machine Learning*, 4413–4423.

Mazumder, R., Hastie, T., and Tibshirani, R. (2010). Spectral regularization algorithms for learning large incomplete matrices. *The Journal of Machine Learning Research*, *11*, 2287–2322.

Nazabal, A., Olmos, P. M., Ghahramani, Z., and Valera, I. (2020). Handling incomplete heterogeneous data using VAEs. *Pattern Recognition*, *107*, 107501.

Oprea, A., and Vassilev, A. (2023). *Adversarial Machine Learning*. https://insidecybersecurity.com/sites/insidecybersecurity.com/files/documents/2023/sep/cs2023_0215.pdf (Access date 15.03.2025).

Prajapati, Y. N., Sharma, M., Azam, F., C, S., and Biradar, A. (2024). Enhancing COVID-19 diagnosis and severity evaluation through machine learning algorithms applied to ct images. *Multidisciplinary Science Journal*, *6*(11), 2024207–2024207.

Raymaekers, J., and Rousseeuw, P. J. (2024). The Cellwise Minimum Covariance Determinant Estimator. *Journal of the American Statistical Association*, *119*(548), 2610–2621.

Ren, Z., Lin, T., Feng, K., Zhu, Y., Liu, Z., and Yan, K. (2023). A systematic review on imbalanced learning methods in intelligent fault diagnosis. *IEEE Transactions on Instrumentation and Measurement*, *72*, 1–35.

Ruff, L. (2021). Deep one-class learning: A deep learning approach to anomaly detection [Doctoral dissertation, Technische Universitaet Berlin].

Sakho, A., Malherbe, E., and Scornet, E. (2025). *Do we need rebalancing strategies? A theoretical and empirical study around SMOTE and its variants* (No. arXiv:2402.03819). arXiv. https://doi.org/10.48550/arXiv.2402.03819 (Access date 27.10.2022).

Sakho, A., Scornet, E., and Malherbe, E. (2024). *Theoretical and experimental study of SMOTE: Limitations and comparisons of rebalancing strategies*. https://hal.science/hal-04438941/ (Access date 23.01.2023).

Schelter, S., Lange, D., Schmidt, P., Celikel, M., Biessmann, F., and Grafberger, A. (2018). Automating large-scale data quality verification. *Proceedings of the VLDB Endowment*, *11*(12), 1781–1794.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., and Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural Computation*, *13*(7), 1443–1471.

Shrestha, N. (2020a). Detecting multicollinearity in regression analysis. *American Journal of Applied Mathematics and Statistics*, *8*(2), 39–42.

Soloff, J. A., Barber, R. F., and Willett, R. (2024). Bagging provides assumption-free stability. *Journal of Machine Learning Research*, *25*(131), 1–35.

Stekhoven, D. J., and Bühlmann, P. (2012). MissForest—Non-parametric missing value imputation for mixed-type data. *Bioinformatics*, *28*(1), 112–118.

Tran, P. V. (2018). Learning to make predictions on graphs with autoencoders. *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, 237–245.

Van Buuren, S., and Groothuis-Oudshoorn, K. (2011). mice: Multivariate imputation by chained equations in R. *Journal of Statistical Software*, *45*, 1–67.

Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. *Proceedings of the 25th International Conference on Machine Learning - ICML '08*, 1096–1103.

Voillet, V., Besse, P., Liaubet, L., San Cristobal, M., and González, I. (2016). Handling missing rows in multi-omics data integration: Multiple imputation in multiple factor analysis framework. *BMC Bioinformatics*, *17*(1).

Wang, S., Kang, B., Ma, J., Zeng, X., Xiao, M., Guo, J., Cai, M., Yang, J., Li, Y., Meng, X., and Xu, B. (2021). A deep learning algorithm using CT images to screen for Corona virus disease (COVID-19). *European Radiology*, *31*(8), 6096–6104.

Woods, A. D., Gerasimova, D., Van Dusen, B., Nissen, J., Bainter, S., Uzdavines, A., Davis-Kean, P. E., Halvorson, M., King, K. M., Logan, J. A. R., Xu, M., Vasilev, M. R., Clay, J. M., Moreau, D., Joyal-Desmarais, K., Cruz, R. A., Brown, D. M. Y., Schmidt, K., and Elsherif, M. M. (2024). Best practices for addressing missing data through multiple imputation. *Infant and Child Development*, *33*(1).

Wu, H., Ruan, W., Wang, J., Zheng, D., Liu, B., Geng, Y., Chai, X., Chen, J., Li, K., Li, S., and Helal, S. (2023). Interpretable Machine Learning for COVID-19: An Empirical Study on Severity Prediction Task. *IEEE Transactions on Artificial Intelligence*, *4*(4), 764–777.

Yan, L., Zhang, H.-T., Goncalves, J., Xiao, Y., Wang, M., Guo, Y., Sun, C., Tang, X., Jing, L., Zhang, M., Huang, X., Xiao, Y., Cao, H., Chen, Y., Ren, T., Wang, F., Xiao, Y., Huang, S., Tan, X., … Yuan, Y. (2020). An interpretable mortality prediction model for COVID-19 patients. *Nature Machine Intelligence*, *2*(5), 283–288.

Yang, Y., Lv, H., and Chen, N. (2023). A Survey on ensemble learning under the era of deep learning. *Artificial Intelligence Review*, *56*(6), 5545–5589.

Yoo, W., Mayberry, R., Bae, S., Singh, K., He, Q. P., and Lillard Jr, J. W. (2014). A study of effects of multicollinearity in the multivariable analysis. *International Journal of Applied Science and Technology*, *4*(5), 9.

Yoon, J., Jordon, J., and Schaar, M. (2018). Gain: Missing data imputation using generative adversarial nets. *International Conference on Machine Learning*, 5689–5698.

U.S. Food and Drug Administration. https://www.fda.gov/ (Access date 19.04.2025).

World Health Organization. https://www.who.int/(Access date 19.06.2025).

Yang, Y., Lv, H., and Chen, N. (2023). A survey on ensemble learning under the era of deep learning. Artificial Intelligence Review, 56(6), 5545–5589.

Yoo, W., Mayberry, R., Bae, S., Singh, K., He, Q. P., and Lillard Jr, J. W. (2014). A study of effects of multicollinearity in the multivariable analysis. International Journal of Applied Science and Technology, 4(5), 9.

Zhang, J., Cao, Y., Tan, G., Dong, X., Wang, B., Lin, J., Yan, Y., Liu, G., Akdis, M., Akdis, C. A., and Gao, Y. (2021). Clinical, radiological, and laboratory characteristics and risk factors for severity and mortality of 289 hospitalized COVID-19 patients. *Allergy*, *76*(2), 533–550.

Zhang, Y., and Long, Q. (2021). Assessing fairness in the presence of missing data. Advances in Neural Information Processing Systems, 34, 16007–16019.

Zhang, Z., Hou, Y., Jia, Y., and Zhang, R. (2024). *An Outlier Detection Algorithm Based on Local Density Feedback Outlier Factor*.

Zheng, R., Liang, Y., Huang, S., Gao, J., III, H. D., Kolobov, A., Huang, F., and Yang, J. (2025). *TraceVLA: Visual Trace Prompting Enhances Spatial-Temporal Awareness for Generalist Robotic Policies* (No. arXiv:2412.10345). arXiv.

Zhou, Y., Aryal, S., and Bouadjenek, M. R. (2024). *Review for Handling Missing Data with special missing mechanism* (No. arXiv:2404.04905). https://doi.org/10.48550/arXiv.2404.04905 (Access date 17.04.2024).

Zoabi, Y., Kehat, O., Lahav, D., Weiss-Meilik, A., Adler, A., and Shomron, N. (2021). Predicting bloodstream infection outcome using machine learning. *Scientific Reports*, *11*(1), 20101.

CURRICULUM VITAE

ORCID ID: 0000-0003-0275-6453l

Name Surname : THIERRY MUGENZI

Foreign Language :ENGLISH

Education and Professional Background:

- 2008 - 2012, National University of RWANDA /Faculty of Applied Sciences, Department of Computer Science
- 2014 - 2016, Gadjah Mada University /Faculty of Mathematics and Natural Sciences, Department of Computer Science
- 2017 - 2025, Eskisehir Technical University /Graduate School of Sciences, Department of Computer Engineering