

# Chapter 4: Patient Treatment Classification

## 4.1 Introduction

Patient treatment classification is a critical aspect of healthcare, aimed at providing effective treatment to patients while optimizing healthcare resources. With the increasing availability of electronic health records and medical data, the use of machine learning and data mining techniques has become popular in healthcare research to develop patient treatment classification models. The objective of patient treatment classification is to classify patients into specific treatment categories based on their medical history, clinical conditions, and other relevant factors. The resulting treatment categories can help healthcare providers determine the most appropriate treatment for a patient, thereby improving patient outcomes and reducing healthcare costs. However, there are still many challenges to overcome, such as developing accurate and reliable patient classification models that can be implemented in real-world clinical settings. This research aims to contribute to the development of effective patient treatment classification models by exploring various machine learning, Deep learning and data mining techniques and evaluating their performance on real-world healthcare dataset. The outcomes of this research can potentially have a significant impact on healthcare, improving patient outcomes and resource utilization in healthcare settings.

In hospitals, medical treatments and surgeries can be categorized into inpatient and outpatient procedures. For patients, it is important to understand the difference between these two types of care, because they impact the length of a patient's stay in a medical facility and the cost of a procedure. The difference between an inpatient and outpatient care is how long a patient must remain in the facility where they have the procedure done. Inpatient care requires overnight hospitalization. Patients must stay at the medical facility where their procedure was done (which is usually a hospital) for at least one night. During this time, they remain under the supervision of a nurse or doctor. Patients receiving outpatient care do not need to spend a night in a hospital. They are free to leave the hospital once the procedure is over. In some exceptional cases, they need to wait while anaesthesia wears off or to make sure there are not any complications. As long as there are not any serious complications, patients do not have to spend the night being supervised.

## 4.2 Implementation

This analysis of the Patient treatment Classification dataset can provide insights into the classification of patient care using their laboratory report. We have used various classification techniques, such as Logistic Regression, RF classifier, Multilayer Perceptron and Supervised Variational Autoencoder, to classify the patient and used the Robust Scaler for feature scaling. The performance of these classification techniques evaluated using different metrics such as

accuracy, sensitivity or recall, precision, specificity, F1 score, F1 Measure, Balanced accuracy, ROC-AUC score and Cohen kappa score, False Positive rate and False Negative rate.

### 4.2.1 Dataset Description

The dataset was gathered through the open-source dataset platform 'Mendeley Data' [31]. The dataset is Electronic Health Record Predicting collected from a private Hospital in Indonesia. It contains the patient's laboratory test results used to determine next patient treatment whether in care or out care patient. The task embedded to the dataset is classification prediction.

Attribute Name	Data Type	Value Sample	Description
HAEMATOCRIT	Continuous	35.1	Patient laboratory test result of haematocrit
HAEMOGLOBINS	Continuous	11.8	Patient laboratory test result of haemoglobins
ERYTHROCYTE	Continuous	4.65	Patient laboratory test result of erythrocyte
LEUCOCYTE	Continuous	6.3	Patient laboratory test result of leucocyte
THROMBOCYTE	Continuous	310	Patient laboratory test result of thrombocyte
MCH	Continuous	25.4	Patient laboratory test result of MCH
MCHC	Continuous	33.6	Patient laboratory test result of MCHC
MCV	Continuous	75.5	Patient laboratory test result of MCV
AGE	Continuous	12	Patient age
SEX	Nominal (Binary)	F	Patient gender
SOURCE	Nominal	{in,out}	The class target in.= in care patient, out = out care patient

<b>Attribute Characteristics:</b>	Integer, Nominal	Real,	<b>Total Number of Attributes:</b>	11
<b>Associated Tasks:</b>	Classification		<b>Number of Instances:</b>	4412
<b>No. of samples in class out</b>	2628		<b>No. of samples in class in</b>	1784

### 4.2.2 System Configuration

This Chapter analysis was done by using the following machine configuration:

- **Operating System:** Windows 11
- **Processor:** AMD Ryzen 3 4300U with Radeon Graphics 2.70 GHz
- **Installed RAM:** 8.00 GB (7.36 GB usable)

- **SSD:** 512 GB (476 GB usable)
- **System type:** 64-bit operating system, x64-based processor

### 4.2.3 Results Analysis and Discussion

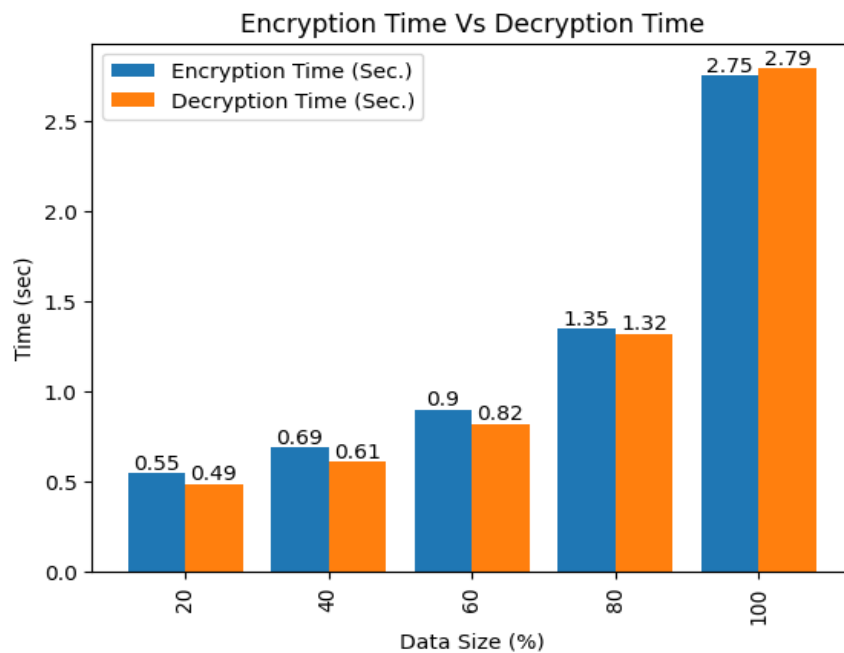
This result analysis provides a summary of the findings and presents them in a clear and concise manner. At the first section we will discuss about encryption and decryption analysis then directly jump to the model diagnosis analysis part. In the model diagnosis part, we will see the dataset overview, feature selection with the correlation matrix, Dataset splitting, cross validation and hyperparameter tuning for model optimization. At last, we choose the best model and discuss about the comparison study.

#### 4.2.3.1 Encryption and Decryption time analysis

In this section we will discuss about the Encryption and Decryption time analysis. Table 4.1 describe the comparison between our SPECK-NOA model and the existing Model.

**Table 4.1:** Encryption and Decryption time analysis

Securing Model	Data size (%)	Decryption Time (sec)	Encryption Time (sec)
SPECK-NOA	20	0.49	0.55
	40	0.61	0.69
	60	0.82	0.90
	80	1.32	1.35
	100	2.79	2.75



**Figure 4.1:** Comparison Analysis of Encryption and Decryption time

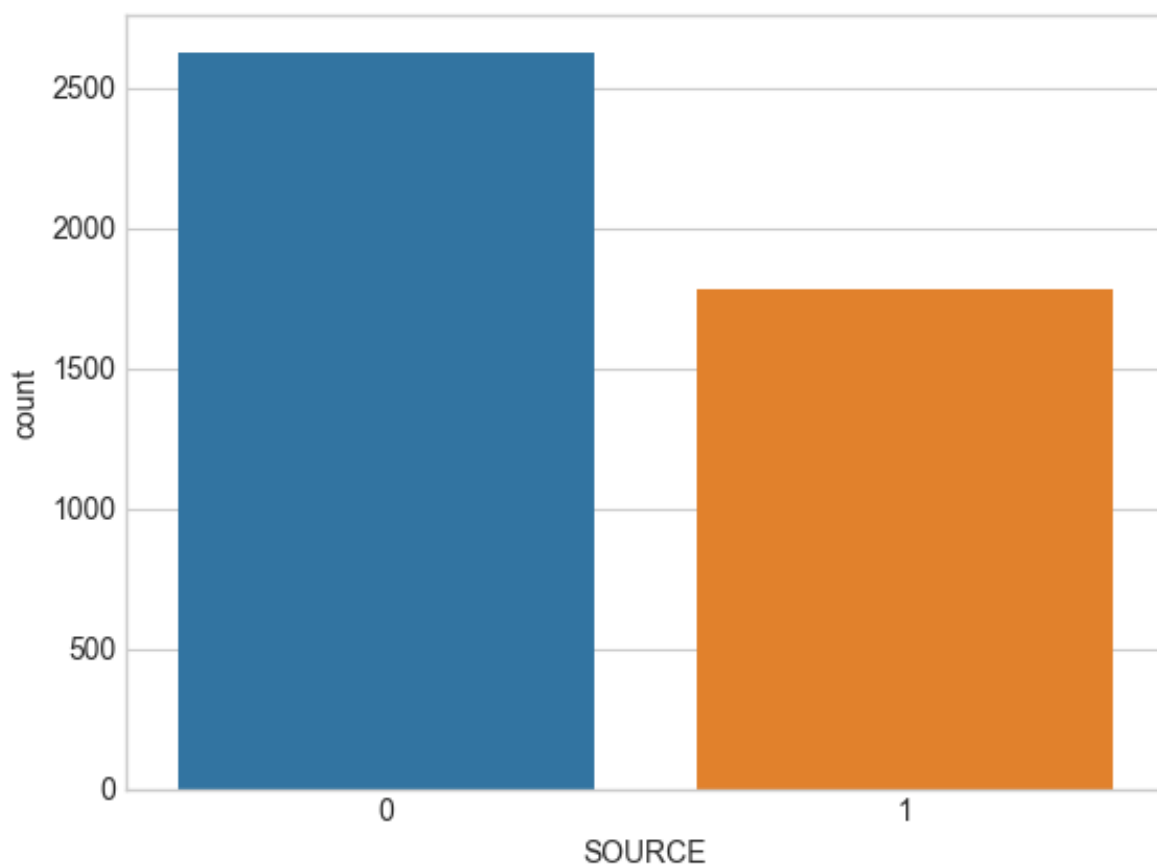
We can see both encryption and decryption time analysis in the Table 4.1 by using SPECK-NOA model. The Table 4.1 is clearly depicted in Figure 4.1 where encryption time and decryption time are showing in different plot basis on percentage of the data size.

After decrypt the dataset, we need to diagnosis the dataset. So now we come the diagnosis part, where we will see the analysis of the diagnosis model. The below all sections are for the diagnosis analysis.

#### 4.2.3.2 Dataset Overview

This section first plots the target distribution then describe the statistics of the dataset with the help of counting of each feature's unique values. Figure 4.2 show the counting plot of the class level in dependent feature and Table 4.2 clearly describe the feature's unique values.

The goal of the dataset overview is to provide readers with a clear understanding of the dataset and how it was prepared for analysis. It is crucial to include this information in research papers as it allows other researchers to replicate the study and verify the findings. A well-described dataset overview section can also help to establish the credibility and reliability of the study.



**Figure 4.2:** Count plot for target feature

**Table 4.2:** Dataset's feature Descriptions

Attributes	No. of unique value
HAEMATOCRIT	326
HAEMOGLOBINS	128
ERYTHROCYTE	433
LEUCOCYTE	276
THROMBOCYTE	554
MCH	189
MCHC	105
MCV	406
AGE	95
SEX	2
SOURCE	2

Now on the next section, we'll see the correlation matrix of the dataset and select the important features based on the correlation if needed.

#### 4.2.3.3 Correlation Matrix and Feature Selection

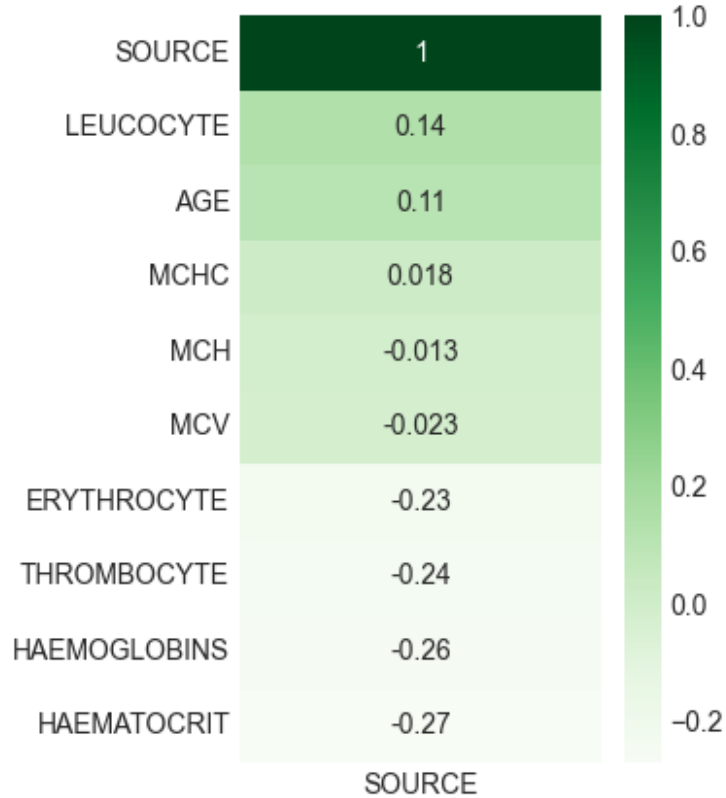
Here we select the independent features based on the relation with the dependent feature. In our study we follow a rule as number of features in our dataset is very low. The rule is if more than 70% of the independent features are correlated at least 5% with the class variable then we discard the features which has less than 5% correlation with the dependent variable.

For the feature selection purpose, we mostly focus on the target variable. So, we will select the features by totally depend on the Figure 4.3 where visible the Pearson's Correlation of all features w.r.t target. We can observe the Figure 4.3 and conclude that there are 6 numerical features which has more than 5% correlation.

Observations:

- LEUCOCYTE and AGE are positively correlated with target
- THROMBOCYTE, ERYTHROCYTE, HAEMOGLOBINS and HAEMATOCRIT are negatively correlated with target
- No notable correlation found between MCHC, MCH, MCV with target.

So, from the Figure 4.3, we can conclude that more than 70% of the independent features are at least 5% correlated with the target variable. So, we select the most relevant features and remove MCHC, MCH and MCV feature.



**Figure 4.3:** Pearson's Correlation of features w.r.t target

As no features are relevant so we need to train the models with all the features. Now, we directly move to the model building part that start with splitting the dataset.

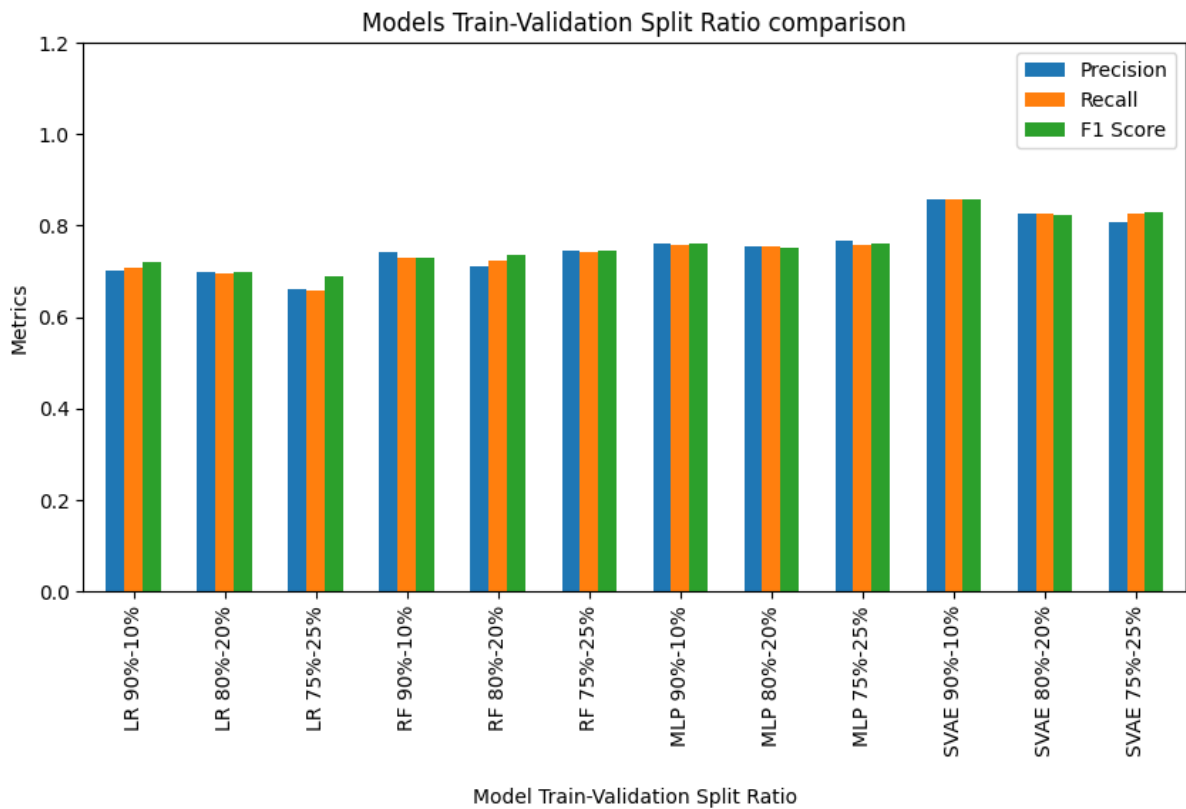
#### 4.2.3.4 Dataset Splitting

For getting the better performance, 1<sup>st</sup> we randomly separate the 20% of the data as test data from the dataset. Now from the rest data we split the data into training data and validation data using 3 different validation split ratio (10%, 20%, 25%). For each splitting run the normal model with the scaled training data and validate with the scaled validation data (Robust Scaler is used to scale the training data then transform the validation data based on that). Performance Validation was done using three different classification evaluation metrics (Precision, Recall and F1 Score) to find the best validation split ratio for cross validation. Table 4.3 describe each model's performance metrics for the validation data.

If we minutely observe the Table 4.3, we can conclude that RF classifier, MLP has highest precision, recall and f1 score on **75%-25%** split ratio than all other split. So, the best split ratio is set for these 3 models as 75%-25%. And for LR and SVAE, 90%-10% split has the highest precision, recall and f1 score than all other split. So, the best split ratio is set for LR and SVAE is 90%-10%.

**Table 4.3:** Splitting Evaluation Metrics on validation data

Model	Train-Validation Split Ratio	Precision	Recall	F1 Score
<i>LR</i>	90%-10%	<b>0.7005</b>	<b>0.7086</b>	<b>0.7195</b>
	80%-20%	0.6984	0.6963	0.6972
	75%-25%	0.6606	0.6593	0.6899
<i>RF</i>	90%-10%	0.7426	0.7285	0.7297
	80%-20%	0.7121	0.7251	0.7346
	75%-25%	<b>0.7464</b>	<b>0.7434</b>	<b>0.7443</b>
<i>MLP</i>	90%-10%	0.7607	0.7568	0.7595
	80%-20%	0.7548	0.7563	0.7527
	75%-25%	<b>0.7686</b>	<b>0.7593</b>	<b>0.7598</b>
<i>SVAE</i>	90%-10%	<b>0.8572</b>	<b>0.8586</b>	<b>0.8575</b>
	80%-20%	0.8268	0.8275	0.8227
	75%-25%	0.8076	0.8275	0.8298



**Figure 4.4:** Comparative analysis of all model basis on dataset splitting

To finding the results of the Table 4.3, we used

- Default parameter for **LR and RF classifier**.
- **For MLP**, number of dense layers 1 with 32 nodes, epochs 10, batch size 128 and the loss function ‘binary crossentropy’
- **For SVAE**, number of encoder and decoder layers 1 with 128 nodes, Number of dense layers 1 with 32 nodes, epochs 10, batch size 128, supervised loss ‘binary crossentropy’

The Figure 4.4 depicted for comparative analysis between each model. From that figure we can say that the SVAE with default parameter works better than others models. Now with the best validation split ratio for each model, we will see the cross-validation performance in the next section.

### 4.2.3.5 Cross Validation

This study used the types of cross-validation that are k-fold cross-validation, and stratified k-fold cross-validation and shuffle split cross-validation. In each Cross-validation technique for each model the training and validation data are also scaled by Robust scaler. Table 4.4 describe the performance metrics basis on the validation data. Here the mean of each evaluation metrics is calculated. Because of multiple combinations of the subsets of 80% train data with the best split ratio, there is generate multiple evaluation metrics for each model, so we take the average of each evaluation metrics. To select the best Cross validation technique for hyperparameter tuning the mean balanced accuracy, mean accuracy and mean AUC score are used for each model.

**Table 4.4:** Cross Validation evaluation metrics

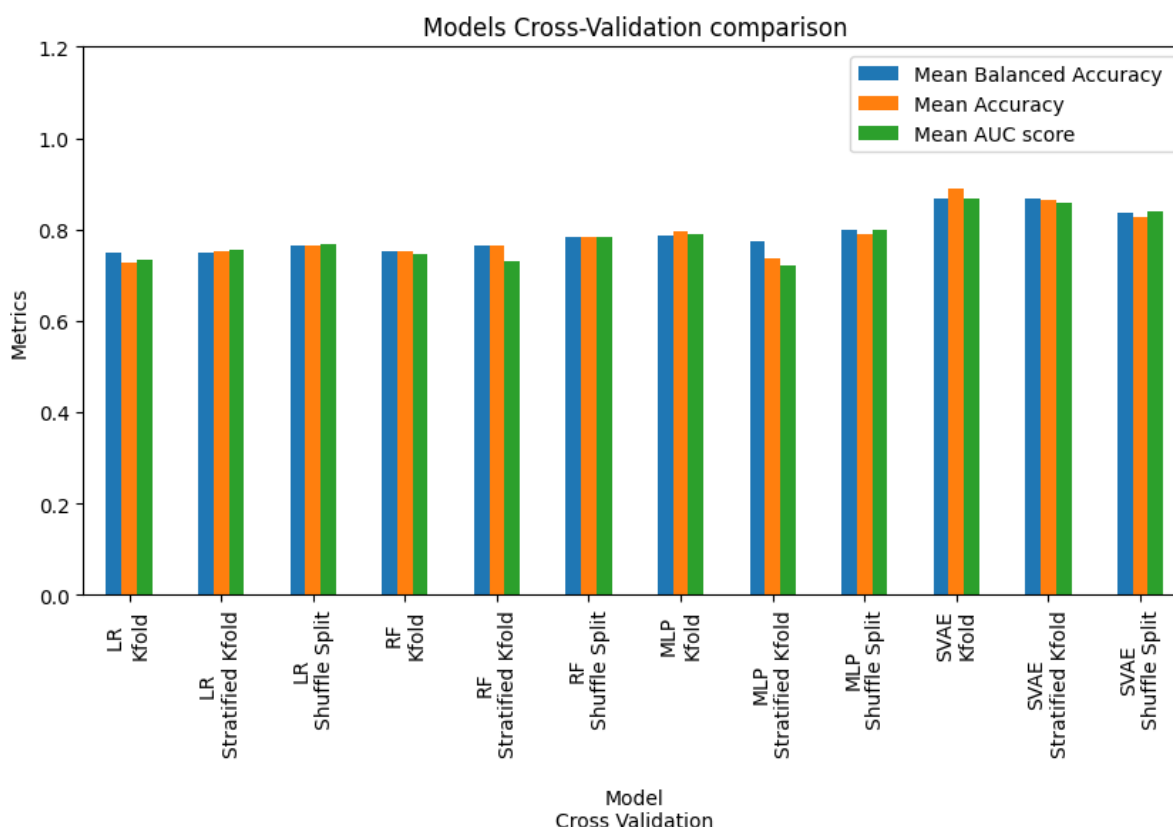
Model	Cross Validation	Mean Balanced Accuracy	Mean Accuracy	Mean AUC score
<i>LR</i>	Kfold	0.7503	0.7286	0.7350
	Stratified Kfold	0.7484	0.7536	0.7570
	Shuffle Split	<b>0.7660</b>	<b>0.7639</b>	<b>0.7689</b>
<i>RF</i>	Kfold	0.7532	0.7525	0.7470
	Stratified Kfold	0.7641	0.7635	0.7304
	Shuffle Split	<b>0.7844</b>	<b>0.7843</b>	<b>0.7841</b>
<i>MLP</i>	Kfold	0.7877	0.7968	0.7893
	Stratified Kfold	0.7746	0.7362	0.7227
	Shuffle Split	<b>0.7986</b>	<b>0.7893</b>	<b>0.7998</b>
<i>SVAE</i>	Kfold	<b>0.8670</b>	<b>0.8886</b>	<b>0.8675</b>
	Stratified Kfold	0.8668	0.8657	0.8572
	Shuffle Split	0.8367	0.8275	0.8398

If we carefully observe the Table 4.4, we can conclude that LR classifier and RF classifier has the highest mean balanced accuracy, mean accuracy and mean AUC score on **shuffle split** than all other CV technique. For MLP, Shuffle split CV technique has highest balanced accuracy



and AUC score than all other CV techniques (maximum 2 metrics are better so chosen). So, the best CV technique is set for these 3 models as **shuffle split**. And for SVAE, **kfold** cross validation has chosen as the best CV technique because **kfold** has highest Balanced Accuracy, Accuracy and AUC score than all other CV techniques.

To finding the results of the Table 4.4, used the same parameter mentioned in the preceding section.



**Figure 4.5:** Comparative analysis of all model basis on cross-validation

The Figure 4.5 can use for comparative analysis between each model. From that figure we can say that the LR classifier with default parameter works better than others models. Now with the best cross-validation technique for each model, we will see the Hyperparameter tuning in the next section.

#### 4.2.3.6 Hyperparameter Tuning

In this part, first we found the best parameter based on the cross validation with the training and validation data using a parameter list for each model. Then Fit the models with the 80% scaled train data and predict the evaluation metrics based on the 20% scaled test data. This test results helps to choose the best model. In Table 4.5, all test results are present. The updated version of all model is following:

##### Updated LR:

**Best Split:** 90%-10%

**Best CV technique:** Shuffle Split

After the performance evaluation--

**Best Model optimization technique:** Optuna

**Reason:** The mention optimization technique has highest F1 Measure, Specificity and Cohen Kappa score than all other optimization techniques.

### **Updated RF:**

**Best Split:** 75%-25%

**Best CV:** Shuffle Split

After the performance evaluation--

**Best Model optimization:** Hyperopt

**Reason:** The mention optimization technique has highest F1 Measure, Specificity and Cohen Kappa score than all other optimization techniques.

### **Updated MLP:**

**Best Split:** 75%-25%

**Best CV:** Shuffle Split

After the performance evaluation--

**Best Model optimization:** Hyperopt

**Reason:** The mention optimization technique has highest F1 Measure, Specificity and Cohen Kappa score than all other optimization techniques.

### **Updated SVAE:**

**Best Split:** 80%-20%

**Best CV:** Kfold

After the performance evaluation--

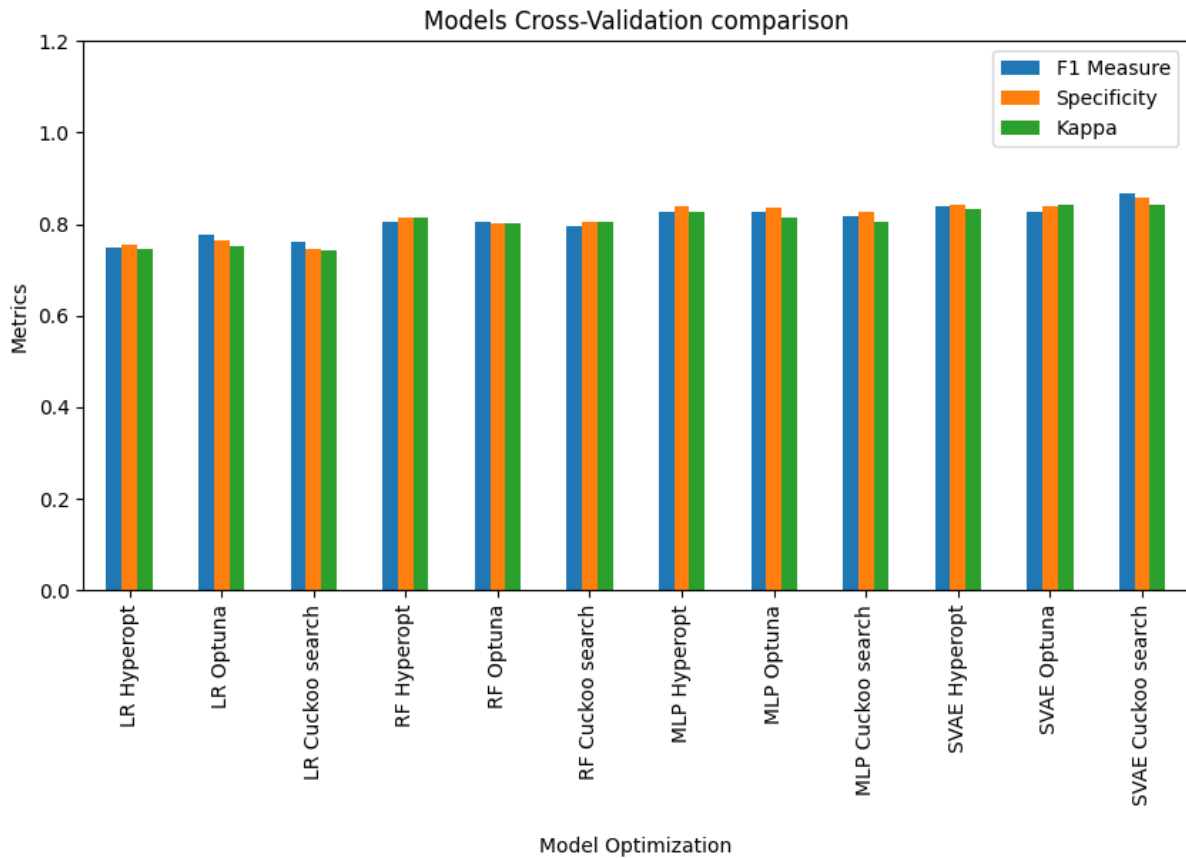
**Best Model optimization:** Cuckoo search

**Reason:** The mention optimization technique has highest F1 Measure, Specificity and Cohen Kappa score than all other optimization techniques.

If we look at the Table 4.5, LR is the best model among all other model. The SVAE model also get better result compare to another model. The LR get best result may be of the non-linearity in the dataset. The Figure 4.6 also use for comparative analysis between each model.

**Table 4.5:** Model Optimization using Hyperparameter Tuning

Model	Model Optimization	F1 Measure	Specificity	Kappa
<i>LR</i>	Hyperopt	0.7484	0.7536	0.7461
	Optuna	<b>0.7767</b>	<b>0.7639</b>	<b>0.7528</b>
	Cuckoo search	0.7627	0.7450	0.7416
<i>RF</i>	Hyperopt	<b>0.8064</b>	<b>0.8134</b>	<b>0.8131</b>
	Optuna	0.8037	0.8028	0.8028
	Cuckoo search	0.7946	0.8062	0.8061
<i>MLP</i>	Hyperopt	<b>0.8270</b>	<b>0.8386</b>	<b>0.8261</b>
	Optuna	0.8258	0.8357	0.8128
	Cuckoo search	0.8167	0.8275	0.8061
<i>SVAE</i>	Hyperopt	0.8384	0.8436	0.8316
	Optuna	0.8260	0.8393	0.8428
	Cuckoo search	<b>0.8666</b>	<b>0.8571</b>	<b>0.8411</b>

**Figure 4.6:** Comparative analysis of all model basis on Hyperparameter tuning

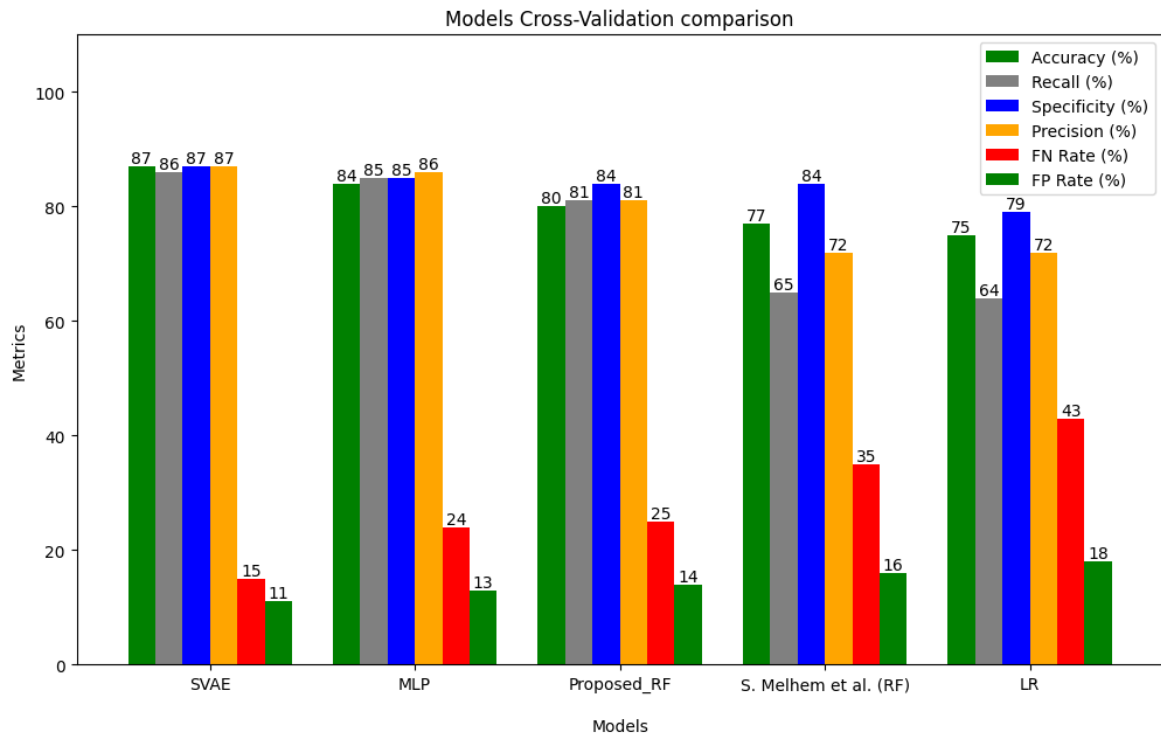
Now we compared our model with all other model using some special evaluation metrics in the next section.

#### 4.2.3.7 Comparative Study

The Table 4.6 describe the comparative analysis that involves analysing and comparing two or more entities or phenomena to identify similarities, differences, and patterns.

**Table 4.6:** Comparative analysis of our model

<b>Models</b>	<b>Accuracy (%)</b>	<b>Recall (%)</b>	<b>Specificity (%)</b>	<b>Precision (%)</b>	<b>FNR (%)</b>	<b>FPR (%)</b>
<b>SVAE</b>	87	86	87	87	15	11
<b>MLP</b>	84	85	85	86	24	13
<b>Proposed_RF</b>	80	81	84	81	25	14
<b>S. Melhem et al. (RF) [13]</b>	77	65	84	72	35	16
<b>LR</b>	75	64	79	72	43	18



**Figure 4.7:** Comparative Analysis with existing model

In the Figure 4.7 we clearly observe that SVAE give the highest in 4 evaluation metrics and lowest in 2 evaluation metrics among six. So, we can conclude that SVAE using Cuckoo search optimization technique give better performance than other model based on six classification evaluation metrics (Precision, Recall, Accuracy, Specificity, False Negative Rate (FNR) and False Positive Rate (FPR)). These six-evaluation metrics are used to compare because of the existing model's [13] performance is measured by the same.