SSRG International Journal of Electrical and Electronics Engineering
ISSN: 2348-8379/ https://doi.org/10.14445/23488379/IJEEE-V11I4P115

Volume 11 Issue 4, 140-150, April 2024

Original Article

Machine Learning Based Assistance to Healthcare Professionals in Disease Prediction and Classification Using Basic Patient Profile

Prachi Palsodkar[1], Deepti Khurge[2], Ashish Bhagat[3], Prasanna Palsodkar[4], P.K. Rajani[5], Varsha Bendre[6]

[1]*Department of Electronics Engineering, Yeshwantrao Chavan College of Engineering, Nagpur, Maharastra, India.*
[2,5,6]*Department of Electronics and Telecommunication, Pimpri Chinchwad College of Engineering, Pune, India.*
[4]*Department of Electronics Engineering, Yeshwantrao Chavan College of Engineering, Nagpur, Maharastra, India.*
[3]*Research Consultant, Jawaharlal Nehru Medical College, Datta Meghe Institute of Higher Education and Research, Maharastra, India.*

[5]*Corresponding Author : rajani.pk@pccoepune.org*

Received: 10 February 2024 Revised: 10 March 2024 Accepted: 09 April 2024 Published: 30 April 2024

Abstract *- Patient profile is critical for medical practitioners, clinicians, and researchers performing clinical evaluations, research studies, and epidemiological investigations. Analyzing patient data provides insights into symptom prevalence and trends across diverse medical illnesses, which aids in trend detection, diagnosis, treatment, and public health improvement. This work investigates the Machine Learning (ML) life cycle, which includes data balancing, feature analysis, K-fold cross-validation, and hyperparameter tuning, to develop classification models for predicting disease presence or absence. Accuracy, Recall, F1 score, Area under the Curve (AUC), and the Jaccard Index are measures used to evaluate ML classifiers like Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), and Boosting Classifier models. Gradient Boost emerges as the best-performing model, blending performance with computational economy, making it ideal for this classification challenge. This comprehensive approach enhances the understanding of illness causes, facilitates personalized treatment, and informs preventive measures.*

Keywords *- Machine Learning, Patient profile, Predicting disease, Support Vector Machine, RF, DT.*

1. Introduction

An increase in medical costs due to the risk of different diseases and an increase in aging creates the need for personalized treatment and early detection. Traditional techniques for illness diagnosis frequently use standardized procedures and broad therapies that may not be suited to an individual's unique genetic makeup, lifestyle circumstances, or specific health issues. This one-size-fits-all strategy can result in inefficiencies, unneeded treatments, and poor outcomes.

Profiling patients based on medical conditions, symptoms like fever, cough, fatigue, and difficulty breathing, as well as demographic and physiological factors like age, gender, blood pressure, and cholesterol levels, has significant implications for medical research, healthcare policy development, and personalized patient care. This technique allows for a deeper awareness of how multiple factors interact, impacting health outcomes, making it easier to identify risk factors, personalize interventions, and optimize therapies for specific patients. The Patient Profiling dataset is useful for medical practitioners, clinicians, and researchers doing clinical analyses, research studies, and epidemiological inquiries into a variety of disorders.

Using this data, they can get insights into the prevalence and patterns of symptoms displayed by individuals with various medical diseases. This understanding is critical for recognizing patterns, guiding diagnostic and treatment plans, and ultimately improving patient care and public health outcomes.

Researchers who focus on specific diseases or disorders indicated in the dataset might use it to investigate correlations between symptoms, age, gender, and other characteristics. Such an investigation can provide new insights, assist the creation of treatment procedures, and drive the design of preventative measures. By investigating these links, researchers can get a better knowledge of disease causes, develop diagnostic techniques, and customize therapies. Machine Learning (ML) technologies are increasingly being investigated for medical data analysis due to their capacity to

evaluate large datasets and detect complex patterns those traditional statistical methods. Various ML approaches are employed in the literature, such as logistic regression (LR), Naïve Bays (NB), SVM, DT, KNN, Artificial Neural Network (ANN), boosting, bagging, RF, Density-Based Clustering (DBSCAN), Stacking, Fuzzy clustering, Ensemble, voting classifier with considering different research questions like a medical problem, method, features, number of samples, different preprocessing and data augmentation techniques, performance metric and clinical implications.

This paper covers the major aspects of the ML life cycle to classify data to predict the possibility of disease or not. Data imbalance [1], Feature analysis [2], K fold Cross validation [3] and hyperparameter tuning [4] are performed for implementation of a classification model. With the best hyperparameters model is retrained and evaluated for SVM [5], DT, and RF [6] and boosting classifier using Accuracy, Recall, F1 score and Jaccard Index. K folds cross-validation and optimum hyperparameters selection create a more generalized model and help in increasing model accuracy. LR, SVM, DT, RF and boosting algorithms are used in this paper with optimum feature selection to increase model generalization.

2. Data Description

The patient Profiling dataset is used in this work to create and refine prediction models for illness diagnosis or monitoring based on symptoms and patient features. This dataset is a useful resource for improving the accuracy and efficacy, resulting in better healthcare decision-making, early illness identification, and more personalized patient treatment.

Independent Features contributed to the data are Disease, Fever, Cough, Fatigue, Difficulty Breathing, Age (in years), Gender, Blood Pressure and Cholesterol Level, which all are categorical variables except age, and the dependent feature is the Outcome Variable (Positive/Negative). The data set utilized for the study included 7% asthma patient data, 5% stroke data, and 89% illness coverage.

The ratio of the test cases that are positive or negative for fever is 50%. There are 48% cough positive instances, 69% fatigue patients, and 25% difficulty breathing cases. Age ranges from 19 to 90 years. With a maximum data count for the 30-40 age range. Gender-wise data distribution is equal between males and females. 48% of high blood pressure and cholesterol patients are considered. This data is used to determine whether or not the patient has illness symptoms based on accessible features. It demonstrates the deep link between patients and illnesses in over 100 cases, as shown in Figure 1.

It was observed in the ‘Disease’ column that there are multiple distinct illnesses, with many having only one to five samples. However, a limited sample size is insufficient to develop an effective illness prediction model. Predicting illnesses based on such minimal information may result in mistakes and misdiagnoses. To guarantee that the model is strong, consider disorders with ten or more samples.

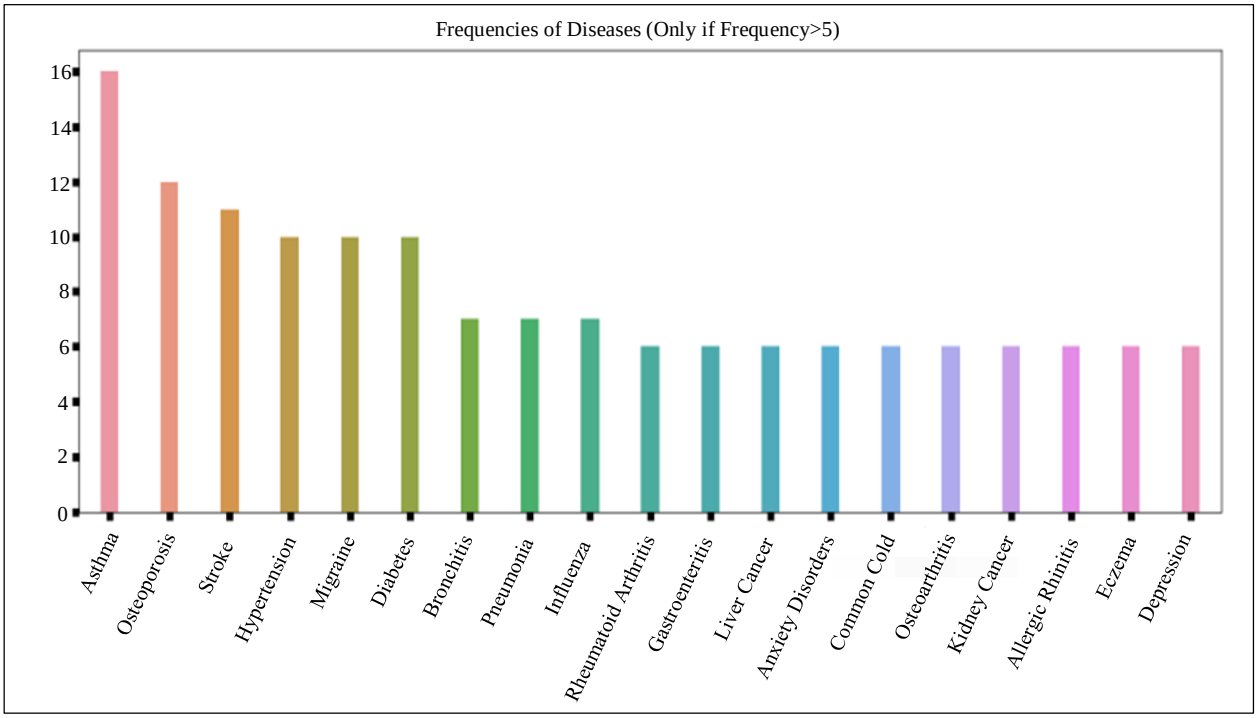


Fig. 1 Disease frequency in the dataset

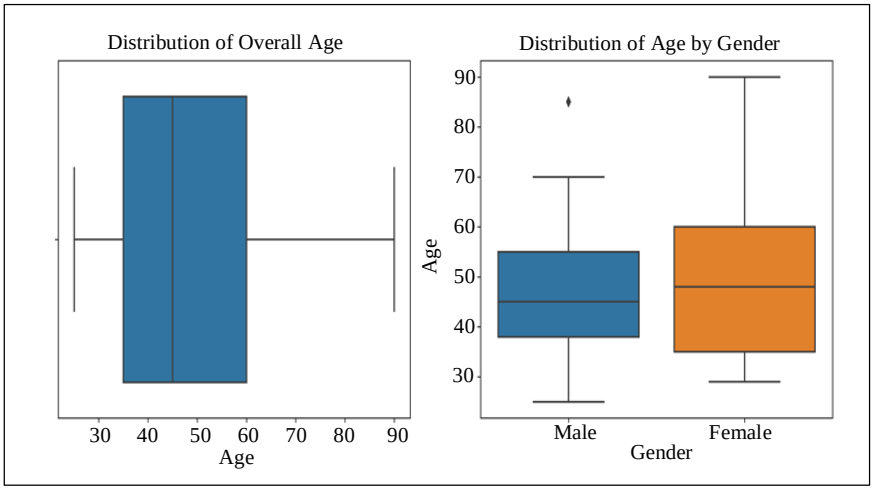


Fig. 2 Distribution of age

Focusing on illnesses with greater sample numbers minimized the number of predicted classes to six. This made the model more controllable and accurate. Priorities quality above quantity, ensuring that projections are based on adequate data to give relevant insights.

Figure 2 infers the dataset's age's span from 18 to 85 years, with the majority lying between 34 and 56 years and a median age of roughly 45 years. The highest age limits fluctuate somewhat across genders, although the median ages are comparable. Males have slightly greater lower age limits.

This implies that testing is widespread between the ages of 35 and 55, which may indicate higher health awareness and check-ups throughout this time period. After running univariate analysis on the 'Age' variable, it is clear that as the age exceeds 80, the risk of the disease becoming a stroke rises. This data supports the widely held belief that the chances of stroke increases with increasing age.

Furthermore, migraine and hypertension are not widespread between the ages of 20 and 30, indicating that they are more common in older age groups. Furthermore, hypertension and osteoporosis grow more common as people become older, implying a link between these conditions and age. These findings emphasize the importance of age as a predicting factor for some illnesses.

However, one must note the small sample size in the dataset, particularly for ages beyond 80. This constraint may provide issues when forecasting new values within this age range. Next, let's analyze how the other variables interact with different diseases. This will help us understand their potential as predictors and identify any patterns or correlations.

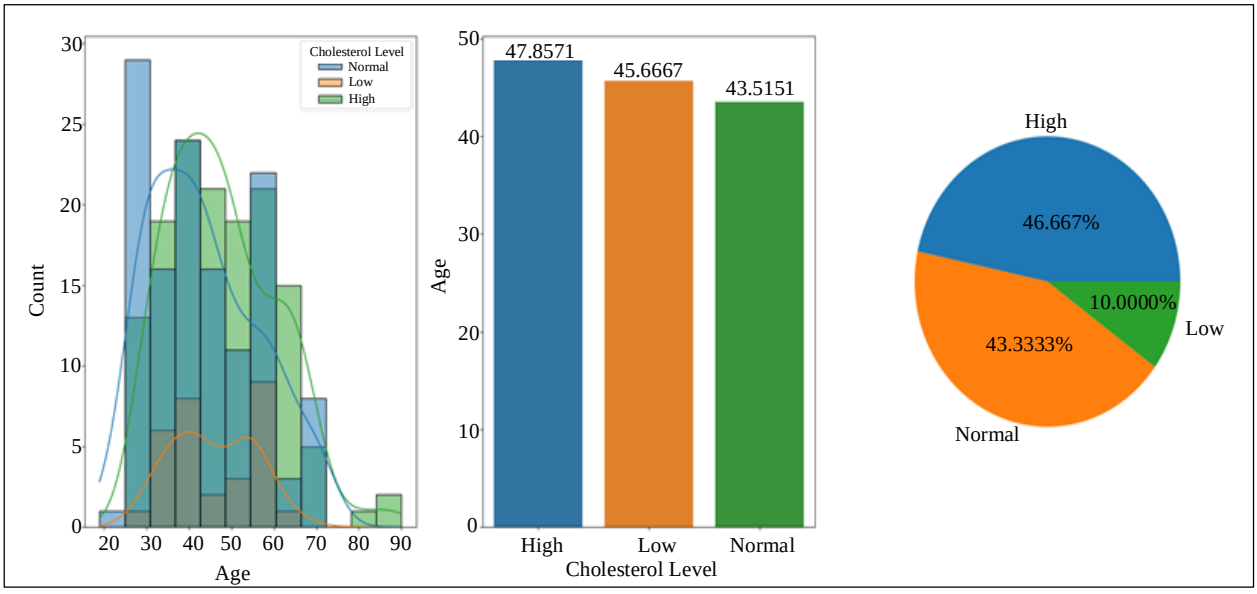
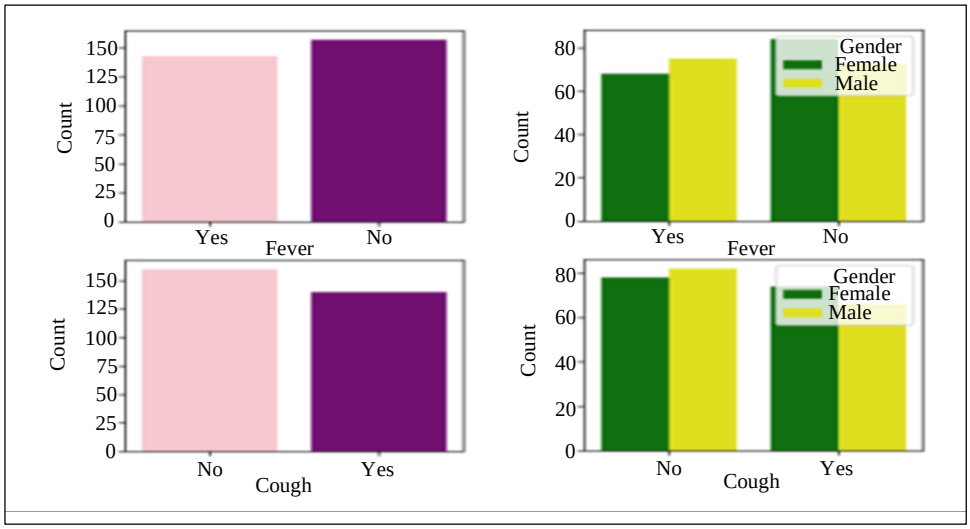
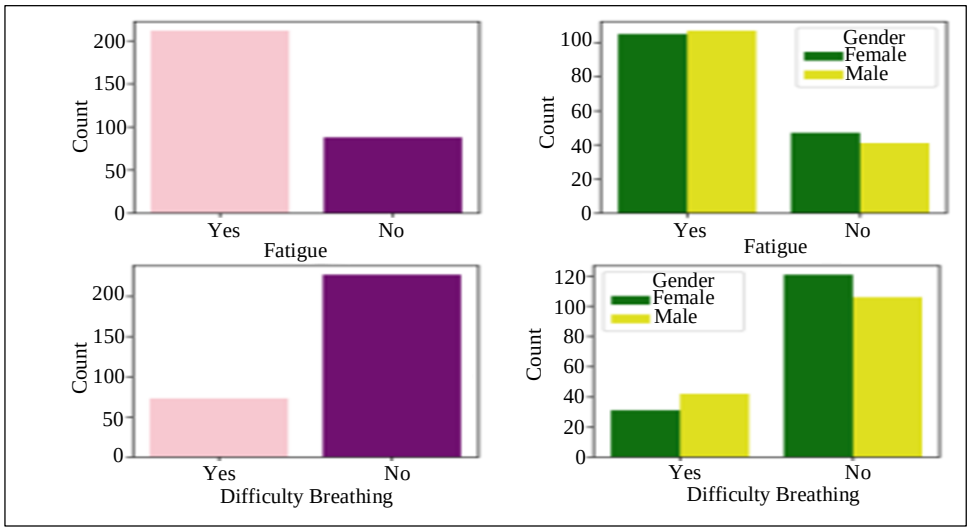
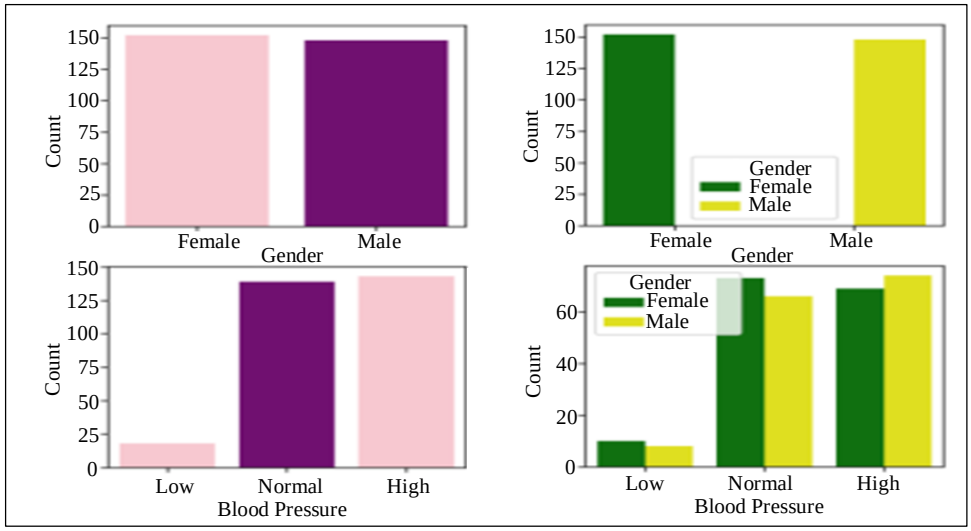


Fig. 3 Blood pressure and cholesterol levels by age


Count
Yes
No
Fever
Gender
Female
Male
Cough


Count
Yes
No
Fatigue
Gender
Female
Male
Difficulty Breathing


Count
Female
Male
Gender
Gender
Female
Male
Low
Normal
High
Blood Pressure

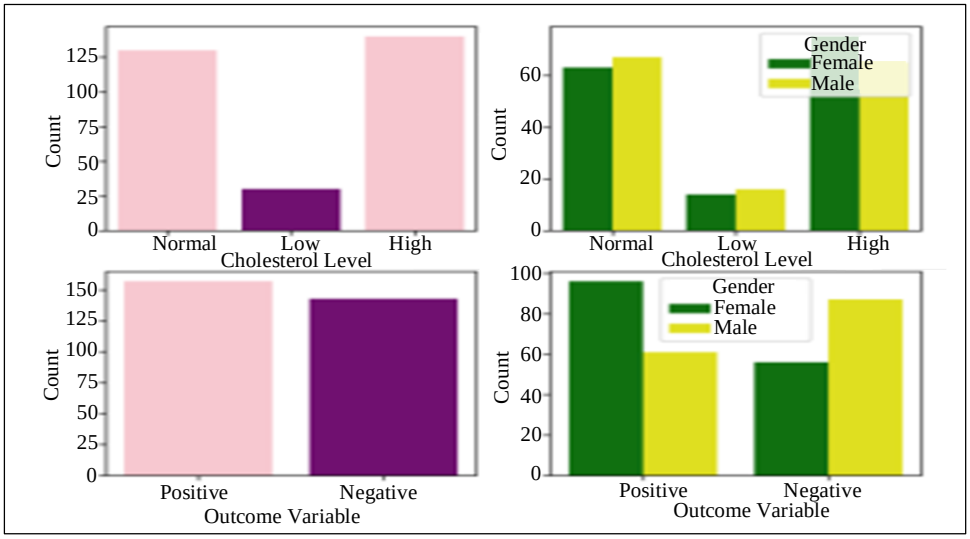


Fig. 4 Frequency plot for fever, cough, fatigue, difficulty breathing, gender, blood pressure, cholesterol level outcome variable

Visual inspection of the other factors reveals significant changes in illness prediction based on the values of each characteristic. For example, cholesterol levels-whether high, normal, or low-can have a considerable influence on illness prediction, reflecting the heterogeneity of diseases in the actual world. Two major insights emerge:

1. Low blood pressure is related to a reduced risk of stroke, which is an important component in stroke prediction.
2. Fatigue, cholesterol level, and blood pressure vary significantly between values, indicating that they might be powerful predictors in our model.

These observations give the importance of these variables in predicting diseases. Blood Pressure (BP) and Cholesterol Levels, as shown in Figure 3 shows, age-wise spread is above age 30 and High, low, and medium BP and cholesterol are equally contributing to the database. Figure 4 shows other independent features for all ages who tend to have issues with their health in any way, be it mild or serious disease. The above analysis shows no Class Imbalance is observed, and hence, no resampling and SMOTE analysis is required here [12].

3. Feature Analysis

To understand the relationships between different variables and to identify the patterns and potential features, correlation analysis is performed. The correlation graph in Figure 5 shows that none of the variables have high relationships with the 'Disease' variable. The variables 'Age' and 'Difficulty Breathing' had the strongest correlations, at 0.4 and -0.4, respectively. When dealing with several variables with poor correlation scores, machine learning becomes a viable option for prediction jobs. However, it's critical to recognize that machine learning algorithms, particularly deep learning models, frequently require large volumes of data to work well.

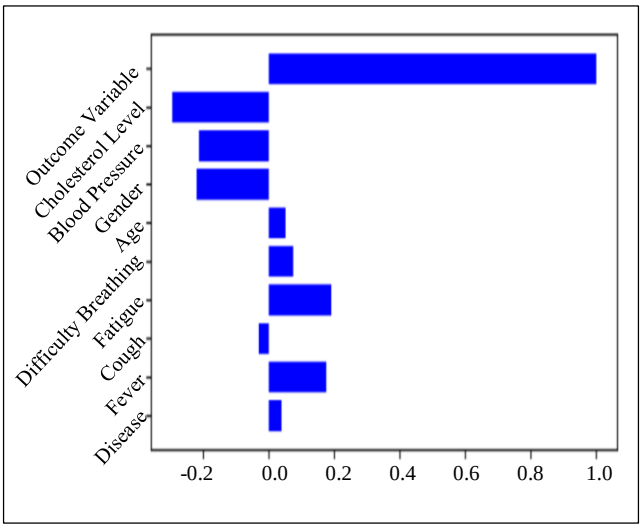


Fig. 5 Correlation graph

Table 1. Correlation among feature

Feature	Scale
Outcome Variable	1.000000
Fatigue	0.191344
Fever	0.175894
Difficulty Breathing	0.074605
Age	0.051587
Disease	0.039327
Cough	-0.030323
Blood Pressure	-0.212771
Gender	-0.219637
Cholesterol Level	-0.294008

Table 2. Feature value greater than zero counts for positively correlated features

Feature	Cases	% Contribution
Fatigue	212	70.67
Fever	143	47.67
Difficulty Breathing	73	24.33
Age	300	100.00
Disease	299	99.67
Cough	140	46.67
Blood Pressure	148	49.33
Gender	148	49.33
Cholesterol Level	160	53.33

4. Methodology

In the earlier section, primary data analysis is performed and determined the value of features is determined through correlation. Notably, here, feature scaling is unnecessary due to the relatively low number of features - only 9 in total. This analysis suggests that using only 3 features could enhance test accuracy, whereas, in this analysis, 9 features are retained to avoid the risk of overfitting. This decision was made to ensure the model's generalizability and robustness.

Figure 6 illustrates that although a subset of features may offer superior test accuracy, retaining all features enhances the odel's capacity to capture diverse patterns in the data.

Figure 7 shows the complete methodology of work; initially, data was split into training and testing data. Training data was cross-validated and pipelined. Cross-validation is an important tool for evaluating model performance, selection, and optimization in machine learning. This analysis uses a 5-fold cross-validation strategy, which frequently achieves a decent mix of bias and variance.

The cross-validation score achieved over all five folds was 0.64, suggesting strong model performance. To achieve accurate model assessment and prevent data loss, we used pipelines. Pipelines are useful in avoiding frequent mistakes, such as accidentally introducing test data into the training process.

After performing data transformation and pipelining, we obtained an MSE was 0.18. Cross validation was completed with default parameter selection, which was future-optimized by using hyperparameter tuning. Cross validated model was retrained with optimum hyperparameters. The trained model was evaluated future for different evaluation metrics like accuracy, F1 score, Recall, AUC and Jaccard index.

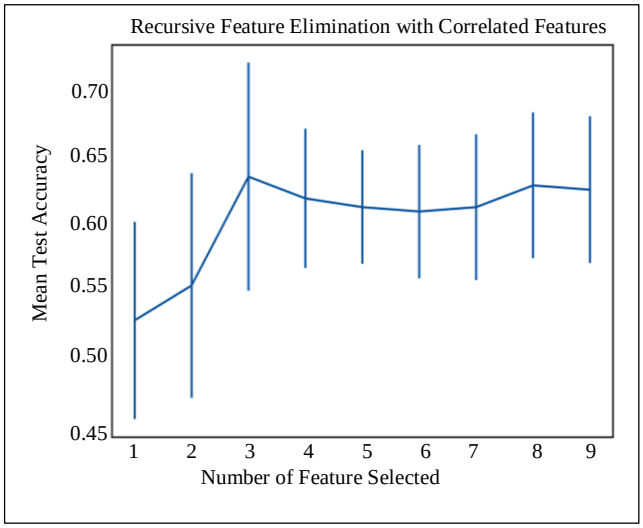


Fig. 6 Feature accuracy relation

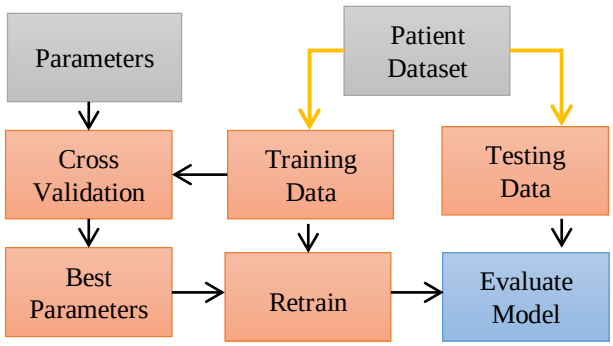


Fig. 7 Methodology for training ML models used are SVM, Decision Tree and Random Forest classifier

4.1. Classifiers and Hyperparameters

Various classifiers are available to train supervised ML models like LR, SVM, DT, RF, Bagging and Boosting Classifiers. Here, we have used SVM, DT and RF for the study to check the effect of hyperparameter tuning on classifier performance.

4.1.1. Logistic Regression (LR)

LR is a statistical model commonly known by various names such as logit regression, Maximum-Entropy classification (MaxEnt), or log-linear classifier. It uses a logistic function to assess the likelihood, outlining the most likely outcomes of a specific experiment or occurrence. In essence, LR models the connection between the dependent variable and one or more independent variables by estimating the likelihood that a given input falls into a certain category or class. This makes it a common choice for binary classification jobs, where the conclusion is either a "success" or a "failure", but it may also be modified to handle multi-class classification issues. The logistic function, also known as the sigmoid function, transforms input values into probabilities between 0 and 1, making it suitable for modeling binary outcomes. [9]. The prediction probability of a positive class for LR is given by,

$$\hat{P}(Z) = \frac{1}{1+e^{-z}} \quad (1)$$

4.1.2. Support Vector Machine (SVM)

SVM is a supervised learning technique used for classification, regression, and outlier identification. SVM's decision function is based on a subset of training data known as support vectors.

The decision function is commonly represented by Equation 2. SVM supports a variety of kernel types, including linear, polynomial, and Radial Basis Function (RBF). These kernels control the transformation performed to distinguish between classes. Key parameters in SVM include:

Regularization Parameter (C)

This parameter determines the trade-off between margin width and classification error. Higher values of C result in narrower margins but potentially reduced classification mistakes, whereas lower values allow for wider margins but may result in more misclassification.

Gamma Parameter (γ)

This parameter influences the flexibility of the decision boundary in the RBF kernel. A higher value of gamma leads to a more complex decision boundary, potentially resulting in overfitting, while a lower value results in a smoother decision boundary.

Kernel Coefficient (Degree)

This parameter defines the polynomial degree in polynomial kernels. It determines the complexity of the polynomial transformation.

Class Weights

This parameter adjusts for class imbalances by assigning different weights to different classes. It helps in handling scenarios where one class might dominate the other in terms of the number of samples.

Kernel Cache Size

This parameter specifies the memory allocation for training optimization. It determines the amount of memory used for storing intermediate results during the training process, affecting the speed and efficiency of training.

Each of these parameters plays a crucial role in determining the performance of an SVM model and should be carefully tuned based on the specific characteristics of the dataset and the desired outcome [10, 12].

$$\sum_{i \in SV} y_i \alpha_i K(x_i x) + b \quad (2)$$

Where α_i represents the duel coefficient, $K(x_i x)$ represents support vectors, and b represents the intercept.

4.1.3. Decision Tree (DT) Classifier

DT is a non-parametric supervised learning technique used for classification and regression applications. Hyperparameters are parameters whose values are determined before the learning process begins and which govern the algorithm's behaviour. Major hyperparameters that control the performance of a Decision Tree model are given below.

Criterion

This hyperparameter defines the quality of a split. Common values include 'gini' for Gini impurity and 'entropy' for information gain. It specifies the impurity measure used to determine the splitting of nodes during the tree-building process.

Max Depth

The maximum depth of the tree prevents overfitting by restricting the depth to which it may develop. A deeper tree can capture more complicated relationships in data, but it may also result in overfitting.

Min Samples Divide

This hyperparameter sets the smallest amount of samples needed to divide an internal node. If the amount of samples at a node is less than this value, it will not be divided further, hence preventing overfitting.

Min Samples Leaf

The number of samples necessary to be present at a leaf node. It guarantees that each leaf node has a minimal number of samples, preventing the formation of leaf nodes with only a few occurrences, which might lead to overfitting.

Max Features

This hyperparameter controls the number of features to consider when looking for the best split at each node. It can be set to a fixed number or a percentage of the total features. Limiting the number of features considered can help in reducing the computational cost and overfitting.

Tuning these hyperparameters appropriately is crucial for achieving a well-generalized Decision Tree model that performs effectively on unseen data. Balancing model complexity and generalization ability is essential for optimal performance

Consider features are represented by$x_i \in R^n, i = 1, \ldots \ldots, l$ and Label data is given by $y_i \in R$, a DT divides the feature space recursively, grouping samples with similar labels or target values together [16].

4.1.4. Random Forest (RF) Classifier

Random Forest (RF) is an ensemble machine learning approach that works for both regression and classification applications. It creates several decision trees and combines

them to provide more reliable and accurate forecasts. By combining individual tree forecasts, RF can reduce overfitting, which is a common problem with standalone Decision Trees.

In a traditional Decision Tree, the algorithm selects the best features from a sample during tree construction, employing a greedy strategy that may result in overfitting. However, Random Forest addresses this concern by creating numerous Decision Trees and considering the collective decisions of these trees. By examining a multitude of trees, the tendency towards overfitting diminishes, as the ensemble approach tends to capture more generalized patterns in the data. In essence, Random Forest leverages the wisdom of multiple decision trees to produce more reliable and robust predictions, thereby overcoming the overfitting problem commonly associated with individual Decision Trees [17-19].

4.1.5. Gradient Boosting (GB) Classifier

GB Classifier, generally outperforms random forests. Gradient-enhanced tree models are built step-by-step like other enhancement methods but generalize the other methods by allowing a differentiable loss function to be optimized [20, 21].

4.2. Hyperparameters Tuning

Hyperparameter tuning, typically through techniques like grid search or randomized search, optimizes the performance of the algorithm is obtained by selecting the best combination of these parameters. Grid search methodically investigates a predetermined grid of hyperparameter values to determine the best combination for a machine learning model. It assesses each combination using cross-validation or a validation set and chooses the one that produces the best results. To find the best cross-validation score, it's essential to search the hyperparameter space. This process typically involves:

- Selecting an estimator (e.g., sklearn.svm.SVC() for Support Vector Classifier).
- Defining the parameter space to explore, including hyperparameters like C, kernel, gamma, or alpha.
- Choosing a method for searching or sampling hyperparameter candidates.
- Setting up a cross-validation scheme to evaluate each candidate's performance.
- Using a score function to assess the model's performance during cross-validation.

In essence, hyperparameter tuning involves systematically exploring various parameter combinations to optimize model performance [4].

5. Results and Analysis

Table 3 shows the performance metrics of various models. The logistic regression algorithm achieved an accuracy of 62.5% on the cross-validation set. The best parameter for regularization was found to be 0.1, resulting in an accuracy of 68% on the validation set.

With this regularization parameter, the recall was 0.72, the AUC was 0.678, the Jaccard index was 0.56, and the F1 score was 0.68. The optimum number of features used was 3. The SVM algorithm achieved an accuracy of 67% on the cross-validation set. Using a regularization parameter (C) of 100 and a gamma value of 10 with an RBF kernel, the accuracy improved slightly to 66%. However, when the regularization parameter was set to 0.60, the accuracy matched the original cross-validation accuracy at 67%.

The recall was 0.5, the AUC was 0.67, the Jaccard index was 0.67, and the F1 score was 0.67. While SVM with an Rbf kernel showed comparable performance to LR, it did not significantly outperform it. The choice of regularization parameter seemed crucial, as it impacted the model's performance. Overall, logistic regression might be preferred due to its simplicity and similar performance. The DT algorithm achieved an accuracy of 69% on the cross-validation set.

The best parameter for regularization, max depth, was determined to be 8. With this parameter, the accuracy improved to 75%. The recall was 0.9, the AUC was 0.73, the Jaccard index was 0.67, and the F1 score was 0.74. The DT algorithm with an optimal max depth of 8 outperformed both logistic regression and SVM in terms of accuracy and other metrics such as recall, AUC, Jaccard index, and F1 score. Therefore, for this particular dataset, the DT algorithm appears to be the most suitable choice. The RF algorithm achieved an accuracy of 72% on the cross-validation set. The best parameters for regularization, max depth (d), minimum samples split (m), and number of estimators (M), were found to be 10, 5, and 7, respectively.

With these parameters, the accuracy improved slightly to 73%. The recall was 0.818, the AUC was 0.723, the Jaccard index was 0.63, and the F1 score was 0.73. Random Forest outperformed the other algorithms in terms of accuracy and various evaluation metrics, indicating its suitability for the dataset. Its ability to handle complex relationships and reduce overfitting contributed to its effectiveness in this scenario. The Adaboost algorithm achieved an accuracy of 64% on the cross-validation set.

Using 12 estimators (M) with a learning rate of 0.1, the accuracy improved to 76%. However, the recall was 0.71, the AUC was 0.64, the Jaccard index was 0.49, and the F1 score was 0.64. While Adaboost showed improvement in accuracy compared to its baseline, its performance in terms of recall, AUC, Jaccard index, and F1 score was not as strong as other algorithms such as Random Forest or Decision Tree.

Further optimization or exploration of different algorithms may be necessary to achieve better results. The

Gradient Boost algorithm achieved an accuracy of 67.8% on the cross-validation set. Utilizing 4 estimators (M) with a learning rate of 1, the accuracy improved to 76%.

The recall was 0.77, the AUC was 0.71, the Jaccard index was 0.57, and the F1 score was 0.72. Gradient Boost demonstrated notable improvement over the baseline accuracy. It performed competitively with other algorithms, exhibiting high recall and achieving a respectable balance of evaluation metrics.

Figures 8, 9, and 10 show that logistic regression achieves moderate performance. It demonstrates a decent accuracy and F1 score, but its Jaccard index indicates a relatively weaker agreement between predicted and actual labels. SVM with an 'rbf' kernel and tuned parameters show a strong recall, indicating its ability to identify positive cases effectively.

However, its overall accuracy is not as high as expected, suggesting potential overfitting or suboptimal parameter tuning. DTs, especially when pruned to a maximum depth of 5, exhibit competitive performance with respectable accuracy, recall, and AUC. Their interpretability and ease of understanding make them attractive choices for this task. Random forests, with an ensemble of decision trees, outperform individual decision trees, achieving the highest accuracy among all models tested. They also demonstrate high recall and AUC, indicating robust performance.

AdaBoost performs reasonably well but falls short compared to Random Forest and Gradient Boost in terms of accuracy and AUC. Its recall is lower, indicating potential challenges in identifying positive cases. Gradient Boost emerges as one of the top-performing models, with competitive accuracy, recall, and AUC.

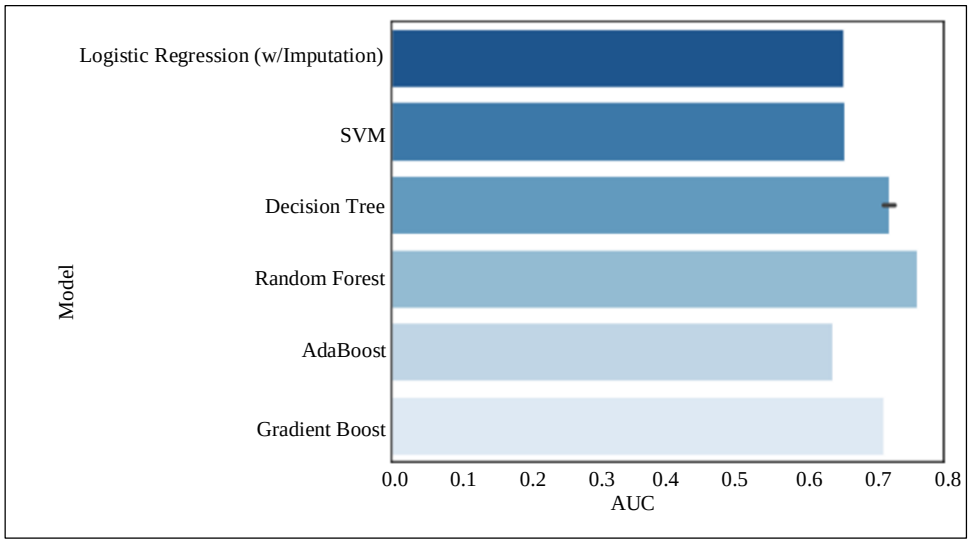


Fig. 8 AUC values for different tested models

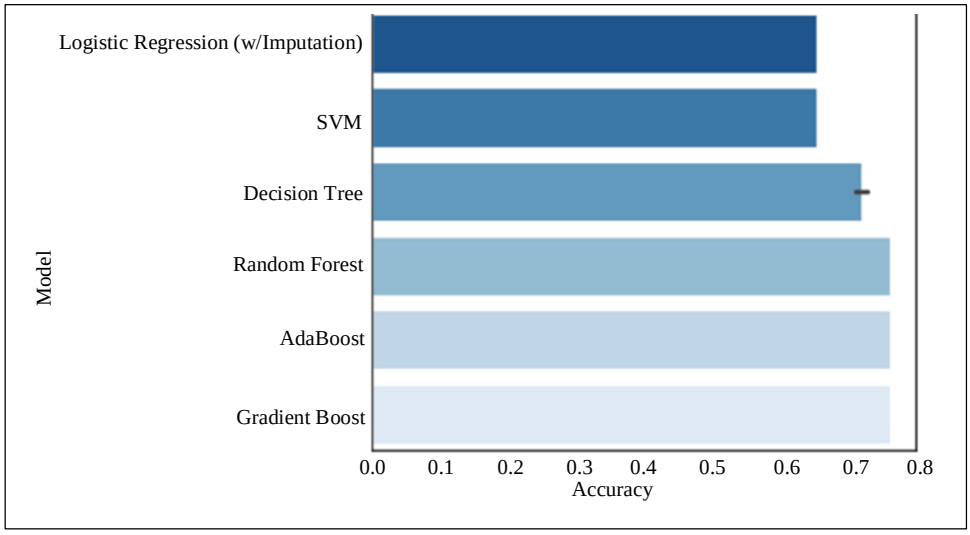


Fig. 9 Accuracy values for different tested models

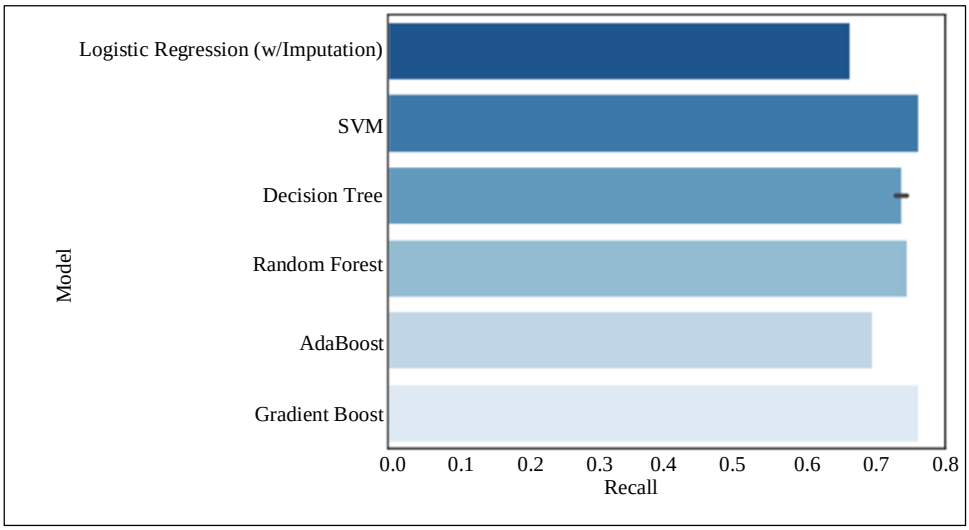


Fig. 10 Recall values for different tested models

Table 3. Performance evaluation table

Algorithm	Accuracy on Cross-Validation Set	Best Parameter for Regularization	Accuracy with the Best Parameter	Recall with the Best C Parameter	AUC with the Best C Parameter	Jaccard Index	F1 Score	Optimum Features
Logistic R	0.625	0.1	0.68	0.72	0.678	0.56	0.68	3
SVM	0.67	C=100, Gamma=10 Rbf Kernel	0.66	0.60	0.67	0.5	0.67	-
Decision Tree	0.69	Max Depth =8	0.75	0.9	0.73	0.67	0.74	-
Random Forest	0.72	M=10, d=5, m=7	0.73	0.818	0.723	0.63	0.73	-
Adaboost	0.64	M=12, LR=0.1	0.76	0.71	0.64	0.49	0.64	-
Gradient Boost	0.678	M=4, LR=1	0.76	0.77	0.71	0.57	0.72	-

6. Conclusion

Advancement of AI provides ease in personalize care to patients based on the predication ML model. ML life lifecycle involves various stages to improve the overall performance of the model. Hyperparameter tuning provides the best parameter choice and provides a more robust model. Different ML models tested show that among the models evaluated, Gradient Boost emerges as the top performer, exhibiting high accuracy, recall, and AUC. It strikes a balance between performance and computational efficiency, making it well-suited for this classification task. Overall, the findings highlight the importance of leveraging machine learning techniques to enhance patient profiling, diagnosis, and treatment in the medical field.

References

[1] Mojdeh Rastgoo et al., "Tackling the Problem of Data Imbalancing for Melanoma Classification," *Proceedings of the 9th International Joint Conference on Biomedical Engineering Systems and Technologies*, vol. 2, pp. 32-39, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[2] Jundong Li et al., "Feature Selection: A Data Perspective," *ACM Computing Surveys*, vol. 50, no. 6, pp. 1-45, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[3] Tzu-Tsung Wong, and Po-Yang Ye, "Reliable Accuracy Estimates from k-Fold Cross Validation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 8, pp. 1586-1594, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[4] Li Yang, and Abdallah Shami, "On Hyperparameter Optimization of Machine Learning Algorithms: Theory and Practice," *Neurocomputing*, vol. 415, pp. 295-316, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[5] Vikramaditya Jakkula, "Tutorial on Support Vector Machine (SVM)," *School of EECS, Washington State University*, vol. 37, 2006. [Google Scholar] [Publisher Link]

[6] Michele Fratello, and Roberto Tagliaferri, "Decision Trees and Random Forests," *Encyclopedia of Bioinformatics and Computational Biology*, vol. 1, pp. 374-383, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[7] Rémi Bardenet et al., "Collaborative Hyperparameter Tuning," *Proceedings of the 30th International Conference on Machine Learning*, vol. 28, no. 2, pp. 199-207, 2013. [Google Scholar] [Publisher Link]

[8] P. McCullagh, and John A. Nelder, *Generalized Linear Models*, 2nd ed., Routledge, 1989. [CrossRef] [Google Scholar] [Publisher Link]

[9] Sara Tehranipoor, Nima Karimian, and Jack Edmonds, "Breaking AES-128: Machine Learning-Based SCA, under Different Scenarios and Devices," *2023 IEEE International Conference on Cyber Security and Resilience (CSR)*, Venice, Italy, pp. 564-571, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[10] Alex J. Smola, and Bernhard Schölkopf, "A Tutorial on Support Vector Regression," *Statistics and Computing*, vol. 14, pp. 199-222, 2004. [CrossRef] [Google Scholar] [Publisher Link]

[11] Trevor Hastie, Robert Tibshirani, and Jerome Friedman, *Elements of Statistical Learning - Data Mining, Inference, and Prediction*, 2nd ed., Springer New York, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[12] Paritosh Jadhao et al., "Prediction of Early Stage Alzheimer's Using Machine Learning Algorithm," *2023 4th International Conference for Emerging Technology (INCET)*, Belgaum, India, pp. 1-5, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[13] Dina Elreedy, and Amir F. Atiya, "A Comprehensive Analysis of Synthetic Minority Oversampling Technique (SMOTE) for Handling Class Imbalance," *Information Sciences*, vol. 505, pp. 32-64, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[14] Laksika Tharmalingam, Disease Symptoms and Patient Profile Dataset, Kaggle, 2024. [Online]. Available: https://www.kaggle.com/datasets/uom190346a/disease-symptoms-and-patient-profile-dataset

[15] Xuchun Wang et al., "Exploratory Study on Classification of Diabetes Mellitus through a Combined Random Forest Classifier," *BMC Medical Informatics and Decision Making*, vol. 21, no. 1, pp. 1-14, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[16] Scikit Learn, Decision Tree Classifier. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

[17] K. VijiyaKumar et al., "Random Forest Algorithm for the Prediction of Diabetes," *2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN)*, Pondicherry, India, pp. 1-5, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[18] Carlos Fernandez-Lozano et al., "Random Forest-Based Prediction of Stroke Outcome," *Scientific Reports*, vol. 11, pp. 1-12, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[19] Derara Duba Rufo et al., "Diagnosis of Diabetes Mellitus Using Gradient Boosting Machine (LightGBM), *Diagnostic*, vol. 11, no. 9, pp. 1-14, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[20] Seung-Bo Lee et al., "Predicting Parkinson's Disease Using Gradient Boosting Decision Tree Models with Electroencephalography Signals," *Parkinsonism and Related Disorders*, vol. 95, pp. 77-85, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[21] K. Sudharani, T.C. Sarma, and K. Satya Prasad, "Brain Stroke Detection Using k-Nearest Neighbor and Minimum Mean Distance Technique," *2015 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, Kumaracoil, India, pp. 770-776, 2015. [CrossRef] [Google Scholar] [Publisher Link]