ISSN No:-2456-2165

# Hybrid Machine Learning Algorithm for Arrhythmia Classification Using Stacking Ensemble, Random Forest and J.48 Algorithm

Onwuka, Ugochukwu C.
Dept. of Computer Science
Ignatius Ajuru Uni. Of Education Iwofe, Nigeria

Asagba, Prince O.
Dept. of Computer Science
University of Port Harcourt, Nigeria

**Abstract:- Arrhythmias also known as dysrhythmia is a heart ailment that arises when electrical signals that coordinate the heartbeats do not work appropriately, they are often precursors to a number of heart diseases which may be terminal, and early detection and adequate treatment can save life, in this paper we propose a classification technique that blends two good performing machine learning algorithms to enhance the accuracy of detecting arrhythmia using Electrocardiogram (ECG) data and Weka machine learning tool, these algorithms include the J.48 and Random Forest algorithms combined with an ensemble algorithm called Stacking; For this experiment the MIT-BIH ECG dataset from Kaggle.com was used to train, test and validate the hybrid algorithm. This dataset used classified ECG data into the 5 super class of arrhythmia approved by the association for the advancement of medical instrumentation (AAMI) to be detectable by equipment and methods, they include normal sinus (N), fusion beat (F), supraventricular ectopic beat (SVEB), ventricular ectopic beat (VEB), and unknown beat (Q). the hybrid algorithm "stacked random forest and j.48) outperformed the other individual algorithms, the performance metrics gotten include 97.63% accuracy, an approximate sensitivity (recall) and Positive predictivity (precision) value of 0.98, other metrics includes a weighted precision recall curve area of 0.97, receiver operator characteristics area of 0.96 and test time of 1.66 seconds and finally a model size of 38.2mb which is suitable for building application for mobile devices.**

***Keywords:-*** *Machine Learning, Arrhythmia Classification, ECG, Random Forest, J.48, Stacking Ensemble.*

## I. INTRODUCTION

Arrhythmia is an ailment that ensues when electrical impulses that controls how the heart beats don't work as required, this makes the heart to beat faster than normal, too slow, flutter, fibrillate, or suffer early heartbeat known as premature contraction. sometimes, arrhythmias are precursors to cardiac arrest which could be fatal; The past two decades have seen considerable advancements in the diagnosis and management of supraventricular and ventricular arrhythmias [1], with digital devices being more available, this paper proposes a classification model for a more accurate detection of arrhythmia by using a hybrid machine learning algorithm.

The term hybrid machine learning algorithm is employed when an ensemble of heterogeneous collection of learners are involved in contrast to other ensemble models where homogenous collection of learners are mostly used as is the case of bagging or boosting.

Ensemble learning is a machine learning theory where two or more learners (machine learning algorithms) are trained or utilized on datasets to solve the same task by extracting several predictions then merged into a single composite prediction [2] Ensemble algorithms coalesces the decisions of separate classifiers that composes it, in order to improve the final prediction. according to [3] It is the procedure of running two or more related but different models and then fusing their outcomes into a single score or spread with the aim of improving the accuracy of predictive analytics and data mining applications.

### *A. Electrocardiogram (ECG)*

The electrical activities of the heart (typically of consisting depolarization and repolarization) is captured by the electrocardiogram, it facilitates the detection and diagnosis of heart anomalies by quantifying electrical potentials on the human body surface, generating a record of the electrical currents associated with heart muscular activities.

The propagation of electrical signals in the heart are pattern like, thus it results to electrical currents ensuing on the surface of the body and electrical potential on the skin surface; consequently, this potential is picked up and/or quantified with the aid of electrode or sensors. The electrical potential difference between the spaces where the electrodes are placed on the skin surface, are normally enhanced using an operational amplifier with optic isolation. Then, the signal is then fed to a high-pass filter; after which it is then also submitted to an antialiasing low-pass filter. Finally, the processed signal shows in an analogical to digital converter. The graphical illustration (a plot of voltage (mV) against time) of this process is called electrocardiogram (ECG). ECG was first demonstrated on humans by Augustus Desiré Waller in 1887 [4], since then, the heart's electrical activities have been recorded, however, the capacity to diagnose the normal cardiac rhythm and arrhythmias became a routine medical check-up from 1960s.

ISSN No:-2456-2165

## II. RELATED WORK

A number of methods for ensemble of classifiers already exists in literature, some techniques like bagging and adaBoost train each classifier with a unique subset of the training data; Dietterich and Bakiri explored this technique in dealing with a problem that required a huge number of classes, they split the number of outputs into different sets, then generated an ensemble of classifier. In another worked they trained each classifier in an altered subset of the input features [5]. Waske and Benediktsson explored the use of ensemble of support vector machines, SVMs for a multi-source land cover classification problem using a balanced dataset [6]. They ensemble support vector machines, training each SVM with a separate data source, this method improved their result extensively when compared with the outcome achieved employing a single SVM that was trained using the entire dataset. Similarly, the effectiveness of ensemble technique was demonstrated by Duin and Tax, they carried out vast experimentations of ensemble learning options with a conclusion that a combination of classifiers trained on diverse feature sets are beneficial, particularly if each classifier offers a yields good result [7].

Various researchers have implemented the use of different machine learning techniques as well as ensembles to classify ECG data, some of which are discussed as follows:

[8] performed a data mining experiment using 3 popular data mining algorithms which are ID3, CART and decision table to build an ensemble prediction model using a large dataset with 10-fold cross validation methods to measure the unbiased estimate. They concluded with a highest accuracy of 86.43%.

[9] implemented a feature enrichment convolution neural network (FE-CNN) classifier to predict 2 class of arrhythmia, they realized the FE-CNN by enriching the ECG signals from MIT-BIH arrhythmia database into time-frequency images using discrete short-time Fourier transform. These images are used as inputs for the CNN, there results showed that FE-CNN obtained a positive predictive rate of 90.1%, sensitivity of 75.6%, and F1 score of 0.82 for the detection of S beats. Sensitivity, positive predictively, and F1 score are 92.8%, 94.5%, and 0.94, respectively, for V beat detection.

[10] put forward a novel learning scheme that encapsulates a hybrid evolutionary fuzzy-rough feature selection model with an adaptive neural network ensemble. The fuzzy-rough method was setup to deal with uncertainty and impreciseness of real valued gene expression dataset and evolutionary search concept optimizes the subset selection process.

[11] proposed a deep neural network-based (DNN) method to predict 5 forms of heartbeat. To achieve good results, they eliminated noise from the ECG signals by applying a low-pass filter on the two-lead heartbeat segments with 2 seconds length generated from the filtered signals, and classified by an adaptive ResNet model. The proposed method was evaluated on the MIT-BIH Arrhythmia Database with the patient-specific pattern. The overall accuracy was 98.6%.

## III. MATERIALS AND METHODS

### A. ECG Dataset

The database used in this research is the MIT-BIH database available at Kaggle.com uploaded and preprocessed by Shayan Fazeli [12]. The MIT-BIH database is often used as it is the most characteristic database for arrhythmia classification and it has been used for most of the published research. Additionally, MIT-BIH is besides the foremost database obtainable for research purposes and has been refined continuously along the years [13]. The ECG signals are sampled to 125Hz and is used for training, testing and validation of the proposed algorithm, table 1, summarizes property the dataset used for our machine learning model.

| | |
|---|---|
| Sample count | 109,446 rows |
| Attributes count | 188 columns |
| Categories (Class) | 5 |
| Sampling Frequency (Hz) | 125 |
| Original Data Source | MIT-BIH Arrhythmia Dataset |
| Classes | [' N: 0, S: 1, V: 2, F: 3, Q: 4 |

Table 1: Dataset Summary

The Dataset provides 5 classes each represented by numerical figures as shown in table 1, Normal Beat (*N*): 0, Supraventricular ectopic beat *(S)*: 1, Ventricular ectopic beat (*V*): 2, Fusion Beat (*F*): 3 and Unknown Beat (*Q*): 4

### B. Preprocessing

Before carrying out the experiment to determine the performance of the algorithms, the raw data is processed to make it usable, here the preprocessing done by applying a nominal to numeric filter, the ECG data last column, which has the class of the dataset is a numeric field, thus will not give us a useful supervised learning classification result, hence, it is necessary to convert the numeric field to a nominal field. To achieve this in the Weka data mining tool (WEKA version 8.3.1), we will utilize the Numeric-to-Nominal filter with the "-R last" as an attribute as illustrated on figure 1 and 2.

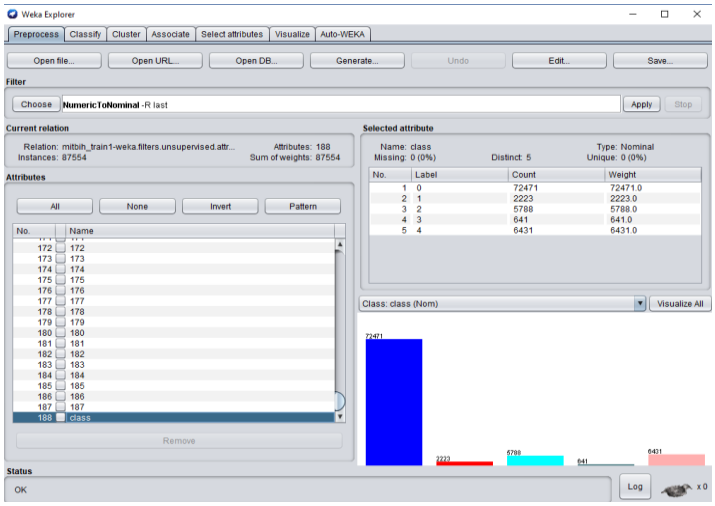


Fig 1: After applying filter NumericToNominal filter with the attributes -R last

Name: class Type: Nominal
Missing: 0 (0%) Distinct: 5 Unique: 0 (0%)

| No. | Label | Count | Weight |
|---|---|---|---|
| 1 | 0 | 72471 | 72471.0 |
| 2 | 1 | 2223 | 2223.0 |
| 3 | 2 | 5788 | 5788.0 |
| 4 | 3 | 641 | 641.0 |
| 5 | 4 | 6431 | 6431.0 |

Fig 2: Class Label Statistics

*C. Machine Learning Algorithms*

This section discusses the classification algorithms choosing to build the proposed hybrid classification algorithms, which consists of 3 machine learning algorithms; these algorithms include one base-learner (Random Forest), one meta-learner (J48) and one ensemble algorithm (stacking). These algorithms are explained as follows:

➢ *Random Forest*

Random Forest is an ensemble algorithm that is composed of decision trees (also called "forest"), thus it is an ensemble of decision trees that uses voting ensemble; as shown in Figure 3, to perform classification on a new object based on some attributes, each tree provides a classification, therefore we say the tree "votes" for that class. The forest chooses the classification having the most votes (over all the trees in the forest).

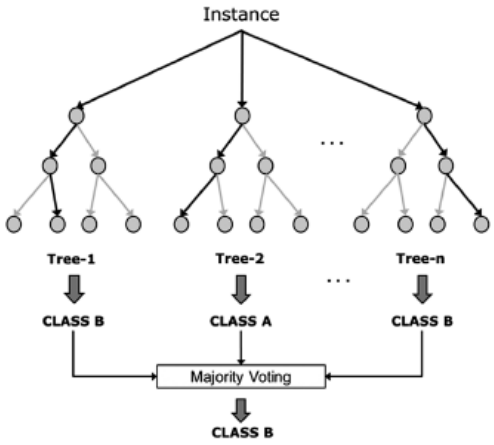


Fig 3: Random Forest Algorithm Depicted

Each tree is planted & grown as follows in random forest:
1. If N cases exists in the training set, then sample of N cases is taken at random but with replacement. This sample will be the training set for growing the tree.

2. If there are V input variables, a number v<<V is specified such that at each node, v variables are selected at random out of the V and the best split on these v is used to split the node. The value of v is held constant during the forest growing.

3. Each tree is grown to the largest extent possible. There is no pruning (that is, reduction of the size of trees).

Random Forest algorithm is explained as follows [14]:

Given a training dataset $D$: = $(a_1, b_1) \ldots (a_n, b_n)$, features (F), and number of trees in forest X.
1. function RandomForest($D$, F)
2. H ← Ø
3. for i ∈ 1. . . X do
4. $S^{(i)}$ ← A bootstrap sample from $D$
5. $h_i$ ← RandomizedTreeLearn $(D^{(i)}, F)$
6. $H \leftarrow H \cup \{h_i\}$
7. end for
8. return H
9. end function
10. function RandomizedTreeLearn $(D, F)$
11. At every node:
12. f ← very small subset of F
13. Split on best feature in f
14 return the learned tree
15 end function

➢ *J.48*

J.48 machine learning algorithm is WEKA data mining tool open source Java implementation of Quinlan's C4.5 algorithm for making pruned or unpruned decision tree; it is an extension of Quinlan's prior ID3 algorithm. "C4.5 was previously ranked number one data mining algorithm in 2008" according to [15]. A given set of training data that is labeled can be used by J.48 to build decision trees using the concept of information entropy. It employs the fact that every attribute of the data can be used to make a decision by splitting the data into smaller subsets. The decision trees generated by J.48 can then be used for classification of new unknown data.

J.48 algorithm is explained as follows [16]:
Input: an attribute valued dataset $D$
1. Tree = {}
2. if $D$ is "Pure" OR other stopping criteria met then
3. terminate
4. else if
5. For all attribute a ∈ $D$ do
6. Compute information-theoretic criteria if we split on 'a'
7. end for
8. $a_{best}$ = Best attribute according to above computed criteria
9. Tree = Create a decision node that tests abest in the root
10. $D_v$ = Induced sub-datasets from $D$ based on $a_{best}$
11. for all $D_v$ do
12. $Tree_v$ = J.48 $(D_v)$
13. Attach $Tree_v$ to the corresponding branch of Tree
14. end for
15. Return Tree

➢ *Stacking*

Stacking also called blending by some literature entails training a learning algorithm to combine the predictions of several other learning algorithms. Firstly, other algorithms are trained using the dataset presented, then a combiner algorithm is trained to generate a final prediction with every other predictions of the other algorithms as added inputs. Stacking ensemble algorithm is elucidated in the following steps:

*D. Algorithm Steps*

*1: Learn first-level (Base-learner) classifiers based on the original training data set.*
base classifiers are learned, based on dataset provided with a weight distribution; parameters can be tuned to generate distinct base classifiers for homogeneous classifiers; different classification methods and/or sampling methods can be applied to generate base classifiers for heterogeneous classifiers.

*2: A new dataset is built based on the output(s) generated by the base classifiers.*
Output(s) or predicted labels of the base classifiers are held as new features, and the original class labels are kept as the labels in the new dataset generated. Instead of using predicted labels.

*3: Learn a second-level classifier based on the newly constructed data set.*

Any learning method could be applied to learn second-level classifier.

Stacking algorithm is a general framework, as such we can plug in various classifiers and learning approaches to create the first-level features and transform the data into a different feature space.

Stacking ensemble algorithm is described as follows;

1. Input: Training Data $D = \{x_i, y_i\}_{i=1}^{m}$
   First-level (base) algorithms $\varepsilon_1, \dots, \varepsilon_T$
   Second-level (meta) algorithms $\varepsilon$
2. Step 1: learn base-leave classifiers
3. For t = 1 to T do
4. Learn $h_t$ based on $D$
5. End for
6. Step 2: construct new data set of predictions
7. For i = 1 to m do
8. $D_h = \{x_i', y_i\}, where\ x_i' = \{h_1(x_i), \dots, h_t(x_i)\}$ % generate a new dataset
9. End for
10. Step 3: learn a meta classifier
11. Learn $H$ based on $D_h$
12. Return $H$
13. Output: ensemble classifier $H$

Stacking algorithm usually yield improved performance than most single trained models. It has been productively used on both supervised learning tasks (regression, classification and distance learning) and unsupervised learning (density estimation).

## IV. ANALYSIS AND RESULTS

WEKA, an open source machine learning software that provides tools for data preprocessing, visualization and data mining was used in this research; Weka, can be downloaded from the link *www.cs.waikato.ac.nz/ml/weka/downloading.html*. The computer system we used in this research is the HP EliteBook 8460p with Intel core i7 processor, 8Gb of RAM, 1Gb of VRAM and Windows 10 OS.

*A. Performance metrics*

To make the right comparison, it is expedient to understand the measures recommended by the association for the advancement of medical instrumentation AAMI for evaluating methods. These methods include the following: Sensitivity (*Se*), Positive predictivity (+*P*), False positive rate (*FPR*), and overall accuracy (*Acc*). Sensitivity and Positive Predictivity are also known in the literature as recall and precision, respectively. The ECG dataset used was split by 70-30, 70% will be used for training and the remaining 30% will be used for testing and validation of the algorithms.

*B. Accuracy, Model size and Test Time*

Accuracy is one the metric for evaluating classification models. Informally, accuracy is the fraction of predictions our algorithm got right. Mathematically, accuracy has the following definition:

$$Accuracy = \frac{(Number\ predicted\ correctly)}{Total\ number\ of\ predictions}$$

For non-binary classification (more than two classes), accuracy is calculated in terms of positives and negatives as follows:

$$Accuracy = \frac{(TP + TN)}{TP + TN + FP + FN}$$

Where *TP* = True Positives, *FP* = False Positives, *FN* = False Negatives and *TN* = True Negatives.

Table 2 shows the accuracy, model sizes and test time of the different algorithms examined; Figure 5, Figure 6 and Figure 7 illustrates the test time, accuracy and model sizes of the algorithms graphically for easier appreciation and understanding.

ISSN No:-2456-2165

| | Accuracy (%) | Kappa statistics | Build Time (Sec) | Test Time (Sec) | Model Size (Kb) |
|---|---|---|---|---|---|
| Random Forest | 97.23 | 0.90 | 175.50 | 1.95 | 38,171 |
| J.48 | 95.98 | 0.87 | 194.72 | 0.09 | 527 |
| Stacked Random Forest & J.48 | 97.63 | 0.92 | 1789.64 | 1.66 | 38,150 |

Table 2: Accuracy, Build time, Kappa Statistics, Model size and Test Time

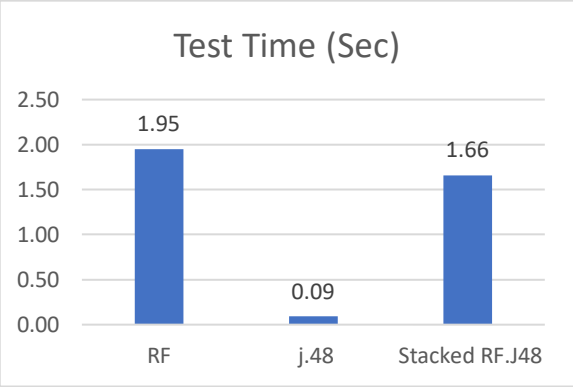


Fig 4: Test time for the various algorithms

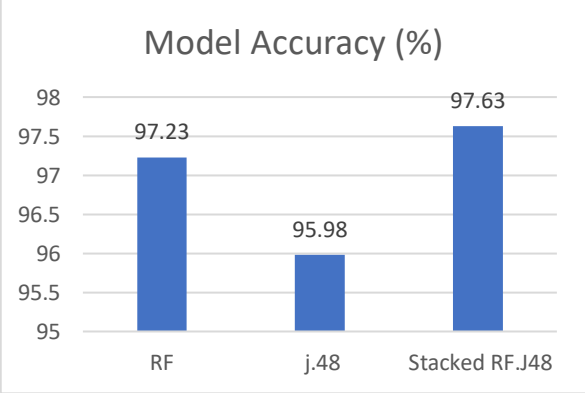


Fig 5: Model Accuracy

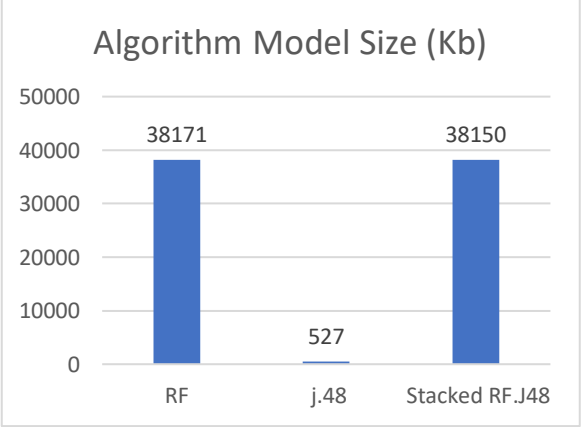


Fig 6: Algorithm model sizes

The results show that the stacked random forest has the best accuracy of 97.63% and a model size of 38.15mb, which is an improved performance. The model size is worthy of note, if one plans to develop for a mobile application (eg android or raspberry pi) with the trained model; this is because machine learning models shouldn't exceed 50mb for mobile application.

*C. Positive Predictivity (+P)*

Positive predictivity (precision) is defined as the sum of true positives (TP) over the sum of true positives and false positives (FP); it is an indicator of how sure we are of our true positive results, high scores for precision indicates that the classifier is returning accurate true positives.

Mathematically, Positive predictivity is given as

$$+P = \frac{T_p}{T_p + F_p}$$

Table 3 shows the Positive predictivity of the experimented algorithms; it can be seen that Stacked Random Forest and j48 performed best with weighted average of 0.98.

| | J.48 | RF | Stacked RF & J48 |
|---|---|---|---|
| 0 (N) | 0.973 | 0.122 | 0.98 |
| 1 (S) | 0.766 | 0.121 | 0.887 |
| 2 (V) | 0.886 | 0.198 | 0.954 |
| 3 (F) | 0.726 | 0.903 | 0.865 |
| 4 (Q) | 0.949 | 0.98 | 0.987 |
| Weighted average | 0.958 | 0.195 | 0.976 |

Table 3: Positive predictivity

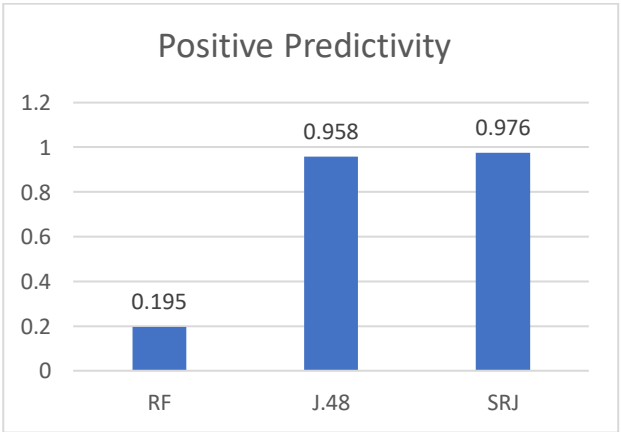


Fig 7: Weighted Positive predictivity

ISSN No:-2456-2165

*D. Sensitivity (Se)*

Sensitivity (Recall) is defined as the number of true positives (Tp) over the number of true positives plus the number of false negatives (Fn). It is a measure how sure we are that our model is not omitting any positive value, high scores for recall indicates that model trained is performing as expected and returning a majority of all positive results.

$$Se = \frac{T_p}{T_p + F_n}$$

Table 4 shows the recall of experimented algorithms, it can also be seen that Stacked Random Forest and J48 performed best with weighted recall average of 0.976

| | **J.48** | **RF** | **Stacked RF & J48** |
|---|---|---|---|
| 0 (N) | 0.984 | 0.999 | 0.994 |
| 1 (S) | 0.627 | 0.635 | 0.724 |
| 2 (V) | 0.862 | 0.865 | 0.905 |
| 3 (F) | 0.582 | 0.566 | 0.689 |
| 4 (Q) | 0.939 | 0.934 | 0.962 |
| Weighted average | 0.96 | 0.972 | 0.976 |

Table 4: Sensitivity

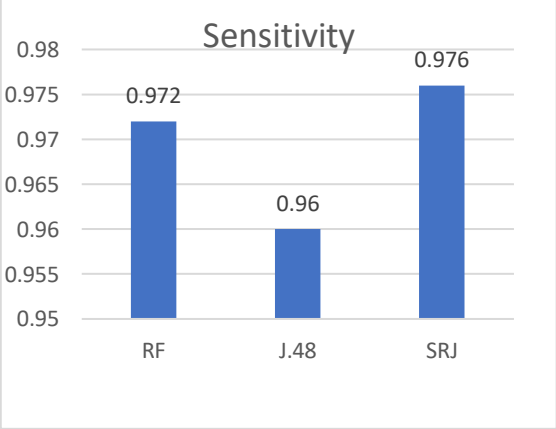


Fig 8: Weighted Sensitivity (recall) of the Algorithms

*E. False Positive Rate (FPR)*

FPR is the ratio of samples not belonging to a given (say class A) that was inaccurately classified as that class (class A), the lower FPR a trained model gives the better the model.

$$FPR = \frac{Fp}{Fp + Tn}$$

Table 5 shows the FPR of the experimented algorithms, Stacked Random Forest and j48 also performed best with an average low false positive rate of 0.08.

| | **J.48** | **RF** | **Stacked RF & J48** |
|---|---|---|---|
| 0 (N) | 0.13 | 0.147 | 0.096 |
| 1 (S) | 0.005 | 0 | 0.003 |
| 2 (V) | 0.008 | 0.001 | 0.003 |
| 3 (F) | 0.002 | 0 | 0.001 |
| 4 (Q) | 0.004 | 0 | 0.001 |
| Weighted average | 0.109 | 0.122 | 0.08 |

Table 5: False Positive Rate

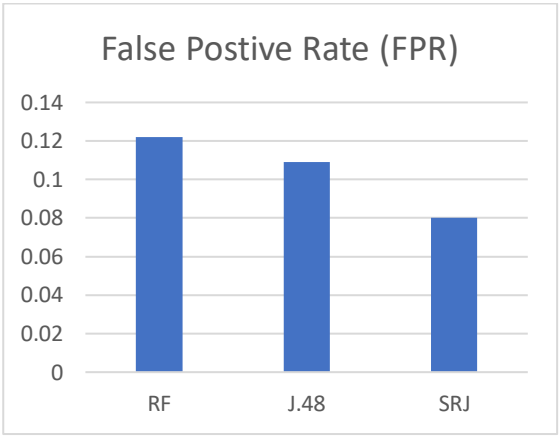


Fig 9: Weighted Average FPR of the Algorithms

*F. Receiver Operator Characteristics (ROC) of the Algorithms*

ROC is a measuring standard in medical and biological machine learning algorithms; that helps one evaluate the effectiveness of the models created. The ROC curve statistically models false positive and false negative detections in noisy environments, ROC area represents the performance averaged over all possible cost ratios.

The ROC area has various prediction level given as follows:

1.0 = Perfect, 0.9 = Excellent, 0.8 = Good, 0.7 = Mediocre, 0.6 = Poor, 0.5 = Totally random, < 0.5 = Invalid

2.0

Shown in table 6 is the ROC of the algorithms, stacked random forest performed excellently with the ROC value of 0.964

| | J.48 | RF | Stacked RF & J48 |
|---|---|---|---|
| 0 (N) | 0.936 | 0.994 | 0.965 |
| 1 (S) | 0.856 | 0.985 | 0.904 |
| 2 (V) | 0.926 | 0.998 | 0.968 |
| 3 (F) | 0.831 | 0.982 | 0.913 |
| 4 (Q) | 0.973 | 0.999 | 0.984 |
| Weighted average | 0.935 | 0.995 | 0.964 |

Table 6: Receiver Operator Characteristics Area of the algorithms

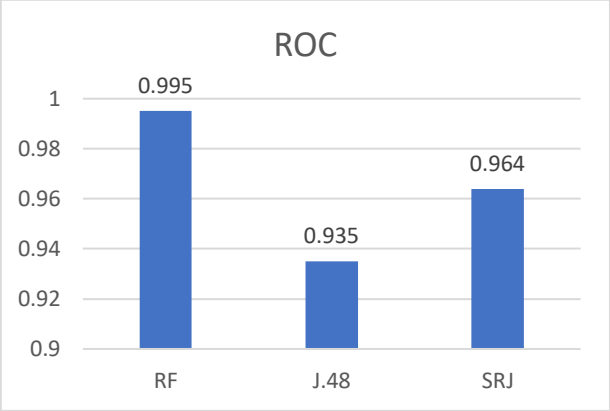


Fig 10: Weighted Average ROC of the Algorithms

*G. Precision Recall Curve Area*

A high area under the curve represents both high recall and high precision, as show in table 7.

| | J.48 | RF | Stacked RF & J48 |
|---|---|---|---|
| 0 (N) | 0.972 | 0.999 | 0.986 |
| 1 (S) | 0.53 | 0.863 | 0.764 |
| 2 (V) | 0.786 | 0.98 | 0.927 |
| 3 (F) | 0.462 | 0.825 | 0.723 |
| 4 (Q) | 0.885 | 0.994 | 0.968 |
| Weighted average | 0.938 | 0.992 | 0.973 |

Table 7: Precision Recall Curve Area

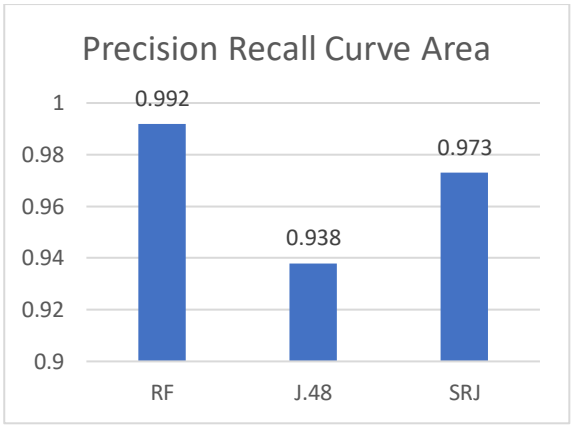


Fig 11: Weighted Average PRC of the Algorithms

## V. CONCLUSION

The experimental results reveals that the hybrid algorithm, stacked random forest and J.48 performed better than the individual algorithms on the MIT-BIH arrhythmia dataset with a good accuracy of 97.63%, an approximate recall and precision value of 0.98, PRC area of 0.97, ROC area of 0.96 reassures its effectiveness at providing excellent results and a test time of 1.66sec. the hybrid algorithm (Stacked Random forest and J.48) performed brilliantly thus is a better choice for automatic arrhythmia application design. Though the model size of 38.2mb is a bit large, it is still a good model size for machine learning application design for mobile devices, given that benchmark size for mobile application machine learning models is 50mb.

## REFERENCES

[1]. Aro, A. L., & Chugh, S. S. (2018). Epidemiology and global burden of arrhythmias (Vol. 1). https://doi.org/10.1093/med/9780198784906.003.0064
[2]. Onwuka, U. (2019). Ensemble Learning. Retrieved July 22, 2019, from http://megacomsnet.com.ng/ensemble-learning
[3]. Burn, E. (2015) What is ensemble modeling? - TechTarget. Retrieved July 23, 2019, from https://searchbusinessanalytics.techtarget.com/definition/Ensemble-modeling
[4]. Oxford D (2004). Waller, Augustus Désiré (1856–1922), physiologist. https://doi.org/10.1093/ref:odnb/38099
[5]. Dietterich, T. G., & Bakiri, G. (1991). Error-correcting output codes: A general method for improving multiclass inductive learning programs. AAAI Press. AAAI. 572–577
[6]. Waske B., Benediktsson, J. A. (2007) Fusion of Support Vector Machines for classification of multisensor data, IEEE Trans. Geosci. Remote Sens. 3858–3866.
[7]. Duin, Robert & Tax, David. (2000). Experiments with Classifier Combining Rules. 16-29. 10.1007/3-540-45014-9_2.

[8]. Chaurasia, Vikas and Pal, Saurabh (2013) Early Prediction of Heart Diseases Using Data Mining Techniques Caribbean Journal of Science and Technology, Vol. 1, 208-217, Available at SSRN: https://ssrn.com/abstract=2991237
[9]. Xie, Qingsong & Tu, Shikui & Wang, Guoxing & Lian, Yong & Xu, Lei. (2019). Feature Enrichment Based Convolutional Neural Network for Heartbeat Classification from Electrocardiogram. IEEE Access. PP. 1-1. 10.1109/ACCESS.2019.2948857.
[10]. Dash, Sujata & Patra, Bichitrananda. (2020). Genetic Diagnosis of Cancer by Evolutionary Fuzzy-Rough based Neural-Network Ensemble. 10.4018/978-1-7998-1204-3.ch036.
[11]. Zhao, W., Hu, J., Jia, D., Wang, H., Li, Z., Yan, C., & You, T. (2019). Deep Learning Based Patient-Specific Classification of Arrhythmia on ECG signal. Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS, (July), 1500–1503. https://doi.org/10.1109/EMBC.2019.8856650
[12]. Kaggle. (2018). ECG Heartbeat Categorization Dataset | Kaggle. Retrieved July 9, 2020, from https://www.kaggle.com/shayanfazeli/heartbeat
[13]. Luz, E. J. da S., Schwartz, W. R., Cámara-Chávez, G., & Menotti, D. (2016). ECG-based heartbeat classification for arrhythmia detection: A survey. Computer Methods and Programs in Biomedicine, 127, 144–164. https://doi.org/10.1016/j.cmpb.2015.12.008
[14]. Maheswari K M. Uma, Ashwin Pranesh, S Govindarajan (2018). "Network Anomaly Detector using Machine Learning", International Journal of Engineering & Technology
[15]. X. Wu, V. Kumar, J.R. Quinlan (2008). Knowledge and Information Systems, 14(1), 1~37.
[16]. The Free Library. (2014). A comparative study of data mining algorithms for decision tree approaches using WEKA tool. - Free Online Library. Retrieved September 10, 2020, from https://www.thefreelibrary.com/A+comparative+study+of+data+mining+algorithms+for+decision+tree...-a0505467547