# Design and Implementation of an Automatic Arrhythmia Classification System Using a Hybrid Machine Learning Algorithm (Stacked Random Forest and J.48 Algorithms)

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:- Arrhythmia is an abnormal condition of the heart that occurs when the electrical impulses that coordinate the heartbeats do not work properly, causing the heart to beat too fast, too slow, irregular or even have premature contractions; accurate and quick diagnosis can reduce fatalities, Thispaper focuses on design and development of a system that will automatically and accurately diagnose five super-class of arrhythmias recommended by theAssociation for the Advancement of Medical Instrumentation (AAMI) to be detectable by equipment/methodswhich includes: Normal(*N*), Supraventricular ectopicbeat (*SVEB*), Ventricular ectopic beat (*VEB*), Fusion beat (*F*) and Unknown beat (*Q*). To achieve this, the system uses a hybrid classification algorithm achieved by the combination of two better performing algorithm (Random Forest and J.48 ensembled with the stacking algorithm). The MIT-BIH ECG arrhythmia database accessible at Kaggle.com was used for training, testing and validation of the system. since this design is intended for edge devices or mobile devices, the design focuses on development of a system that uses less computation power, less application size and ability to give correct classification with less time. Our results showed that the designed system has an overall accuracy of 97.67%, average precision value of 0.977, average recall value of 0.977 and average F1-measure value of 0.977.**

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

## I. INTRODUCTION

In this present day, digitized data collection is increasing, biomedical data analysis is attracting more interest; however, designing an efficient and easily available support systems for heart condition monitoring based on machine learning is still an on-going research. Due to the delicate nature of our heart, an accurate, reliable, easily available and mobile heart monitoring system able to support decision of specialist in diagnosis and treatment of heart diseases even in remote places is an exigent need. A number of machine learning diagnosis support systems capable of providing reliable accurate diagnosis already exists, but most often very expensive, not easily able and requires large computing needs.

This research focuses on designing an improved arrhythmia classification and detection system using edge devices (fog computing) to carry out substantial amount of the computation instead of relying on cloud computers and applying hybrid machine learning technique to improve the accuracy of the diagnosis. This will help foster supports systems for physicians in remote places and also guard the security of biomedical data collected.

Ensemble methods are often referred to as meta-algorithms that combine several machine learning techniques into one predictive model in order to decrease variance (bagging), bias (boosting), or improve predictions (stacking).

Ensemble methods have been known to improve classification results, but few researchers have explored its use particularly in the field of improving ECG classification results using classical machine learning techniques. This research, will explore this option, using stacking method to ensemble our classifiers in other to achieve an improved prediction. Machine learning algorithms used in this research are the random forest and J.48 algorithms, both which are classical machine learning algorithm.

Weka machine learning tool, will be used for model generation (building) in this research, the model generated from the ensemble algorithm (hybrid algorithm) will be used to develop a software for our target edge device, the raspberry pi; using Java programing language.

➢ *Aim and Objectives of the Study*

The aim of this research is to design and implement a system using a hybrid classification model (Stacked Random Forest and J.48 algorithm) for differentiating between several arrhythmias more accurately.

The Objectives includes to:

1. Design a new improved cardiac arrhythmia classification system to correctly predict the presence of 5 main superclass of arrhythmia

2. Train a model using an ensemble of classification algorithms in other to obtain a better classification result and accuracy.
3. To develop a software using the model from the trained data.
4. To compare the old system with the proposed system.

➢ *Electrocardiogram (ECG)*

The electrical activity of the heart (depolarization & repolarization) is measured by an instrument called electrocardiogram, its aids the detection and diagnosis of heart abnormalities by measuring electrical potentials on the body surface and generates a record of the electrical currents associated with heart muscle activity. Augustus Desiré Waller was the first to demonstrate human ECG in 1887 (OxfordDNB, 2004), since then, the electrical activities of the heart has been recorded, the ability to recognize the normal cardiac rhythm and/or arrhythmias did not become routine in medical check-ups until 1960. Nowadays, there are many approaches to measurement / recording ECG. da Silva et al., provided a taxonomy of state-of-the-art ECG measurement methods: in-the-person, on-the-person and off-the-person (Silva et al, 2014). Within the in-the-person category, there are equipment designed to be used inside human body, such as surgically implanted ones, subdermal applications or even ingested in the form of pills. These devices are used when less invasive approach is not applicable. Contrasting with the in-the-person category, there is off-the-person category. Devices on this category are designed to measure ECG without skin contact or with minimal skin contact. According to this category is aligned with future trends of medical application where pervasive computer systems are a reality. Examples of such equipment are the ones based on capacitive devices which measure the electric field changes induced by the body allowing ECG measurement at distance of 1cm or more even with clothing between the body and the sensor.

➢ *Arrhythmia Overview*

The Heart Rhythm Society defines arrhythmia as any change from the normal sequence of electrical impulses (HRS, 2018.) or abnormality in the timing or pattern of the heartbeat. Arrhythmias may cause the heart to beat too rapidly, too slowly or irregularly. They are common and may cause a wide variety of symptoms, such as a racing, skipping or fluttering sensation (called palpitations) in your chest. Cardiac arrhythmias also may cause lightheadedness, fainting, chest pain, shortness of breath, fatigue or no symptoms at all. Many types of arrhythmia are merely nuisances; other types may be serious problems because they cause the patient to develop heart failure, pass out or even die suddenly when the heart beats too slowly or too rapidly to pump blood to the body according to J. Hokins services (2014).The association of advanced medical instrumentation (AAMI), the recommended types of arrhythmia detectable by equipment or methods are 15 classes of arrhythmia, grouped into 5 super-classes which includes: Normal (*N*), Supraventricular ectopicbeat (*SVEB*), Ventricular ectopic beat (*VEB*), Fusion beat (*F*) and Unknown beat (*Q*). these five classes will be the types of arrhythmia for our new system to dragonize using ECG data.

## II. RELATED WORK

Application of machine learning in area of bio-signals and medical solutions is currently a widely researched topic in computer and information science; different authors and scholars have researched different ways to improve heart disease classification, with many focusing on arrhythmia.

Following their researches, different methods have been proposed and made available to develop automatic recognition and diagnosis of disease using ECG data. Some of this method include Self Organizing Maps (SOM), Support Vector Machines (SVM), Multilayer Perceptron (MLP), Markov Models, Fuzzy or Neuro-fuzzy Systems, decision level fusion, Convolution neural network and different procedures have been recommended to enhance execution (Raut et al, 2008) and (Asl et al, 2008).

Till date, a few analysts have made undertakings to apply learning ensembles to diagnosis cardiac beats. Various strategies have been introduced over the years for working up the automatic structures to absolutely order the ECG information.

(Savalia et al, 2018) Proposed distinguishing between several arrhythmias by using deep neural network algorithms such as multi-layer perceptron (MLP) and convolution neural network (CNN). They employed the use TensorFlow library from Google for deep learning and machine learning is used in python to acquire the algorithms proposed. The ECG databases accessible at PhysioBank.com and kaggle.com were used for training, testing, and validation of the MLP and CNN algorithms. They were able to classify various cardiovascular diseases with 88.7% accuracy for MLP and 83.5% for CNN.

(Osowski, Hoai, & Markiewicz, 2004) Proposed the use of support vector machine (SVM) working in the classification mode heartbeat recognition. they used two different preprocessing methods for feature generation. One method involves the higher order statistics (HOS) while the second the Hermite characterization of QRS complex of the registered electrocardiogram (ECG) waveform. Combining the SVM network with these preprocessing methods yields two neural classifiers, which have been combined into one final expert system. The combination of classifiers utilizes the least mean square method to optimize the weights of the weighted voting integrating scheme. The results of the performed numerical experiments for the recognition of 13 heart rhythm types on the basis of ECG waveforms confirmed the reliability and advantage of the proposed approach.

(Chen et al, 2019) Proposed a novel two-step predictive framework for ECG signal processing, where a global classifier recognizes severe abnormalities (red alarms) by comparing the signal against a universal reference model. The seemingly normal signal samples undergo a subsequent deviation analysis and yellow alarms are called by identifying mild and yet informative signal morphology distortions comparing to the learned patient-specific baseline that can be indicative of upcoming heart conditions. To facilitate an accurate deviation analysis, a controlled

ISSN No:-2456-2165

nonlinear transformation with optimized parameters was proposed to increase the symmetry of signals for different abnormality classes in the feature space. this method achieved a classification accuracy of 96.6% and provides a unique feature of predictive analysis by generating precaution warning messages about the elevated risk of heart abnormalities to take preventive actions according to physician orders.

(Favieiro & Balbinot, 2019) proposed an approach to aggregate the power of hybrid classifiers, the low noise susceptibility of the random forest and the robustness of paraconsistent logic to provide an intelligent treatment of contradictions and uncertainties in datasets which they called paraconsistent random forest. Computational results demonstrated that paraconsistent random forest could classify several databases with satisfactory accuracy in comparison with state-of-The-Art (Zhou et al., 2020) Proposed an attention-based ResNet processing of ECG data, they used MIT-BIH arrhythmia database and PTB Diagnostics datasets for validation of their model, they obtained accuracy as follows: in MIT datasets is 96.2%. And in PTB datasets the accuracy is 99.6%.

## III. MATERIALS AND METHODS

➢ *Methodology*

The object-oriented programming (OOP) methodology is adopted in this research; this methodology which is data-centric, provides program modeling, reusability, and type-checking. It describes a programming methodology which implements many object-oriented features with in a conventional programing environment. In this approach, problems are decomposed into a number of entities called objects, then builds data and functions around these entities.The object-oriented programming methodology adopted in this research is recommended for the development of artificial intelligence (AI) and expert systems; whose problems are often data centric with well understood requirements and clearly defined objective, thus can be broken down into classes and functions.

➢ *Analysis of the Existing System*

There are several existing systems in literature but we considered the work done by Savalia et al., 2018 who used the convolution neural networks and multilayer-perceptron algorithm with a number of hidden layers for its sequence-to-sequence learning task; both of which are central part of the deep neural network. The algorithm uses the rectifier linear unit (ReLU) activation tool in all convolution layers and was implemented using the GoogleTensorFlow library for deep learning. The MIT-BIH ECG database accessible at PhysioBank.com and Kaggle.com was used for the training and testing. To achieve their classification, first, the network reads the datasets, and then defines their features and labels. In the MLP algorithm, the labels will be arrhythmia and normal sinus, while in the CNN algorithm, the labels different arrhythmia diagnosable by the system.The next step is to encode the dependent variable the dataset labels for the deep network. As the dataset is categorical, containing different arrhythmia names as labels, it is mandatory to encode the dataset because the labels are not numerical and cannot be read directly by the algorithm.

In the following step, the TensorFlow data structures were defined for holding features, labels, which includes defining weights, biases, hidden layers, activation tools, filters, filter size, placeholders for inputs, and desired output. There is also another tensor defined to store trained model output. This was followed by implementation and training of the model with the training dataset. Once the network is trained, it will calculate how far the trained model's output is from the actual output. Then, the cross-entropy function will try to reduce this error to a minimum point. Once it reaches the minimum value, the trained model will give testing accuracy by performing training with a test dataset.

Figure1 shows the architecture of the existing system, which clearly explains the sequence-to-sequence system for the ECG arrhythmia detection.

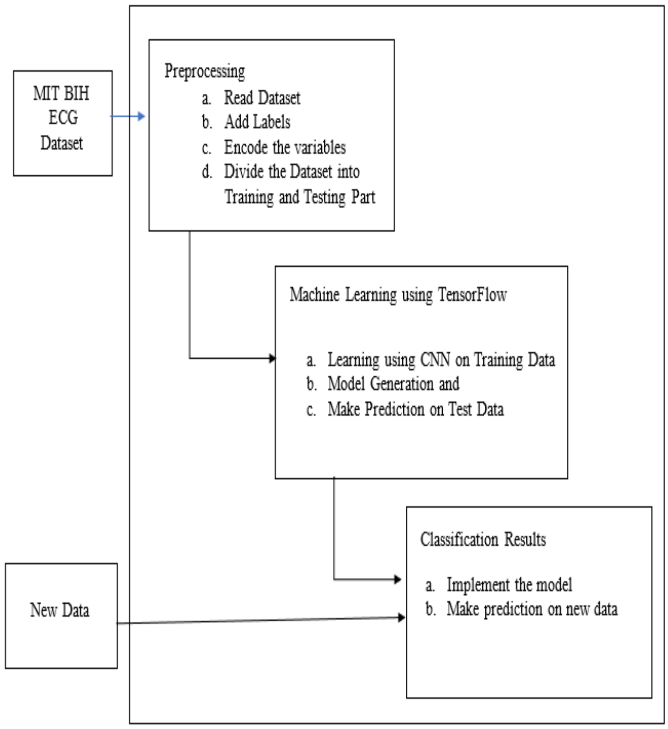


Figure 1: Architecture of the existing system (Adapted from Savalia et al., 2018)

➢ *Architecture of the Proposed System*

The proposed system architecture which show its conceptual model, structure and view shown Figure 2, reveals the various algorithms used in the design of the system which includes save predicted results to re-train the algorithm to improve classification results.

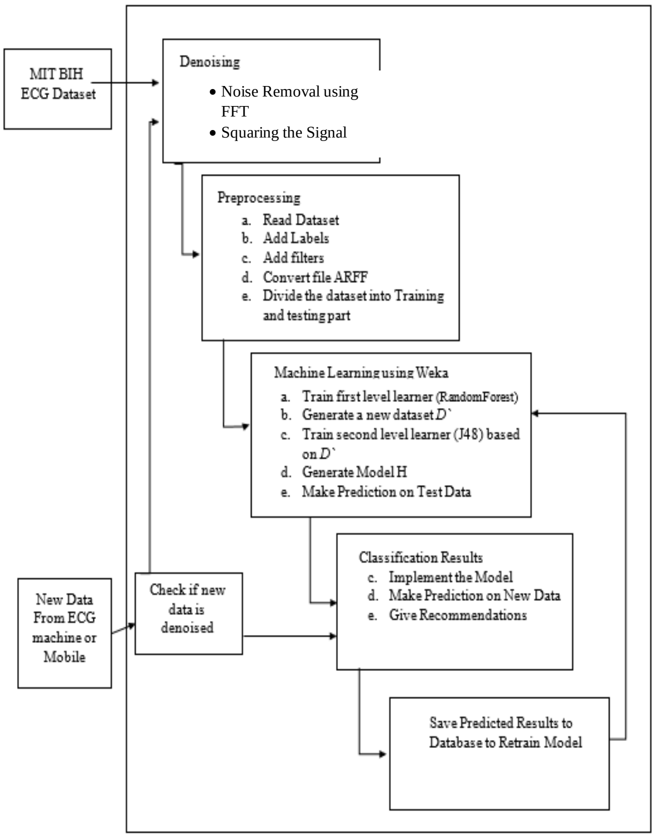


Figure 2: Architecture of the Proposed System

➢ *Architectural Details of the New System*

The details of each module in the new system as illustrated in figure 1 are explained as follows:

*1. ECG Dataset*

ECG signals from the MIT-BIH arrhythmia database (Kaggle, 2018)resampled at a frequency of 125Hz is used in training, testing and validation of the proposed system, Table1 summarizes the details of the dataset used for training our machine learning model.

**Table1: Summary of Dataset**

| | |
|---|---|
| Number of Samples: | 109,446 |
| Number of attributes | 188 |
| Number of Categories | 5 |
| Sampling Frequency | 125Hz |
| Data Source: | Physionet's MIT-BIH Arrhythmia Dataset |
| Classes: | 'N': 0, 'SVEB': 1, 'VEB': 2, 'F': 3, 'Q': 4 |

From our training dataset, we are provided with these five (5) classes of arrhythmia represented by numbers Normal (*N*): 0, 'SVEB *(S)*': 1, 'VEB (*V*)': 2, 'Fusion Beat (*F*)': 3, 'Unknown Beat (*Q*)': 4 as shown in table 1.

*2. ECG Data Resampling*

To improve the classification performance of the chosen classifiers, raw ECG data is squared and down sampled at 125Hz, which has already been done by the provider of the ECG dataset, but a necessary step for new ECG data to be classified,All data samples were cropped and padded with zeroes to fit a dimension of 188 columns. This is done to achieve a uniform signal independent of source and also remove noise artifacts.

*3. Preprocessing Module*

Before carrying out our experiments, the raw data needs to be processed to the format that computer can work with using Weka.

a. Adding attribute label row (1-187 and Class) to the csv data

Our original raw CSV data, does not have attribute label, to be able to import the data into Weka, we edit our CSV data using a text editor, add label data for each attribute, (1,2…187, Class) and then save it.

b. Apply numerical to nominal filter to the class column

The class attribute is a numerical data, but we need to convert it to a nominal data to aid our data visualization and get a confusion matrix (a better classification result).

c. Save the preprocessed Data to convert it to ARFF format.

*4. Machine Learning using Weka API*

In this module, we will input our trained hybrid algorithm model generated from our machine learning tool, Weka and stored on disk to make classification or predictions on new patients' data

*5. Classification Module*

In this module as shown in figure 2, the hybrid classification model built using our dataset and machine learning / data mining tool (Weka) will be provided for the application to make prediction on new patient data, This module of the application will also give expert level recommendations based on the output of the classification results from the new ECG patient data, it will also display statistics of the machine learning model and as well provide an interface to print the summary of the diagnosis reports and the patient data.

*6. Save Predicted Result*

Here Predicted Results are saved to system systematically, for easy retrieval and then for retraining our machine learning model, to improve the classification of our machine learning algorithm.

➢ *Algorithm of the New System*

The algorithm of the system explains the finite sequence steps taking to develop the new system, three machine learning algorithms was used for training our classification model, this is done to improve the accuracy of the model in predicting the results; these algorithms include one base-learner (Random Forest), one meta-learner (J48) and one ensemble algorithm (stacking). These algorithms were implemented using the Weka open source software program. The hybrid machine learning model generated will be used to develop the improved arrythmia detection system.

The algorithm steps are detailed as follows:

Step 1: Start
Step 2: Read the Dataset
Step 3: If dataset is denoised/Resampled
Add Labels
Else

Apply denoising algorithm
Step 4: Apply numerical-to-nominal filter for class label
Step 5: Divide Dataset to Training and Testing set
Step 6: Specify N learners (Random Forest, J.48, N=2)
Step 7: Train and Test dataset using learner n
Step 8: Produce the learning Model ($H_n$)
Step 9: if learner n = N
Then Step 10
Else Goto Step 7
Step 10: merge models ($\sum_1^N H_n$)
Step 11: Produce Best H
Step 12: Implement Model (to make prediction on test data or new data)
Step 13: Display Classification Result and Recommendations
Step 14: Save classified data to database and use to re-train model
Step 15: End

➢ *Flowchart of the New System*

A system flowchart diagram is a design that shows the process of flow of data form dataset for training to production of algorithm model to classification, display and storage of the data of classification result. This flowchart is shown on Figure 3 and 4.

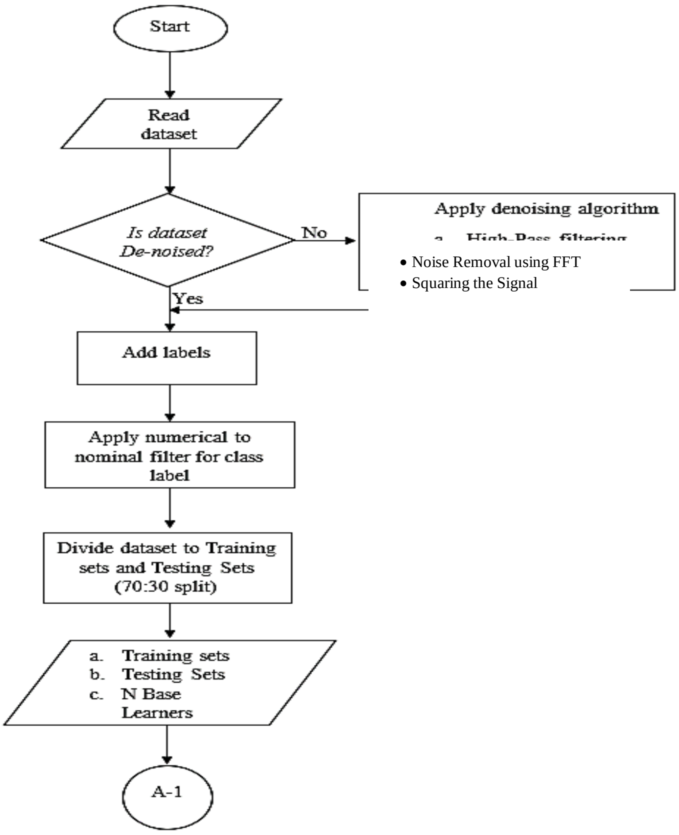


Figure 3: Flowchart of the New System.

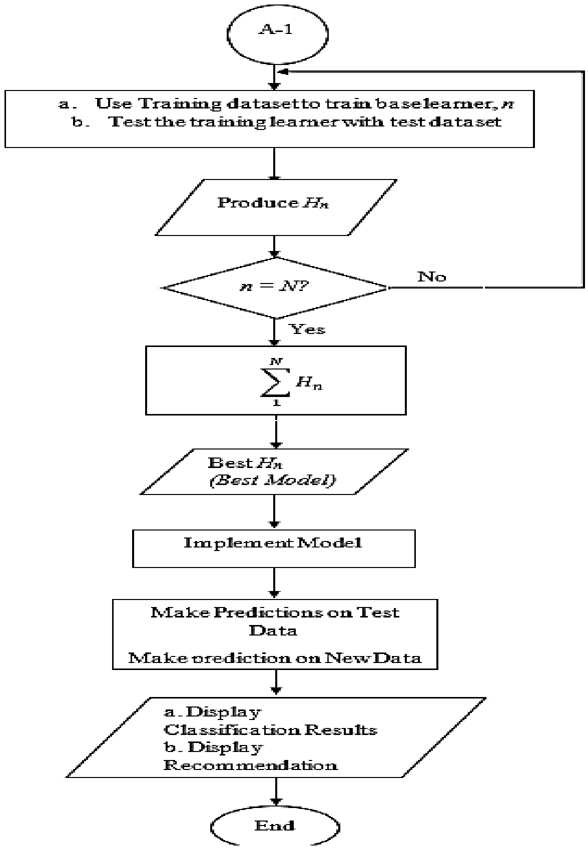


Figure 4: Flowchart of the New System.

➢ *UML Diagram the system*

Unified Modeling Language (UML) is a standard language for specifying, visualizing, constructing, and documenting the objects of software systems, the UML diagram visualizes and specifies the several classes that are interrelated and functions together as separate modules for the system to be implemented. Figure 5 shows the UML diagram of the system.

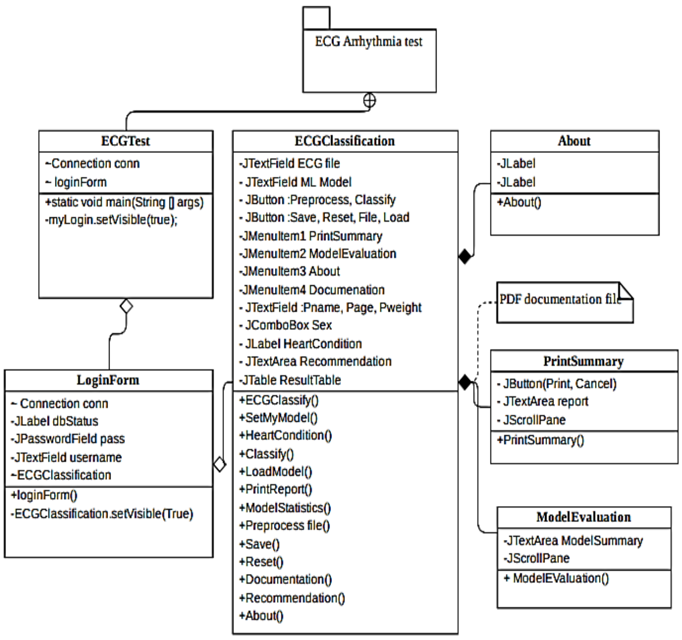


Figure 5: UML Diagram of the System

## IV. IMPLEMENTATION

Implementation here describes how the information systemwill be built which includes the hardware, software and user requirements; to ensure that the information system is operational, useable and as well meets the benchmarks stated by AAMI.

➢ *Hardware Requirements*

The system hardware requirements needed for the implementation for the ECG classification system are divided into two (2), first is the development system and second is the target system.

a.*Hardware Requirements for Development/Machine Learning System*

In implementing the project, experiments for machine learning algorithms, training, testing, creating the machine learning model and designing the software have a minimum requirement of Processor: Dual-core CPU with 1.2GHz or higher, RAM: 4.0 GB and above, Storage: 20GB free space and above

OS: Windows 7 or Higher, Ubuntu 14 or Higher or any comparable Linux distro etc.

However, the computer system we used in this research is the HP EliteBook 8460p with Intel Core i7 processor, 8Gb of RAM, 350Gb Hard Drive, 1Gb of VRAM and Windows 10 OS.

b.*Hardware Requirements for Target System*

Designing for fog or mobile devices such as Android device, Banana Pi or Raspberry Pi, may need special considerations which we will see in the next section. For this paper, our target device is the Raspberry Pi.

Device Category: Raspberry Pi, Operating system: Linux (Raspbian), CPU:4x ARM Cortex-A53, 1.2GHz, RAM: 1GB, Storage: microSD 16GB minimum, GPU: Broadcom VideoCore IV.

Our target device has some limited hardware requirements such as the 1GB ram and small 16GB microSD card. Thus, in our software design this limitation will be considered. Using machine learning models from the cloud providers, you simply rely on network connectivity and a cloud provider API to access models and make predictions. In our case of using edge devices to store prebuilt models, a different approach is required which make it very necessary to understand the limitations of our target device.

➢ *Software Requirements*

Just like the hardware requirements, there are separate minimum software requirements for the development system and the target system.

*Software Requirement for the Development System*

The minimal software requirement for the development system is as follows

Operating System: Microsoft Windows operating system (Windows 7 and higher), Mac OS or any recent 64-bit Linux distro with X window capability.

IDE: Java NetBeans 8 and above, Java Runtime, Environment JR7 and above, JDK: Java compiler or Development Kit version 8 and above

➢ *Target System Software Requirement*

Operating System: Raspbian Linux distro for raspberry pi from 2018, Java SE: Java SE embedded 7

➢ *Programming Language*

Java programming language is used to implement the hybrid machine learning technique in this research; Java is a class-based; object-oriented general-purpose language, designed in such a way it has few implementation dependencies. It's the most used programming language in software development for edge devices and IoT, it is the most popular for being cross-platform, thus you write once it runs on Windows, Mac, and Linux. The Apache NetBeans (integrated development environment) IDE version 11.2, is used for the java programming in this research.

➢ *Other Software Tools Used for Project Implementation*

Other software tools and Java APIs used for the project implementation are

a. Weka
b. Weka.jar
c. SQLite Studio
d. SQLite JDBC driver
e. Python

### *A. Weka (Waikato Environment for Knowledge Analysis)*

Weka (Waikato Environment for Knowledge Analysis) is an open-source software developed by the University of Waikato that provides tools for data preprocessing, implementation of several machine learning algorithms, and visualization tools so that you can develop machine learning techniques and apply them to real-world data mining problems.Weka is used for building the machine learning model for our software designed

### *B. Weka Java Application Program Interface (API)*

Weka Java API is a collection of java routines, data structures, object classes and variables making up all the algorithms and data one can use to implement Machine Learning classification and predictions in our java code.

### *C. SQLite JDBC Driver*

SQLite JDBC driver is a Java API library developed by Taro L. Saito for accessing and creating SQLite database file in Java software. It requires no configuration as native libraries of most common operating systems (Windows, Linux, Mac OSX and Android) are assembled to form a single JAR (Java archive). To use SQLite JDBC driver in our Java programming codes, we would need to download the JAR file and then add it to our class-path.

### *D.* Python

Python is used for denoising our data, which includes down sampling and squaring of the ECG signal file, it was used because of the limitation of Java programming language in signal manipulations and data science.

➢ *Application Screenshots*

The screenshots from the software develop are shown in figure 4 to 11

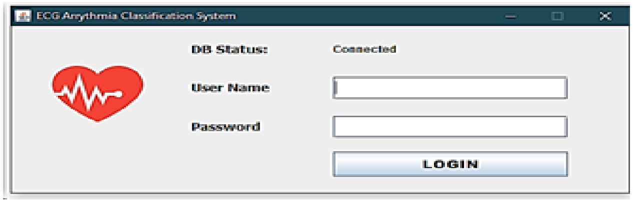


Fig 6: Login Screen of the Application.

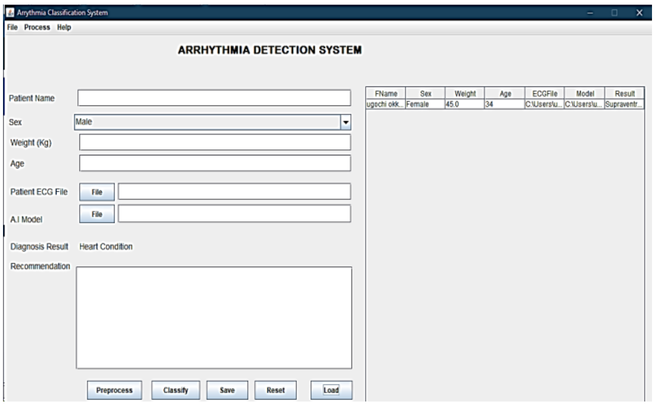


Figure 7: Screenshot of the Arrhythmia Detection System.

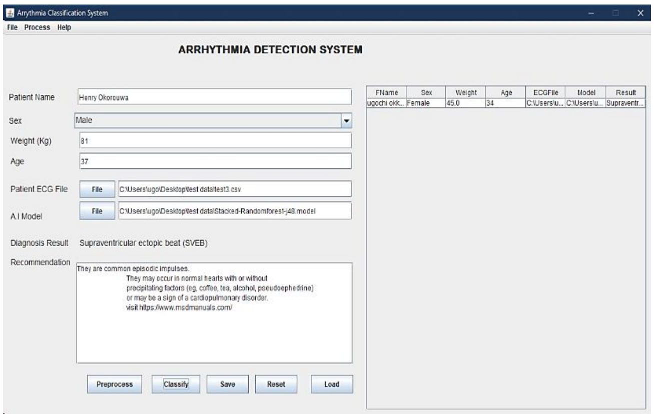


Figure 8: Screenshot of the Arrhythmia Detection System Showing Diagnosis Result

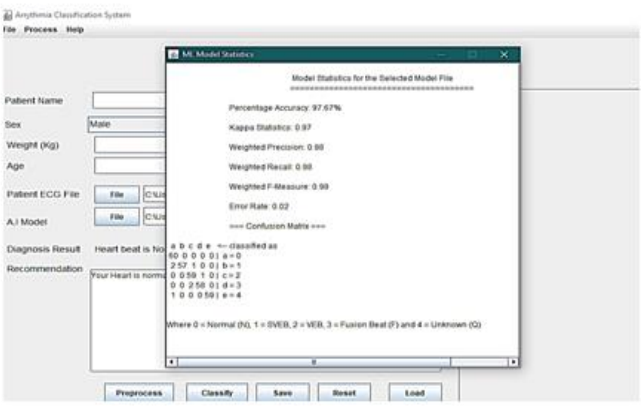

Figure 9: Screenshot of the Arrhythmia Detection System Showing Model Statistics of the selected model

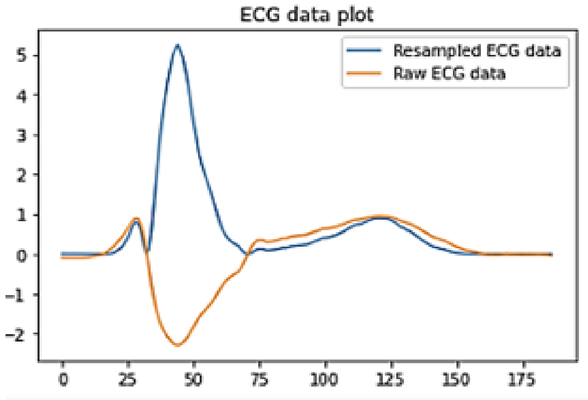


Figure 10:Sample Denoised Data for Classification - 1

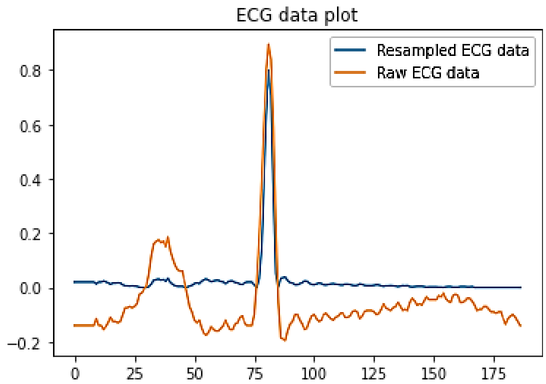


Figure 11:Sample Denoised Data for Classification - 2

## V. RESULTS AND DISCUSSION

The result produced by the system is gotten by testing 300 ECG patient data, 60 from each class that is 60 Normal (N), 60 Supraventricular ectopic beat (SVEB), 60 Ventricular ectopic beat (VEB), 60 Fusion beat (F) and 60 Unknown beat (Q) chosen at random from a test dataset. This is done to determine the accuracy and performance metrics of newly designed system using the hybrid algorithm and compare it to that of the algorithm used on the existing system as well as the individual algorithms combined to generate our hybrid algorithm.

➢ *Performance Metrics and Model Creation Factors for Edge Devices*

Measures recommended by the association for the advancement of medical instrumentation AAMI for evaluating 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; since the overall accuracy can be strongly distorted by the results of the majority class, taking into account the aforementioned performance metrics is very necessitous in making right choice of algorithm and not limiting to only accuracy.

Other factors considered during model creation for machine learning applications for edge devices (example Raspberry pi and mobile smartphones) are explained in table 2.

Table 2: Factors affecting machine learning application on mobile devices

| Factor | Priority | Reasons |
|---|---|---|
| Model training time | Low | Time taken to train models are important; nevertheless, as we deploy static models within applications at the edge like in our case, this priority is low because you can always apply more resources when necessary, possibly even in the cloud, to train the model. |
| Model test time | Medium | If an algorithm produces a complex model requiring relatively long testing times, this could result in latency or performance issues on the device when making predictions. |
| Model accuracy | High | Model accuracy must be sufficient to produce results required by your well-defined problem. |
| Model size | High | When deploying pertained Machine Learning models onto devices, the size of the model must be consisting with the memory and processing resources of the target device. |

For smartphones and Raspberry Pi devices, a good guideline for model size is 5MB -50MB. When considering larger classical machine learning models, its ideal to make certain a sufficiently greater accuracy to justify the larger size.

Table 3: Confusion Matrix of Classified Data from the New System Using the Hybrid Algorithm (Stacked Random Forest and J.48 Algorithm)

| Class | Normal (N) | SVEB | VEB | Fusion (F) | Unknown (Q) |
|---|---|---|---|---|---|
| N | 60 | 0 | 0 | 0 | 0 |
| SVEB | 2 | 57 | 1 | 0 | 0 |
| VEB | 0 | 0 | 59 | 1 | 0 |
| F | 0 | 0 | 2 | 58 | 0 |
| Q | 1 | 0 | 0 | 0 | 59 |

- *Accuracy, Model Size and Kappa Statistics of the Machine Learning Models*

Accuracy is one the metric for evaluating classification models defined as the fraction of predictions our machine learning model got right. Mathematically, accuracy has the following definition:

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

For binary classification, accuracy can also be calculated in terms of positives and negatives as follows:

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

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

Kappa Statistic is a measure that compares the accuracy of a given system to that of a random system.

$$Kappa = \frac{Total\ Accuracy - Random\ Accuracy}{1 - Random\ Accuracy}$$

Random Accuracy is described as the sum of the products of reference likelihood and result likelihood for each class, it is a hypothetical expected probability of agreement under an appropriate baseline constraint

$$Random\ Accuracy = \frac{Actual\ False * Predicted\ False + Acutal\ True * Predicted\ True}{Total * Total}$$

Or

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

Table 4 shows the accuracy, kappa statistics and model sizeof the designed system, results shows that the hybrid algorithm (Stacked Random Forest and J48) has an accuracy of 97.67% on the provided patients data, more so, the model size and kappa statistics is within the range allowed for a good machine learning for mobile application.

Table 4: Accuracy, Kappa Statistics and Model size of Algorithm

| Accuracy (%) | Kappa statistics | Model Size (Kb) |
|---|---|---|
| 97.67 | 0.97 | 38,150 |

- *Precision (Positive Predictivity) of the Machine Learning Models*

Precision or positive predictivity (+P) is the is defined as the number of true positives (Tp) over the number of true positives plus the number of false positives (Fp). High scores for precision indicate that the classifier is returning accurate results. Mathematically precision is given as

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

- *Recall (Sensitivity) of The Machine Learning Models*

Recall also known as True-positive rate or sensitivity (*Se*) is defined as the number of true positives (*Tp*) over the number of true positives plus the number of false negatives (*Fn*). High scores for recall indicate the trained model is performing well and returning a majority of all positive results.

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

- *F-Measure of the Machine Learning Models*

F-Measure or F1-score defines a balance between precision and recall, it is the weighted average or harmonic mean of precision and recall; and gives a better measure of the incorrectly classified cases than the accuracy metric. the F-Measure is important as it takes both the false negatives (FN) and false positives (FP) into consideration when describing the performance matric of a machine learning algorithm.

$$F1 = 2 * (\frac{Precision * Recall}{Precision + Recall})$$

**Table 5: Performance Indicators of the System**

| Class | Precision | Recall | F-Measure |
|---|---|---|---|
| N | 0.952 | 1 | 1 |
| SVEB | 1 | 0.95 | 0.95 |
| VEB | 0.952 | 0.983 | 0.983 |
| F | 0.983 | 0.967 | 0.967 |
| Q | 1 | 0.983 | 0.983 |
| Weighted average | 0.977 | 0.977 | 0.977 |

➢ *Result Comparison*

Comparing the developed system to the existing system, using the same dataset from Phyisonet.com and Kaggle.com the designed system shows an improved result in accuracy and other performance metrics as summarized in table 7

**Table 6: Result Comparison of The Existing and New System.**

| Parameters | Existing system | New System | Remark |
|---|---|---|---|
| Database | MIT-BIH arrhythmia database | MIT-BIH arrhythmia database | Same |
| Machine Learning tool | TensorFlow Library / Python | Weka API/ Java | |
| Algorithm | Convolution neural network (CNN) Checks if Arrhythmia is present or not /Multilayer Perceptron (MLP) (For Multi-class prediction) | Random Forest (RF) and J.48 ensembled with Stacking algorithm (For Multi-class prediction | |
| Accuracy achieved | 83.5% and 88.7% | 97.67% | Improved |

➢ *Discussion of Results*

As reported in table 4, the overall accuracy of the system is 96.67%, which showed an improved performance when compared to that of the existing system which had an overall accuracy of 83.5% and 88.7%. The system achieved a positive predictivity (precision) of 0.95, 1, 0.95, 0.98, 1 and 0.98 for N, SVEB, VEB, F and Q classes respectively, while for sensitivity (Recall) it achieved 1, 0.95, 0.98, 0.97 and 0.98 for N, SVEB, VEB, F and Q classes respectively.

In summary the system designed with the hybrid algorithm (stacked random forest and J48), performed excellently when compared to others in all performance metrics considered. Thus, it is reliable and more suitable for accurate medical diagnosis of arrhythmia heart disease.

## VI. CONCLUSION

In this research, we designed and implemented an automatic arrhythmia classification system using the hybrid machine learning algorithm studied, made up of Random Forest, J.48 and ensembled with Stacking algorithm. This study is performed on the reputable MIT-BIH ECG datasets from Phyisonet and Kaggle.com. The system was evaluated using the performance evaluation recommended by AAMI; demonstrated that the system designedperformed with a good accuracy of 97.67%, an approximate recall and precision value of 0.977, with a good model size of 38.2mb less than the maximum bench mark of 50mb after training thus is suitable for our edge device classification algorithm and arrhythmia diagnosis system.

## VII. FUTURE WORK

Further researches will be made on this work to be able to detect other subclasses of arrhythmia not included in this research, totaling the number of predictable class to fifteen (15) recommended by the AAMI; more researches will also be made on how to integrate this design on an ECG device for immediate prediction of the presence of arrhythmia.

## REFERENCES

[1]. Asl, B. M., Setarehdan, S. K., & Mohebbi, M. (2008). Support vector machine-based arrhythmia classification using reduced features of heart rate variability signal. , 44(1), 51-64.

[2]. Chen, J., Valehi, A., & Razi, A. (2019). Smart Heart Monitoring: Early Prediction of Heart Problems Through Predictive Analysis of ECG Signals. *IEEE Access*, *7*(December), 120831–120839. https://doi.org/10.1109/access.2019.2937875

[3]. Favieiro, G. W., & Balbinot, A. (2019). Paraconsistent Random Forest: An Alternative Approach for Dealing with Uncertain Data. *IEEE Access*, *7*, 147914–147927. https://doi.org/10.1109/ACCESS.2019.2946256

[4]. Heart Rhythm Society. (2018). Electrical System | Heart Rhythm Society. Retrieved June 25, 2019, from https://www.hrsonline.org/electrical-system

[5]. Kaggle. (2018). ECG Heartbeat Categorization Dataset | Kaggle. Retrieved July 9, 2020, from https://www.kaggle.com/shayanfazeli/heartbeat

[6]. Osowski, S., Hoai, L. T., & Markiewicz, T. (2004). Support Vector Machine-Based Expert System for Reliable Heartbeat Recognition. *IEEE Transactions on Biomedical Engineering*, *51*(4), 582–589. https://doi.org/10.1109/TBME.2004.824138

[7]. Oxford DNB. (2004, September 23). Waller, Augustus Désiré (1856–1922), physiologist. https://doi.org/10.1093/ref:odnb/38099

[8]. Savalia, S., & Emamian, V. (2018). Cardiac Arrhythmia Classification by Multi-Layer Perceptron and Convolution Neural Networks. *Bioengineering*, *5*(2), 35. https://doi.org/10.3390/bioengineering5020035

[9]. Silva, H.´acido, Carreiras, C., & Andr´e Lourenc¸o. (2014). *Off-the-Person Electrocardiography Off-the-Person Electrocardiography*. (September 2013).

[10]. Zhou, Y., Zhang, H., Li, Y., & Ning, G. (2020). *ECG Heartbeat Classification Based on ResNet and Bi-LSTM*. https://doi.org/10.1088/1755-1315/428/1/012014