Scalable and Upgradable AI for Detected Beat-By-Beat ECG Signals in Smart Health

I Hua Tsai
Department of Computer Science
Texas Tech University, Lubbock, TX
79409, USA
Student Member, IEEE

Bashir I. Morshed
Department of Computer Science
Texas Tech University, Lubbock, TX
79409, USA
Senior Member, IEEE

2023 IEEE World AI IoT Congress (AIIoT) | 979-8-3503-3761-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/AIIoT58121.2023.10174482

***Abstract*— The accurate and efficient classification of electrocardiogram (ECG) signals is crucial in identifying cardiac conditions for effective remote heart monitoring and telemedicine systems. However, the large-scale data from various ECG databases presents a challenge to traditional artificial intelligence (AI) algorithms. To address this issue, we use multi-stage classification as a promising method in sHealth. In this study, we aim to evaluate the performance and power consumption of various machine learning and deep learning algorithms in single-stage and multi-stage classification. Specifically, we analyzed beats combined from three different ECG databases and trained decision tree (DT), artificial neural network (ANN), support vector machine (SVM), Naïve Bayes, K-nearest neighbors (KNN), bagged tree, recurrent neural network (RNN), convolutional neural network (CNN), and long short-term memory (LSTM) algorithms using time series feature extraction library (TSFEL). By dividing the data into training and testing sets, we were able to obtain the best accuracy for each algorithm and evaluate their memory usage, CPU usage, and running time. Our study highlights the advantages of multi-stage classification with DT and ANN in accurately detecting cardiovascular diseases while also consuming less power, making it a scalable and upgradable approach. The algorithm developed through this research can be implemented in wearable devices as a pre-trained model to efficiently monitor heart health and detect potential cardiac issues. The use of such IoT devices for remote heart monitoring can significantly improve access to healthcare for patients in remote or underserved areas, allowing for early detection and treatment of cardiovascular diseases.**

Keywords- Multi-stage classification, Single-stage classification, Cardiac episodes, Unsupervised monitoring, Machine learning.

I. INTRODUCTION

Cardiovascular diseases (CVDs) are a major global health issue and monitoring heart disease is crucial in improving patient outcomes. With the rise of the Internet of Things (IoT), wearable devices and smart health (sHealth) applications have become increasingly popular for monitoring heart disease and promoting heart health. These devices and apps use sensors and algorithms to collect and analyze electrocardiogram (ECG) signals, which measure the electrical activity of the heart and can detect irregular heartbeats or other abnormalities. The integration of IoT technology into heart disease monitoring has revolutionized the way we approach cardiovascular health. With these devices and applications, patients can take control of their heart health and receive real-time feedback, leading to better management of heart disease and improved patient outcomes.

The concept of multi-stage classification has emerged as a promising approach to address the challenges of adapting AI models to new sensor data or decision-making processes in smart systems. Breaking down the AI model into multiple stages allows for independent modifications, making it more scalable and upgradable compared to single-stage classification. While single-stage classification can present challenges when changes occur, multi-stage classification enables a more flexible and adaptable approach. The benefits of multi-stage classification are demonstrated in the example of cardiac disease detection, showcasing its scalability and upgradability for smart health systems.

An ECG is a non-invasive medical test that measures the electrical activity of the heart. It's quick, safe, and comfortable for patients. The test captures and records the electrical signals generated each time the heart muscle contracts. By analyzing these signals, ECG can help healthcare professionals understand the heart's health and rhythm [2]. An electrocardiogram (ECG) can detect abnormal beats and discover type and origin [3-12]. Interpreting ECG signals is challenging and requires expertise [3]. Automatic heartbeat classification can help physicians analyze ECG recordings [4]. The Stationary Wavelets Transform and Support Vector Machine are used to extract features and detect heart conditions [5]. A study has been conducted on the performance of convolutional and deep neural networks on mobile and embedded platforms [6]. Gradient blending is used to address overfitting in multimodality and compute an optimal blend [7]

The use of LSTM and CNN algorithms for ECG feature analysis and classification is increasing due to advancements in ECG data and DNNs [8]. The performance of training deep learning models on mobile devices should be investigated in terms of memory consumption, hardware utilization, and power consumption [9]. Deep learning CNN and LSTM frameworks stack similar networks to generate a robust model [10]. Different methods are used for ECG feature extraction, then machine learning algorithms like Decision Trees, Random Forests, and Gradient Boosted Trees are applied for ECG classification [11]. Multiple CNN models can diagnose cardiac arrhythmias by transferring knowledge from deep learning models into one model using ECG signals [12].

This work explores the potential of multi-stage classification for data analysis, in order to identify limitations in traditional single-stage classification algorithms. The study focuses on detecting cardiac conditions in patients from single-lead ECG

Table I: The criteria of AAMI labeling class of MIT-BIH Database with the full database, and training and testing dataset.

AAMI Heartbeat class	**N**	**S**	**V**	**F**	**Q**	**Total**
Description	Normal beat	Supraventricular ectopic beat (SVEB)	Ventricular ectopic beat (VEB)	Fusion beat	Unknown beat	
Label	N, L, R	S, e, j, A, a, J	V, E	F	Q, /, f	
Full data *(Total 41 records)*	654,628	71,223	81,356	6,663	3912	817,782
Training ***(21 records)***	**298,177**	**31,925**	**46,017**	**1,746**	**1,403**	**379,268**
chfdb (chf01-05)	173,975	26,409	1,141	1,025	131	202,681
mitdb (100-109, 111-113)	17,253	5,231	7,354	552	1,054	31,444
ltdb (14046, 14157, 14184)	106,949	285	37,522	169	218	145,143
Testing ***(20 records)***	**356,451**	**39,298**	**35,339**	**4,917**	**2,509**	**438,514**
chfdb (chf06-10)	203,154	31,542	1,245	957	357	237,255
mitdb (114-119, 121-124, 200)	20,014	6,541	7,521	1,221	1,253	36,550
ltdb (14134, 14172, 14149, 15814)	133,283	1,215	26,573	2,739	899	164,709

data, using machine learning and deep learning algorithms. The aim is to classify different heartbeats with more precise training and testing data. The performance and power consumption of top-performing classifiers, such as decision tree and artificial neural network, are evaluated using statistical metrics. The feature vectors for classification are formed using the time series feature extraction library (TSFEL). The study also compares the effectiveness of multi-stage and single-stage classification in analyzing heartbeats and applying the results to wearable devices. The example of cardiac disease detection highlights the benefits of multi-stage classification in terms of scalability and the ability to upgrade. Table I shows the criteria of AAMI labeling class of MIT-BIH database with the full database, and training and testing dataset.

II. METHODOLOGY

This study utilized the MIT BIH database, which follows the criteria set by AAMI to categorize heartbeat types [1]. The five heartbeat categories in the MIT-BIH database include Normal and Bundle Branch Block beats (class N), Supraventricular Ectopic Beats (SVEB, class S), Ventricular Ectopic Beats (VEB, class V), Fusion beats (class F), and Unknown beats (class Q). Data was sourced from the open-source PhysioNet database, which was a combination of the BIDMC Congestive Heart Failure database, the MIT BIH Arrhythmia database, and the MIT BIH Long-term ECG database [13].

A. ECG data

In our study, we used data from three ECG databases: the BIDMC congestive heart failure database (chfdb), the MIT BIH arrhythmia database (mitdb), and the MIT BIH long term ECG database (ltdb). The chfdb contained two lead ECG signals with a 250 Hz sampling rate and a bandpass filter applied, and we used 10 records for our analysis. The mitdb contained two lead ECG signals with a 360 Hz sampling rate and a bandpass filter applied, and we used 24 records for our analysis. The ltdb contained two lead ECG signals with a 128 Hz sampling rate and a bandpass filter applied, and we used 7 records for our analysis. We divided the records into a training dataset (the 21 ECG data records) and a testing dataset (the remaining 20 ECG data records). We have 379,628 beats for training data, and 438,514 beats for testing data. Fig 1 shows the flowchart of ECG signal demonstrating machine learning algorithm for preprocessing, analysis, and classification. The classifier's performance is assessed using the training dataset, and the final evaluation of the heartbeat classification system is carried out on the test dataset.

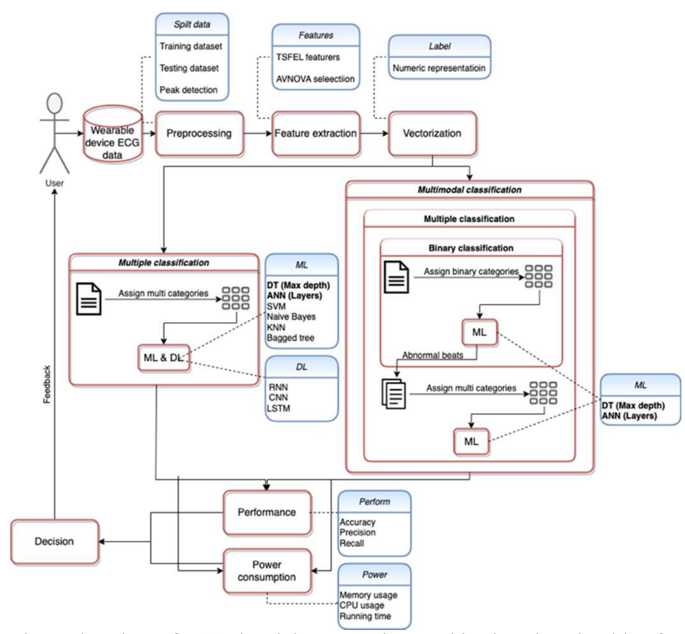


Fig 1: Flowchart of ECG signal demonstrating machine learning algorithm for preprocessing, analysis, and classification.

B. Signal preprocessing

In our work, we aim to analyze the human electrocardiogram (ECG) signal which is a weak physiological signal characterized by nonlinearity, non-stationarity, and strong randomness. The ECG signal is vulnerable to various types of noise and distortions, such as baseband drift, EMG interference, power frequency noise, and other noise interferences, which results in a low signal-to-noise ratio (SNR). These disturbances cause deformation of the ECG waveform and make it difficult for doctors to accurately interpret the signal. To tackle the problem of baseline drift, a low-pass filter is commonly used to remove the low-frequency noise in the ECG signal. Other traditional methods for removing baseline drift include median filtering, wavelet transform, averaging filtering, and EMD decomposition [14].

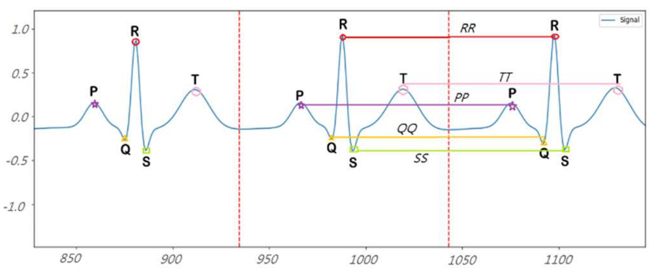


Fig 2: P, Q, R, S, T peaks in detail to form a heartbeat.

Table II: Summary the total of tsfel and ANOVA selection features.

Type of features	Features
Time series feature extraction library (Tsfel) (175)	Total 175 features be extracted
Analysis of variance test (ANOVA) Selected features (133)	signal distance, slope, wavelet energy, wavelet entropy, spectral centroid, ECDF percentile count, empirical cumulative distribution function (ECDF), root mean square, autocorrelation, max power spectrum, mean absolute deviation, spectral skewness, spectral roll-off, skewness, spectral decrease, median frequency, negative turning points, neighborhood peaks, peak to peak distance, wavelet standard deviation, wavelet variance, fast Fourier transform mean coefficient, power bandwidth, spectral distance, median absolute diff, median diff, spectral entropy, absolute energy, human range energy, Kurtosis, MEL cepstral coefficients (MFCC), maximum frequency, …etc..

C. Peak detection

R peak detection plays a crucial role in ECG heartbeat recognition. In this study, the Pan Tompkins algorithm was utilized to accurately detect the R peak [15]. The open-source code "The Python Toolbox for Neurophysiological Signal Processing" provided by neurokit 2 was used to implement the algorithm [16]. Once the R peak was detected, the RR interval and mean RR interval were calculated, which were then used to find the P, Q, S, T, and T' peaks. The algorithm detected three R peaks to form one heartbeat. The middle value between the first and second R peaks was used as the starting point, while the middle value between the second and third R peak was used as the ending point of the beat. Fig 2 shows P, Q, R, S, T peaks in detail to form a heartbeat.

D. Feature extraction

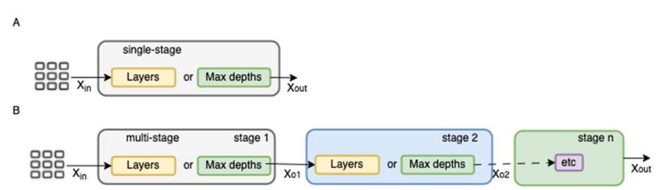
In this work, two ECG beats are processed at a time using the sliding window technique where each beat is processed and then slides to the next beat. A total of 175 features are extracted from these beats using the time series feature extraction library (TSFEL) in Python [17]. The feature selection process involves applying analysis of variance (ANOVA) algorithms, which results in the selection of 133 features for use in classification. ANOVA is a collection of statistical models and estimation procedures used to analyze differences in group means. Table II Summaries the total of tsfel and ANOVA selection features.

E. Classification

We categorize normal heartbeats (N) as 0, supraventricular ectopic heartbeat (S) as 1, ventricular ectopic beats (V) as 2 and fusion beat with unknown beat as 3. To classify the data, we use various machine learning algorithms such as Decision Tree (DT), Artificial Neural Network (ANN), Support Vector Machine (SVM), Naive Bayes, K-Nearest Neighbors (KNN), and Bagged Tree, as well as Deep Learning models such as Recurrent Neural Network (RNN), Convolutional Neural Network (CNN), and Long Short-Term Memory (LSTM). We employ 10-fold cross-validation technique for training and evaluation. Both single-stage multi-classification and two-stage multimodal classification methods are tested to obtain accurate results.

E. 1. Single-stage classification

We perform single-stage classification using one classifier to distinguish between four classes. We train machine learning models such as decision tree (DT), artificial neural network (ANN), support vector machine (SVM), Naive Bayes, K-nearest neighbors (KNN), bagged tree. As well as deep learning models like recurrent neural network (RNN), convolutional neural network (CNN), and long short-term memory (LSTM). We use 10-fold cross-validation to evaluate the performance of the classifier. We perform experiments with both DT and ANN, which perform best among the machine learning models, and adjust parameters to test their performance. The structure of the classifier is shown in Figure 3-A.


3-A: The structure of single-stage classification. 3-B: The structure of multi-stage classification
Fig 3: Present the structure of classifier.

E. 2. Multi-stage classification

In multi-stage classification, there are two classifiers involved in two-step classification. First, a binary classification is performed to distinguish normal from abnormal beats. The second step involves building a new classifier for multi-class classification, focused only on the three outlier beats. We conduct experiments using DT and ANN, which perform the best among ML models, and adjust parameters to assess performance. The classifier structure is depicted in Figure 3-B.

F. Performance

Various statistical metrics are used to assess model performance and evaluate selected TSFEL features, including accuracy, precision, and recall. Record refers to the correct identification of the proportion of true negatives. Precision represents recognition accuracy, while accuracy determines how closely a measured value corresponds to the true value. We also measure power consumption in terms of memory usage, CPU usage, and running time, based on the best accuracy achieved by ML and DL algorithms. Tensorflow's "Keras Model Profiler" package is used to obtain basic but critical information about model parameters and memory requirements on the GPU, while the "Psutil" package is used to retrieve information on running processes and system utilization, including CPU, memory, disks, network, and sensors, for system monitoring, profiling, and process management.

III. RESULTS

The Pan Tompkins algorithm is used to detect the R peak, and a sliding window with two beat intervals move one beat at a time until the end. The resulting peaks are labeled as P, Q, R, S, and T and are illustrated in Figure 4 for the interval from 1050 to 1250 at the apex. The blue line represents the P peak, the yellow line represents the Q peak, the green circle represents the R peak, the red line represents the S peak, and the purple line represents the T peak.

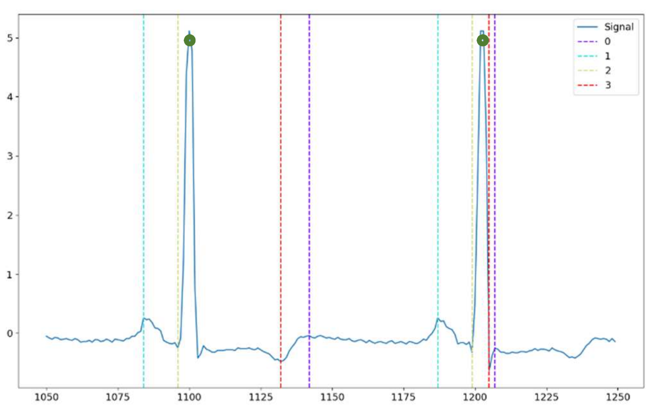


Fig 4 : P, Q, R, S, T peaks detected for 2 ECG beats

For the machine learning, DT got the overall accuracy 91%, overall precision and recall are 91% and 90%. ANN got the overall accuracy 96%, overall precision and recall are 97% and 98%. For deep learning, the performance of RNN, CNN, LSTM exceeds 97%, among which CNN performs best. CNN got the overall accuracy 99%, overall precision and recall are 99% and 99%. Table III shows the comparison of ML & DL performance of single-stage classification base on different classifiers.

Table IV shows the power consumption results for single-stage classifications using ML and DL. The ANN classifier in the ML model achieved a higher accuracy of 96%, but consumed more memory, CPU usage, and running time than DT, which had an accuracy of 91% with quicker running time and lower memory and CPU usage. Among the DL models, the CNN classifier showed the highest accuracy, less memory and CPU usage, and running time. The DL models, however, consumed more energy than the ML models.

Table III: Comparison of ML & DL performance for single-stage classification base on different classifiers

Data: chf01~ chf10, ltdb1~ ltdb7, mitdb100~ mitdb124 Feature vector: 133 x 675,358			
Single-stage classifier		Overall Accuracy (%)	Overall Precision / Recall (%)
Machine learning	DT(10)	91	91 / 90
	ANN(64)	96	97 / 98
	SVM	86	82 / 80
	Naïve Bayes	82	86 / 82
	KNN	70	68 / 73
	Bagged tree	88	80 / 84
Deep learning	*RNN*	98	97 / 97
	CNN	99	99 / 99
	LSTM	98	99 / 98

Table IV: Power consumption of single-stage classification for ML and DL

Data: chf01~ chf10, ltdb1~ ltdb7, mitdb100~ mitdb124 Feature vector: 133 x 675,358					
Single-stage classification	*Machine learning*		*Deep learning*		
Classifier	ANN(64)	DT(10)	RNN	CNN	LSTM
Accuracy (%)	96	91	98	99	98
Memory usage (MiB)	518.3	463.1	585.2	570.1	572.8
CPU usage (%)	5.5	1	19.6	17.5	19
Running time (s)	33.5	1.5	78	56	58

Table V : Summary of power consumption for ANN and DT single-stage classification based on different parameters.

Single-stage classification (one classifier, one step)				
Classifier	**Accuracy (%)**	**Memory usage (MiB)**	**CPU usage (%)**	**Running time (s)**
ANN (Layers)				
256 Increase	100	529.6	6.5	40
128	100	528.4	6.5	35
64 Standard	96	518.3	5.5	33.5
32	84	527.8	5.5	32
16	72	523.5	5.0	32
8	68	521.2	5.0	21
4	57	519.1	4.6	11
2	55	517.4	4.3	10
1 Decrease	45	516.3	4.0	6
DT (MaxDepth)				
25 Increase	100	463.2	1	1.6
24	100	463.4	1	1.6
23	100	463.6	1	1.6
20	99	463.3	1	1.6
15	96	463.7	1	1.5
10 Standard	91	463.1	1	1.5
6	86	456.5	1	1.6
2	83	463.4	1	1.2
1 Decrease	83	462.8	1	1.2

Table V summaries of power consumption for ANN and DT single-stage classification based on different parameters like layers and maximum depth. We vary the ANN parameters to find out what performance will be exhibited using different numbers of layers. We set 64 layers as standard, increasing and decreasing from 256 layers to 1 layer. The 128-layer and 256-layer ANN classifiers achieved 100% accuracy without much change in memory, CPU usage, and runtime. After 128 layers, the accuracy and power consumption are reduced, and the runtime becomes faster. We vary the DT parameters to find out what performance will be exhibited using different numbers of maximum depth. We set 10 max depth as standard, increasing and decreasing from 25 max depth to 1 max depth. The 25-max depth, 24-max depth and 23-max depth DT classifiers achieved 100% accuracy without much change in memory, CPU usage, and runtime. After 23-max depth, the accuracy and memory usage are reduced, and the CPU usage and runtime remain the same.

Table VI: Summary of power consumption for ANN and DT combination as multi-modal classification based on different parameters and arrangment.

Multi-stage classification (two classifiers, two steps)					
	First stage classification		**Second stage classification**		**Overall**
Classifier	**Accuracy (%)**	**Memory usage (MiB)**	**Accuracy (%)**	**Memory usage (MiB)**	**CPU usage (%) ; Running time (s)**
ANN&ANN (Layers)					
256 Increase	100	429.1	100	446.8	17 ; 87
128	100	428.0	100	444.1	16.5 ; 80
64 Standard	100	429.4	95	446.3	17.5 ; 77
32	100	428.8	90	445.4	14 ; 70
16	100	427.5	70	444.5	13.7 ; 64.5
8	100	427.1	58	444.2	13 ; 54
4	76	427.1	50	443.3	12 ; 34
2	75	427.4	48	444.1	11.5 ; 33.5
1 Decrease	75	426.2	47	442.8	16 ; 22
DT&DT (MaxDepth)					
25 Increase	100	383.9	100	373.3	1 ; 1.6
24	100	384.6	99	374.3	1 ; 1.6
23	100	383.9	98	373.4	1 ; 1.6
20	100	384.1	94	373.5	1 ; 1.6
15	100	384.5	84	373.8	1 ; 1.6
10 Standard	100	383.9	72	373.3	1 ; 1.6
6	100	384.7	59	374.2	1 ; 1.6
2	100	384.5	49	374.2	1 ; 1.6
1 Decrease	100	383.2	47	373.0	1 ; 1.6
Arrangement					
DT (25) & ANN (128))	100	385.4	100	440.2	16 ; 42.12
ANN (128) & DT (25)	100	452.3	100	381.2	12 ; 41.8

Table V summaries of power consumption for ANN and DT combination as multi-stage classification based on different parameters and arrangement. In multi-stage ANN&ANN classifier from 256 layers to 8 layers, we achieve 100% accuracy without much change in memory usage, but after 8 layers the accuracy decreases in the first stage. In 256 layers and 128 layers, we achieve 100% accuracy without much change in memory usage, but after 128 layers the accuracy decreases in the second stage. When layers are reduced, overall CPU usage increases and run times are faster. In multi-stage DT&DT classifier from 25 max depths to 1 max depth, we achieve 100% accuracy without much change in memory usage in the first stage. At 25 max depths we achieve 100% accuracy, but after 25 max depths the accuracy drops off in the second stage. When max depths are reduced, overall CPU usage and run times remain the same. We also select the most accurate classifier based on the changed parameters to combine the classifier as ANN(128)&DT(25) or DT(25)&ANN(128). When the first stage is 25 max depth DT and the second stage is 128 layers ANN, we get 100% accuracy in both first and second stages, low memory usage for the first stage, but high-power consumption in second stage. The overall CPU usage and elapsed time are also higher than the other sort order.

IV. DISCUSSION

In our study, we evaluated the performance of machine learning and deep learning models using equal amounts of data. Our results indicate that deep learning models have higher memory usage, runtime, and CPU usage than machine learning models. Although the accuracy of deep learning models improves as the amount of data increases, they are more complex than machine learning models. Interestingly, we found that single-stage classification outperforms multi-stage classification in terms of running time, as it generates data with only one model and runs fewer layers. We also discovered that normalizing the data effectively can significantly improve running time. Moreover, our findings suggest that single-stage classification is faster than multi-stage classification, as it is less prone to overfitting and demonstrates differing rates of overfitting and generalization among different modalities.

V. CONCLUSION.

Our study compares the performance of single-stage and multi-stage classification methods for detecting ECG heartbeats using machine learning and deep learning algorithms. We use publicly available ECG data from various databases and analyzes it using time series feature extraction library (TSFEL) to form feature vectors. We find that multi-stage classification is beneficial for handling large-scale data, as it allows for independent modification of each stage, making the model more scalable and upgradable. This approach helps optimize specific stages without affecting others, leading to more efficient power consumption and better accuracy. We examine the performance of various machine learning algorithms such as decision tree, artificial neural network, support vector machine, Naive Bayes, K-nearest neighbors, and bagged tree, and deep learning algorithms such as recurrent neural network, convolutional neural network, and long short-term memory. The parameters of these algorithms are varied to determine the best performance. The results of our study show that multi-stage classification is more efficient than single-stage classification. In single-stage classification with ANN, reducing the number of layers can save power but not improve accuracy. Increasing the number of layers can improve accuracy but consume more power. In single-stage classification with DT, changing the maximum depth has little effect on power consumption but can improve accuracy. Increasing the maximum depth can improve accuracy without increasing power consumption. For multi-stage classification, the study uses DT and ANN to test performance and vary parameters. In multi-stage classification with ANN, reducing the number of layers can save power but not improve accuracy. In multi-stage classification with DT, changing the maximum depth has little effect on power consumption and accuracy in the first step, but reducing the maximum depth can decrease accuracy in the second step. Our study shows that multi-stage classification can improve the performance and efficiency of AI models for detecting heart conditions, and the findings have implications for the development of smarter and more efficient healthcare systems.

ACKNOWLEDGMENT

This material is based upon work supported by the National Science Foundation under Grant No. 2105766.

REFERENCES

[1] ANSI/AAMI. Testing and Reporting Performance Results of Cardiac Rhythm and ST Segment Measurement Algorithms, 1998. Association for the Advancement of Medical lnstrumentation.

[2] Udyavara Rajendra Acharya, Yuki Hagiwara, Joel En Wei Koh, Shu Lih Oh, Jen Hong Tan, Muhammad Adam, Ru San Tan, “Entropies for automated detection of coronary artery disease using ECG signals: A review,” Biocybernetics and Biomedical Engineering, Volume 38, no. 2, pp. 373-384, 2018.

[3] U. R. Acharya, Y. Hagiwara, J. E. W. Koh, S. L. Oh, J. H. Tan, M. Adam, and R. S. Tan, ``Entropies for automated detection of coronary artery disease using ECG signals: A review," Biocybern. Biomed. Eng., vol. 38, no. 2, pp. 373 384, 2018.

[4] Z. Zhang, J. Dong, X. Luo, K.-S. Choi, and X. Wu, “Heartbeat classi cation using disease-speci c feature selection,” Comput. Biol. Med., vol. 46, pp. 79 89, Mar. 2014.

[5] Asgari S, Mehrnia A, Moussavi M. Automatic detection of atrial fibrillation using stationary wavelet transform and support vector machine. Comput Biol Med. 2015 May;60:132-42. doi: 10.1016/j.compbiomed.2015.03.005. Epub 2015 Mar 14. PMID: 25817534.

[6] Nicholas D. Lane, Sourav Bhattacharya, Petko Georgiev, Claudio Forlivesi, and Fahim Kawsar. 2015. An Early Resource Characterization of Deep Learning on Wearables, Smartphones and Internet-of-Things Devices. In Proceedings of the 2015 International Workshop on Internet of Things towards Applications (IoT-App '15). Association for Computing Machinery, New York, NY, USA, 7–12.

[7] Wang, W., Tran, D., & Feiszli, M. (2019). What Makes Training Multi-Modal Classification Networks Hard? *ArXiv* N.

[8] S. Liaqat, K. Dashtipour, A. Zahid, K. Assaleh, K. Arshad, and N. Ramzan, “Detection of Atrial Fibrillation Using a Machine Learning Approach,” Information, vol. 11, no. 12, p. 549, Nov. 2020, doi: 10.3390/info11120549.

[9] Irfan, S.; Anjum, N.; Althobaiti, T.; Alotaibi, A.A.; Siddiqui, A.B.; Ramzan, N. Heartbeat Classification and Arrhythmia Detection Using a Multi-Model Deep-Learning Technique. *Sensors* **2022**, *22*, 5606.

[10] Liu, J., Liu, J., Du, W., & Li, D. (2019). Performance Analysis and Characterization of Training Deep Learning Models on Mobile Devices. *ArXiv*.

[11] Alarsan, F.I., Younes, M. Analysis and classification of heart diseases using heartbeat features and machine learning algorithms. J Big Data 6, 81 (2019). https://doi.org/10.1186/s40537-019-0244-x.

[12] Hammad, M.; Meshoul, S.; Dziwiński, P.; Pławiak, P.; Elgendy, I.A. Efficient Lightweight Multimodel Deep Fusion Based on ECG for Arrhythmia Classification. *Sensors* **2022**, *22*, 9347.

[13] Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PCh, Mark RG, Mietus JE, Moody GB, Peng C-K, Stanley HE. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. Circulation 101(23):e215-e220 Circulation Electronic Pages; http://circ.ahajournals.org/content/101/23/e215.full]; 2000 (June 13).

[14] M. M. R. Momota and B. I. Morshed, "Inkjet Printed Flexible Electronic Dry ECG Electrodes on Polyimide Substrates Using Silver Ink," 2020 IEEE International Conference on Electro Information Technology (EIT), 2020, pp. 464-468, doi: 10.1109/EIT48999.2020.9208322.

[15] Pan and W.J. Tompkins, “A Reaf-time QRS Detection Algorithm,” Ieee Transactions on Biomedical Engineering, vol. 32, no. 3, 1985, pp. 230-236.

[16] Makowski, D., Pham, T., Lau, Z.J. et al. NeuroKit2: A Python toolbox for neurophysiological signal processing. Behav Res 53, 1689–1696 (2021). https://doi.org/10.3758/s13428-020-01516-y.

[17] Barandas M., Folgado D., Fernandes L., Santos S., Abreu M., Bota P., Liu H., Schultz T., Gamboa H. Tsfel: Time series feature extraction library SoftwareX, 11 (2020), Article 100456, 10.1016/j.softx.2020.100456.