REPUBLIC OF TURKEY
ADANA ALPARSLAN TÜRKEŞ SCIENCE AND TECHNOLOGY UNIVERSITY

INSTITUTE OF GRADUATE EDUCATION
DEPARTMENT OF COMPUTER ENGINEERING

HYPER-PARAMETER OPTIMIZATION OF DEEP NEURAL NETWORKS WITH METAHEURISTIC ALGORITHMS

MUSTAFA EVREN KIYMAÇ
MASTER OF SCIENCE

ADANA 2022

REPUBLIC OF TURKEY
ADANA ALPARSLAN TÜRKEŞ SCIENCE AND TECHNOLOGY UNIVERSITY

INSTITUTE OF GRADUATE EDUCATION
DEPARTMENT OF COMPUTER ENGINEERING

HYPER-PARAMETER OPTIMIZATION OF DEEP NEURAL NETWORKS WITH METAHEURISTIC ALGORITHMS

MUSTAFA EVREN KIYMAÇ
MASTER OF SCIENCE

SUPERVISOR
ASST. PROF. DR. YASİN KAYA

ADANA 2022

ABSTRACT

HYPER-PARAMETER OPTIMIZATION OF DEEP NEURAL NETWORKS WITH METAHEURISTIC ALGORITHMS

Mustafa Evren KIYMAÇ
Department of Computer Engineering
Supervisor: Asst. Prof. Dr. Yasin KAYA
July 2022, 39 pages

Cardiac arrhythmias indicate cardiovascular disease, which is the leading cause of mortality worldwide, and can be detected by an electrocardiogram (ECG). Automated deep learning methods have been developed to overcome the disadvantages of manual interpretation by medical experts. The performance of artificial neural networks strongly depends on hyperparameter optimization (HPO), and this NP-hard problem is suitable for metaheuristic (MH) methods. In this study, a novel method is proposed for the HPO of a convolutional neural network (CNN) arrhythmia classifier using an MH algorithm. The approach utilizes our variant of an MH method, named the memory-enhanced artificial hummingbird algorithm, which has an additional memory unit that stores the evaluations of the solutions and reduces the computation time significantly. The study also proposes a novel fitness function that considers both the accuracy rate and the total number of parameters of each candidate network. Experiments were conducted on raw ECG samples from the MIT-BIH arrhythmia database. The proposed method was compared with three other MH methods and achieved equal or outperforming results, with classification accuracy reaching 98.87%. The proposed method yielded promising results in finding a high-performing solution with relatively lower complexity.

Keywords: arrhythmia classification, ECG, deep learning, CNN, hyperparameter optimization, artificial hummingbird algorithm

ÖZET

META-SEZGİSEL ALGORİTMALAR İLE DERİN SİNİR AĞLARININ HİPER-PARAMETRE EN İYİLEŞTİRMESİ

Mustafa Evren KIYMAÇ
Bilgisayar Mühendisliği Anabilim Dalı
Danışman: Dr. Öğr. Üyesi Yasin KAYA

Temmuz 2022, 39 sayfa

Kalp ritim bozuklukları (aritmiler), dünya çapında önde gelen ölüm nedeni olan kalp-damar hastalığının bir göstergesidir ve bir elektrokardiyogram (EKG) ile saptanabilir. Tıp uzmanlarınca elle yorumlamanın elverişsizliklerinin üstesinden gelmek için otomatik derin öğrenme yöntemleri geliştirilmiştir. Yapay sinir ağlarının başarımı, büyük ölçüde hiper-parametre en iyileştirmesine (HPEİ) bağlıdır ve bu NP-zor sorun, meta-sezgisel (MS) yöntemler için uygundur. Bu çalışmada, bir MS algoritması kullanan bir evrişimsel sinir ağı (ESA) aritmi sınıflandırıcısının HPEİ'si için yeni bir yöntem önerilmiştir. Yaklaşım, çözümlerin değerlendirmelerini saklayan ve hesaplama süresini önemli ölçüde azaltan ek bir bellek birimine sahip, bellek-geliştirmeli yapay sinek kuşu algoritması adlı bir MS yöntem varyantımızı kullanmaktadır. Çalışma ayrıca, her aday ağın hem doğruluk oranını hem de toplam parametre sayısını göz önüne alan yeni bir uygunluk işlevi önermektedir. Deneyler, MIT-BIH aritmi veri tabanından alınan işlenmemiş EKG örnekleri üzerinde gerçekleştirildi. Önerilen yöntem, üç başka MS yöntemiyle karşılaştırıldı ve %98,87'ye ulaşan sınıflandırma doğruluğu ile eşit ya da daha iyi başarım gösteren sonuçlar elde etti. Önerilen yöntem, göreceli olarak daha düşük karmaşıklık ile yüksek başarımlı bir çözüm bulma konusunda umut verici sonuçlar vermiştir.

Anahtar Kelimeler: aritmi sınıflandırma, EKG, derin öğrenme, ESA, hiper-parametre en iyileştirmesi, yapay sinek kuşu algoritması

I dedicate this study to my beloved family for their endless support …

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Asst. Prof. Dr. Yasin KAYA, for providing guidance and feedback throughout my thesis study. I also want to express my gratitude to my family and friends for their support in my studies.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

AAMI	: Association for the Advancement of Medical Instrumentation
ABC	: Artificial Bee Colony
AHA	: Artificial Hummingbird Algorithm
ANN	: Artificial Neural Network
BN	: Batch Normalization
BSA	: Backtracking Search Optimization Algorithm
CAD	: Computer-aided Diagnosis
CNN	: Convolutional Neural Network
CRO	: Coral Reef Optimization
CS	: Cuckoo Search
CV	: Cross-validation
CVD	: Cardiovascular Disease
DBM	: Deep Boltzmann Machine
DBN	: Deep Belief Network
DE	: Differential Evolution
DL	: Deep Learning
DNN	: Deep Neural Network
DO	: Dropout
DS	: Down-sampling
ECG	: Electrocardiogram
FA	: Firefly Algorithm
FN	: False Negative
FP	: False Positive
GWO	: Grey Wolf Optimizer
HP	: Hyperparameter
HPO	: Hyperparameter Optimization
HS	: Harmony Search
LSTM	: Long Short-term Memory
MAHA	: Memory-enhanced Artificial Hummingbird Algorithm

MH	: Metaheuristic
MIT-BIH	: Massachusetts Institute of Technology, Beth Israel Hospital
ML	: Machine Learning
MPA	: Marine Predators Algorithm
PSO	: Particle Swarm Optimization
RF	: Random Forest
ReLU	: Rectified Linear Unit
RMSE	: Root Mean Square Error
RNN	: Recurrent Neural Network
SA	: Simulated Annealing
SCA	: Sine Cosine Algorithm
SVM	: Support Vector Machine
TGA	: Tree Growth Algorithm
TN	: True Negative
TP	: True Positive
WSN	: Wireless Sensor Network

1. INTRODUCTION

Cardiovascular disease (CVD) is disorders of the heart and blood vessels. Unfortunately, CVD is the leading cause of mortality worldwide, accounting for about one-third of all deaths. More than four out of five incidents are due to heart attacks and strokes, and one-third of these occur prematurely in people under the age of 70. Thus, it is extremely vital to detect CVD as early as possible so that their treatment can begin (World Health Organization, 2022, May 25; Kaya, 2021).

A common indicator of CVD is a cardiac arrhythmia which is a problem with the rate or rhythm of the heartbeats. During an arrhythmia, the heart may beat too fast, too slow or with an irregular pattern. It is caused by problems with the electrical system that regulates the heartbeats, or with the circulation or structure of the heart (Heart Rhythm Society, 2022, May 25; Mohebbanaaz et al., 2022). It does not usually cause any symptoms. If the condition is left untreated, the heart may not be able to pump enough blood to the body, which can cause damage to the heart, brain, or other organs (National Heart, Lung, and Blood Institute, 2022, May 25a).

The most common test used to diagnose an arrhythmia is the electrocardiogram (ECG) which is a simple, painless, and non-invasive test that detects and records the electrical activity of the heart. An ECG plots the amplitudes and intervals of the electrical impulses in the heart. Therefore, the rate and rhythm of the heartbeats can be interpreted as steady or irregular with proper examination of these values (National Heart, Lung, and Blood Institute, 2022, May 25b).

Arrhythmia cases can be divided into two groups: life-threatening and non-life-threatening. In the Association for the Advancement of Medical Instrumentation (AAMI) standard, non-life-threatening arrhythmias have five main classes: normal (N), supraventricular ectopic (S), ventricular ectopic (V), fusion (F) and unknown (Q) (Association for the Advancement of Medical Instrumentation, 2012; Ramasamy et al., 2022). Each group has a different implication and needs to be treated differently. Therefore, correct identification of the type of abnormality is of utmost importance before treatment can be given.

Due to the very high variance in morphology, accurate identification of ECG waveforms is complicated. In practice, the current standard is visual assessment of the plots, which is subjective to the observer. To overcome the disadvantages of manual interpretation of the ECG, intensive work is being done to develop computer-aided diagnosis (CAD) systems for automation using machine learning (ML) methods (Acharya et al., 2017).

ML is the use of algorithms to analyze data and then make decisions and predictions about real-world events. Commonly used ML algorithms include k-nearest neighbors (Ramasamy et al., 2022; Kaya, 2021; Kaya & Pehlivan, 2015a), decision trees (Mohebbanaaz et al., 2022), random forests (RF) (Asadi et al., 2021), support vector machines (SVM) (Kaya & Pehlivan, 2015a; Houssein et al., 2021; Geweid & Chen, 2022; Nguyen et al., 2021), and artificial neural networks (ANN) (Kaya, 2021; Zeng et al., 2021; Kaya & Pehlivan, 2015b). Previous studies used traditional methods that relied on feature extraction and selection performed by experts. This manual approach had numerous drawbacks, such as susceptibility to subjective bias of the researcher and high labor and time costs.

A subfield of ML called deep learning (DL) has emerged based on the substantial advancement in computing hardware, development of specialized algorithms, and immense amounts of data becoming available in various domains. In this approach, necessary features are gathered directly from the raw data and in the classifier itself, as opposed to traditional methods (Murat et al., 2020).

The main model used in DL is the deep neural network (DNN) which is an ANN with more than one hidden layer between the input and output layers. DNNs are significantly better at handling large-scale, high-dimensional data and are more robust and resilient to data noise compared to traditional ML methods (Chen et al., 2022). The convolutional neural network (CNN) is the most common type of DNN used in arrhythmia classification to distinguish between ECG waveform morphologies of heartbeats (Nguyen et al., 2021; Kumar et al., 2022; Dey et al., 2021; Petmezas et al., 2021).

The performance of a neural network is highly dependent on the configuration of its hyperparameters (HP), and there is no general method for determining the optimal set of them. The process of finding the right set of HPs for a target task is both computationally intensive and time-consuming and requires a high level of knowledge and experience in addition to a trial-and-error approach (Yang & Shami, 2020). The problem belongs to the NP-hard class; therefore, deterministic methods are not suitable for it. However, stochastic methods such as metaheuristic (MH) algorithms have proven successful in solving NP-hard optimization problems (Kılıç et al., 2021).

In light of all the points described above on this subject, our work focuses on hyperparameter optimization (HPO) of a CNN model using an MH algorithm for arrhythmia classification on ECG signals. The main contributions of this study can be summarized as follows:

1. An automated HPO method and fitness function for high performance and low complexity was proposed for a CNN arrhythmia classifier using raw ECG signals as input.
2. The proposed method uses our novel variant of a recent MH algorithm to search a vast structural and operational HP configuration space of about 21 million possible combinations, which is not feasible with deterministic approaches due to computational and time constraints.
3. The method was evaluated using 48,507 heartbeat samples with five AAMI classes from the MIT-BIH arrhythmia database and compared with three different MH algorithms.
4. The experiments yielded promising results in finding optimized networks for arrhythmia classification compared with other state-of-the-art approaches.

The rest of this study is organized as follows: Section 2 reviews related work in the literature, Section 3 explains the material and method of the study, Section 4 gives the results of the experiments and the discussion of them, Section 5 states the conclusion, and Section 6 gives recommendations for future study.

2. LITERATURE REVIEW

There are numerous studies in the literature that use various MH methods to optimize HPs of DNNs in an automated manner. These MH methods include: the simulated annealing (SA) (Kirkpatrick et al., 1983), particle swarm optimization (PSO) (Kennedy & Eberhart, 1995), harmony search (HS) (Geem et al., 2001), cuckoo search (CS) (Yang & Deb, 2009), firefly algorithm (FA) (Yang, 2009), backtracking search optimization algorithm (BSA) (Civicioglu, 2013), differential evolution (DE) (Storn & Price, 1997), coral reef optimization (CRO) (Salcedo-Sanz et al., 2014), tree growth algorithm (TGA) (Cheraghalipour et al., 2018), artificial bee colony (ABC) (Karaboga & Basturk, 2007), marine predators algorithm (MPA) (Faramarzi et al., 2020), and grey wolf optimizer (GWO) (Kumar et al., 2022; Mirjalili et al., 2014). In these studies, different types of DNNs such as the plain DNNs, recurrent neural networks (RNN) (Luo et al., 2021), long short-term memory (LSTM) networks (Petmezas et al., 2021), deep belief networks (DBN), deep Boltzmann machines (DBM) (Pandey et al., 2021), and CNNs (Nguyen et al., 2021; Kumar et al., 2022; Dey et al., 2021) were utilized for classification tasks in many domains.

Tsai et al. (2020) used the SA metaheuristic algorithm to optimize the number of neurons in the hidden layers of a fully-connected DNN. They used the DNN to predict the number of bus passengers on a bus at the stops of selected routes in the Taichung city of Taiwan. They determined all other hyperparameters of the DNN ahead of the optimization process. The number of hidden layers was set to three, the activation function was set as rectified linear unit (ReLU), etc. The input data for the study was a dataset provided by the Taichung smart transportation big data research center that gives information about the routes, stops, buses, and passengers of the transportation system. A pre-processing was done on the dataset comprising of a feature selection, and the ten most influential features on the number of passengers were selected.

Their proposed algorithm started with generating an initial solution of a vector of three integers representing the number of neurons in each hidden layer. These integers were chosen from a selected interval of [5, 50]. The solutions were used for the construction of candidate DNNs. The candidate DNNs were evaluated for their performance using the root mean square error

(RMSE) metric. After the decision about its quality, the current solution was either kept as the best solution or got discarded. The solution vector's elements were changed in each iteration, using a neighbor selection operator with a number selected from a normal distribution taking mean as the previous value of element and standard deviation as 1. The iterations continued until the pre-determined termination criterion was met, which was selected in the study as the total number of iterations being 5.

They made a comparison of their experimental results in terms of RMSE and prediction time, with traditional forecasting methods of SVM, RF, and extreme gradient boosting; and other deep learning methods of a standard DNN and a DNN with dropout. The results showed that, their proposed method achieved the lowest RMSE and prediction time values among the methods compared. Their method had nearly 4% lower RMSE and 15% lower prediction time from the standard DNN.

Elmasry et al. (2020) utilized the PSO algorithm to optimize a DNN, LSTM, and DBN for computer network intrusion detection. The number of hidden layers and their number of nodes, dropout rate, batch size, learning rate, momentum, decay, activation function, optimizer, initialization function, and number of epochs were optimized. Two common intrusion detection system databases were used in the experiments. The proposed method outperformed the previous studies on the subject. Nevertheless, the method achieved a 4-6% increase in detection rate and a 1-5% decrease in false alarm rate using feature selection with PSO.

Passos & Papa (2020) implemented the CS, FA, BSA, and variants of the HS, PSO, and DE for the HPO of a DBN and DBM for binary image reconstruction. The number of hidden units, learning rate, weight decay, and momentum were searched for optimization. Three public datasets were used for the study: MNIST, Semeion handwritten digit, and Caltech 101 silhouettes. The BSA and the HS and DE variants achieved the best results among the MHs in the comparison. It was discussed that, since the fitness landscapes of the optimized methods have flat zones, the evolutionary-based optimization techniques with higher exploration were able to avoid local minimums. It was also noted that DBM performed better compared to DBN on two of the three datasets used.

In another study, Martín et al. (2020) used another metaheuristic that is nature-inspired and called the statistically-driven coral reef optimization. The algorithm is an extension of the CRO algorithm, which is inspired from the reproduction process of coral reefs. This version does not require to adjust any parameters, since they are automatically and dynamically chosen based on the statistical characteristics of the evolution, therefore the need to set hyperparameters to perform the metaheuristic search is being avoided.

The researchers tried to optimize the number of nodes and the type of activation functions in the fully-connected layers of a well-regarded CNN called VGG-16, which is used for image classification and has 16 layers. The classification was performed on two well-known datasets: CIFAR-10 and CINIC-10, which are collections of 60,000 and 270,000 images, respectively, belonging to 10 classes each.

Their main goal was to find simpler architectures of existing deep learning models, having a smaller number of units and parameters, and that were close in performance to the original models. The purpose of this simplification was to help the usage of CNN models in devices with low computational resources. In their experimental results, they showed that they were able to achieve their goal with models that have only about 7 and 8% of the number of parameters, and only about 9 and 10% of the number of nodes of the original model, with the first and second datasets, respectively. The accuracy rate was down by less than 0.2 and 7%, and the execution time was up by more than 32 and 54-fold, again with the first and second datasets, respectively.

Bacanin et al. (2020) used improved versions of the TGA and FA for the optimization of the hyperparameters of CNN models. They focused on optimizing: the number of convolutional layers, the number and size of kernels at each convolutional layer, the number of fully-connected layers, and the number of nodes at each fully-connected layer. They did not include the type of pooling layer, activation function, and regularization technique in the search, because of the limitations of computing power available.

They performed their experiments on the well-known MNIST benchmark dataset, which is comprised of 70,000 labeled images of handwritten digits from zero to nine, with each image in grayscale consisting of 28×28 pixels. As a pre-processing step, the pixel values of images that were in the range of [0, 255] were normalized to [0, 1].

They made comparisons on other metaheuristics (PSO, CS, and bat algorithm), the original versions of the TGA and FA, and manually designed state-of-the-art networks (LeNet-5, deeply supervised net, shallow CNN, etc.). Their improved versions of the TGA and FA gave the lowest loss rates among all the methods in comparison, with 0.19 and 0.21%, respectively; and highest classification accuracy rates among all the metaheuristics in comparison, with 99.18 and 99.16%, respectively.

Erkan et al. (2022) implemented the ABC algorithm to optimize a CNN for plant species identification. The optimum values were searched for the filter size and stride in convolutional and pooling layers. The input was the Folio dataset consisting of leaf images from various species. The results were compared with different studies using SVM and CNN based on VGG-19. The proposed optimized CNN achieved an accuracy rate of 98.99% and a weighted average F-score of 99.93%, outperforming the others.

Houssein et al. (2022) utilized an improved version of the MPA to optimize a CNN for arrhythmia classification. The optimum values were searched for the learning rate, optimizer, activation function, dropout rate, number of epochs, patience, number of filters in convolutional layers, and number of nodes in dense layers. In the experiments, the MIT-BIH arrhythmia, European ST-T, and St. Petersburg INCART databases were used, and the proposed method achieved accuracy rates of 99.33, 99.75, and 99.43%, respectively.

Lastly, Karthiga et al. (2022) used the ABC, GWO, and a proposed hybrid of both to optimize CNNs for the development of a wireless sensor network (WSN) enabled ECG monitoring system. The optimization was performed for the number of convolutional layers and their number and size of filters, number of dense layers and their number of nodes, number of epochs, and learning rate. The study focused on optimized routing for efficient ECG data transmission and arrhythmia detection. The ANN, RNN, and CNNs optimized with the above MHs were compared using the samples from the MIT-BIH arrhythmia database. The proposed hybrid method achieved the highest performance with an accuracy rate of 94.27%.

As can be seen from these studies, the MH methods have demonstrated their significant ability for stochastic optimization of the HPs of various deep learning methods. Thus, this approach is highly promising for further exploration of the subject in general and relates to a wide range of data science domains for various tasks in particular. This comprehensive view of the field formed the main focus of the study presented in this thesis.

3. MATERIAL AND METHODS

3.1. Dataset

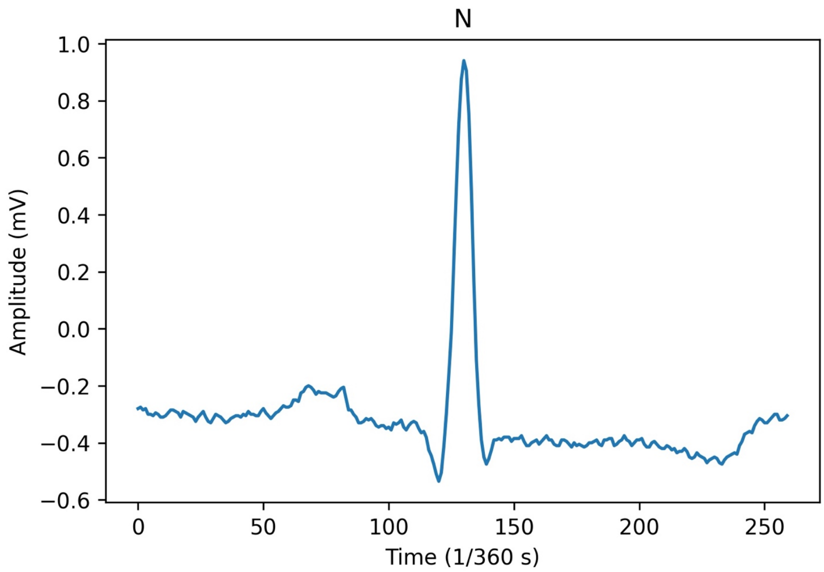
The dataset of ECG signals used in the experiments of this study was obtained from the open access Massachusetts Institute of Technology, Beth Israel Hospital (MIT-BIH) arrhythmia database (Moody & Mark, 2001). The database contains 48 half-hour portions of ECG recordings, obtained from 47 subjects at a sampling rate of 360 Hz. Each record was independently annotated by two or more cardiologists (PhysioNet, 2022, May 25). The data were segmented for use in the experiments with a window size of 260 values centered at the annotated R-peaks, with each segment representing a single heartbeat.

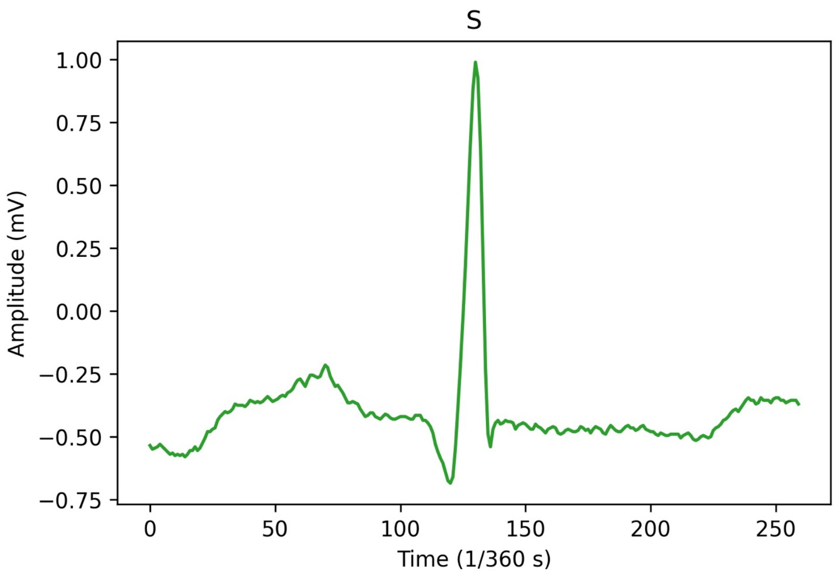
The database contains approximately 109,000 samples. In this study, an upper limit of 800 samples was set for each class in a record to reduce class imbalance, resulting in a total number of 48,507 samples. The beat types, classes, codes, and number of samples in the dataset are listed in Table 3.1.

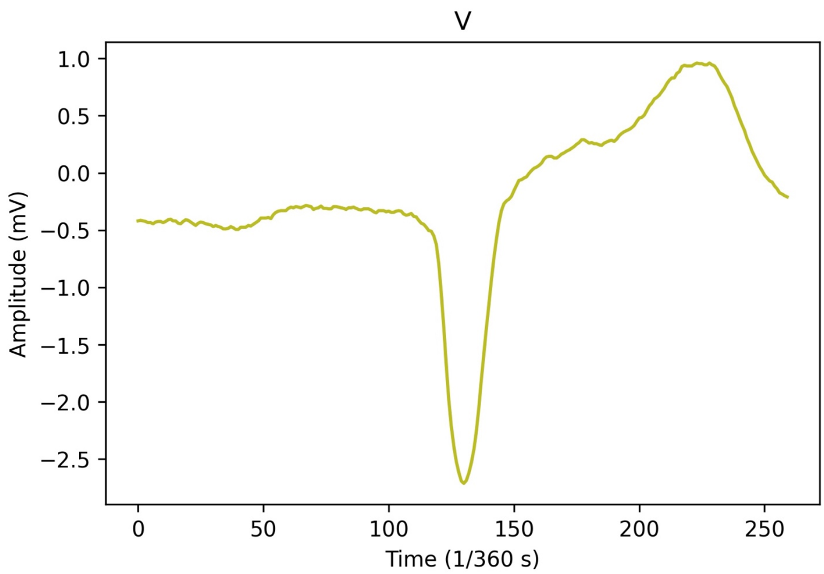
Table 3.1. Beat types, classes, codes, and number of samples in the dataset

MIT-BIH beat type	AAMI class	Code	Samples
Normal Left bundle branch block Right bundle branch block Atrial escape Nodal (junctional) escape	Normal	N	35,304
Supraventricular premature Atrial premature Aberrated atrial premature Nodal (junctional) premature	Supraventricular ectopic	S	2,199
Premature ventricular contraction Ventricular escape	Ventricular ectopic	V	6,987
Fusion of ventricular and normal	Fusion	F	802
Unclassifiable Paced Fusion of paced and normal	Unknown	Q	3,215

The plots of examplary samples from each class is given in Figure 3.1.


N
Amplitude (mV)
Time (1/360 s)


S
Amplitude (mV)
Time (1/360 s)


V
Amplitude (mV)
Time (1/360 s)

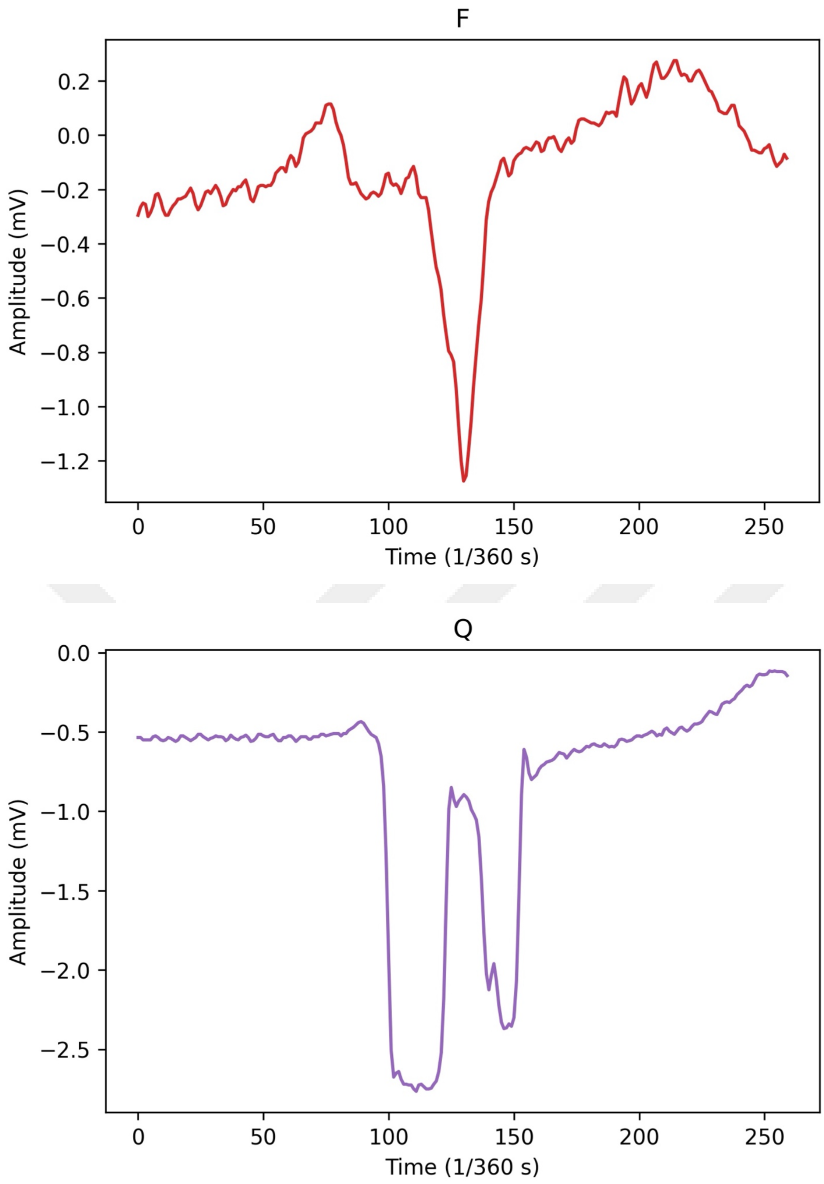


Figure 3.1. Plots of examplary samples from each class

3.2. Experimental Settings

The experiments were conducted on a computer with an Intel Xeon Gold 2.10GHz processor, 512GB memory, Nvidia Tesla V100 32GB graphics processor, and MS Windows Server 2019 operating system. The implementations of the methods used in the experiments were coded with the Python programming language and Keras deep learning library in the Jupyter Notebook integrated development environment.

3.2.1. Evaluation Metrics

The evaluation metrics used in this study are given with the confusion matrix below:

Actual / Predicted	Class	¬ Class
Class	true positive (TP)	false negative (FN)
¬ Class	false positive (FP)	true negative (TN)

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

$$\text{Precision} = TP / (TP + FP)$$

$$\text{Recall} = TP / (TP + FN)$$

$$\text{F-score} = TP / (TP + (FP + FN) / 2)$$

3.2.2. General Procedure

The general procedure of the study is given in Figure 3.2. First, the ECG beat samples are obtained from the ECG database. Then, candidate CNN models are trained and evaluated using these samples, and a hyperparameter optimization is performed to find the optimum configuration for the classifier. After the optimization process, the optimized model is validated with the samples for a final performance evaluation on arrhythmia classification.

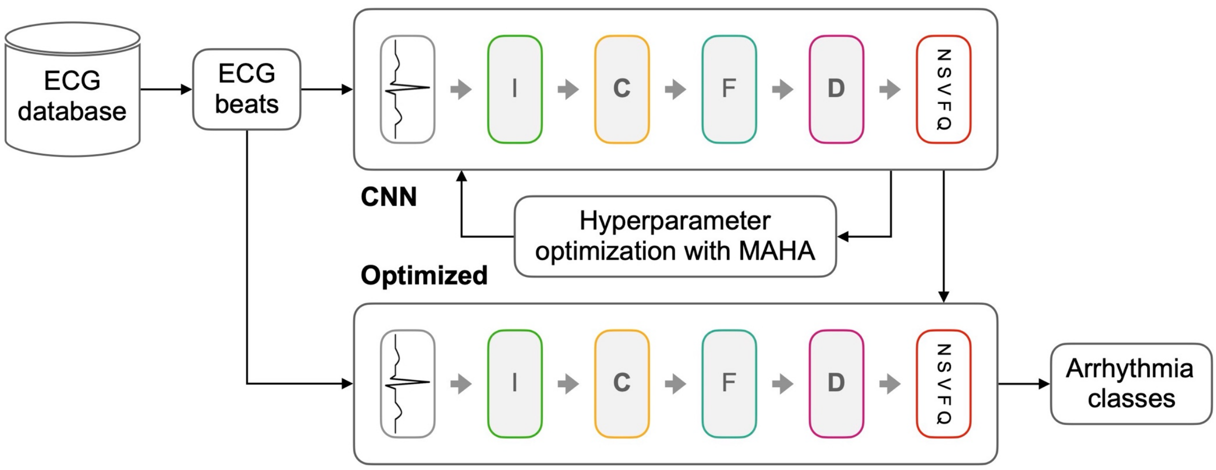


Figure 3.2. General procedure of the study

3.3. Classifier

3.3.1. Artificial Neural Network

Artificial neural networks are one of the most important inventions in the field of machine learning. In this method, mathematical structures that are inspired by the functioning of nerve cells in the human brain during impulse transmission and their arrangement for information formation are used. The first examples were applied in the 1940s (McCulloch and Pitts, 1943).

In the literature, several types of the method have been proposed: feedforward neural networks (Bebis and Georgiopoulos, 1994), iterative neural networks, etc. In feedforward neural networks, there is a series of unidirectional data transmissions from the input layer to the output layer. In recurrent neural networks, there is a two-way data sharing between layers, back and forth.

Despite the differences in their functioning, the common features of ANNs are their learning abilities. Learning means that they can derive knowledge from experience. Similar in a way to natural neural networks, ANNs have a mechanism that adapts to the input data set given to them. The method that performs the learning of an ANN is called a trainer. The trainer is responsible for training the ANN to achieve the highest performance against input datasets that it has not seen before. As a supervised learning method, the trainer first gives training examples to the ANN. Then, at each training step, it changes the structural parameters of the ANN to improve its performance. After the training phase is completed, the trainer is deactivated, and the ANN is ready for use.

There are two types of learning methods in the literature: deterministic and stochastic methods. Back-propagation and gradient-based methods are considered deterministic. In such methods, if the training samples do not change, separate training trials result in identical performance. On the other hand, separate training trials with identical training samples result in more or less varying performances in stochastic methods.

The advantages of deterministic methods are their simplicity and speed. However, performance is dependent on initial solutions, and they are more likely to get stuck with local optimums and miss the global optimum. Whereas, in stochastic methods, since the selection of initial solutions and the training process occur randomly, they are more likely to approach the global optimum by avoiding the local optimums.

Feedforward neural networks are neural networks in which data transmission takes place in a single and forward direction. In these networks, nodes are arranged in sequential layers. The first layer is called the input layer and the last layer is called the output layer. The layers between these two layers are called hidden layers. In this structure, each node in one layer is linked to each node in the next layer, or in other words, they are fully-connected (see Figure 3.3). Feedforward neural networks with more than one hidden layer are called deep neural networks.

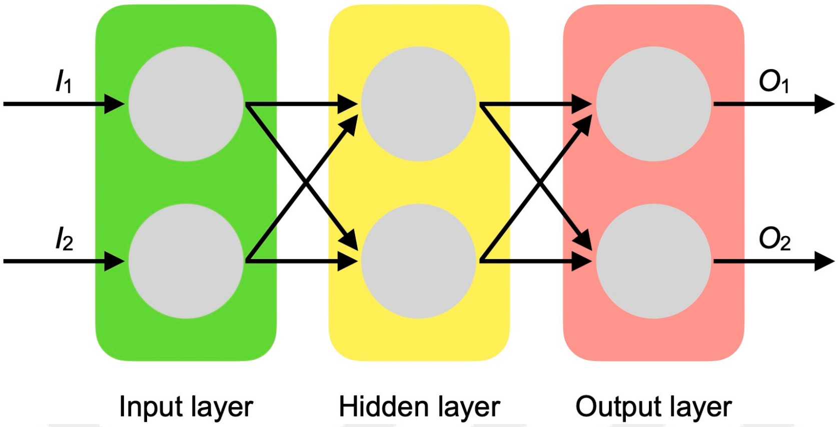


Figure 3.3. An ANN with one hidden layer

For an ANN, after the input data and weights and thresholds between the nodes are provided, by taking the weighted sum of the inputs and passing them through the activation function, the output data is obtained with the following equation:

$$O_j = \sum_i I_i \cdot W_{i,j} + B_j$$

where O_j: output of j^{th} node in the current layer, I_i: input from i^{th} node in the previous layer, $W_{i,j}$: weight between i^{th} node in the previous layer and j^{th} node in the current layer, B_j: bias of j^{th} node in the current layer.

3.3.2. Convolutional Neural Network

A CNN is an DNN which has special layers that extract necessary features from the input data and prepare them for the classification task to be performed in the subsequent classical neural network layers. The network can be one-dimensional or higher-dimensional depending on the number of dimensions of the input data. Since the input data used in this work are one-dimensional (1D) vectors of decimal numbers, a 1D CNN is utilized as the classifier.

The most important layer in this type of network is the convolutional layer which performs the convolution operation, that is, an element-wise multiplication and addition of the input and filter vectors, as in the equation below:

$$O_j = \sum_{i=1}^{k} I_{j+i} \cdot F_i$$

where O_j: j^{th} element of the output, k: kernel size, I_{j+i}: $(j+i)^{th}$ element of the input, F_i: j^{th} element of the filter kernel.

This layer consists of filters with a vector of a certain size, called the kernel, which is used to perform the convolution operation. The input vector may be padded with zeros at both ends (using the 'same' option) to keep its original length throughout these operations. At each operation, the window of the input vector for the operation, with a length equal to that of the kernel, slides forward for a certain number of elements named the stride (see Figure 3.4). These layers detect local features from lower-level to higher-level attributes of the data successively (Goodfellow et al., 2016).

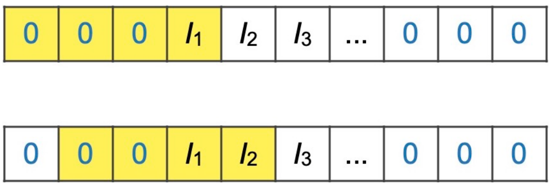


Figure 3.4. Slide of convolution window with padding=same, kernel size=4, and stride=1

In the non-linearity layer, an activation function is applied to the output of the previous layer (e.g., ReLU, see Figure 3.5). The down-sampling (DS) layer reduces the size of feature representation, number of parameters, and computations (e.g., maximum pooling (max-pooling), see Figure 3.6). Batch normalization (BN) and dropout (DO) are applied to reduce the sensitivity of network initialization and avoid over-fitting. A flattening layer converts the feature map into a vector. Finally, the dense (fully-connected) layers assign weights and biases to the units (nodes) and calculate probabilities for the input data belonging to the target classes as the output of the system.

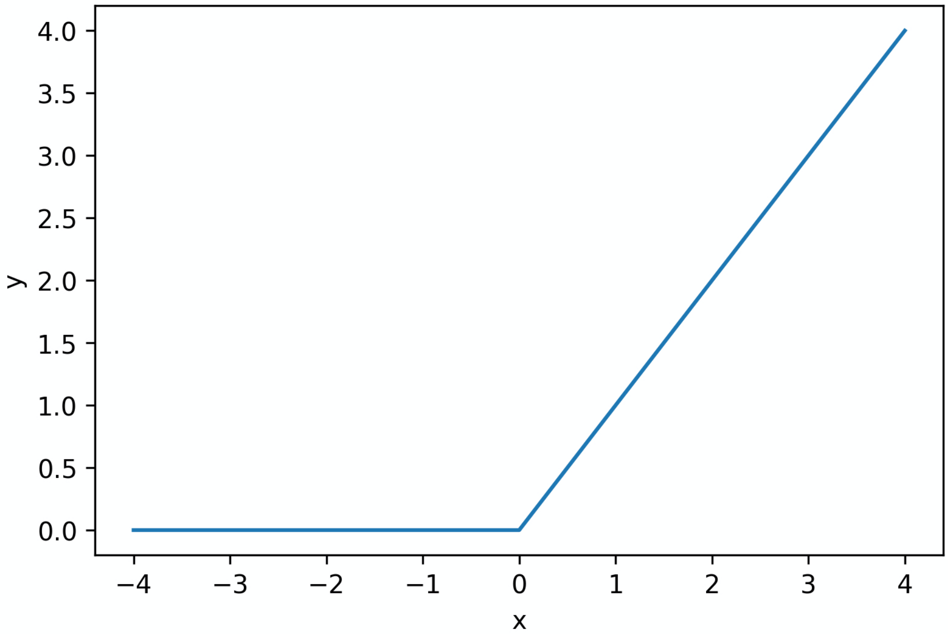


Figure 3.5. Plot of the ReLU activation function: $y = \max(0, x)$

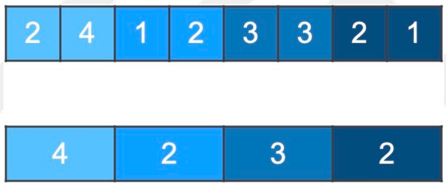


Figure 3.6. An examplary maximum pooling operation with kernel size=2 and stride=2

3.4. Hyperparameter Optimization

In this study, several selected hyperparameters of a CNN architecture are optimized for arrhythmia classification on ECG signals. These HPs are searched in sets of discrete numbers selected empirically or from studies on this topic in the literature. The rest of the HPs are predefined or used with their default values. The HPs and corresponding search sets or predefined values are listed in Table 3.2.

Table 3.2. HPs and their search sets or predefined values

Hyperparameter	Search set
Conv. layers	{3, 4, 5, 6}
Dense layers	{1, 2}
Filters	{32, 64, 128, 256}
Units	{256, 512, 1024, 2048}

Batch size	{32, 64, 128}
Dropout rate	{0.2, 0.3, 0.4, 0.5}
Kernel size	{4, 6, 8, 10}
Learning rate	{0.0001, 0.001, 0.01, 0.1}

Hyperparameter	Predefined value
Padding	same
Activation	relu, softmax
Down-sampling	max-pooling
Optimizer	adam
Loss	categorical crossentropy
Metrics	accuracy
Epochs	10
Validation split	0.2

Sets of HPs as different configurations for the classifier are represented in the form of encoded solutions and these solutions are optimized using the selected metaheuristic methods.

3.4.1. Encoded Solutions

The encoded solutions of HP sets consist of a structural and an operational part. The structural part consists of the number of convolutional and dense layers, and number of filters or units each layer has in sequence. The operational part consists of the batch size, dropout rate, kernel size, and learning rate. A representation of an encoded solution can be seen below:

$$solution = \{co, de, f_1, f_2, f_3, f_4, f_5, f_6, u_1, u_2, b, d, k, l\}$$

where *co*: number of convolutional layers, *de*: number of dense layers, *f*: number of filters in each convolutional layer, *u*: number of units in each dense layer, *b*: batch size, *d*: dropout rate, *k*: kernel size, *l*: learning rate.

The encoded solutions are initialized randomly from the uniform distribution of integers in the [*minimum*, *maximum*] intervals of element indices of HP search sets. These solutions are first decoded and then passed to the CNN method for training and evaluation of the candidate networks.

The general architecture of the 1D-CNN arrhythmia classifier optimized in this study is shown in Figure 3.7, with the optimized HPs indicated in orange. First, ECG beat data are given to the input layer (I). The following convolutional blocks (C), each consisting of a convolutional, BN, and DS layer, take the data from the previous layer. Then, the feature vector is formed by the flattening layer (F) and passed to the following dense blocks (D), each consisting of a dense, BN, and DO layer. Finally, the output of the last dense block is passed to the output layer to complete the classification task.

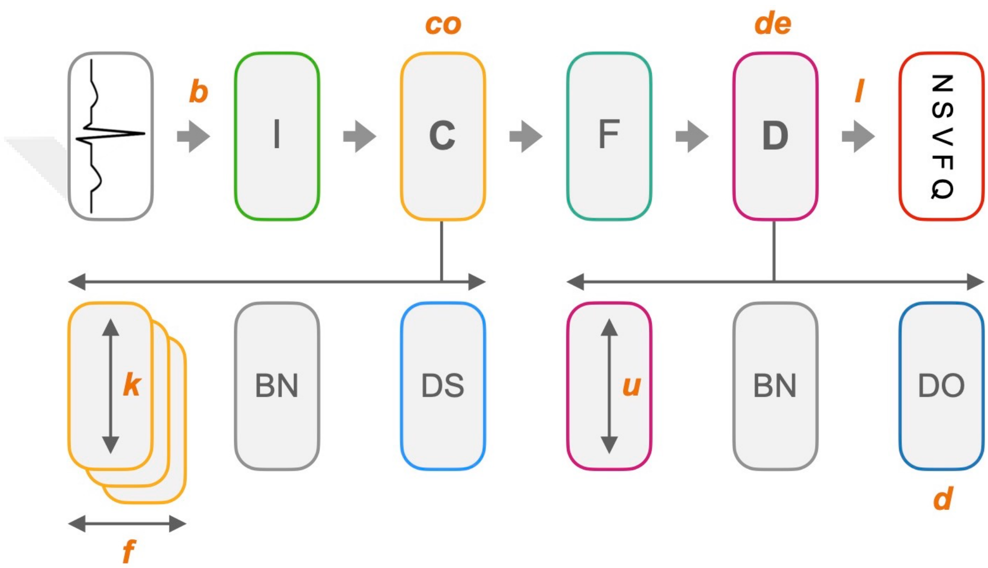


Figure 3.7. General architecture of the 1D-CNN arrhythmia classifier

3.4.2. Metaheuristic Methods

Heuristic means 'to find' or 'to discover by trial- and- error'. High quality solutions to a difficult optimization problem can be found in a reasonable amount of time, but there is no certainty that the absolute optimum values are to be reached in a certain number of runs. These methods are expected to work most of the time, but not always. In addition, they are useful when the best solutions are not necessarily needed, but rather good solutions that are easy to reach.

Further development of heuristic methods is the so-called metaheuristic methods. The meta- prefix is used to give the meaning of 'beyond' or 'at a higher- level', and these methods achieve higher performances than heuristics in general. Moreover, all metaheuristic methods use certain trade-offs of randomization and local search. There is no consistent definition of heuristics and metaheuristics in the literature, and these terms are seen to be used interchangeably. However, the recent trend is to name all stochastic methods with randomization and local search as metaheuristic, and this convention is used in this study. Randomization provides an effective way to move from local search to global-level search. Therefore, metaheuristic methods are generally suitable for global optimization.

Two main components of metaheuristic algorithms are intensification and diversification, or exploitation and exploration. Diversification means generating different solutions to explore the search space at global level. Intensification means focusing on the search in a local region by exploiting the information that a good solution has already been found in that region. This is done in combination with selecting the best solutions among the candidate solutions. Best solution selection ensures that solutions converge to optimality, while diversification through randomization prevents solutions from being trapped in local optimums and increases the diversity of solutions.

Metaheuristic algorithms can be classified in various ways, and a common approach is to classify as population-based or trajectory-based. For example, genetic algorithms, PSO, FA, and CS are population-based, which all use multiple search agents. A collective intelligence of the group leads to near-optimal solutions in the search space. SA, on the other hand, is trajectory-based which uses a single search agent that moves piecewise through the search space. A better move is always accepted, while a not-so-good move may be accepted with some probability. (Yang, 2020).

3.4.3. Artificial Hummingbird Algorithm

The optimization of the HPs of the classifier is performed using the proposed variant of the artificial hummingbird algorithm (AHA) (Zhao et al., 2022). This method is a bio-inspired algorithm inspired by the excellent memory, flight capabilities, and foraging behaviors of hummingbirds. These birds feed on the nectar of flowers and can remember individual flowers in a region along with their location, nectar-refilling rate, and time of last visit.

With their tiny bodies and high-frequency wing beats, aside from the flight like other birds, they can also fly at three different attitudes: axial, diagonal, and omnidirectional, as shown in Figure 3.8, and simulated in the upper, middle, and lower parts of Eq. (1), respectively:

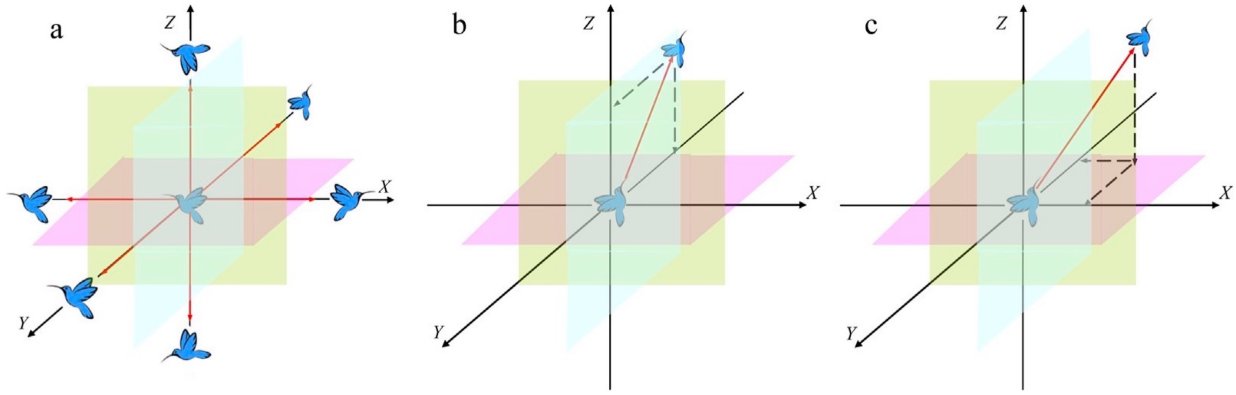


Figure 3.8. Flight attitudes of hummingbirds, (a) axial, (b) diagonal, (c) omnidirectional

$$D_i = \begin{cases} \begin{cases} 1 & i = random(d) \\ 0 & \text{otherwise} \end{cases} & r_1 < 1/3 \\ \begin{cases} 1 & i = P_j,\ P = per(k) \\ 0 & \text{otherwise} \end{cases} & 1/3 \le r_1 < 2/3 \\ 1 & 2/3 \le r_1 \end{cases} \tag{1}$$

where D: direction switch vector, i: index of dimension, $random(d)$: a random integer in $[1, d]$, d: number of dimensions, $per(k)$: a random permutation of integers in $[1, k]$, j: an integer in $[1, k]$, k: an integer in $[2, \lceil r_k \cdot (d-2) \rceil + 1]$, r_k and r_1: random decimal numbers in $[0, 1)$.

In this method, each food source has the same number and type of flowers, its position is a solution vector, and its nectar-refilling rate is its fitness value. Each bird is assigned to a source, knows its position and nectar-refilling rate, shares this information with other birds, and remembers its visit level of other birds' assigned sources in the form of a visit table (see Figure 3.9).

The table is initialized with each bird's cell for its assigned source as *null* and the remaining cells as 0. At each iteration, each bird's cells for unvisited sources are incremented by 1, and its cell for a visited source is initialized to 0. When a bird gets assigned to a new source, the cells of this source for the remaining birds are updated as each bird's maximum value of its cells incremented by 1.

		Source			
		x_1	x_2	x_3	x_4
Bird	x_1	--	2	1	3
	x_2	1	--	1	2
	x_3	1	0	--	2
	x_4	1	0	1	--

Figure 3.9. Visit table of a population of four hummingbirds

The birds perform three foraging behaviors: guided, territorial, and migration foraging. Firstly, in guided foraging, a bird explores the search space with guidance from the visit table, which it uses to find its target source among other birds' assigned sources with the highest unvisited time by itself. If there are multiple highest times, it chooses the source with the highest nectar-refilling rate. Then, the bird visits its target source and finds a candidate source for itself during the flight. A bird gets assigned to its candidate source if it has a higher nectar-refilling rate than its assigned source.

Secondly, in territorial foraging, a bird exploits the territory of its assigned source to find a candidate source instead of visiting another bird's assigned source. Lastly, in migration foraging, a bird migrates to a distant source when there is a shortage of food in its region. The bird at the source with the lowest nectar-refilling rate migrates to a new source selected randomly from the entire search space.

In the algorithm, flight attitudes have an equal probability of 1/3, and guided and territorial foraging have an equal probability of 1/2. Migration foraging is performed when the number of iterations exceeds the migration coefficient, which is determined as $2n$, where n is the population number. The candidate sources for guided and territorial foraging are generated using the upper and lower parts of Eq. (2), respectively:

$$v_i = \begin{cases} t_i + a \cdot D \cdot (x_i - t_i) & r_2 < 1/2 \\ x_i + b \cdot D \cdot x_i & \text{otherwise} \end{cases} \tag{2}$$

where v: candidate source, t: target source, x: assigned source, i: index of bird, a and b: random decimals from the standard normal distribution, r_2: a random decimal in $[0,1)$.

3.4.4. Proposed Fitness Function

For this study, a fitness function was proposed to improve the HPO process of the classifier. This function combines the two important factors of the classification accuracy and the total number of parameters of each candidate network. As mentioned in Section 1, the HPO process of neural networks involves very high computational complexity and incurs significant time and energy costs consequently, forming a major obstacle for associated studies. Therefore, it is extremely important to find configurations with relatively lower number of parameters while maintaining the highest possible classification performance. This approach is very beneficial to achieve better efficiency for the target tasks in this research area.

The total number of parameters of a candidate network is subjected to minimum-maximum normalization, which is performed with the equation below:

$$p_{scaled} = (p - p_{min}) / (p_{max} - p_{min})$$

where p_{scaled}, p, p_{min}, and p_{max}: scaled, total, minimum, and maximum number of parameters, respectively.

And the fitness value of a candidate network is calculated as in the equation below:

$$fit = \alpha \cdot acc + \beta \cdot (1 - p_{scaled})$$

where fit: fitness value, acc: accuracy rate, α and β: fitness contribution coefficients for accuracy rate and number of parameters, selected as 0.99 and 0.01, respectively.

3.4.5. Proposed Memory-enhanced AHA

The original method of AHA uses the visit table as the memory unit and can be considered as the heart of the algorithm. To improve the efficiency of the method for the HPO of a CNN, which is the focus of this study, we integrated an additional memory unit to the algorithm. This addition provides a significant enhancement to the memory capability of the original method; hence, the proposed variant of the method is named as the memory-enhanced AHA (MAHA). The new memory unit is designed to store three essential components: the classification accuracy, fitness value, and number of repeated appearances of solutions in the search process.

The utilization of the new memory unit in MAHA significantly improves the time required for a CNN's HPO by training a candidate configuration of the network only once in a single run and avoiding unnecessary repetitive training. In addition, the involvement of the total number of parameters in the fitness value promotes finding network configurations with relatively lower computational complexity and time cost. Lastly, knowing the number of repeated appearances of solutions provides the opportunity to compare the global optimum with the most frequently seen solutions.

The pseudocode of the memory(·), fitness memory function is given in Algorithm 1, where *source*: source to be evaluated, fitness(·): fitness function, *mem*: evaluations of sources, *rep*: number of repetitions.

Algorithm 1. memory(·), fitness memory function

input: $source$, fitness(·)
output: fit

if $source \notin mem$
  $acc, fit = \text{fitness}(source)$
  $rep = 1$
  $mem_{source} = acc, fit, rep$
else
  $fit = mem_{source,2}$
  $mem_{source,3} \mathrel{+}= 1$

And the pseudocode of the proposed MAHA is given in Algorithm 2, where *l* and *u*: lower and upper boundaries of dimensions, *iterations*: number of iterations, *max*: index of the source with maximum fitness value, *f*: fitness values of the assigned sources, r_0: a random decimal number in $[0,1)$, *visit*: visit table matrix, *tar*: index of the target source, *min*: index of the source with minimum fitness value.

Algorithm 2. Memory-enhanced artificial hummingbird algorithm

input: $l, u, d, n, iterations$, memory$(\cdot)$
output: x_{max}, f_{max}

for i^{th} bird $\leftarrow$ 1 to n
  $x_i = l + r_0 \cdot (u - l)$
  $f_i = \text{memory}(x_i)$
  for j^{th} source $\leftarrow$ 1 to n
    if $i = j$
      $visit_{i,j} = null$
    else
      $visit_{i,j} = 0$

for $t \leftarrow$ 1 to $iterations$
  for i^{th} bird $\leftarrow$ 1 to n
    compute D with Eq. (1)
    compute v_i with Eq. (2)
    for j^{th} source $\leftarrow$ 1 to n, $j \neq i$
      $visit_{i,j}$ += 1
    if $r_2 < 1/2$
      $visit_{i,tar} = 0$
    $new = \text{memory}(v_i)$
    if $new > f_i$
      $x_i = v_i$
      $f_i = new$
      for j^{th} source $\leftarrow$ 1 to n, $j \neq i$
        $visit_{j,i} = \max_{k \in 1..n,\, k \neq j} (visit_{j,k}) + 1$
  if mod$(t, 2n) = 0$
    $x_{min} = l + r_0 \cdot (u - l)$
    $f_{min} = \text{memory}(x_{min})$
    for j^{th} source $\leftarrow$ 1 to n, $j \neq min$
      $visit_{min,j}$ += 1
      $visit_{j,min} = \max_{k \in 1..n,\, k \neq j} (visit_{j,k}) + 1$

4. RESULTS AND DISCUSSION

The optimization experiments were conducted on the selected dataset and classifier with the utilization of four MH methods, namely: PSO, GWO, sine cosine algorithm (SCA) (Mirjalili, 2016), and the proposed method of MAHA. The runs were performed with a population size of 30 and an iteration number of 50. The parameter settings of the MH methods are given in Table 4.1.

Table 4.1. Parameter settings of the MH methods

Method	Parameter	Value
PSO	c_1, c_2, v_{max}, w_{min}, w_{max}	2, 2, 6, 0.2, 0.9
GWO	a	$2 \rightarrow 0$
SCA	a	2
MAHA	migration coefficient	$2n$

In each run, a total of $n \cdot iterations = 30 \cdot 50 = 1500$ candidate solutions were evaluated, which requires a substantial amount of computation and time. The proposed method of MAHA benefited immensely from its additional memory unit and required only 933 fitness evaluations, which is a significant, almost 40% reduction in number. This result is a concrete proof of the proposed approach and shows its advantages in this research area. With further studies on this topic, other MH methods can also be improved in a similar manner to increase their optimization efficiency for various domains.

The best solutions in the runs are given with their HPs in Table 4.2. Firstly, for the structural HPs, it can be seen that the best performing networks have the highest levels of depth in their feature learning convolutional layers. Three best solutions have the highest possible number of convolutional layers, and one has the second-highest. These layers have different number of filters consisting of all possible numbers. All best solutions have one dense layer, which is the shallowest option. In addition, these layers have a varying number of units, and none has the highest number in the search set.

Table 4.2. Best solutions found with the MH methods

Method	Filters	Units	B	D	K	L
PSO	32, 256, 256, 256, 256, 256	256	32	0.3	8	0.001
GWO	32, 32, 32, 64, 64	1024	32	0.3	10	0.001
SCA	32, 32, 32, 32, 128, 256	512	64	0.5	10	0.001
MAHA	64, 64, 64, 64, 128, 128	256	32	0.3	8	0.001

Secondly, for the operational HPs, three best solutions have the lowest, and one has the second-lowest batch size. For the dropout rate, three have the second-lowest option, and one has the highest. For kernel sizes, two have the highest possible value and two have the second-highest. And very significantly, all best solutions have the same, second-lowest learning rate in the search set. Sensibly, this number is also the most commonly used and observed value for similar networks in the literature.

The convergence curves of the accuracy rate and fitness value (%) in the runs are shown in Figure 4.1. First, it can be seen that PSO converged to its maximum accuracy rate in about ten iterations and the other three methods converged in about 30 iterations. For the fitness value, PSO and SCA converged to their maximum in about ten iterations, and GWO and MAHA converged in about 30 iterations.

On closer look, it can be observed that the accuracy rate of some iterations is slightly lower than that of the previous iterations, since the optimization process is based on the proposed fitness function. These are expected cases, because a network with a sufficiently lower number of parameters may have a higher fitness value than a network with a higher accuracy rate.

For the performance metrics, MAHA and PSO achieved equal highest accuracy rates of 98.87, SCA the second-lowest with 98.80, and GWO the lowest with 98.78. And MAHA achieved the highest fitness value of 98.87, PSO the second-highest with 98.81, SCA the second-lowest with 98.78, and GWO the lowest with 98.77. The proposed MH method of MAHA has successfully demonstrated its high performance in exploring and exploiting the vast search space of an NP-hard optimization problem such as the one focused in this study. This promising result of the approach shows many application potentials in various search domains.

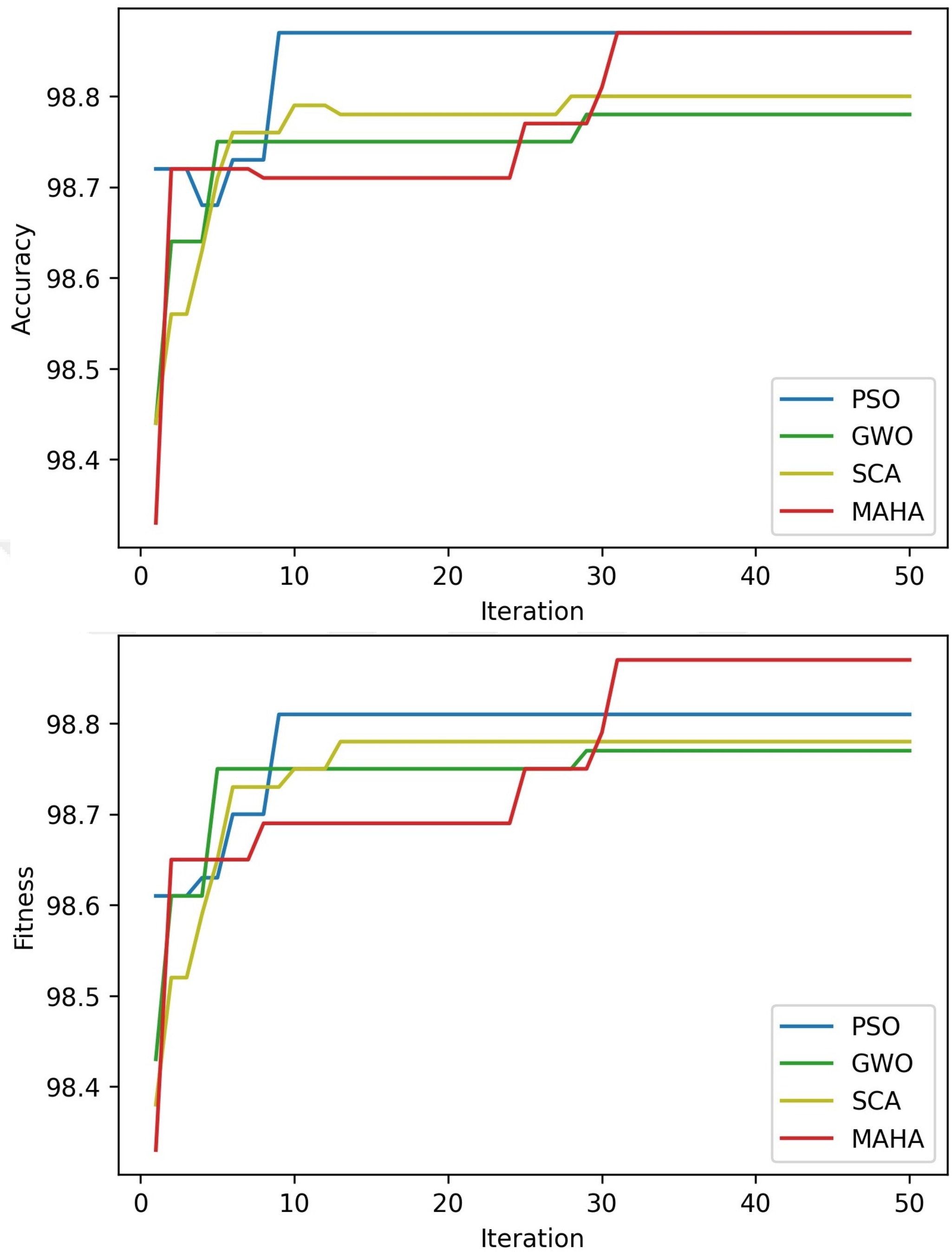


Figure 4.1. Convergence curves of metrics in the searches with MH methods

The stratified 10-fold cross-validation (CV) training and testing metrics (%) of the best solution found with MAHA are shown in Figure 4.2 and Table 4.3, respectively. As can be seen from these results, the performance of the proposed algorithm is further validated by using all samples in the dataset as training and testing subsets for the prediction task. The CV mean training accuracy and loss rate plots show that the training of the network was performed successfully without over-fitting. Moreover, the CV mean testing accuracy rate is very close to the one achieved in the search phase (only about 0.2% lower).

The CV confusion matrix (% of true labels) of the best solution found with MAHA is shown in Figure 4.3. On the inspection of the values, it can be clearly seen that the target classes with a relatively lower number of samples (that are, the classes F and S) are the least correctly classified ones compared to the others in the dataset.

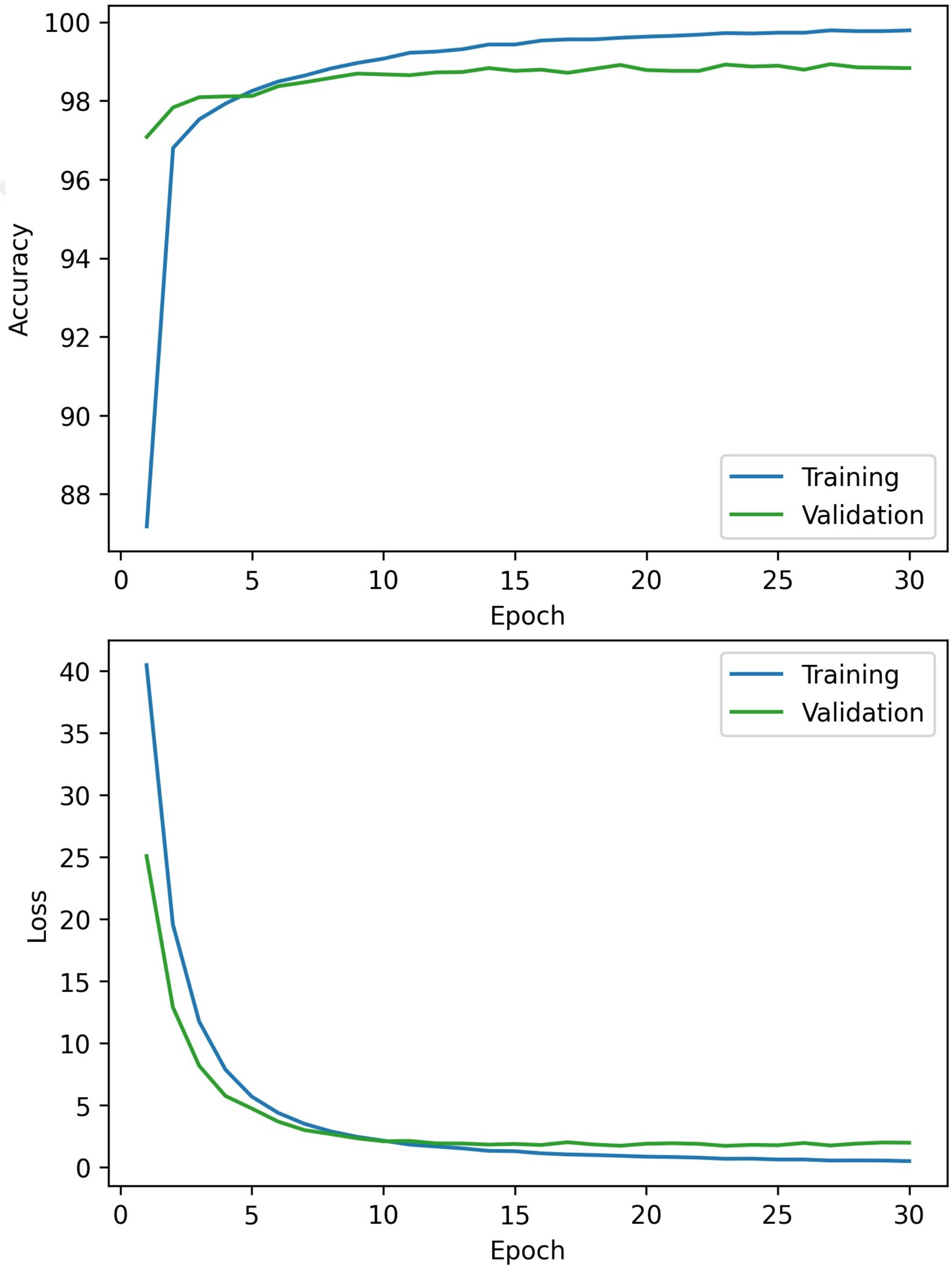


Figure 4.2. CV mean training metrics of the best solution of MAHA

Class F with the lowest number of samples, possessing only about 1.7% of all samples in the dataset, is the least correctly classified class. Its samples are mostly mislabeled as belonging to classes N and V. Class S, with the second-lowest number of samples, consisting of about 4.5% of all samples, is the second-least correctly classified class. Its samples are mostly mislabeled as being in classes N and V, also.

Interestingly, class Q with 6.6% of the samples, has a higher correct prediction rate than class V which has a significantly higher share with 14.4%. Lastly, as expected, class N, which has the highest number of samples, consisting of 72.8% of all samples, has the highest correct prediction rate among all target classes in the dataset.

Table 4.3. CV testing metrics of the best solution of MAHA

Fold	Accuracy	Precision	Recall	F-score
1	98.72	98.71	98.72	98.70
2	98.87	98.86	98.87	98.86
3	98.41	98.47	98.41	98.41
4	98.91	98.90	98.91	98.90
5	98.47	98.45	98.47	98.46
6	98.74	98.73	98.74	98.72
7	98.56	98.54	98.56	98.54
8	98.87	98.85	98.87	98.85
9	98.64	98.63	98.64	98.63
10	98.66	98.64	98.66	98.64
Mean	98.69	98.68	98.69	98.67

As a comparison with previous studies, in (Houssein et al., 2022), the data used were largely altered for obtaining class balance with random undersampling and synthetic minority oversampling, and the sets had a total of 80,000 samples each. In addition, six types of feature extractions were applied to the samples. With the significant help from these factors, promising results were achieved. In (Karthiga et al., 2022), the dataset used had three different arrhythmia classes of atrial fibrillation, atrial flutter, and ventricular fibrillation, with a total of only 3,000 samples.

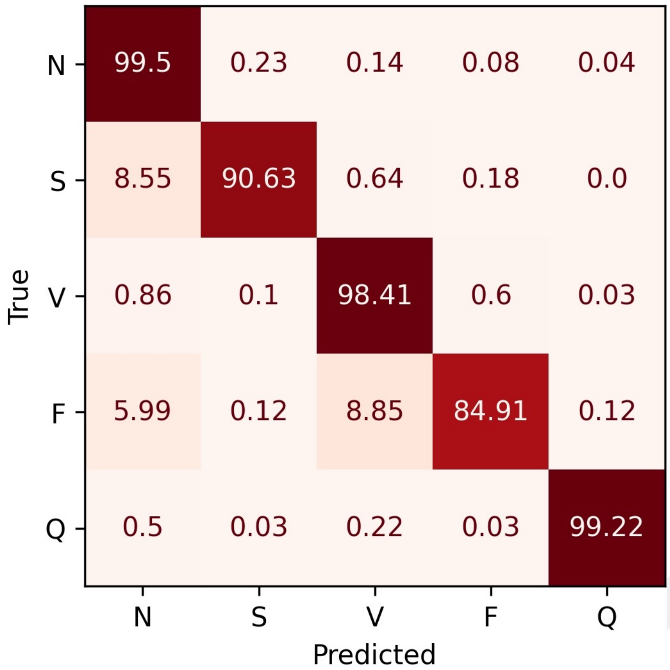


Figure 4.3. CV confusion matrix of the best solution of MAHA

5. CONCLUSION

In this study, an automated HPO method was proposed to search and find the optimal configuration of a CNN arrhythmia classifier. The input data were obtained from the ECG heartbeat recordings of the MIT-BIH arrhythmia database. The approach used our novel variant of a very recent MH method called the memory-enhanced artificial hummingbird algorithm. The method explored and exploited an immensely large search space of structural and operational HP configurations. With the utilization of its new and additional memory unit, the computation and time requirements were reduced significantly.

In the experiments, the proposed method of MAHA was compared with three other MH methods of PSO, GWO, and SCA. The results showed that MAHA shared the best performance with PSO in terms of accuracy rate. It performed as the best for the proposed fitness function considering both the accuracy rate and the total number of parameters of each candidate network. The proposed method yielded promising results in finding a high-performing network with relatively lower complexity, with classification accuracy reaching 98.87%. The CV metrics further proved the quality of the best solution found by a training step without over-fitting and a testing step demonstrating very close success with the one obtained in the search phase, with F-score reaching 98.67%.

6. RECOMMENDATIONS

The proposed approach can be applied in different datasets for cardiac arrhythmia classification and other domains for future work. Moreover, the additional memory unit can be integrated into other MH algorithms to investigate efficiency improvements for optimization tasks in various domains.

REFERENCES

Acharya, U. R., Oh, S. L., Hagiwara, Y., Tan, J. H., Adam, M., Gertych, A., & San Tan, R. (2017). A deep convolutional neural network model to classify heartbeats. *Comput. Biol. Med.*, *89*, 389–396. doi: 10.1016/j. compbiomed.2017.08.022

Asadi, S., Roshan, S., & Kattan, M. W. (2021). Random forest swarm optimization-based for heart diseases diagnosis. *J. Biomed. Inform.*, *115*, 103690. doi: 10.1016/j.jbi.2021.103690

Association for the Advancement of Medical Instrumentation (2012). *Testing and reporting performance results of cardiac rhythm and ST segment measurement algorithms*. Standard EC57 AAMI.

Bacanin, N., Bezdan, T., Tuba, E., Strumberger, I., & Tuba, M. (2020). Optimizing convolutional neural network hyperparameters by enhanced swarm intelligence metaheuristics. *Algorithms*, *13*(3), 67. doi: 10.3390/ a13030067

Bebis, G., & Georgiopoulos, M. (1994). Feed-forward neural networks. *IEEE Potentials*, *13*(4), 27–31. doi: 10.1109/45.329294

Chen, S. W., Wang, S. L., Qi, X. Z., Samuri, S. M., & Yang, C. (2022). Review of ECG detection and classification based on deep learning: Coherent taxonomy, motivation, open challenges and recommendations. *Biomed. Signal Process. Control*, *74*, 103493. doi: 10.1016/j.bspc.2022.103493

Cheraghalipour, A., Hajiaghaei-Keshteli, M., & Paydar, M. M. (2018). Tree growth algorithm (TGA): A novel approach for solving optimization problems. *Eng. Appl. Artif. Intell.*, *72*, 393–414. doi: 10.1016/j.engappai. 2018.04.021

Civicioglu, P. (2013). Backtracking search optimization algorithm for numerical optimization problems. *Appl. Math. Comput.*, *219*(15), 8121–8144. doi: 10.1016/j.amc.2013.02.017

Dey, M., Omar, N., & Ullah, M. A. (2021). Temporal feature-based classification into myocardial infarction and other CVDs merging CNN and bi-LSTM from ECG signal. *IEEE Sens. J.*, *21*(19), 21688–21695. doi: 10.1109/jsen.2021.3079241

Elmasry, W., Akbulut, A., & Zaim, A. H. (2020). Evolving deep learning architectures for network intrusion detection using a double PSO metaheuristic. *Comput. Netw.*, *168*, 107042. doi: 10.1016/j.comnet.2019.107042

Erkan, U., Toktas, A., & Ustun, D. (2022). Hyperparameter optimization of deep CNN classifier for plant species identification using artificial bee colony algorithm. *J. Ambient Intell. Humaniz. Comput.*, . doi: 10.1007/ s12652-021-03631-w

Faramarzi, A., Heidarinejad, M., Mirjalili, S., & Gandomi, A. H. (2020). Marine predators algorithm: A nature-inspired metaheuristic. *Expert Syst. Appl.*, *152*, 113377. doi: 10.1016/j.eswa.2020.113377

Geem, Z. W., Kim, J. H., & Loganathan, G. V. (2001). A new heuristic optimization algorithm: Harmony search. *Simulation*, *76*(2), 60–68. doi: 10.1177/003754970107600201

Geweid, G. G. N., & Chen, J. D. Z. (2022). Automatic classification of atrial fibrillation from short single-lead ECG recordings using a hybrid approach of dual support vector machine. *Expert Syst. Appl.*, *198*, 116848. doi: 10.1016/j.eswa.2022.116848

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press. https://www.deeplearningbook.org

Heart Rhythm Society (2022, May 25). Heart rhythm disorders. https://upbeat.org/heart-rhythm-disorders

Houssein, E. H., Hassaballah, M., Ibrahim, I. E., AbdElminaam, D. S., & Wazery, Y. M. (2022). An automatic arrhythmia classification model based on improved marine predators algorithm and convolutions neural networks. *Expert Syst. Appl.*, *187*, 115936. doi: 10.1016/j.eswa.2021.115936

Houssein, E. H., Ibrahim, I. E., Neggaz, N., Hassaballah, M., & Wazery, Y. M. (2021). An efficient ECG arrhythmia classification method based on manta ray foraging optimization. *Expert Syst. Appl.*, *181*, 115131. doi: 10.1016/j.eswa.2021.115131

Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *J. Glob. Optim.*, *39*(3), 459–471. doi: 10.1007/s10898-007-9149-x

Karthiga, M., Santhi, V., & Sountharrajan, S. (2022). Hybrid optimized convolutional neural network for efficient classification of ECG signals in healthcare monitoring. *Biomed. Signal Process. Control*, *76*, 103731. doi: 10.1016/j.bspc.2022.103731

Kaya, Y. (2021). Detection of bundle branch block using higher order statistics and temporal features. *Int. Arab J. Inf. Technol.*, *18*(3), 279–285. doi: 10.34028/iajit/18/3/3

Kaya, Y., & Pehlivan, H. (2015a). Classification of premature ventricular contraction in ECG. *Int. J. Adv. Comput. Sci. Appl.*, *6*(7). doi: 10. 14569/ijacsa.2015.060706

Kaya, Y., & Pehlivan, H. (2015b). Feature selection using genetic algorithms for premature ventricular contraction classification. In *2015 9th international conference on electrical and electronics engineering (ELECO)* (pp. 1229–1232). IEEE. doi: 10.1109/eleco.2015.7394628

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95 - international conference on neural networks* (pp. 1942–1948). IEEE volume 4. doi: 10.1109/icnn.1995.488968

Kılıç, F., Kaya, Y., & Yildirim, S. (2021). A novel multi population based particle swarm optimization for feature selection. *Knowl.-Based Syst.*, *219*, 106894. doi: 10.1016/j.knosys.2021.106894

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, *220*(4598), 671–680. doi: 10.1126/science.220.4598.671

Kumar, A., Kumar, S., Dutt, V., Dubey, A. K., & García-Díaz, V. (2022). IoT-based ECG monitoring for arrhythmia classification using coyote grey wolf optimization-based deep learning CNN classifier. *Biomed. Signal Process. Control*, *76*, 103638. doi: 10.1016/j.bspc.2022.103638

Luo, X., Yang, L., Cai, H., Tang, R., Chen, Y., & Li, W. (2021). Multiclassification of arrhythmias using a HCRNet on imbalanced ECG datasets. *Comput. Methods Programs Biomed.*, *208*, 106258. doi: 10.1016/j.cmpb. 2021.106258

Martín, A., Vargas, V. M., Gutiérrez, P. A., Camacho, D., & Hervás-Martínez, C. (2020). Optimising convolutional neural networks using a hybrid statistically-driven coral reef optimisation algorithm. *Appl. Soft Comput.*, *90*, 106144. doi: 10.1016/j.asoc.2020.106144

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics 1943*, *5*(4), 115–133. doi: 10.1007/bf02478259

Mirjalili, S. (2016). SCA: A sine cosine algorithm for solving optimization problems. *Knowl.-Based Syst.*, *96*, 120–133. doi: 10.1016/j.knosys.2015.12.022

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Adv. Eng. Softw.*, *69*, 46–61. doi: 10.1016/j.advengsoft.2013.12.007

Mohebbanaaz, Kumari, L. V. R., & Sai, Y. P. (2022). Classification of ECG beats using optimized decision tree and adaptive boosted optimized decision tree. *Signal Image Video Process.*, *16*(3), 695–703. doi: 10.1007/ s11760-021-02009-x

Moody, G. B., & Mark, R. G. (2001). The impact of the MIT-BIH arrhythmia database. *IEEE Eng. Med. Biol. Mag.*, *20*(3), 45–50. doi: 10.1109/51. 932724

Murat, F., Yildirim, O., Talo, M., Baloglu, U. B., Demir, Y., & Acharya, U. R. (2020). Application of deep learning techniques for heartbeats detection using ECG signals-analysis and review. *Comput. Biol. Med.*, *120*, 103726. doi: 10.1016/j.compbiomed.2020.103726

National Heart, Lung, and Blood Institute (2022, May 25a). Arrhythmias. https://www.nhlbi.nih.gov/health/arrhythmias

National Heart, Lung, and Blood Institute (2022, May 25b). Heart tests. https://www.nhlbi.nih.gov/health/heart-tests

Nguyen, Q. H., Nguyen, B. P., Nguyen, T. B., Do, T. T. T., Mbinta, J. F., & Simpson, C. R. (2021). Stacking segment-based CNN with SVM for recognition of atrial fibrillation from single-lead ECG recordings. *Biomed. Signal Process. Control*, *68*, 102672. doi: 10.1016/j.bspc.2021.102672

Pandey, S. K., Janghel, R. R., Dev, A. V., & Mishra, P. K. (2021). Automated arrhythmia detection from electrocardiogram signal using stacked restricted Boltzmann machine model. *SN Appl. Sci.*, *3*(6), 624. doi:10.1007/s42452-021-04621-5

Passos, L. A., & Papa, J. P. (2020). A metaheuristic-driven approach to fine-tune deep Boltzmann machines. *Appl. Soft Comput.*, *97*, 105717. doi: 10.1016/j.asoc.2019.105717

Petmezas, G., Haris, K., Stefanopoulos, L., Kilintzis, V., Tzavelis, A., Rogers, J. A., Katsaggelos, A. K., & Maglaveras, N. (2021). Automated atrial fibrillation detection using a hybrid CNN-LSTM network on imbalanced ECG datasets. *Biomed. Signal Process. Control*, *63*, 102194. doi: 10.1016/j.bspc.2020.102194

PhysioNet (2022, May 25). MIT-BIH arrhythmia database. https://www.physionet.org/content/mitdb/1.0.0/

Ramasamy, K., Balakrishnan, K., & Velusamy, D. (2022). Detection of cardiac arrhythmias from ECG signals using FBSE and Jaya optimized ensemble random subspace k-nearest neighbor algorithm. *Biomed. Signal Process. Control*, *76*, 103654. doi: 10.1016/j.bspc.2022.103654

Salcedo-Sanz, S., Del Ser, J., Landa-Torres, I., Gil- López, S., & PortillaFigueras, J. A. (2014). The coral reefs optimization algorithm: A novel metaheuristic for efficiently solving optimization problems. *Sci. World J.*, *2014*, 739768. doi: 10.1155/2014/739768

Storn, R., & Price, K. (1997). Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.*, *11*(4), 341–359. doi: 10.1023/a:1008202821328

Tsai, C.-W., Hsia, C.-H., Yang, S.-J., Liu, S.-J., & Fang, Z.-Y. (2020). Optimizing hyperparameters of deep learning in predicting bus passengers based on simulated annealing. *Appl. Soft Comput.*, *88*, 106068. doi: 10.1016/j.asoc.2020.106068

World Health Organization (2022, May 25). Cardiovascular diseases. https://www.who.int/health-topics/cardiovascular-diseases

Yang, L., & Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, *415*, 295–316. doi: 10.1016/j.neucom.2020.07.061

Yang, X.-S. (2009). Firefly algorithms for multimodal optimization. In *International symposium on stochastic algorithms (SAGA 2009)* (pp. 169–178). Springer. doi: 10.1007/978-3-642-04944-6_14

Yang, X.-S. (2020). *Nature-inspired optimization algorithms*. Academic Press.

Yang, X.-S., & Deb, S. (2009). Cuckoo search via Lévy flights. In *2009 world congress on nature and biologically inspired computing (NaBIC)* (pp. 210–214). IEEE. doi: 10.1109/nabic.2009.5393690

Zeng, W., Yuan, J., Yuan, C., Wang, Q., Liu, F., & Wang, Y. (2021). A novel technique for the detection of myocardial dysfunction using ECG signals based on hybrid signal processing and neural networks. *Soft Comput.*, *25*(6), 4571–4595. doi: 10.1007/s00500-020-05465-8

Zhao, W., Wang, L., & Mirjalili, S. (2022). Artificial hummingbird algorithm: A new bio-inspired optimizer with its engineering applications. *Comput. Methods Appl. Mech. Eng.*, *388*, 114194. doi: 10.1016/j.cma.2021.114194