DEVELOPMENT OF AN OPTIMAL FEATURE SELECTION SCHEME FOR EPILEPSY CLASSIFICATION FROM DISRUPTIVE ELECTROENCEPHALOGRAM SIGNAL USING AN IMPROVED GRASSHOPPER OPTIMIZATION ALGORITHM

BY

UMAR, Buhari Ugbede PhD/SEET/2016/869

DEPARTMENT OF COMPUTER ENGINEERING
FEDERAL UNIVERSITY OF TECHNOLOGY, MINNA,
NIGERIA

FEBRUARY, 2023

DEVELOPMENT OF AN OPTIMAL FEATURE SELECTION SCHEME FOR EPILEPSY CLASSIFICATION FROM DISRUPTIVE ELECTROENCEPHALOGRAM SIGNAL USING AN IMPROVED GRASSHOPPER OPTIMIZATION ALGORITHM

ABSTRACT

Epilepsy is a common type of disorder that causes recurrent seizures and affects approximately 70 million people worldwide. One of the common diagnostic tools is the Electroencephalogram (EEG). EEG is an extremely complex signal that holds information about the various activities of the human brain and neurologists inspect the EEG recordings of an epilepsy patient to identify and analyze epileptic seizures. However, most seizures occur unexpectedly, and finding ways to detect a possible seizure before it happens has been a challenging task for many researchers. This is because the detection of epileptic seizures requires visual monitoring of a patient's EEG recordings for hours or even days, thus making it a laborious and time-consuming process, and whose outcome may be affected by the experience of the neurologists. As each channel or electrode implanted in the brain provides different statistical measures. A critical issue in epilepsy classification is the selection of suitable statistical features. This necessitated the development of a metaheuristic-based effective and improved grasshopper optimization algorithm (IGOA) using elite opposition-based learning and exponential switching parameters between local and random walks for updating the value of the Grasshopper Optimization Algorithm for the optimizations of feature selection for epilepsy classification from disruptive EEG signals. The original Grasshopper Optimization Algorithm (GOA) was developed using linear switching parameters for updating the iteration value of the Grasshopper Optimization Algorithm, which lead to premature convergence in some complex optimization techniques and drawbacks in exploiting the search space. The IGOA was tested on 14 test functions (unimodal and multimodal benchmark functions) and used to optimize a feedforward artificial neural network for epilepsy classification. From the result, the IGOA outperformed the original GOA in terms of best optimal value, worst, mean and standard deviation and effectively balancing the exploitation and exploration search space. Grasshopper Optimization AlgorithmArtificial Neural Network (GOA-ANN), Particle Swamp Optimization-Artificial Neural Network (PSO-ANN), Salp Swarm Optimization Algorithm-Artificial Neural Network (SSOAANN), Bat Algorithm-Artificial Neural Network (BA-ANN) and Grey Wolf Optimization Algorithm-Artificial Neural Network (GWOA-ANN) were evaluated and compared with IGOA-ANN for their classification accuracies, the number of search agents and features extraction. Also, the result was compared with similar results in the literature. The results showed the classification accuracies performance: IGOA-ANN (99.6%), GOA-ANN (99.40%), GWOA-ANN (98.40%), SSOA-ANN (98.40%), BAANN (98.80%) and PSO-ANN (99.0%) respectively. Based on the previous studies presented in the literature using the University of Bonn EEG dataset, the IGOA-ANN method produced better sensitivity (99.60%), precision (99.60%), and accuracy (99.60%). Considering these metrics and the fact that it requires minimum feature extraction, the IGOA-ANN optimized approach makes it an efficient method for epilepsy classification. The method will help neurologists with efficient and accurate epilepsy classification, thereby saving time.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER ONE

1.0 INTRODUCTION

1.1 Background to the Study

Epilepsy, which ranks fourth behind Alzheimer's disease, migraines, and stroke, is the most common neurological disorder and the most deadly ailment. It affects millions of people worldwide and is brought on by a malfunction of the brain's Central Nervous System (CNS). Seizures, which are bursts of aberrant brain electrical activity, are its defining feature (Abbasi *et al*, 2019; Currey *et al.*, 2021; Oliva & Rosa, 2019; Rasheed *et al*., 2020; Singh *et al.*, 2019; Walter *et al.*, 2020).

Since most seizures occur unexpectedly, it is difficult to predict when someone could have one. Many researchers have struggled to come up with methods to predict a seizure before it occurs. Using the EEG data from an epilepsy patient, classification algorithms can be utilized to forecast whether or not someone will experience a seizure (Almustafa, 2020). Contrarily, classifying epileptic seizures requires hours or even days of visual observation of significant chunks of an individual patient's EEG records, making it a tedious and timeconsuming process. Furthermore, because manual scanning-based diagnoses can differ when carried out by several neurologists based on their observations and experiences, they may be inaccurate (Chakraborty & Mitra, 2021). The adoption of accurate seizure classification and forecasting algorithms can considerably lessen the problem of epilepsy, as reliable and simple seizure classification is the key to developing devices to treat epilepsy (Larmuseau, 2016).

The creation of computer-aided diagnostic (CAD) processes is essential for handling big datasets, cutting down on the time needed to classify seizures, and giving doctors a second perspective on how to diagnose the condition (Chakraborty & Mitra, 2021). Metaheuristic algorithms are one such method for streamlining the categorization process. To select the best features for improved epilepsy classification from EEG signals, this research presents an improved Grasshopper Optimization Algorithm (IGOA) and artificial neural network for optimal feature selection for enhanced epilepsy classification from EEG signals.

1.2 Statement of the Research Problem

Using an electrode or channel put in the brain, which provides unique statistical measurements, the EEG data is obtained. One of the trickiest parts of diagnosing and categorizing epilepsy is choosing the right statistical features. Undoubtedly, earlier scholars worked very hard to determine the optimum characteristics, and some researchers used a variety of attributes. Amin *et al*., (2015); Logesparan *et al*., (2012), others applied a few features (Boonyakitanont *et al*., 2020; Esteller *et al*., 2001; Hadoush *et al*., 2019; Quintero-Rincón *et al*., 2019) for detecting the seizure. By examining the statistical qualities of the features, it is crucial to recognize the various statistical viewpoints that might be taken on each brain signal. The main goal is to acquire only the crucial features by preventing a low-dimensional dataset situation that would make the knowledge discovery process ineffective (Amin et al., 2015; Logesparan *et al*., 2015; Zhang *et al*., 2015). This is crucial for the classification process since, depending on the qualities and needs of the dataset, each classifier has advantages and disadvantages (Fernandez-Delgado, *et al*., 2014). To identify the capable classifier (one that excels at solving seizure detection and imparting information finding), several classifiers have been tried on EEG datasets, and their performance has been evaluated using both a "black-box" method and a "whitebox" approach (flaw in them is their inability to provide adequate explanations for the patterns and logic rules concealed within the models). The "non-black-box technique" is also being used increasingly and frequently (Birjandtalab *et al*., 2017; Donos *et al*., 2015).

As the study for clinical epilepsy detection and classification continues, the number of parameters used to construct the classifier, independent of the kind of classifier, has a substantial impact on the classification performance. Therefore, the search for efficient biomarkers is crucial, particularly when EEG recordings are influenced by a multitude of physiological parameters, making it more difficult to differentiate between different brain states (Ong *et al*., 2018; Zainuddin *et al*., 2012). As a result, this research offers an Improved Grasshopper Optimization Algorithm (IGOA) for optimal feature selection from EEG signals using Artificial Neural Networks for epilepsy classification. It is predicted that using Grasshopper Algorithms for feature selection and an artificial neural network (using its data-driven approach) to classify epilepsy will be effective and efficient.

1.3 Research Question

In carrying out this research, some pertinent issues require answers. These includes:

i. Can the first-order and second statistical features be extracted from the EEG signal for epilepsy classification?

ii. Can the standard Grasshopper optimization Algorithm (GOA) be improved using elite opposition-based learning and exponential switching parameters between local and random walks for updating its value?

iii. Can the improved Grasshopper Optimization Algorithm using elite oppositionbased learning and exponential switching parameters between local and random walks perform better than the standard Grasshopper Optimization Algorithm, Bat

Algorithm (BA), Particle Swarm Optimization (PSO), Grey Wolf Optimization Algorithm (GWOA) and Salp Swarm Optimization Algorithm (SSOA) using

accuracy, sensitivity, specificity, precision, F1 score, recall as metrics? iv. Can a graphical user interface-based system for epilepsy classification based on objective two be developed using the University of Bonn Epilepsy dataset?

v. Can a graphical user interface-based system develop using an Improved Grasshopper Optimization Algorithm and Artificial Neural Network (IGOA-ANN) for epilepsy classification problems perform better results or similar results when compared to existing research in the literature that uses the University of Bonn epilepsy dataset?

1.4 Research Aim and Objectives

The aim of this research is to develop an optimal feature selection scheme for epilepsy classification based on IGOA using elite opposition-based learning and exponential switching parameters between local and random walks for updating the value of the Grasshopper Optimization Algorithm using the Artificial Neural Network approach. To realize the aforementioned aim, the following objective will be explored:

i. To acquire epilepsy dataset from the University of Bonn Epilepsy dataset and extract first and second-order statistical features for faster and more accurate epilepsy detection.

ii. To develop an improved GOA using elite opposition-based learning and exponential switching parameters between local and random walks for updating its value.

iii. To evaluate the performance of the IGOA on different benchmark test functions in comparison with the standard GOA, Bat Algorithm (BA), Particle Swarm Optimization (PSO), Grey Wolf Optimization Algorithm (GWOA), and Salp Swarm Optimization Algorithm (SSOA) using accuracy, sensitivity, specificity, precision, F1 score, recall as metrics.

iv. To develop a graphical user interface-based system for epilepsy classification based on 2 and evaluate its performance using the University of Bonn Epilepsy dataset.

v. To evaluate the performance of the developed GUI by benchmarking it against existing research in the literature that uses the University of Bonn epilepsy dataset.

1.5 Research Justification

The term "epileptic seizure" describes a group of conditions marked by recurrent cerebral cortex discharges that result in erroneous brain activity. Each year, 2.4 million new cases are anticipated to be reported worldwide (Rahman & Karim, 2015). Medication-resistant epilepsy affects about 30–40% of the population and may require surgical surgery for either curative or palliative purposes. The thought of suffering uncontrollable seizures while going about their daily lives can be unsettling for many people. Recent years have seen a significant number of studies on methods for detecting and anticipating seizures. Patients with refractory epilepsy will benefit greatly from this, since it may allow them to take precautionary measures to avoid damage or attempt to take medicine to prevent seizures.

Neurologists always use EEG records to investigate suspected seizure occurrences (Alam

& Bhuiyan, 2013; Gadhoumi *et al*., 2015; Kumar *et al*., 2014; Omerhodzic *et al*., 2013; Rajeev & Pachori, 2015). Traditionally, an expert neurologist would perform this task by visually scanning EEG patterns, which is time-consuming and a potentially erroneous approach. The errors are especially noticeable in EEG signals that last for a long time (Rahman & Karim, 2015). To reduce the likelihood of these errors, an automated seizure prediction system can transform primarily qualitative diagnostic criteria into a more objective quantitative signal feature categorization task. The system's objective is not to take the role of the neurologist, but rather to relieve him of the laborious (and potentially error-prone) observation task and enhance the detection and classification procedures as a whole.

1.6 Research Scope

For feature selection, epilepsy detection, and classification, many methods and algorithms were applied. Currently, the epilepsy detection and prediction community are yet to produce a clinical outcome, and developing a device for clinical prediction is quite costly. As a result, the new optimization feature selection will be limited to testing utilizing ANN accuracy, sensitivity, specificity, precision, F1 score, and recall in the MATLAB environment. The method does not necessitate the development of an experimental system for clinical testing, rather it will use epilepsy data set from the university of Bonn.

1.7 Thesis Organization

Introduction, research problem, research questions, the general context of the research, research aim and objectives, research justification as well as the scope of the research are presented in the first chapter. The fundamentals of epilepsy and its types, EEG signals and their types, and current research on epilepsy detection and classification are covered in chapter two. The basic concepts of feature extraction and performance are also discussed. The approach used to achieve each of the research work's objectives are outlined in chapter three. In chapter four, the obtained results, and discussion are provided. In chapter five, the whole research work's summary, conclusions, and suggestions are presented.

CHAPTER TWO

2.0 LITERATURE REVIEW

2.1 Introduction

Recent research efforts have focused on applying machine learning to provide improved clinical, reliable, and efficient epilepsy detection. This chapter provides an overview of various epilepsy classification methodologies, feature selection methods, optimization algorithms, and their various forms, as well as the benchmark function. The subjects of epilepsy, EEG monitoring, machine learning, and a review of relevant publications are discussed. In particular, the application of optimization algorithms for feature extraction and selection in recent years.

2.2 Epilepsy Seizures

Epilepsy Seizures refer to a phenomenon that occurs when there are abnormal and excessive electrical discharges in the brain network, triggering transient or short-lived sudden attacks, spasms, or events characterized by clinical symptoms (Panayiotopoulos, 2010). A seizure can occur mildly in the form of a muscle jerk of infinitesimal magnitude, or in severe form as convulsions that are widespread and last a long time (Sharmila, 2018). The various types of seizures, based on symptoms, as shown in Figure 2.1, are partial and generalized seizures.

i. **Partial Seizures**- This is the most common of all seizures. It affects one side of the brain when it occurs. As such, if the seizure affects the part of the brain responsible for speech, the affected person is unable to talk. It is of two types:

   a) Simple partial seizure: In the period of experiencing this type of seizure, the affected person is awake and aware (Sivasankari *et al*., 2010).

   b) Complex partial seizure: This type of partial seizure usually occurs in one of the two temporal lobes of the brain.

There are, however, situations when partial seizures graduate into generalized seizures. A partial seizure of this nature is known as a partial seizure secondarily generalized.

ii. **Generalized Seizures**- In this specific seizure type, the "early clinical alterations indicate beginning movement of both hemispheres." (Chang *et al*., 2017). It affects both sides of the brain. It is of many types which can be classified into two categories, which are; Generalized Convulsive Seizure and Generalized NonConvulsive Seizure.

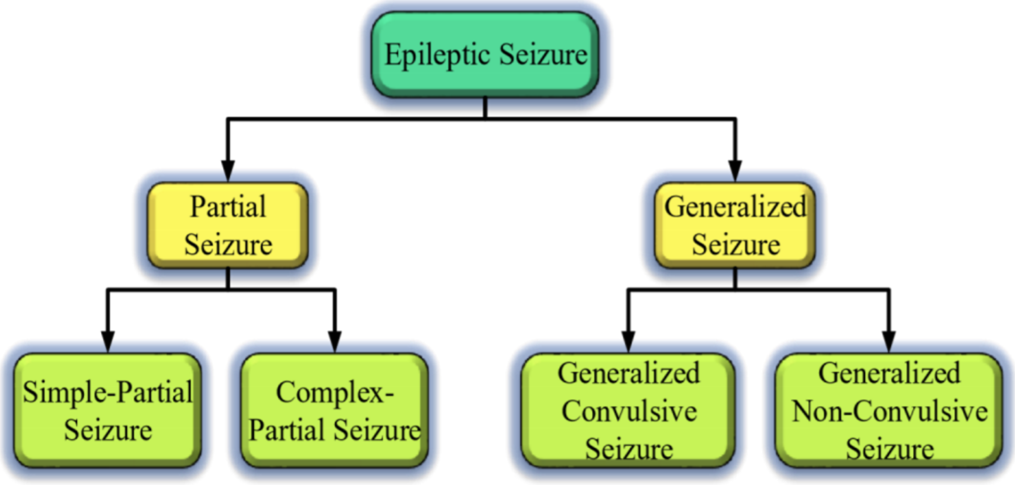

Figure 2.1: Types of Epileptic Seizures

However, Fisher, (2017) states that "any seizure type classification will be operational (practical) and observational" and therefore, presented an expanded organization of the various types of seizures, which are; Onset types include Focal, Generalized, and Unknown.

a) Focal Onset Seizures.

This is a type of seizure that starts on one side of the brain with a change in the level of consciousness of the affected person. It occurs in the form of automatisms or autohypnosis which may include actions like chewing movements, smacking of lips, and rubbing of hands.

b) Generalized Onset Seizures.

This is a type of seizure that is generalized from the start. It begins simultaneously in both hemispheres of the brain. The patient is usually unable to describe the seizure as they have no memory of it.

c) Unknown Onset Seizures.

This refers to the types of seizures which cannot be placed in any of the other categories due to inadequate information (Fisher, 2017). Figure 2.2 better shows these various types of classification.

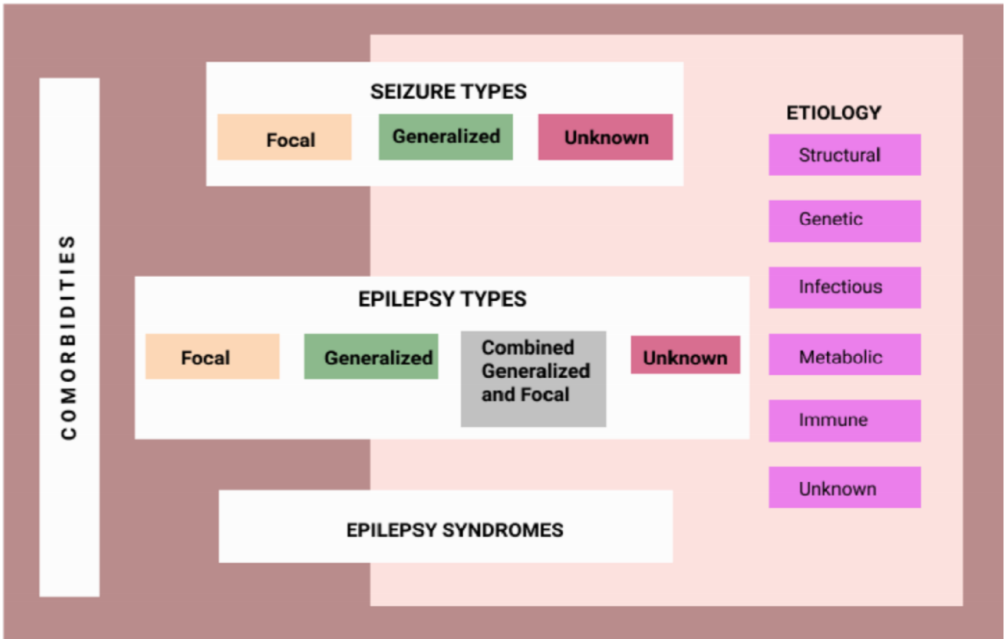

Figure 2.2: Seizure Types Classification

2.3 Electroencephalography (EEG)

Electroencephalography is a technique for determining the electrical potential of the brain (Kumar & Bhuvaneswari, 2012). To capture electrical activity, a device with electrodes is positioned on the scalp of the brain. The EEG produces a signal that is a high dimension, noisy, and redundant when it records electrical activity (Li *et al*., 2018). These waves, which range in amplitude and frequency and represent different activities like sleep, rest, waking, diseases, and so forth, make up the signals. Specific EEG patterns, in general, represent a normality standard, while deviations from this standard represent abnormality (Hively & Protopopescu, 2003). The superposition of brain activities that are recorded as electrical potential fluctuations on the scalp is represented by the EEG signal. The EEG waves capture a wealth of information about how the brain functions. A significant number of electrical potentials from diverse sources, including brain cells, neurons, and artifacts, are superimposed in the EEG acquired from scalp electrodes (Nayak & Cholayya, 2006). The electrooculogram (EOG) signal is activated by eye movements or blinks, making it the main and most prevalent artifact in EEG analysis (Sanei & Chambers, 2013).

2.3.1 EEG signals

The signal produced by the EEG falls under a class of some pre-defined frequency bands in the frequency spectrum. These includes; Alpha, Beta, Delta, and Theta. They are also known as EEG patterns (Snyder, 1990). Table 2.1 gives a summary of these EEG patterns (signal types) as well as the behavioral state associated with each of them, while an illustration of many forms of typical EEG rhythms is shown in Figure 2.3.

Table 2.1*: EEG Patterns and Behavioral States (Khosla* et al., 2020; Snyder, 1990)

S/N	EEG Pattern/Rhythm	Frequency Range (Hz)	Corresponding Behavioral State
1.	Delta	0.1 – 3.5	Hypnagogic, visual imagery, light sleep.
2.	Theta	4 – 7.5	Vague dream states, deep, restful sleep.
3.	Alpha	8 - 13	Low level of environment arousal, awake, non-focused, comfortable, drowsy, or nonvigilant (eyes closed).
4.	Beta	14 - 30	Dream/REM sleep, being awake and alert, paying close attention and solving problems, and being highly stimulated by the environment (eyes open).
5.	Gamma	>30	Increased focus and attention.

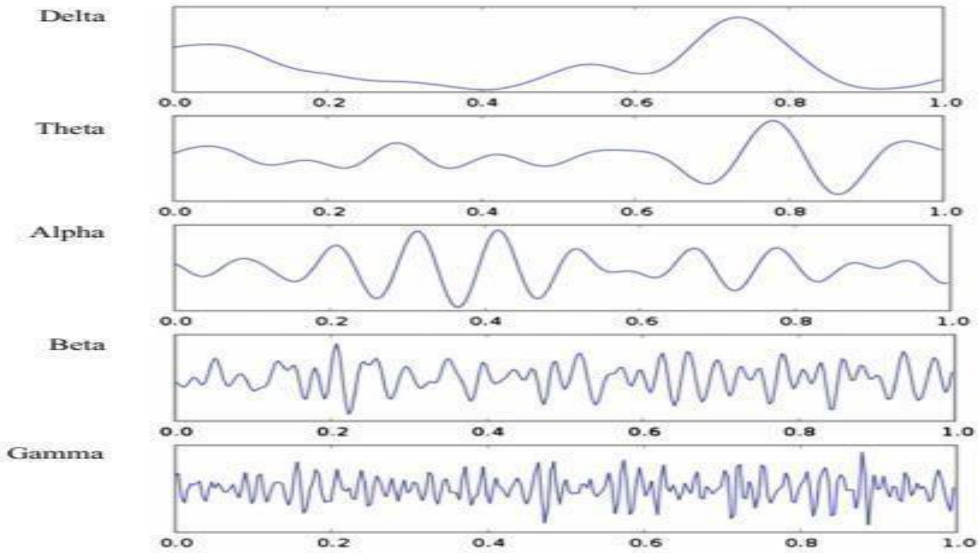

Figure 2.3: Example of Different Types of Normal EEG Rhythm

2.3.2 EEG signal analysis

Because there is a lot of data obtained when an EEG measurement is taken, there is a need for the provision of schemes and tools that allow for automated feature extraction, analysis, and classification of the data contained in the EEG signals (Siuly *et al*., 2016). There have been several approaches proposed, which can be categorized into five groups: similarity measurements, neural networks, synchronization measures, statistical analysis, and correlation-based techniques (Sivasankari *et al*., 2010). The procedure for algorithm-based

EEG signal analysis is depicted in Figure 2.4.

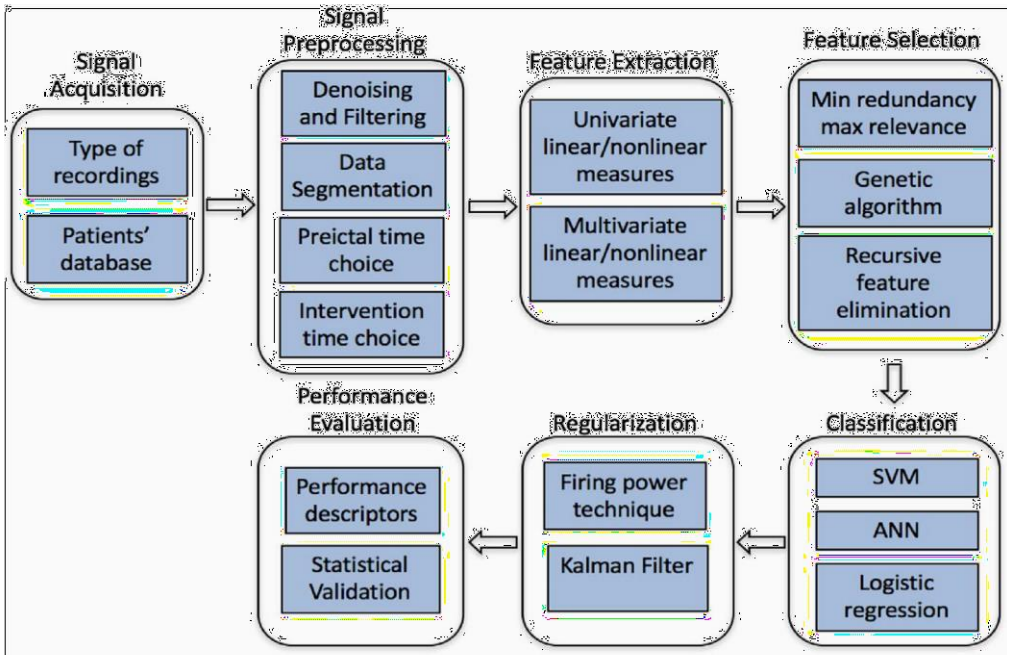

Figure 2.4: Overview of the process of algorithmic-based EEG signal analysis (Assi *et al.*, 2017)

2.4 Epilepsy Dataset

There are several datasets of EEG signals available today to researchers. These datasets provide researchers with an opportunity to extract robust features and understand the general structure of epilepsy seizures (Hossain *et al.*, 2019) so that they can come up with better models and systems for the prediction and detection of epilepsy using EEG signals.

This section, therefore, reviews the various publicly available EEG datasets.

2.4.1 University of bonn dataset

This dataset has five subjects with a total of 100 items each and a sampling frequency of 173.61 Hz. Each of the 100 sets contains 4096 samples from a single ASCII-coded EEG time series (*Epileptologie Bonn / Forschung / AG Lehnertz / EEG Data Download*). About

40 minutes of EEG data was provided for each dataset. Ihle *et al*., (2012); Türk & Özerdem,

(2019) adopted this dataset with a scalogram based Convolutional Neural Network (CNN) to allow for the detection of an epileptic seizure. Ahammad *et al*., (2014) adopted this dataset with wavelet-based features to allow for the detection of an epileptic seizure. Zhao *et al*., (2020) also, adopted this same dataset with a deep neural network with a 5-type layer to allow for robust detection of epileptic seizures.

2.4.2 Freiburg-EEG dataset

This dataset was proposed in the year 2000 and was made available to researchers working on seizure prediction (Gadhoumi *et al*., 2016). It contained invasive long-term intracerebral EEG recordings obtained from 21 focal epilepsy patients (13 patients with 24 hours continuous and 8 patients with discontinuous) interictal recordings, containing a total of

582 hours of EEG data. The recordings had a 256 Hz sampling rate (Ihle *et al*., 2012). Sharif & Jafari, (2017) adopted this dataset together with an SVM classifier and fuzzy rules distribution as predictive characteristics to allow for the prediction of epileptic seizures. Yang *et al*., (2018) used the same dataset and an SVM classifier to predict epileptic seizures using permutation entropy as a predictive factor.

2.4.3 Bern-Barcelona-EEG dataset

Consists of 83 hours of 512 Hz-sampled intracranial EEG recordings made from five epileptic patients (Shoeibi *et al*., 2020). This dataset was recorded using intracranial electrodes placed on five patients with focal epilepsy (Lu & Triesch, 2019). It is made up of different parts that different EEG signal analysis specialists have accepted. Lu & Triesch, (2019) adopted this dataset with a CNN that had residual connections, to allow for the classification of EEG signals.

2.4.4 Flint-hills dataset

EEG recordings at a sample rate of 239.74 Hz make up this dataset with a total of 1419 hours of continuous intracranial recordings from 10 patients and 59 seizures (Ihle *et al*., 2012; Assi *et al*., 2017).

2.4.5 Kaggle dataset

This dataset consists of Intracranial EEG recordings obtained from 5 dogs (having naturally occurring epilepsy) with a 400Hz sampling frequency and from two epileptic patients with a sampling rate of 5kHz (Zhang & Parhi, 2015). Larmuseau, (2016) adopted standard deviation techniques with a Recurrent Neural Network (RNN) as a classifier to allow for the prediction of epileptic seizures. Dadgar-Kiani *et al*., (2016) also adopted this dataset with Fast Fourier Transform (FFT) techniques and an SVM classifier to allow for the prediction of an epileptic seizure.

2.4.6 Zenodo dataset

Contains Multi-channel EEG recordings obtained from 79 neonatal epileptic seizure patients, with an average record duration of 74 minutes and a sampling rate of 256Hz (Shoeibi *et al*., 2020; Stevenson, Tapani *et al*., 2019). Clinical data collation, conversion to EDF format, and the reading of the converted EDF into annotated montage are some of the processes that were carried out in obtaining the dataset (Stevenson *et al*., 2019).

2.4.7 Hauz khas dataset

Consists of scalp EEG recordings made at a sampling rate of 200Hz from 10 patients with epilepsy (Shoeibi *et al*., 2020).

2.4.8 CHB-MIT dataset

This dataset contains about 844 hours of EEG-ECG data that was obtained from 24 patients and synchronized , with a 256 Hz sampling rate (Shoeb & Guttag, 2010). *Alotaiby et al*.,

(2015) adopted this dataset together with a CSP-based feature extraction technique, with a Linear Discriminant Analysis classification to allow for the ability to anticipate epileptic seizures. On the other hand, Usman & Hassan, (2018) adopted this dataset together with a Large Laplacian Spatial Filter, with a Naïve Bayes classifier to allow for prediction of epileptic seizure. Fergus *et al*., (2015) adopted it with a Root mean square and Bandpass filter techniques together with an SVM classifier to allow for the prediction of epileptic seizures. Usman & Hassan, (2018) also adopted the dataset with variance and skewness as predictive characteristics and an SVM classifier to achieve the prediction of epileptic seizures.

2.4.9 Boston dataset

Consists of 256 Hz-sampled scalp EEG recordings made over 940 hours from 23 pediatric epileptic patients. Researched with this dataset to allow for early detection of autism spectrum disorder (Assi *et al*., 2017; Bosl, Tager-Flusberg, & Nelson, 2018). Table 2.2 gives a summary of some of the various available datasets, showing their sampling rate, the total length of the recordings, and the number of patients. The frequency with which each dataset has been used in various evaluated papers is shown in Figure 2.5.

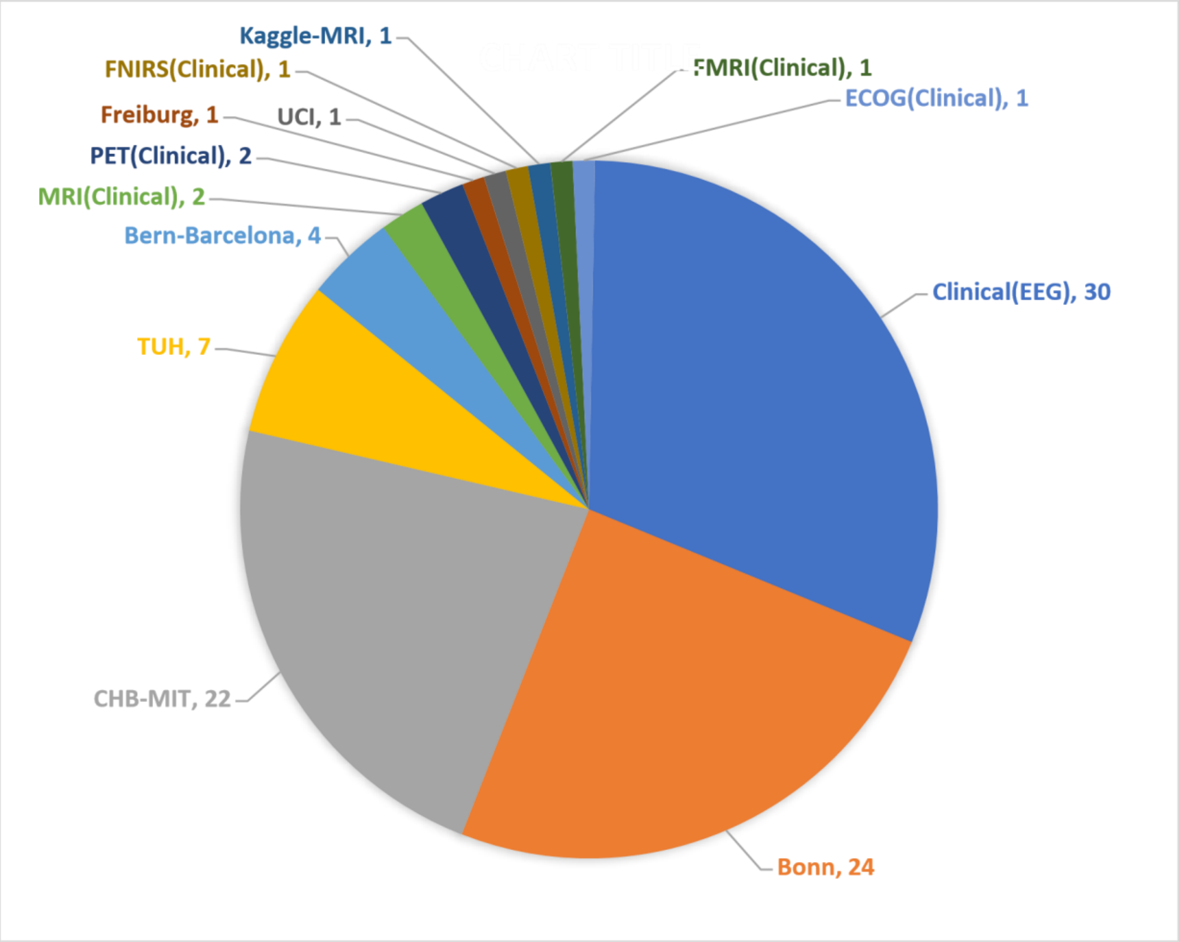

Figure 2.5: Number of Times each Dataset has been used in Various Research Works

Table 2.2: Various Epileptic Seizure Datasets (Assi *et al*., 2017; Bosl *et al*., 2018)

S/N				Dataset Number Duration of (Hours)		Recording Sampling Frequency of Patients	Total Number (Hz)	Seizures
1.	University of Surface and Bonn EEG minutes			39	5	173.61	-	
2.	Freiburg EEG	Intracranial EEG (iEEG)	582	21	256	87		
3.	Bern Barcelona	iEEG 83	5	512	3750 EEG			
4.	Flint-Hills	Continuous Intracranial long-term ECoG		1419	10	239.74	59	
5.	2 Kaggle 5 dogs	5000 48 iEEG 627 400						
6.	Zenodo	Scalp EEG (sEEG)	74	79 minutes	256	460 Neonatal		
7.	Hauz Khas	sEEG -	10	200	-			
8.	CHB-MIT	sEEG	844	24	256	163		
9.	Boston	Scalp EEG	940	23 Pediatric	256	198		
10.	TUH iEEG	iEEG - 10874	250	~14777 Surface and		173.61	-	

[11] Bonn 39 min 10

2.5 Feature Selection Methods

Features that are inappropriate, unnecessary, or extraneous are the focus of feature selection. It is a method for obtaining the most valuable features from datasets (Liu & Yu, 2005). The problem of feature selection in machine learning is one of the most crucial and challenging. The feature selection problem has many applications in a variety of industries. Among the challenges are those related to biomedicine, such as selecting the best gene from a candidate gene (Ahmed *et al*., 2013), text mining (Aghdam *et al*., 2009), picture analysis (Ghosh *et al*., 2013), and choosing the best visual contents (pixels, color), among others.

The workings of the feature selection procedure are depicted in Figure 2.6. The original dataset, feature subset selection, feature selection algorithm, selection criterion, and validation are the five essential steps in the feature selection process that are depicted in the diagram. To choose the best subset of features, numerous feature selection techniques have been developed. The three categories of techniques include filter, wrapper, and embedding approaches, as shown in Figure 2.6 (Hoque *et al*., 2014). Filtering methods are unaffected by learning or classification algorithms. It is constantly concentrated on the general attributes of the data (Xu *et al*., 2010).

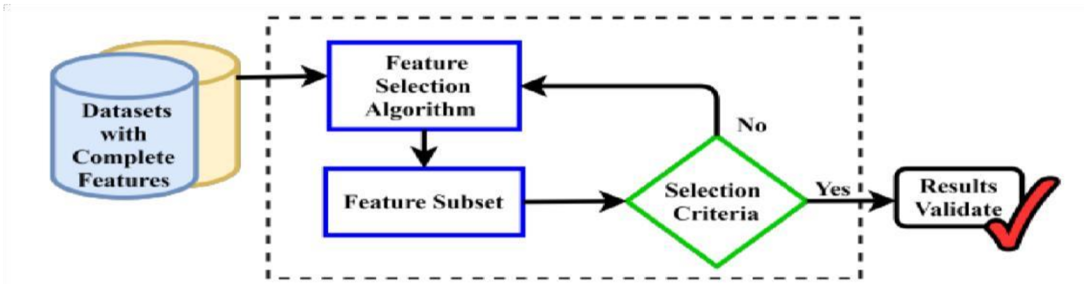

Figure 2.6: The process through which features are chosen is operational (Agrawal *et al*., 2021)

Wrapper approaches usually include the classifier's classification algorithm and interact with it. These algorithms require more processing power than filters, but they also produce more accurate results. Filters and wrapper methods are combined to generate embedded methods. The training procedure for embedded techniques includes feature selection and is carried out concurrently with the classifier. Additionally, because embedded methods operate by using a learning algorithm, they will be categorized as wrapper approaches (Tang *et al*., 2014).

Wrapper approaches are slower than filter methods but offer better results. The modeling technique, which creates and assesses each subset, is used by wrapper approaches. To produce subsets, wrapper techniques employ a separate search methodology and sort the three categories of exponential, sequential, and randomized search strategies (Jovi *et al*., 2015). In the exponential technique, the number of features that are investigated grows exponentially with feature size. Even though this method yields trustworthy results, its high computing cost makes it impractical to use. Exhaustive search, branch, and bound approach are examples of exponential search strategies (Sun *et al*., 2004). Sequential algorithms logically add or remove attributes. Local optima are produced when a feature is added to or removed from a subset because it cannot be modified again. Sequential algorithms include things like best first, linear forward selection, floating forward or backward selection, and others. Randomness is used to move around the search space in randomized algorithms to prevent them from getting stuck in local optima. Randomized algorithms include population-based methods like simulated annealing, random generation, and metaheuristic algorithms. Types of Feature Extraction Techniques are shown in Figure 2.7.

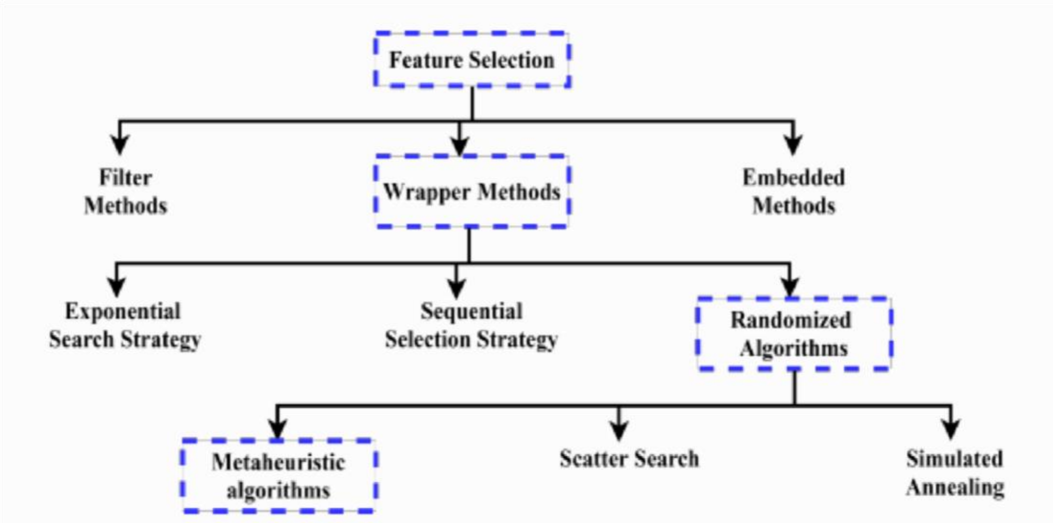

Figure 2.7: Types of Feature Extraction Techniques (Agrawal *et al*., 2021)

Because of their properties, metaheuristic algorithms attract a lot of attention from researchers. Different types of issues have been solved using a variety of algorithms. Even though metaheuristic algorithms have had a lot of success in tackling feature selection difficulties, there are some hurdles and issues with scalability and stability in feature selection. A dataset in real-world problems can have dozens or even millions of features. The suggested approach must be scalable to handle huge datasets in the feature selection challenge. A good scalable classifier that can handle enormous datasets is required in the method (Bolón-Canedo *et al*., 2018). So, when developing an algorithm to handle the feature selection problem, scalability is essential. When designing an algorithm to manage feature selection difficulties, stability is an important factor to take into account. A method is said to be stable for feature selection when it consistently identifies the same subset of features across various dataset samples. In most cases, the feature selection approach becomes unstable when attempting to get the best categorization. Instability occurs when features with a strong correlation are deleted to get the best classification accuracy. As a result, consistency is just as crucial as categorization accuracy. Khaire & Dhanalakshmi, (2019) describe a possible plan of action for the feature selection challenge's algorithm stabilization.

Despite the shortcomings of metaheuristic algorithms, improved versions of the algorithms have been developed and effectively applied to solve feature selection issues.

2.6 Feature Extraction from EEG Signal

Feature extraction is a process carried out on EEG signals to obtain specific information or attributes from the signal, in a way that makes it possible for the features to be easily interpreted (Azlan & Low, 2014). The extraction of features from the various EEG datasets has been done using different techniques, with the majority of the techniques falling into one of four categories (Acharya *et al*., 2013), which are:

i. Frequency domain methods.

The various techniques under this approach use statistical and Fourier transform (FT) to examine in-depth, information and details that exist in the frequency domain (Acharya *et al.*, 2013).

ii. Time-domain methods.

Component analysis and linear prediction are two essential methods for time-domain analysis. The component analysis is an unsupervised technique for linking a set of data to a set of features. Principal, linear, and independent component analysis is one of the component analysis techniques utilized in epilepsy diagnosis (Duda & Hart, 1973), whereas the linear prediction method focuses on forecasting the outcome of a linear system using the current input and the prior outputs (Acharya *et al.*, 2013). iii. Time-Frequency domain methods.

This group of methods for EEG signal analysis is made up of 2 categories; Wavelet Transform and Hilbert-Huang Transform. The Wavelet Transform is a multi-scale extension of the Fourier Transform that solves non-stationary signal problems (Kumar, Alam, & Siddiqi, 2017). On the other hand, the Hilbert-Huang Transform allows for instantaneous frequency data to be obtained from a signal by the decomposition of the signal into Intrinsic mode functions (IMFs) (Acharya *et al*., 2013). iv. Nonlinear methods.

The various methods listed beforehand cannot properly detect phase locking or nonlinear coupling that may exist among harmonics of the same spectrum (Acharya *et al*., 2009). To overcome this limitation, various nonlinear methods like Higher Order Spectra (HOS), Sample Entropy (SampEn), and Recurrence Quantification Analysis (RQA), have been developed (Acharya et al., 2013).

2.7 Types of Features in EEG Signals

Since the quality of the features that are extracted from the EEG data will have a significant impact on the system's performance, feature extraction is given a lot of consideration throughout the creation of any EEG seizure detection and prediction system (Boubchir *et al*., 2017). Consequently, this section provides a summary of the many properties that can be drawn from EEG data. The different properties that can be retrieved from EEG data can be categorized into one of three groups, as suggested by (Boubchir *et al*., 2017); a) EEG Time-domain features.

The features that are taken out of the raw EEG data are contained in this category (EEG in the time domain). The various features under this category can be further divided into 3 classes (Boubchir *et al*., 2017), which are:

i. Time-domain features based on Statistical Moment: Features such as normalized moments, coefficient of EG signal variation, first moment, and second central EEG signal moment are the various features that fall under this class of time-domain features (Aarabi *et al*., 2006).

ii. Time-domain features based on Amplitude: These features fall under the category of time-domain features and include the inter-quartile range, root mean square amplitude, and median absolute deviation (Aarabi *et al*., 2006; Löfhede *et al.*, 2010). iii. Time-domain features based on Entropy: Shannon entropy belongs to this class of time-domain features (Greene *et al*., 2008).

b) EEG Frequency-Domain Features.

This category of features is obtained using Fourier transform from a frequency domain representation of an EEG signal. The features in this domain are based on spectral information contained in the EEG signal. The features in this category are of different classes which are (Boubchir *et al*., 2017):

i. Frequency-domain features based on Power Spectrum: This category of frequencydomain properties includes the feature of maximum power in frequency bands

(Aarabi *et al*., 2006).

ii. Frequency-domain features based on Spectral information: Spectral flux, flatness, and centroid are the various frequency-domain features that fall under this category (Löfhede *et al*., 2010).

iii. Frequency-domain features based on Entropy: Spectral entropy feature of the EEG signal belongs to this class of frequency-domain features (Greene *et al*., 2008).

c) EEG Time-Frequency domain features.

This category of features provides additional information due to their ability to take into account, dynamical changes in non-stationary signals. Discrete Wavelet Transform (DWT), is used in obtaining this category of features from an EEG signal (Hernández *et al*., 2018). The features that fall under this category can be placed in one of two classes which are:

i. Time-Frequency signal-related features: This class of features is extracted using the Quadratic Time-Frequency Domain formula.

ii. Time-Frequency image related-features: This category refers to the features that c visually describe identified seizure activity patterns using image descriptors. Local Binary Patter descriptor and Haralick descriptor are some of the image descriptors that fall under this category (Boubchir *et al*., 2017).

The feature extraction techniques and features employed on the EEG Dataset are summarized in Table 2.3. A summary of the numerous frequency-domain features is shown in Table 2.4, while time-domain features and several key equations are shown in Table 2.5.

Table 2.3: Summary of Feature Extraction Methods and Features used on EEG Dataset

S/N	Feature Extraction Methods	Relevant Feature
1	Time-Domain feature	Mean, Variance, Mode, Median, Skewness, Kurtosis, Max, Min, Zero Crossing, Line Length, Energy, Power, Shannon Entropy, Sample Entropy, Approximate, Entropy, Fuzzy Entropy, Hurst Exponent, Standard Deviation
2	Frequency-Domain feature	Spectral Power, Spectral Entropy, Energy, Peak Frequency, Median Frequency
3	Time-frequency Domain feature	Line Length, Min, Max, Shannon Entropy, Approximate Entropy, Standard Deviation, Energy, Median, Root Mean Square
4	Discrete Wavelet Transform (DWT)	Bounded Variation, Coefficients, Energy, Entropy, Relative Bounded, Variation, Relative Power, Relative Scale Energy, Variance, Standard deviation
5	Continuous Wavelet Transform (CWT)	Energy's Standard Deviation, Energy, Coefficient Z-score, Entropy
6	Fourier Transform (FT)	Median Frequency, Power, Peak Frequency, Spectral Entropy Power, Spectral Edge Frequency, Total Spectral Power

Table 2.4: EEG Frequency-domain Features (Boubchir *et al*., 2017)

S/N	EEG Frequency- Domain Features $F^{(f)}$

( )

1. Feature-based on power spectrum: Maximum power of the frequency bands.

$$\left| \quad \right|$$

$$(_f)\ \square\square\ Z\square k\square \left| \ ^{2} \right| \qquad (2.8)$$

$$F =$$

$$1\ k=1$$

$$2$$

$$(_f)\ \square_M\ Z\square k\square \qquad (2.9)$$

$$F_2 = \ ^{=\square +}{}_{k}\quad {}_{1}$$

2. Features based on spectral information:
Spectral Roll-off: Spectral concentration below threshold λ

$$\left| \quad \right|$$

$$F_{6(f)} = \square \square_{Mk=1}\ Z\square k\square \qquad (2.10)$$

3. Features based on entropy. Spectral entropy.

$$\left| \quad \right|$$

$$\frac{1}{\log M} \quad \square\ \square \qquad \square\ \square$$

$$F_7^{(f)} = \square_{k=1}^{M} P\left(\ \right)\left(\ Z \quad k\right)\log P\left(Z \quad k\right) \qquad (2.11)$$

Table 2.5: EEG Time-domain Features (Boubchir *et al*., 2017)

S/N ~~EEG Time-Domain Features~~ $\left(F_{(t)}\right)$

1. i First moment and second central moment of EEG Signal.

Mean: $F_1^{(t)} = \square = N\ 1\ \square_n{}^{N}{}_{=1}\ Z\square n\square \quad \left| \quad \right|$ (2.1)

$\square_2 = N\ 1 \square_{n}{}^{N}{}_{=1} (\square - Z \square n \square)_2$

Variance: $F_2^{(t)} =$ — (2.2)

ii. Normalized Moments: Third and Fourth central moments of EEG signal.

$1 \square^N (Z \square n \square \quad \square$

$^{(t)} \quad -)_3$ (2.3)

Skewness: $F_3 = N \square\ 2 \quad n=1$

Coefficient Of variation of EEG Signal:

$_{(t)}$

$F_5^{(t)} = \square \square = F F_{1\ (^2\ t)}$ (2.4)

2. Features based on amplitude.

Median absolute deviation of EEG amplitude

$^{(t)}\ 1 \quad N = (Z \square n \square \quad \square \quad -)$ (2.5)

$F_6 = n = \square_{n\ 1}$

Root mean square amplitude.

$F_7^{(t)}$

$= \square_{nN=1} Z \square n \square\ _2$

(2.6) N

3. Features based on entropy. Shannon entropy

$F_8^{(t)} = - \square_{n}{}^{N}{}_{=1} Z \square n \square \log_2 (Z \square n \square)$ (2.7)

2.8 Fundamental Concepts of Optimization

This section covers the fundamentals of optimization. This includes many forms of optimization approaches, and optimization problems, including metaheuristic optimization algorithms that are biologically inspired.

2.8.1 Optimization

In nature, optimization is considered optimal searching, in which problem-dependent objectives (performance index) must be evaluated or attained, as well as restrictions, must be met (Siddique & Adeli, 2015). The following are the general ways of solving optimization problems (OP) (Antoniou and Lu, 2007): Analytical, Experimental, Graphical, and Numerical are the four types of analysis. Combinatorial optimization is the use of an objective function to find the minimal or highest value of a countable collection of alternative solutions (Zhang, Lu, & Gao, 2015). In most cases, there are multiple solutions to an optimization problem; yet, any optimization process aims to discover the optimum solution among all possible options (Rothlauf, 2011).

2.8.2 Optimization problems

Optimization problems are problems with several solutions, variables, restrictions, and a performance measure to determine whether a chosen solution is optimal (Antoniou and Lu, 2007). Optimization problems have the following properties (Rothlauf, 2011): The availability of several decision-making choices; the existence of constraints on both equality and inequality, which limits the options for forming decisions; The presence of an evaluation function that may be used to assess the impact of each decision alternative. Each decision alternative has a different impact on the evaluation function. The nature of choice variables, types of constraints, number of objective functions, the character of equations, and physical structure of the problem are all used to classify optimization problems. Figure

2.8 displays the categorization of optimization issues.

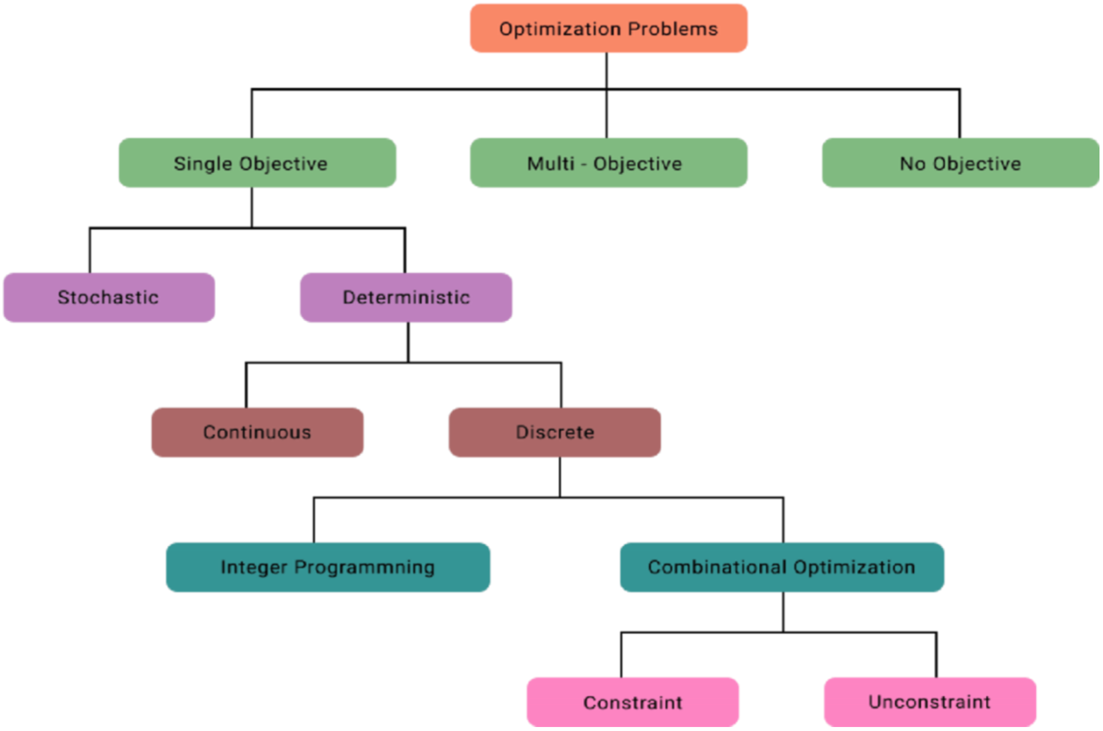

Figure 2.8: Classification of Optimization Problems (Lawler, 2001)

Issues with no defined objective function are known as non-objective optimization problems. A difficulty that arises during a feasibility study is finding values for a particular variable that satisfy certain limitations (Lawler, 2001). Single-objective optimization issues are problems with a single objective function or performance evaluation criterion, whereas multi-objective optimization problems have many objective functions.

Some or all of the variables in deterministic optimization problems are deterministic, whereas some or all of the variables in stochastic optimization issues are probabilistic. Continuous optimization issues have variables that correspond to any actual value, whereas discrete optimization problems have variables that belong to a collection of discrete values that are subsets of particular integers (Bertsekas, 1998). Combinatorial optimization problems are optimization issues where the objective is to accomplish the desired result by optimally allocating a finite or limited collection of resources to a discrete, bounded set of decision variables. Combinatorial frameworks, like those in (Antoniou & Lu, 2007), are subsets of discrete optimization whose variables assume some sets in the form of combinatorial structures. Arrangements, Assignments, Routes, Schedules, Sequences, and Combinations are all terms used to describe how things are put together.

Because of the enormous number of variables and restrictions, as well as the non-linear nature of real-world optimization problems, they are unique and challenging to solve.

Furthermore, because of their multi-modal objective function nature, they are computationally expensive, necessitating the use of novel optimization approaches to solve them (Glover & Kochenberger, 2006; Sumathi & Kumar, 2018). Most disciplines, including Engineering, Mathematics, Physics, Chemistry, Biology, Economics, Social Sciences, Commerce, Politics, and Administration, have optimization difficulties (Antoniou & Lu, 2007). Optimization challenges can be found in all sectors of engineering, including electrical, civil, mechanical, chemical, telecommunications, biomedical, and Mechatronics. Modeling; building and structure designs; system, device, and circuits design; design of equipment, instruments, and tools; function optimization and approximation; digital image processing; process, inventory, and quality control; forecasting and scheduling are some of the specific application areas of optimization techniques in Engineering (Antoniou & Lu, 2007).

2.8.3 Classification of optimization techniques

Depending on the outcome, optimization algorithms can be either deterministic or stochastic. Because they produce the same results for every iteration, traditional deterministic optimization approaches frequently fail to produce optimal and complete solutions. Integer programming, Non-Linear Programming (NLP), Gradient-Based (GB) and Gradient Free (GF) algorithms, convex programming, and Linear Programming (LP) are examples of algorithms (Siddique & Adeli, 2015). Because of the random character of their strategy, stochastic algorithms, which might be heuristic or metaheuristic, outperform deterministic solutions. They are capable of solving difficult issues and obtaining global solutions since they produce various answers for different iterations and examine several regions of the search space concurrently (Glover & Kochenberger, 2006). Because the possibility of discovering optimal solutions is not guaranteed, heuristic algorithms that seek optimal solutions by trial and error (for example, Scatter Search) tend to get stuck in local optimums. They focus more on exploitation and are problem-specific (Siddique & Adeli, 2015).

In the past three decades, interest in meta-heuristic optimization has grown among academics and researchers, and numerous meta-heuristics have been frequently suggested for use in domains including engineering, computer science, health, economics, and others to solve challenging real-world problems. The two types of meta-heuristics are singlebased and population-based algorithms, as shown in Figure 2.9. The fundamental concept of single-based meta-heuristic algorithms sometimes referred to as trajectory algorithms, is the construction of a single solution at each run. The neighborhood method improves this responsiveness (Agrawal *et al.*, 2021). Simulated Annealing (SA) is one of the most wellknown single-based meta-heuristics (Kirkpatrick *et al.*, 1983).

In contrast to single-based meta-heuristic algorithms, population-based meta-heuristic algorithms provide a population of multiple solutions for each run. The four main subcategories of population-based meta-heuristics are evolutionary-based meta-heuristics, swarm intelligence-based meta-heuristics, event-based meta-heuristics, and physics-based meta-heuristics. Evolutionary algorithms are the first class of population-based algorithms (EA). They are driven by natural evolutionary occurrences and use three major operators: selection, recombination, and mutation.

Swarm intelligence (SI) methods, which derive their knowledge from the group behavior of living organisms, are included in the second category. Birds, ants, bees, and other insects are examples. The Artificial Bee Colony (ABC) Algorithm and Particle Swarm Optimization (PSO) are two of the most used algorithms in this area (Agrawal *et al*., 2021; Ma *et al*., 2017). A few examples of search algorithms include the Cuckoo Search Algorithm (CS), Krill Herd (KH), Fruit Fly Optimization (FFO), Grey Wolf Optimizer (GWO), Ant Lion Optimizer (ALO), Dragonfly Whale Optimization Algorithm (WOA), Salp Swarm Algorithm (SSA), and Crow Search Algorithm. In the third category, inspiration comes from human deeds rather than from natural occurrences. The Teaching Learning-Based Algorithm (TLBA), the Imperialist Competitive Algorithm (ICA), and the Harmony Search (HS) algorithm were all inspired by educational practices, societal imperialism, and musical principles, respectively. The final class of meta-heuristics is those based on physical principles (PA). For instance, the Gravitational Search Algorithm (GSA) models gravitational forces between masses whereas the Multi-Verse Optimizer (MVO) is based on notions of numerous universes (Abualigah *et al*., 2021; Hatta, *et al*., 2019; Meraihi *et al*., 2020; Mirjalili *et al*., 2014; Shehab *et al*., 2017; Wang *et al*., 2019; Zhang & Geem, 2019). Big-Bang Big-Crunch (BBBC), Henry Gas Solubility Optimization, Optics Inspired Optimization (OIO), Thermal Exchange Optimization (TEO), Water Evaporation Optimization (WEO), Vibrating Particles System Algorithm (VPSA), and Electromagnetic Field Optimization (EFO) (HGSO), Magnetic Chargeability Optimization (Abedinpourshotorban *et al*., 2016; Kashan, 2015; Kaveh & Dadras, 2017; Kaveh & Ghazaan, 2017).

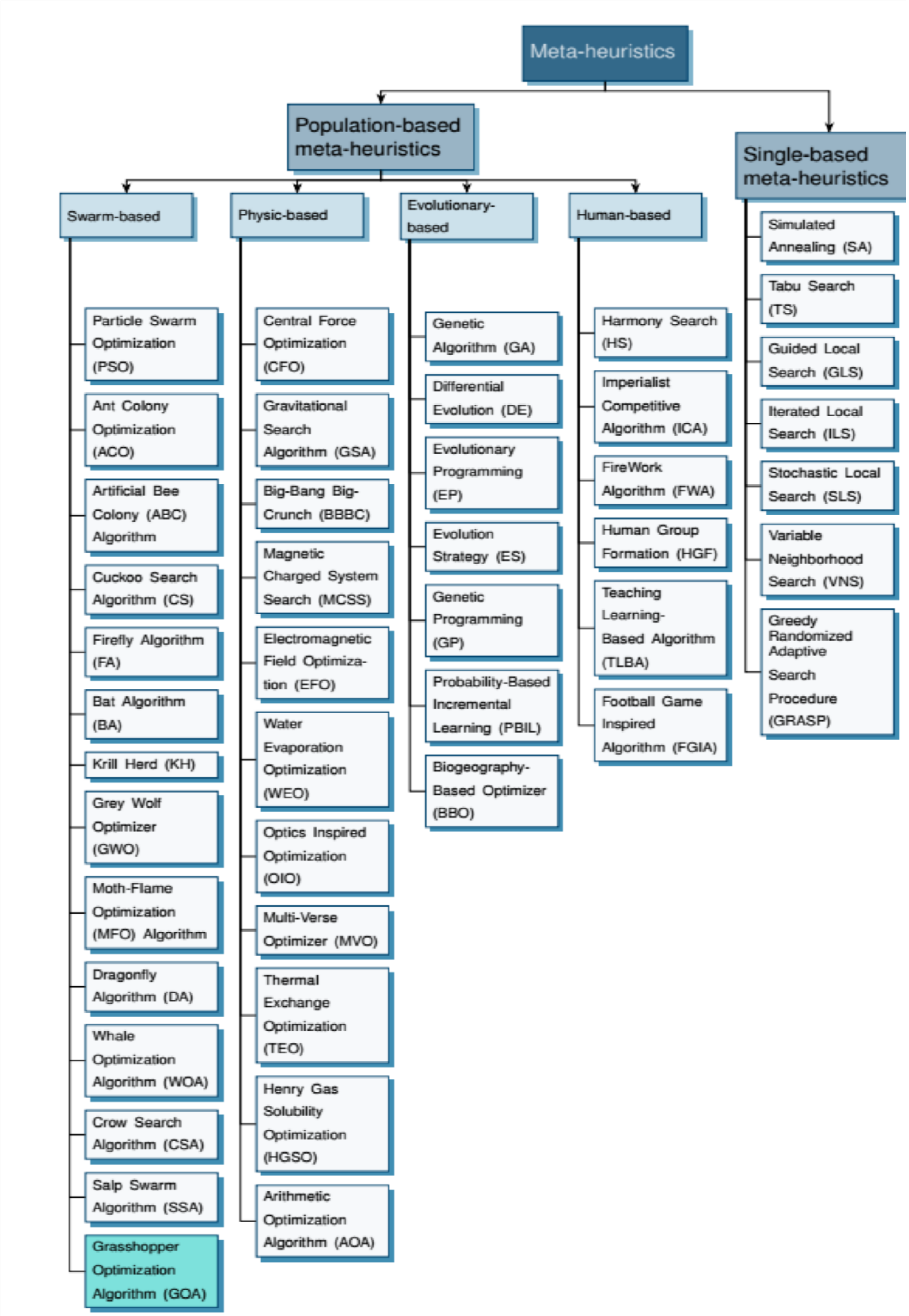
Meta-heuristics
Population-based meta-heuristics
Single-based meta-heuristics
Swarm-based
Physic-based
Evolutionary-based
Human-based
Particle Swarm Optimization (PSO)
Ant Colony Optimization (ACO)
Artificial Bee Colony (ABC) Algorithm
Cuckoo Search Algorithm (CS)
Firefly Algorithm (FA)
Bat Algorithm (BA)
Krill Herd (KH)
Grey Wolf Optimizer (GWO)
Moth-Flame Optimization (MFO) Algorithm
Dragonfly Algorithm (DA)
Whale Optimization Algorithm (WOA)
Crow Search Algorithm (CSA)
Salp Swarm Algorithm (SSA)
Grasshopper Optimization Algorithm (GOA)
Central Force Optimization (CFO)
Gravitational Search Algorithm (GSA)
Big-Bang Big-Crunch (BBBC)
Magnetic Charged System Search (MCSS)
Electromagnetic Field Optimiza-tion (EFO)
Water Evaporation Optimization (WEO)
Optics Inspired Optimization (OIO)
Multi-Verse Optimizer (MVO)
Thermal Exchange Optimization (TEO)
Henry Gas Solubility Optimization (HGSO)
Arithmetic Optimization Algorithm (AOA)
Genetic Algorithm (GA)
Differential Evolution (DE)
Evolutionary Programming (EP)
Evolution Strategy (ES)
Genetic Programming (GP)
Probability-Based Incremental Learning (PBIL)
Biogeography-Based Optimizer (BBO)
Harmony Search (HS)
Imperialist Competitive Algorithm (ICA)
FireWork Algorithm (FWA)
Human Group Formation (HGF)
Teaching Learning-Based Algorithm (TLBA)
Football Game Inspired Algorithm (FGIA)
Simulated Annealing (SA)
Tabu Search (TS)
Guided Local Search (GLS)
Iterated Local Search (ILS)
Stochastic Local Search (SLS)
Variable Neighborhood Search (VNS)
Greedy Randomized Adaptive Search Procedure (GRASP)

Figure 2.9: Types of Meta-Heuristics (Agrawal *et al.*, 2021)

2.8.4 Standard optimization benchmark functions

The effectiveness of optimization techniques is evaluated and assessed using test functions (Jamil & Yang, 2013). Because test functions are designed to imitate real-world problems that are growing more complicated, multi-modal, non-linear, and non-convex, they are used to test newly developed optimization methods (Ali *et al*., 2005).

Fundamentally, benchmark test functions are optimization issues expressed as mathematical numerical functions. These functions are optimized using a set of best-fit parameter values that aid in achieving the best solution, where *D* denotes the issue dimensions. The optimum answer is concealed among a huge number of sub-optimal alternatives scattered throughout a problem landscape with a variety of hills and valleys. Any optimization technique, including metaheuristic algorithms, strives to discover the optimum answer as rapidly as possible (though this is not always guaranteed). The global searchability and local convergence ability of any metaheuristic are used to determine its efficiency. Better global searchability algorithms are difficult to trap in sub-optimal (local minima or maxima) locations. Simultaneously, metaheuristics with good convergence ability make it difficult to overlook any optimum solution within the neighborhoods.

In the literature, various test functions have been employed while introducing new metaheuristic algorithms or proposing modified variants of current methods. Not only that, but different literature contains a diversity of dimensional situations. This means that there is no agreed-upon test-bed with standard configurations that new researchers can use to conduct tests to validate their suggested approaches. Indeed, it is argued that such diverse techniques reveal an algorithm's problem-solving capabilities on a certain set of challenges. However, the majority of benchmark test functions have a similar nature, such as numerical optimization problems or, to be more exact, unconstrained numerical problems. Researchers have also discovered that they have a set of functions that they frequently repeat throughout several experimental trials. This limited review scoured the literature for experiments on benchmark test functions to measure metaheuristic algorithm performance. The following is a summary of some of the work that shows how many test functions have been utilized with which configurations to get a sense of existing methodologies (Hussain *et al*., 2017).

There have been several test benchmark test functions constructed, each with various modalities, separability, and linearity. The majority of test functions are categorized according to their modality, or the number of optimal solutions they contain. They fall into three categories: composite, multimodal, and unimodal (Mirjalili & Lewis, 2016).

2.8.5 Unimodal benchmark test functions

For unimodal benchmark functions, there is only one global optimal solution. They are utilized to evaluate how well an algorithm can utilize data. (Mirjalili & Lewis, 2016) Some of the examples are shown in Table 2.6.

Table 2.6: Unimodal Benchmark Test Functions

Function	Dim	Range	Shift Position	fmain
$F_1(x) = \sum_{i=1}^{n} x_i^2$	20	[-100,100]	[-30, -30, -30]	0
$F_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	20	[-10,10]	[-3, -3, -3]	0
$F_3(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} X_j \right)^2$	20	[-100,100]	[-30, -30, -30]	0
$F_4(x) = \max\{\lvert x_i \rvert, 1 \leq i \leq n\}$	20	[-100,100]	[-30, -30, -30]	0
$F_5(x) = \sum_{i=1}^{n-1} \left[100\left(x_{i+1} - x_i^2\right)^2 + \left(x_i - 1\right)^2 \right]$	20	[-30,30]	[-15, -15, -15]	0
$F_6(x) = \sum_{I=1}^{n} \left(\left[x_i + 0.5 \right] \right)^2$	20	[-100,100]	[-750, -750, -750]	0
$F_7(x) = \sum_{i=1}^{n} i x_i^4 + random[0,1]$	20	[-1.28,1.28]	[-0.25 -0. 25.., -0.25]	0

2.8.6 Multimodal benchmark test functions

Due to the high number of local optima that multimodal test functions have and the fact that their optimal number increases exponentially with the number of variables, they are used to evaluate the exploration potential of algorithms (Mirjalili & Lewis, 2016). Some of them allow specified dimensions, while others accept any dimension. Table 2.7 presents some of these examples.

Table 2.7: Multimodal Benchmark Test Functions

Function	Dim Range	Shift Position	fmain
$F_8(x) = \square_{I=1}^{n} -x\sin\left(\sqrt{\lvert x_i \rvert}\right)$	20 [-500,500]	[-300, -300]	$-418.9829\square5$
$F_9(x)=\square_{i}^{\square}\square x_i^{1} - 10\cos(2\square x_i)+10^{\square}\square$	20 [-5.12,5.12]	[-2, -2, -2]	0
$F_{10}(x)=-20\exp\square\square\sqrt{\square-0.2}\ \frac{1}{n}\square_{=n} x_{i2}\square\square\square-\exp\square\square\ 1\ n\square_{i=n1}\cos(2\ x_i)\ 20+e$ $\square\ \square\square\square+$	20 [-32,32]		0

1 .9 Feature Classification

This is a process that is concerned with assigning the vectors (containing the selected features) to their appropriate class, as to whether the vector is a seizure or a non-seizure (Boubchir *et al.*, 2017). Due to the development of various low-cost interfaces, there has been an evolution of EEG recordings with various channels and thus, the development of various channel selection algorithms. Figure 2.10 shows an overview of the general process of EEG signal classification, While the overview of feature classification based on the

$F_{11}(x)= 40001\ \square_{i=n1} x_{i2} - \square_{i=n1}\cos\sqrt{\square}\ \square\square\ x1_i\ \square\square\ \square+1$ 600,600]20 [- [400]-400 , -
0

20 [-50, - 50] 30] [-30, -3-, - 0

$(y_i - 1)^2$ $F_{12}(x) =$ □□ □10sin(□y_i) + □ $^{n-1}$ □□ 1+10sin^2

(□y_{i+1})□□+(y_n − 1)^2 □□

n □ $_{i=1}$ □

n

+□u($x_{i,}$10,100,4)

$_{i=1}$

x

$y_i = 1 + \quad _i + 1\ 4$

□ $k(x_i - a)^m x_i$ □

a □ □ $u(x_{i,}\ a,k,m) =$ □ □

$0 - a$□x_i □a □ □ □

□ $_m$ □ □□$k(-x_i - a)$

x_i □ $_-$ a□□

□ $_n$ ^{2} 20 [-50, - 50] □ [-100, - 0 100]

$F_{13}(x) =$ 0.1□sin^2(3□x_i) + □(x_i − 1)^2 □□1 + sin^2(3 x_i □

+1)□□+(x_n − 1) □□1 + sin^2(2□x_n)□□□

□ $_{i=1}$ □

n

□ + $u(x_i$,5100,4)

$_{i=1}$

number of EEG data channels is shown in Figure 2.11. these are the various to select from for epilepsy classification. Some of the classifiers adopted in this process of feature classification are: Support Vector Machines (SVM), Artificial Neural Networks, and Logistic Regression. The reason for their adoptions is due to their high accuracy value in the literature.

2.9.1 Support vector machines (SVM)

This margin classifier optimizes the distance between training points by using a separating hyper-plane. For such a decision boundary, two hyper-parameters must be defined: cost and cost factor. Cross-validation (Park *et al*., 2011) or grid check (Moghim & Corne, 2014) can be used to determine which of these parameters to combine. It is well known that they perform better than other classifiers in terms of sensitivity and specificity (Boubchir *et al*., 2017).

2.9.2 Artificial neural networks

Artificial neural network are a group of connected units or nodes that mimic the neurons in a biological brain (Lai, 2021). Given enough neurons and layers, they are universal approximators that can estimate any continuous function. However, they are susceptible to overtraining and are prone to underperform in conditions where there are insufficient input features (Boubchir *et al*., 2017).

2.9.3 Logistic regression

This is a category of classifiers that are parameterized by weights and biases. To train them, there is a need to find adequate weights that have been optimized by minimizing a predefined loss function (Boubchir *et al*., 2017).

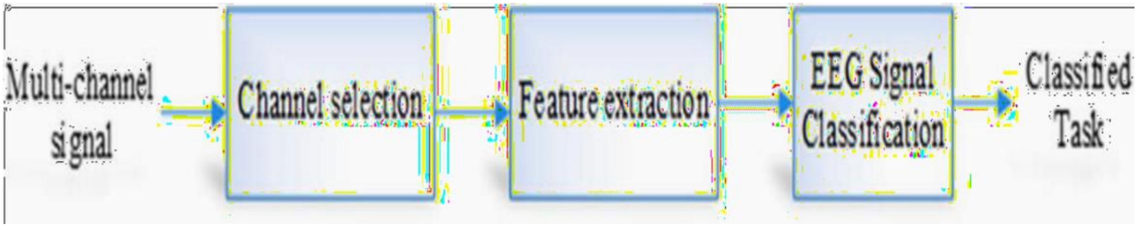

Figure 2.10: General Process of EEG signal classification (Alotaiby *et al.,* 2015)

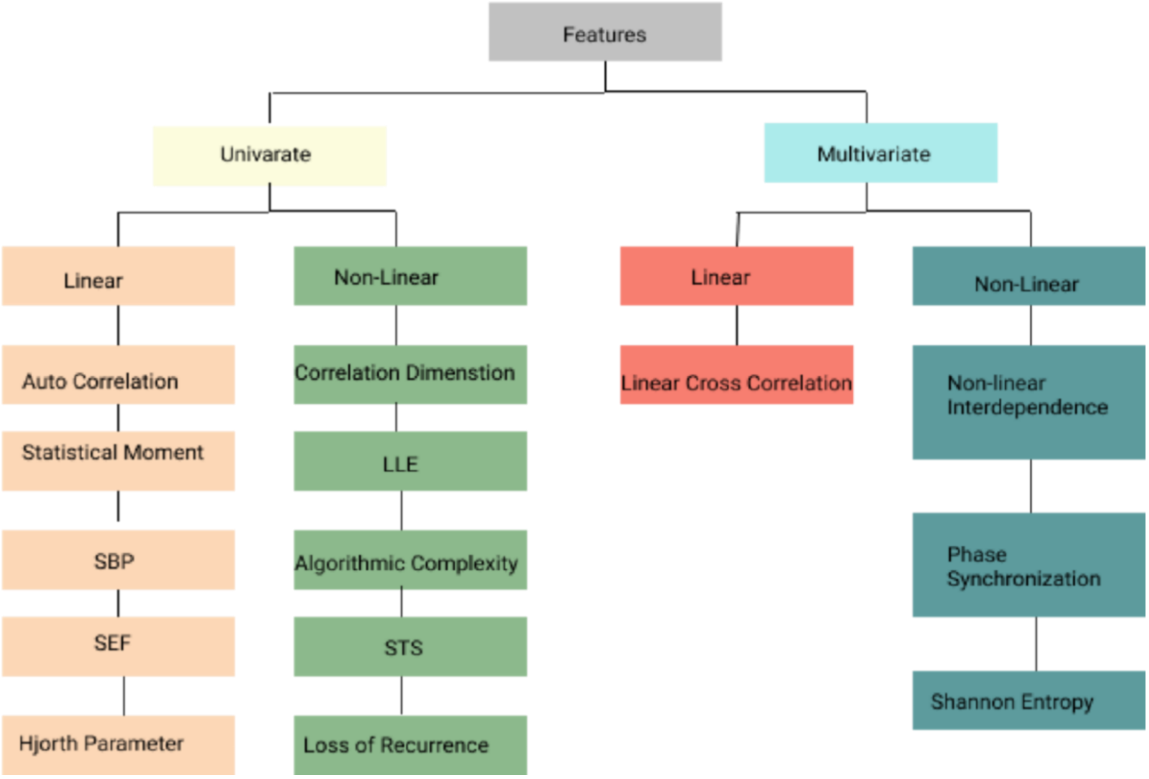

Figure 2.11: Feature Classification based on the number of channels in EEG Data (Rasheed *et al*., 2020)

2.10 Epileptic Seizure Detection and Prediction

Several strategies are used to anticipate and identify an epileptic seizure. Figure 2.12 provides a summary of the different methods.

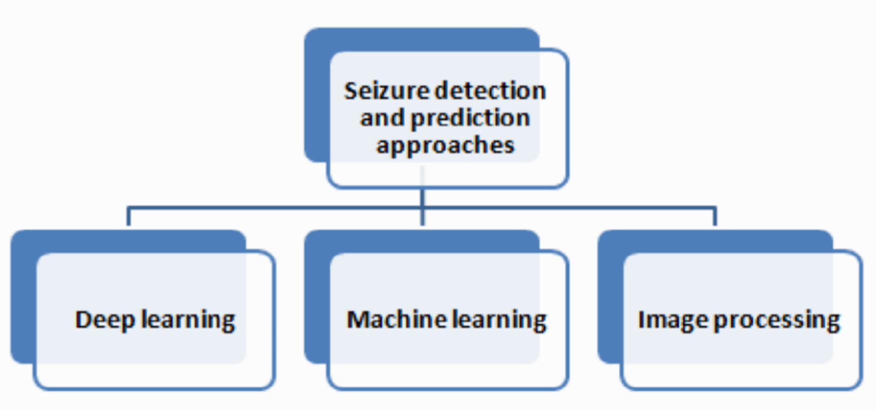

Figure 2.12: Approaches to Prediction and Detection of Epileptic Seizures (Khati &

Ingle, 2020)

2.10.1 Epilepsy seizure detection

The research on the identification of epileptic seizures using various machine learning and deep learning techniques is reviewed in this part. The procedure for identifying and categorizing epileptic seizures using EEG signals is shown in Figure 2.13.

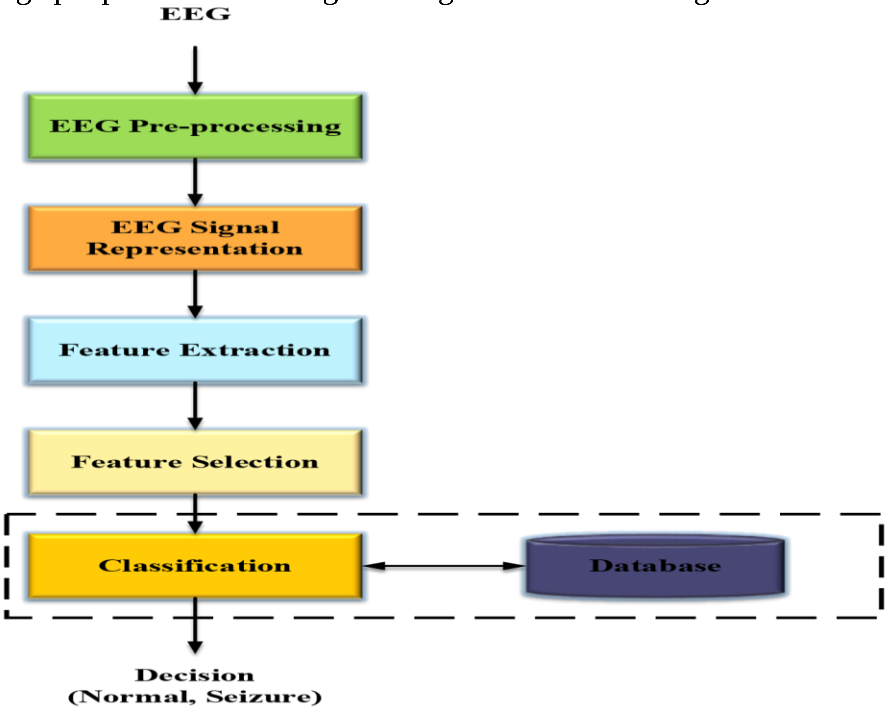

Figure 2.13: Process of EEG Epileptic seizure detection and classification (Boubchir *et al.*, 2017)

2.10.1.1 Epilepsy Detection using Machine Learning

Researchers have created several techniques for machine learning-based epilepsy detection. This section carries out a brief review of some of these works.

Lasefr, Ayyalasomayajula, & Elleithy, (2017) proposed a scheme for epilepsy detection through wavelet transforms and a supervised machine learning neural network, Support Vector Machine (SVM), allowing for accuracy as high as 96% using an SVM and 98% using ANN. Siddiqui, Islam, & Kabir, (2019) proposed the use of decision forest to reduce the time taken to carry out seizure detection while still ensuring a high value of accuracy in the detection. The scheme allowed for the identification of the part of the brain that was most affected by the seizure. Song & Liò, (2010) investigated the use of Sample Entropy

(SampEn) for the detection of an epileptic seizure, using a back-propagation neural network (BPNN) and an extreme learning machine (ELM), allowing for fast detection of epilepsy seizure with a classification accuracy of 95.67%. Donos *et al.*, (2015) proposed a scheme for seizure detection using a random forest classifier, resulting in a low detection delay time of 1.75 seconds. Koolen et al., (2014) presented an approach for seizure detection using a 'line length-based algorithm', thus allowing for 84.27% accuracy in detection. Gill, Fatima, Akram, Khawaja, & Awan, (2015) proposed a system for seizure detection using spectral and temporal feature extraction, allowing for accuracy as high as 86.93%. Shoeb & Guttag, (2010) presented an approach for the detection of an epileptic seizure by the adoption of machine learning for feature vector classification, resulting in a high detection rate of 96%. Zhang *et al*., (2015) adopted a scheme for epilepsy detection using wavelet package decomposition with a Support Vector Machine (SVM), thereby resulting in great accuracy and performance with low training time. Guo, Rivero, Dorado, Rabunal, & Pazos, (2010) put forward a method for epilepsy detection by supporting a wavelet decomposition process with an ANN, thus providing a higher value of accuracy in the detection process. Ahmad, Khan, and Majeed, (2014) proposed an approach for epilepsy detection using an SVM, allowing for an average accuracy of 95.12%. Sharma, Shah, & Achuth, (2019) proposed a method for epileptic seizure detection using an SVM for the classification of bi-orthogonal wavelet-filtered EEG signals, thereby yielding good classification accuracy. The performance of the system was however not validated using a large dataset.

Lee & Kim, (2016) adopted SVM and ANN with the fuzzy-logic-based classification of

EEG signals in the detection of epileptic seizures, achieving accuracy as high as 94.39%. It however failed to put in place modalities for no-precipitation echoes. Chen, Wan, Xiang, & Bao, (2017) implemented a scheme for the detection of epileptic seizures through the decomposition of EEG signals into 7 wavelet families using Discrete Wavelet Transform and SVM, with the MIT and Bonn dataset. The work achieved accuracy as high as 90%.

The approach was however highly time-consuming. Manzouri, Heller, Dümpelmann, Woias, and Schulze-Bonhage, (2018) put forward the use of a Random Forest Classifier for the extraction of features and classification of EEG signals in the process of detection of seizures using the European Epilepsy Dataset. It obtained a mean AUC score of 0.89. Fasil & Rajesh, (2019) proposed the classification of EEG signals for the detection of epileptic seizures using the exponential energy feature in the time domain on the BernBarcelona EEG and Ralph Andrzejak dataset. It achieved accuracy as high as 99.5%. It however did not put in place modalities for real-time applications and there was no testing done using more realistic datasets.

The performance of the various systems relied and hinged on how well features were extracted from the EEG recordings by a person with domain knowledge in the majority of these publications on the use of machine learning for epilepsy identification. Table 2.8 gives a summary of various works on Machine learning methods for Epilepsy Detection.

Table 2.8: Summary of ML Methods for Detection of Epileptic Seizures

Author	Database	Detective (%) (%)	Accuracy (%)	Sensitivity	Specificity	Comments	Characteristics
(Koolen et al., 2014)	University Hospitals of Leuven	Burst	84.27	84	85.70	The data set are few in numbers	
(Ahmad et al., 2014)	CHB-MIT and PIMH	Discrete Wavelet Transform	95.12	91.7	95.7	The data set are few in numbers	
(Gill et al., 2015)	CHB-MIT	Gaussian mixed models	86.93	86.26	87.58	The data set are few in numbers	
(Donos et al., 2015)	Freiburg	Seizure Onset		93.84		The data set are few in numbers and not available publicly	
(Guo et al., 2010)	Five set singlechannel EEG segments.	Wavelet Transform	99.60	99.40	100	Good accuracy was achieved	
(Shoeb & Guttag, 2010)	CHB-MIT	Feature Vector Design	96	92.3		The data set are few in numbers	
(Y. Zhang et al., 2015)	Bonn	Wavelet Package Decomposition	99.21			Good accuracy, and publicly available	

(Siddiqui et al., 2019)	EcoG	Decision Forest	100			Good accuracy with few data sets
(Song & Liò, 2010)	The publicly available, individually collected dataset	Sample Entropy	95.67	97.26	98.77	The data set are few in numbers and not available publicly.

2.10.1.2 Epilepsy detection using deep learning

The research that has been done to enable the identification of epileptic seizures using deep learning is reviewed in this section.

Hussein *et al*., (2018) proposed a scheme that adopted Deep neural networks to allow for direct learning from EEG recordings without the need for extraction of features by an individual with domain knowledge, thus resulting in inaccuracy as high as 97.75%, however, the testing of the proposed scheme was not carried out using many datasets. Wei *et al*., 2018) presented a method for automatically detecting epileptic seizures that processed the collected EEG signals using a 3D convolutional neural network, achieving a performance accuracy of 90% and specificity of 93.78%. Using a Long Short-Term Memory (LSTM) network. Hussein *et al*., (2018) proposed a method for the early identification of epileptic seizures, enabling a high level of detection in both favorable and unfavorable conditions. Using a pyramidal one-dimensional convolutional neural network (P-1D-CNN), Ullah *et al*., (2018) developed a system for epileptic seizure detection, achieving a detection accuracy of 99.1% with a small dataset. The system however could only detect the seizures after they had occurred, not before they occurred. Achilles *et al*., (2018) proposed employing CNN and video-EEG devices to identify epileptic episodes. The Area Under Curve (AUC) was attained at 78.33%. Park *et al*., (2018) put forward an approach for the detection of epileptic seizures using a deep convolution network for analysis of multi-channel EEG signals with Spatiotemporal correlation using the SNUHHYU dataset. It achieved accuracy as high as 90.5%. It however did not put in place modalities for region location of epileptic seizure occurrence.

Liu & Woodson, (2019) presented a technique for the 90% accurate detection of epilepsy through the classification of EEG signals using a Convolution Neural Network (CNN).

However, the system's testing was limited to a tiny set of data. Convolutional Neural

Network (CNN)-based multi-class classification of EEG dataset was proposed by (Akut, 2019), which strengthened the process of epileptic seizure detection. Aliyu *et al*., (2019) described how to classify an EEG signal using a recurrent neural network (RNN), which has a 99% accuracy rate for detecting epileptic seizures. However, a full comparison between the system and other deep learning models was not done. Türk & zerdem, (2019) suggested using a convolutional neural network to learn attributes from scalogram images derived from continuous wavelet transformation of EEG records, leading to a performance accuracy of 93.60% in the identification of epileptic episodes.

According to (Akut, 2019), epileptic seizures can be detected using sophisticated feature analysis of EEG signals using CNN with Discrete Wavelet Transform (WT). By doing away with feature extraction, it was able to attain an accuracy of 99.4%. Using the BernBarcelona dataset, it was suggested by (San-Segundo *et al*., 2019) to employ two convolutionallayered deep learning models for the extraction of features from EEG signals for the detection of epileptic seizures. The work achieved an accuracy of 98.9%. It however did not obtain statistically tangible results in the combination of various transforms. Ansari *et al*., (2019) adopted a Deep CNN combined with a random forest classifier, with handengineered features, from multi-channel EEG data. It achieved a seizure detection rate of

77%. It however failed to carry out testing on a multi-rated dataset. Emami *et al*., (2019) put forward the adoption of CNN analysis of long-term EEG signals for image-based detection of seizures. It achieved a median seizure detection rate of 100%. It however failed to put in place modalities for the evaluation of EEG signals in time series. Sui *et al*., (2019) put forward the classification of EEG data in the detection of epileptic seizures by using CNN and Short-time Fourier transform (STFT) on spectrograms of iEEG signals in the time-frequency domain. It was able to classify objects with an accuracy of up to 91.8%. A Temporal graph convolutional network (TGCN) was suggested by (Covert *et al*., 2019) for the extraction of characteristics from EEG signals in the identification of epileptic seizures. Specificity and sensitivity were 97% and 99%, respectively, for the work. Convolutional Neural Network (CNN) with fast Fourier transform and wavelet packet decomposition was suggested by (Tian *et al*., 2019) for application in the epileptic seizure detection process. Up to 92.95% accuracy was attained.

Uyttenhove *et al*., (2020) proposed the use of a convolutional neural network known as

Tiny Visual Geometry Group (t-VGG) for epilepsy detection, resulting in a 95.52% Area Under the Precision-Recall (AUPR). The accuracy of the system however required some improvements. Table 2.9 gives a summary of various works on Deep learning methods for Epilepsy Detection.

Table 2.9: Summary of Some DL methods in Detection of Epileptic seizure

Author	Database	Detective Characteristics	Acc	Sen	AUC	Spec	Comments
(Aliyu et al., 2019)	Bonn	Discrete Wavelet Transform	98	N/M	N/M	N/M	Good accuracy, and publicly available
(Uyttenhove et al., 2020)	TUEP	Various Epilepsy Markers	N/M	N/M	93.02	N/M	The data set are few in numbers and not available publicly
(J. Liu & Woodson, 2019)	Five set Features EEG segments.	Various single channel	99.6	N/M	N/M	N/M	The data set are few in numbers and not available publicly
(M. Sun, Wang, Min, Zang, & Wang, 2018)	Dog EEG	Discrete Fourier Transform	78.6	N/M	N/M	N/M	The data set are few in numbers and not available publicly
(Wei et al., 2018)	First Affiliated Hospital Xinjiang EEG	Temporal and Spatial Features	90	88.90	N/M	93.78	The data set are few in numbers and not available of publicly
(Hussein et al., 2018)	Bonn	Robust features	100	100	N/M	100	Good accuracy, and publicly available
(Akut, 2019)	Bonn	Discrete Wavelet Transform	99.4	98.5	N/M	99.45	Good accuracy, and publicly available
(Türk & Özerdem, 2019)	Five set single channel EEG segments.	Short Time Fourier Transform (STFT)	99.36	99.00	98.50	99.46	The data set are few in numbers and not available publicly

(Hussein et al., 2019)	Freiburg	Raw data	97.75	N/M	N/M	N/M	The data set are few in numbers and not available publicly

2.11 Epilepsy Seizure Prediction

This section looks at some of the work that has been done to predict epileptic seizures using machine learning and deep learning techniques. Figure 2.14 shows the method of epilepsy prediction using EEG data and classification algorithms.

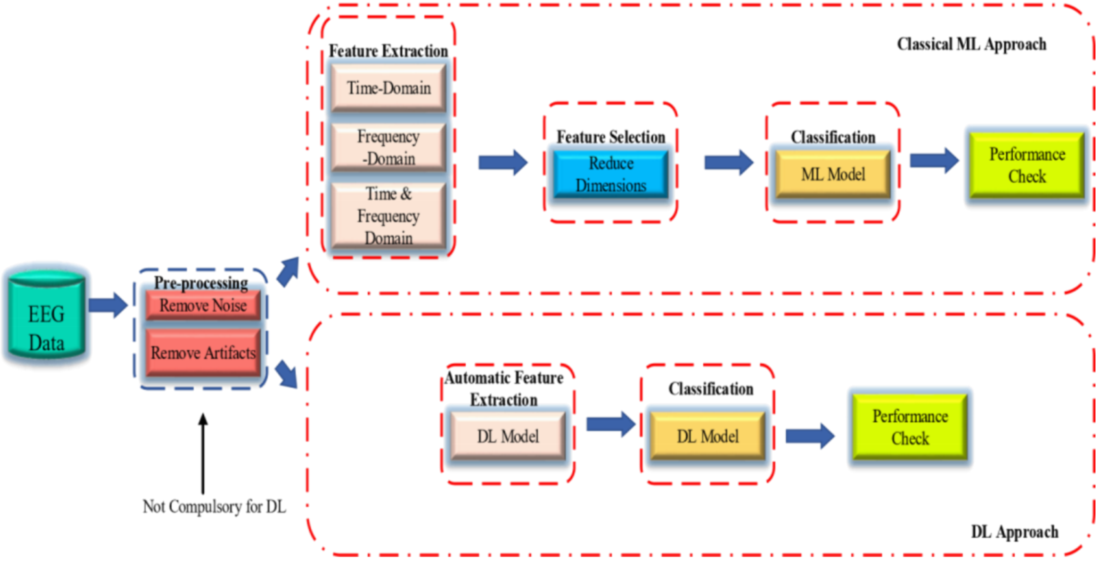

Figure 2.14: Process of Epilepsy prediction using EEG data and classification algorithm (Rasheed *et al*., 2020)

2.11.1 Epilepsy prediction using machine learning

The research that has been done on machine learning-based methods for epileptic seizure prediction is reviewed in this section.

Park *et al*., (2011) developed a method for categorizing the spectral power aspects of the acquired EEG data using Support Vector Machines (SVM) to forecast epileptic seizures. The sensitivity of this method was 97.5%, and the overall percentage of incorrect predictions was 13%. A method for the prediction of epileptic seizures utilizing a Support Vector Machine for analysis of acquired EEG signals is presented (Rosas-Romero & Guevara, 2020). Liu *et al*., (2018) proposed a system for the prediction of low-grade gliomas (LGG) epilepsy that used machine learning. The system achieved a validation cohort of 0.8152. Sujitha *et al*., (2010) put forth a model for the forecasting of epileptic seizures using Support Vector Machines and a Radial Basis Function (RBF), resulting in a 93.87% prediction accuracy. Usman *et al*., (2017) put forward a scheme for a reliable preprocessing and extraction of features from EEG data through the use of empirical mode decomposition (EMD), resulting in a high true positive rate of 92.23%. Usman & Hassan, (2018) proposed an algorithm to allow for the prediction of an epileptic seizure by extraction of various univariate features from the EEG data, allowing for a true positive rate of 89.90% to be achieved. Yang *et al*., (2018) described a means for the prediction of epileptic seizures through permutation entropy and SVM for classification, thereby resulting in an average sensitivity of 94%. Bandarabadi *et al*., (2015) proposed an algorithm for the prediction of epileptic seizures through continuous long-term evaluation of multichannel EEG data, thereby achieving a sensitivity of 75.8%. Sharif & Jafari, (2017) defined an approach for the prediction of epileptic seizures through feature extraction from the Poincare plane samples, thereby allowing for an average sensitivity of 96.6%. Direito *et al*., (2017) proposed the prediction of epileptic seizures through SVM classification of multi-channel high-dimensional feature sets, resulting in an overall sensitivity of 38.47%. Assi *et al*., (2018) proposed the feasibility of seizure detection using higher-order statistics processing of EEG data with ANN, and as a result, produced accuracy as high as 78.11%. Kitano *et al*., (2018) put forward the use of a polling-based unsupervised learning algorithm with wavelet transform preprocessing to allow for the prediction of epileptic seizures, thereby resulting in sensitivity as high as 98%, with a 91% accuracy.

Leszczyński, (2018) proposed a channel unification-based algorithm for seizure prediction using the EPILEPSIAE dataset with a denoising Autoencoder. It achieved a median AUC value as high as 0.552. Teijeiro, Shokrekhodaei, & Nazeran, (2019) put forward a framework for workstation setup for the prediction of seizures through for analysis of raw EEG dataset. Savadkoohi *et al*., (2020) investigated the use of time and frequency domain analysis for feature engineering of EEG signals with SVM and KNN learning algorithms for the prediction of seizures. The work achieved flexibility with a wide range of frequencies and sensitivity as high as 99%. Table 2.10 presents a summary of various works on Machine learning methods for Epilepsy Prediction.

Table 2.10: Summary of ML Methods in the Prediction of Epileptic Seizures

Author	Database	Predictive Characteristics	Model	Pred	Sen	False Positive/ hr	Acc	Comments
								available

			SOM	N/M	(Sharif	Freiburg	Distribution of	publicly SVM
(Usman *et al.*, 2017)	CHB-MIT	Entropy	SVM	23.48 min	92.23	N/M	N/M	The data set are few in numbers and not available publicly
(Usman & Hassan, 2018)	CHB-MIT	Variance, Skewness	KNN Naïve Bayes SVM	34 min	97.44 90.66 97.07	N/M	N/M	The data set are few in numbers and not available publicly
(Kitano *et al.*, 2018)	CHB-MIT	Zero-crossing of DWT coefficients			98	N/M	N/M	The data set are few in numbers and not
& Jafari, 2017)		6 fuzzy rules		42 min	96.6	0.05-0.08	N/M	The data set are few in numbers and not available publicly few in numbers
(Y. Yang *et al.*, 2018)	Freiburg	Permutation Entropy	SVM	61.93 min	94	0.111	N/M	The data set are few in numbers and not available publicly
(Bandarabadi *et al.*, 2015)	Epilepsia	Amplitude Distribution Histogram & Spectral Power	N/M	8 sec	73.98	0.06	N/M	The data set are few in numbers
(Direito *et al.*, 2017)	Epilepsiae	22 univariate features	SVM	N/M	38.5	0.2	N/M	The data set are few in numbers

(Assi *et al*., 2018)	IEEE.org	Bi-spectral Entropy	MLP	N/M	N/M	N/M	N/M	The dat a set are few in numbers
(Park *et al*., 2011)	Freiburg	Spectral power	SVM	N/M	97.5	0.20	N/M	The dat a set are few in numbers
(Rosas-Romero & Guevara, 2020)	Notre-Dame du Centre Hospitalier	Functional nearinfrared spectroscopy (fNIRS)	SVM	N/M	N/M	N/M	N/M	The dat a set are few in numbers
(Sujitha *et al*., 2010)	Siemens Magnetom Symphony	Entropy, Contrast, Correlation.	SVM	N/M	N/M	N/M	93.87	The dat a set are
(Usman *et al*., 2017)	CHB-MIT	Time and Frequency Domain Features	N/M	23.61 min	92.23	N/M	N/M	The data set are few in numbers and not available publicly

2.11.2 Epilepsy prediction using deep learning

Khan *et al*., (2017) With a sensitivity of 87.8%, convolutional filters were used to study the prediction of epileptic seizures on EEG signals that had undergone the wavelet transformation procedure. Truong *et al*., (2017) adopted a convolutional neural network (CNN) with Short-Time Fourier Transform (STFT) to allow for the prediction of epileptic seizures from iEEG data, thereby allowing for sensitivity in detection as high as 81.4%. Hosseini *et al*., (2017) proposed a dimensionality-reduction technique with stack autoencoders for unsupervised feature extraction from the EEG dataset for the prediction of an epileptic seizure. It achieved accuracy, precision, and sensitivity of 94%, 95%, and 93% respectively.

Sun *et al*., (2018) outlined an approach for the prediction of epileptic seizures, through the use of a two-layer convolutional neural network (CNN) for the conversion of time-domain EEG signals to a frequency domain, thereby allowing for a high level of performance in epileptic preictal state prediction. However, the system was unable to implement a continuous variable that would have served as a signal of the likelihood of an oncoming seizure (Tsiouris *et al* 2018). Long Short-Term Memory (LSTM) networks and CNN were used in the method proposed by (Tsiouris *et al*., 2018), resulting in a false prediction rate of 0.11 to 0.02 for epileptic seizures. Khan et al., (2018) investigated the adoption of convolutional filters with wavelet transformation of EEG signals for extraction of features in the prediction of epileptic seizures. It achieved a sensitivity of 87.8% (Truong et al., 2018) and adopted Short-Time Fourier Transform (STFT) with CNN for analysis and extraction of features from a 30s EEG signal window. It achieved sensitivity as high as

81.4%. It however did not make provision for real-time analysis.

Yuan *et al*., (2019) presented a process for the prediction of an epileptic seizure by using a Convolutional Neural Network (CNN) and a Common spatial pattern (CSP) for analysis obtained of EEG signals, thereby allowing for a high level of sensitivity of 92.2% and a false prediction rate of 0.12/h. Wei *et al*., (2019) proposed the use of a Long-Term

Recurrent Neural Network (LRCN for spatiotemporal-based extraction of features from EEG data for the prediction of epileptic seizures. It achieved accuracy and sensitivity of 93.40% and 91.88% respectively. It however did not provide modalities for the fusion of multimodal image information. Truong *et al*., (2019) adopted a Generative Adversarial Network (GAN) with Short-Fourier Transform to predict epileptic seizure from EEG data, thereby producing an operating characteristic curve (AUC) as high as 77.68%. A deep learning-based technique was put forward by (Daoud & Bayoumi, 2019), using 4 deep learning-based models to allow for the prediction of epileptic seizures, thereby producing an accuracy of 99.6%. Hussein *et al*., (2019) developed a scheme for the prediction of epileptic seizures using CNN fed with image-like EEG data, thereby resulting in an average sensitivity as high as 87.85%. Rosas-Romero & Guevara, (2020) proposed the use of Convolutional Neural Networks for the analysis and classification of functional nearinfrared spectroscopy (fNIRS) signals. Usman *et al*., (2020) put forward a method for the automated extraction of features from EEG signals using a Convolutional Neural Network, allowing for sensitivity as high as 92.7% and 90.8% specificity. Table 2.11 presents a summary of various works on Deep learning methods for Epilepsy Prediction.

Table 2.11: Summary of Review on Deep Learning for Prediction of Epileptic

Work	Database	Predictive Characteristics	Model Positive/	Pred hr	Sen	False	Acc	Comments
(Khan *et al.*, 2017)	MSSM CHB-MIT	Wavelet Transform	CNN	8 min 6 min	87.8	0.142	N/M	The data set are few in numbers and not available publicly
(Truong *et al.*, 2017)	Freiburg CHB-MIT	STFT	CNN	5 min	81.4 81.2	0.06 0.16	N/M	The data set are few in numbers and not available publicly
(Hussein *et al.*, 2019)	Melbourne Seizure prediction competition dataset	STFT	CNN	5	87.8 min	N/M	N/M	The data set are few in numbers
(Tsiouris *et al.*, 2018)	CHB-MIT	Various Time and Frequency Features	LSTM	15120 min	99.28	0.11-0.02	N/M	The data set are few in numbers
(Truong *et al.*, 2019)	Freiburg CHB-MIT	STFT	GAN	5 min	N/M	N/M	N/M	The data set are few in numbers
(Daoud & Bayoumi, 2019)	CHB-MIT	Raw data	DCAE + Bi- LSTM	1 hr	99.72	0.004	N/M	The data set are few in numbers
(M. Sun *et al.*, 2018)	Dog iEEG	Discrete Fourier Transform	CNN RNN	N/M	N/M	N/M	N/M	The data set are few in numbers
(Rosas-Romero & Guevara, 2020)	NotreDame du Centre Hospitalie	Functional near-infrared spectroscopy (fNIRS)	CNN	N/M	N/M	N/M	N/M	The data set are few in numbers

(Yuan Zhang *et al.*, 2019)	CHB-MIT	Wavelet Packet Decomposition	CNN	N/M	92.2	N/M	N/M	The data set are few in numbers
(Usman *et al.*, 2020)	CHB-MIT	STFT	CNN		92.7	N/M	N/M	The data set are few in numbers

2.12 Optimization Algorithms

Various machine learning and neural networks are faced with the challenging problem of optimization. Optimization is the process that involves obtaining minimum or maximum function evaluation by finding a set of inputs to an objective function (Kochenderfer & Wheeler, 2019). In machine learning, optimization algorithms are adopted to reduce a function known as the loss function, while in neural networks, optimization is achieved through a process known as backpropagation (Kochenderfer & Wheeler, 2019). Several optimization algorithms exist, this section, therefore, reviews some of them.

2.12.1 BAT algorithm (BA)

The Bat algorithm (BA), a population-dependent stochastic search technique, has primarily been used to solve different kinds of optimization problems; however, one of the main problems that BA has encountered is that, when dealing with complex real-world problems, it is frequently caught in local optima (Bangyal, *et al*., 2018). It is a natural-inspired method that works well for both continuous and discrete optimization issues. It was created using natural processes, such as the echolocation of microbats, and is effective at resolving extremely challenging optimization issues (Bangyal *et al*., 2018; Fister *et al*., 2014). The pseudocode for the bat algorithm is displayed in Table 2.12.

Table 2.12: Pseudocode of BAT Algorithm (Yang, 2010)

```
Objective function $f$(x), $x = (x_1,..., x_d)^T$
Initialize the bat population $x_i$ ($i$ = 1, 2, ..., n) and $v_i$
Define pulse frequency $f_i$ at $x_i$
Initialize pulse rates $r_i$ and the loudness $A_i$ while (t
< Max number of iterations)
Generate new solutions by adjusting frequency,  and
updating velocities and locations/solutions        if
(rand > ri)
    Select a solution among the best solutions
    Generate a local solution around the selected best solution        end if
    Generate a new solution by flying randomly
```

```
if (rand < $A_i$ & $f(x_i) < f(x_*)$)        Accept the new solutions        Increase ri and reduce Ai
        end if
Rank the bats and find the current best $x_*$ end while
Postprocess results and visualization
```

2.12.2 Grasshopper optimization algorithm (GOA)

The GOA is a swarm intelligence system that takes cues from the swarming and foraging habits of grasshoppers. Numerous optimization issues have been solved by the GOA algorithm in a variety of domains (Meraihi *et al*., 2021). A three-bar truss, a 52-bar truss, and a cantilever beam were all subjected to structural optimization to identify the optimal shape, and the GOA is a algorithm that was introduced in 2017 (Steczek *et al*., 2020). The pseudocode for the Grasshopper Optimization Algorithm is displayed in Table 2.13.

Table 2.13: Pseudo-codes for the GOA

```
Generate the initial population of Grasshoppers $P_i$ (i =1, 2, ..., n) randomly
Initialize $c_{min}$, $c_{max}$ and maximum number of iterations $t_{max}$
Evaluate the fitness f ($P_i$) of each grasshopper $P_i$
T = the best solution
while (t < tmax ) do
    Update c1 and c2
    for i = 1 to N (all N grasshoppers in the population) do
        Normalize the distance between grasshoppers in the range [1,4]
        Update the position of the current grasshopper
        Bring the current grasshopper back if it goes outside the boundaries
    end for
    Update T if there is a better solution
    $t = t + 1$
end while
Return the best solution T
```

2.12.3 Grey wolf optimization algorithm (GWO)

GWO is a meta-heuristic optimization technique that mimics the hierarchical essence of grey wolves by naming the strongest solution alpha, then beta, and finally delta in decreasing order. In addition, the hunting strategy of detecting, encircling, and attacking is mathematically modeled to find the best-optimized response (Hatta *et al*., 2019). Table

2.14 shows an overview of the GWO algorithm.

Table 2.14: Pseudocode of GWO Algorithm (Faris, Aljarah, Al-Betar, & Mirjalili, 2018)

```
Initialize the grey wolf population Xi (i – 1,2, …, n)
Initialize d, A, and C
Generate the Randomly Positions of Search Agent
Calculation of the fitness of each search agent
$X_\alpha$ = the best search agent
$X_\beta$ = the second-best search agent
$X_\delta$ = the third-best search agent While
(t<max number of iterations)     for
each search agent
        Update the position of the current search agent by $\vec{X}(t+1) = \vec{X}_1 + \vec{X}_2 + {\vec{X}_3}/{3}$
    End for
    Update d, A, and C
    Calculation of the fitness of all search agents
    Update $X_\alpha$, $X_\beta$ and $X_\delta$
End while  Return
$X_\alpha$
```

2.12.4 Particle swarm optimization algorithm (PSO)

The optimal solution to a problem based on an objective function is found using the swarm behavior of bird flocks in PSO, a heuristic optimization technique (Xu *et al*., 2018). One of the most crucial techniques in swarm intelligence is this one. The Particle Swarm Optimization Algorithm is a powerful tool for nonlinear, constrained, and unconstrained optimization problems. While solving multi-modal optimization problems, its rapid convergence causes it to frequently enter local optima, which can cause particle swarms to form too soon (Xu *et al*., 2018). The general PSO's pseudocode is displayed in Table 2.15.

Table 2.15: Pseudocode for PSO (El-Shorbagy & Hassanien, 2018)

```
Step 1. Initialization
(a)  Set constants $k_{max}$, $c_1$, $c^1$.
```

[1] **.12.5 Salp swarm algorithm (SSA)**

(b) Initialize randomly particles positions x^i_0s in Rn for I = 1, …, p.

(c) Initialize randomly particles velocities $0 \leq \mathrm{v}^i_0 \leq \mathrm{v}^i_{max}$ for *I = 1, …, p.*

(d) Set *k* = 1

Step 2. Optimization

(a) Evaluate the function value f^i_k.

. *(b)* If $f^i_k \leq f^i_{best}$ then $f^i_{best} = f^i_k, p^i_k = x^i_k$

$f^i_k \leq f^g_{best}$ then $f^g_{best} = f^i_k, p^g_k = x^i_k$. (c) If

(d) If the stopping criterion is satisfied go to step 3

(e) All velocities of the particle are updated $\mathrm{v}^i_k$for *i = 1, …, p* by

$$v^i_{k+1} = v^i_k + c_1 \times r_1 \times \left(p^i_k - \mathrm{x}^i_k\right) + c_2 \times r_2 \times \left(p^g_k - \mathrm{x}^i_k\right)$$

(f) All positions of the particle are updated $\mathrm{x}^i_k$for *i = 1, …, p* by $_k\dot{\mathrm{x}}_{+1} = \mathrm{x}_{ki} + v_{ki+1}$ (g) *k = k + 1*

(h) Go to step 2(a)

Step 3. Termination

carried out on the prediction and detection of an epileptic seizure, showing some of their performance evaluation.

Table 2.16: Pseudocode for SSA (Singh *et al*., 2020)

```
Initialize population $x_i$  $i$=1, 2, …, $n$
for generation from 1 to max_$iter$ do
```

It's a metaheuristic algorithm based on the swarming activity of Salp in oceans, which aims at developing a population-based optimizer (Faris *et al*., 2020). It is an algorithm that is suitable for obtaining fast-speed searching results in challenging problems (Singh *et al*., 2020). SSA is a stochastic algorithm that begins the optimization process by generating a collection of initial random solutions to begin the population and then improves these solutions over time in two phases: exploration (diversification) and exploitation (intensification) (Abusnaina *et al*., 2018). Table 2.16 shows the Pseudocode of the Salp swarm algorithm, while Table 2.17 gives a summary of the various works that have been

```
Calculate all salp in the crowd Represent the best salp as F
Update the value of c1 by using $c_1 = 2e^{-(\frac{4l}{L})^2}$

for (all salp $x_i$) do
if $x_i$ is a leader then
Update the position of the leader by using $x_j^1 = \begin{cases} F_j + c_1((ub_j - lb_j)c_2 + lb_j)\ c_3 \geq 0.5 \\ F_j - c_1((ub_j - lb_j)c_2 + lb_j)\ c_3 < 0.5 \end{cases}$ else

Update the position of the followers by using the mathematical equation $x_j^i = \frac{1}{2}(x_j^i + x_j^{i-1})$
Return F
```

Table 2.17: Summary of Works on Prediction and Detection of Epileptic Seizure

Author/ Year	SVM	CHBMIT	Sensitivity = 92.23% Transform	
(Usman *et al*., 2017)	kNN	CHBMIT	Sensitivity = 97.44%	**Comments**
(Usman & Hassan, 2018)	SOM	CHBMIT	Sensitivity = 98%	The data set are few in numbers
(Kitano *et al*., 2018)	SVM	Freiburg	Sensitivity = 96.6%	The data set are few in numbers
(Sharif & Jafari, 2017)	Random Forest (RF)	Bonn =	Average Accuracy 98.45%	The data set are few in numbers The data set are few in numbers Contains variety of patient data, and are good. Contains variety of patient data, and are good
(Mursalin, Zhang, Chen, & Chawla, 2017)	kNN	Bonn	With Wavelet, Accuracy= 96% With Hilbert, Accuracy = 100%	
(Polat & Ozerdem, 2016)				
(Niknazar, Mousavi, Vahdat, & Sayyah, 2013)	ECoG	Bonn 98.67%	Accuracy =	Contains variety of patient data, and are good The data set are few in numbers
(Zhou, Liu, Yuan, & Li, 2013)	Bayesian Linear Discriminant analysis (BLDA)	Freiburg	False Detection Rate = 0.13/h, Sensitivity = 96.25%	

CHBMIT Sensitivity = 91-96% (Tzallas,
The data set are few in Tsipouras, ANN
(Shoaib, Lee, Jha, & SVM
numbers Contains Bonn Accuracy
Verma, 2014) (Desai,
Bonn Accuracy = 96% variety of patient = 100%
2017) ANN
data, & Fotiadis,
CHBMIT Sensitivity = 99.72% and are good 2009)
The data set are
(Daoud & Bayoumi, Raw Data
few in numbers
2019)
Freiburg Sensitivity = 94% The data set are few in
(Y. Yang *et al*., SVM
numbers The data
2018)
CHBMIT Sensitivity = 99.28% set are few in numbers
(Tsiouris *et al*.,
2018) Time and
Frequency Bonn Accuracy = 97.77% Contains variety
Features of
(Guo *et al*., 2010)
patient data.

Method **Dataset** Discrete **Performance**
Wavelet

2.13 Performance Evaluation of Works Reviewed

There is a need to monitor and evaluate the performance of various systems and algorithms for the detection and prediction of epileptic seizures. This section, therefore, presents an overview of the metrics with which these systems are evaluated as well as a performance review of previous works that have been carried out. Some of the metrics used in the evaluation of the various seizure prediction and detection systems are (Giannakakis *et al*., 2014; Rasheed *et al*., 2020):

i. Accuracy

The accuracy of a seizure detection system is calculated as shown in Equations 2.12 and 2.13

$$\text{Accuracy} = \frac{\text{NumberCorrectlyDetectedSeizures} + \text{CorrectlyNormanlState}}{\text{TotalNumberof Cases}} \qquad (2.12)$$

$$\text{Accuracy} = \frac{TP + TN}{TP+FP+FN+TN} \qquad (2.13)$$

Where TP= true positive, TN= true negative, FP= false positive and FN = false negative.

ii. Precision

The precision of a seizure detection system is calculated as shown in Equations 2.14 and 2.15.

$$\text{Precision} = \frac{\text{NumberCorrectlyDetectedSeizures}}{\text{TotalNumberof seizure}} \qquad (2.14)$$

$$\text{Precision} = \frac{TP}{TP+FP} \qquad (2.15)$$

iii. Sensitivity

The sensitivity of a seizure detection system is calculated as shown in Equations 2.16 and 2.17.

$$\text{Sensitivity} = \frac{\text{NumberCorrectlyDetectedSeizures}}{\text{TotalNumberof AlgorithmPositiveOutcome}} \qquad (2.16)$$

TP

$$\text{Sensitivity} = \frac{}{TP+FN} \qquad (2.17)$$

iv. Specificity

The specificity of a seizure detection system is calculated as shown in Equations 2.18 and 2.19.

$$\text{Specificity} = \frac{\text{Numberof CorrectlyStates}}{\text{TotalNumberof AlgorithmNegativeOutcome}} \qquad (2.18)$$

$$\text{Specificity} = \frac{TN}{TN+FP} \qquad (2.19)$$

2.14 Chapter Summary

This chapter presents a review of epilepsy detection and prediction using various machine algorithms. Specifically, a machine learning algorithm was discussed. Feature selection and the use of EEG for monitoring were discussed. A review of some epilepsy detection methods was discussed. From the review, the prediction and detection of epileptic seizures are found to be a very important process that greatly affects the quality of life of patients with epileptic disorders. Various researchers have come forward to propose and develop schemes and systems that help improve the process of prediction and detection of epileptic seizures. As shown in the literature review, various Machine learning and Deep learning schemes have been adopted together with various EEG datasets to develop better and improved systems. Additionally, it is demonstrated that no single machine learning or deep learning method is much better than others because the effectiveness of the various schemes depends on the characteristics and structure of the EEG dataset that was used. Again, it was discovered that the framework for seizure detection and prediction has room for development. Since a variety of physiological factors can alter EEG recording and make it more difficult to discern between distinct brain states, the quest for efficient biomarkers is crucial. The Improved Grasshopper Optimization Algorithm and Artificial Neural Network technique for detection and classification were used in this study to provide the best feature selection sets for epilepsy detection. This improves the overall efficiency of epilepsy detection toward clinical application.

CHAPTER THREE

3.0 RESEARCH METHODOLOGY

3.1 Introduction

The resources and research techniques are described in this section. This chapter also covers the research methodology, materials needed, techniques and approaches used, data set, data collection process, and metrics for evaluation using epilepsy classification and the

IGOA-ANN approach.

3.2 Research Design and Methods

The steps of the methodology mapped to the set of objectives (as defined in section 1.4) are shown in Figure 3.1. For research objective one, the following step was taken:

1. Acquire the Epilepsy Dataset

   The epilepsy dataset was acquired from the University of Bonn. The sample of the dataset is shown in Table 3.1.

Table 3.1: Individual Datasets and the Number of EEG Segments used in each Detection Task

Class	Settings	A	B	C	D	E
1.	Subject	5 Healthy	5 Healthy	5 Epilepsy Patient	5 Epilepsy Patient	5 Epilepsy Patient
2.	Number of Epoch	100 Segments	100 Segments	100 Segments	100 Segments	100 Segments
3.	Epoch Duration	23.6s	23.6s	23.6s	23.6s	23.6s
4.	Patient Recording	Eyes Open State	Eyes Close Recording	Pre-seizure Recording from the Healthy Region	Pre-seizure Recording from Epileptic Area	Recording During Seizure
5.	Electrode Type	Surface	Eyes Close Recording	Pre-seizure Recording from the Healthy Region	Pre-seizure Recording from Epileptic Area	Recording During Seizure
6.	Electrode Placement	10-20 System	10-20 System	Hippocampal Formation	Epileptogenic Zone	Epileptogenic Zone

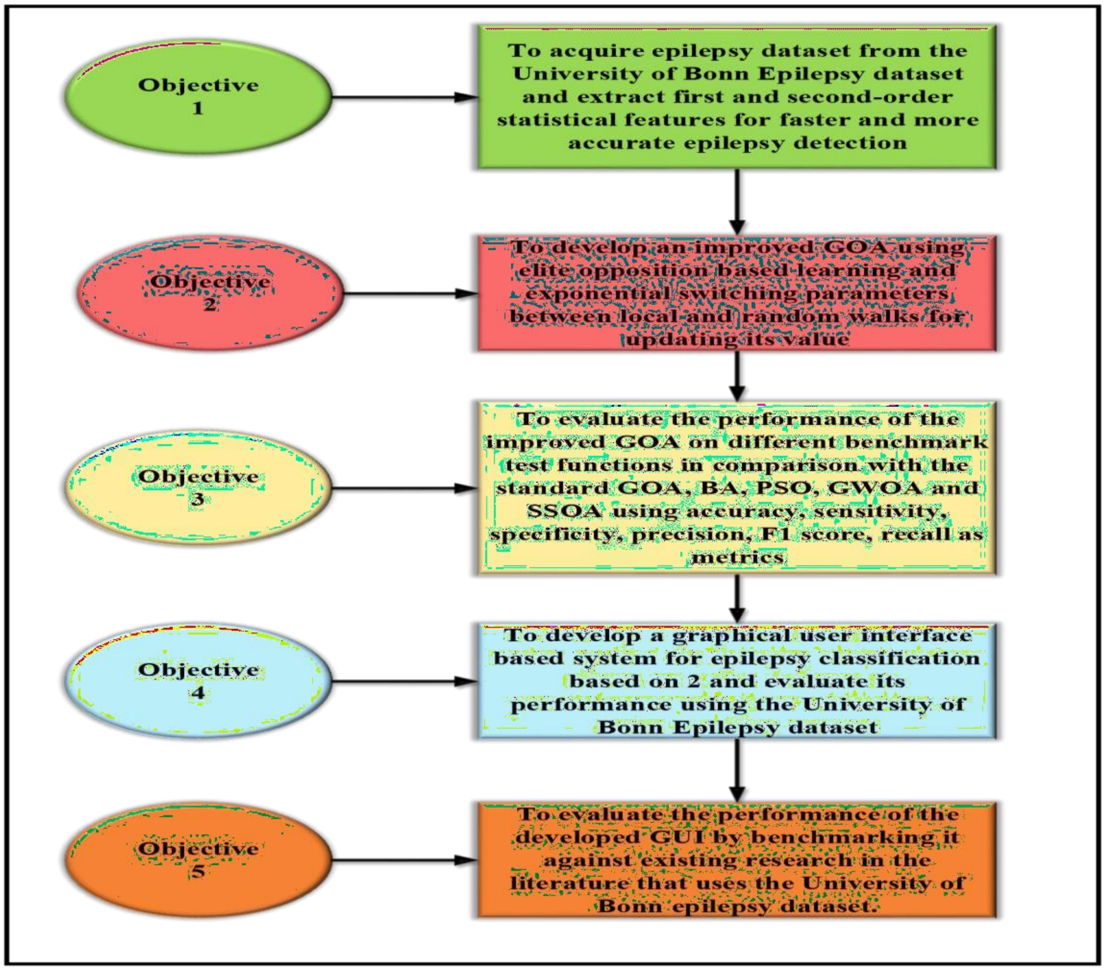
Objective 1
To acquire epilepsy dataset from the University of Bonn Epilepsy dataset and extract first and second-order statistical features for faster and more accurate epilepsy detection
Objective 2
To develop an improved GOA using elite opposition based learning and exponential switching parameters between local and random walks for updating its value
Objective 3
To evaluate the performance of the improved GOA on different benchmark test functions in comparison with the standard GOA, BA, PSO, GWOA and SSOA using accuracy, sensitivity, specificity, precision, F1 score, recall as metrics
Objective 4
To develop a graphical user interface based system for epilepsy classification based on 2 and evaluate its performance using the University of Bonn Epilepsy dataset
Objective 5
To evaluate the performance of the developed GUI by benchmarking it against existing research in the literature that uses the University of Bonn epilepsy dataset.

Figure 3.1: Research Methodology

2. Preprocessing the Dataset

To effectively suit the networks, this entails assessing the gathered data. The data was altered and transposed to achieve this. Utilizing the data (load) in the network for effective subject classification is made simple by pre-processing. There are five sets of EEG data in this study, each comprising 100 single channels that last for 23.6 seconds. There are 4,096 samples total from one EEG time series included in every channel. The time series' acquisition device's spectral bandwidth ranged from

0.5 Hz to 85 Hz.

With low-pass filter settings ranging from 0.5 to 40 Hz, the signals were recorded at a frequency of 173.61 Hz. Low pass filters block signals that travel above a cutoff frequency but let signals flow within a pass-band (also known as the pass-band) below the cutoff frequency (known as the stop-band). As a result, the filter alters the output values to facilitate the detection of trends and minimize signal loss while increasing the overall signal-to-noise ratio. Computation of the First-order and second-order statistical Feature extraction for epilepsy classification.

In the algorithm, discrete wavelet transforms are essential. Continuous wavelet transform decomposes the signal into scaled and shifting parameters. ($S_{,,b}(t)$) of a single function $S(t)$, called mother wavelet. If $x(t)$ is the signal, then its CWT is given as (Chen *et al.*, 2017):

$$S_{ab,}(t) = \square_{-}{}^{\square}{}_{\square} x(t)\overline{\sqrt{|a|}}\square \;\square\; \overline{a}\; t \quad \square t - b\,\square\; \square\square$$ (3.1) were,

$a =$ Scale parameter

b = Translation parameter
$a, b \in \mathbb{R}$and$a \neq 0$

The discrete wavelet transform (DWT) was created by discretizing the parameters a and b. In its most typical setup, the DWT uses a dyadic sample with parameters a and b based on powers of two: $a = 2^j$ and $b = k2^j$ with $j, k \in \mathbb{Z}$. By substituting in equation 3.1, this is called dyadic wavelets (Chen *et al.*, 2017):

$$-j$$

$$—$$

$$S_{j,k}(t) = 2_{\,z} S\left(2_{-j}\, t - k\right)$$ (3.2) of

note, DWT could be written as (Chen et al., 2017):

$$-j$$

$$—$$

$$=\square\square \qquad \square\square$$

$$\square_{-}{}^{\square}{}_{\square} x(t) 2^{\,z} \quad (2 \qquad S^{-j}\, t - k\,)dt\, S(t),$$

$S_{j,k}(t)$ (3.3) were, $d_{j,k}$ $d_{j,k} =$

=Wavelet coefficients, $j =$ Level, and $k =$ Location

The automated detection of EEG data typically involves the two main goals of feature extraction and classification. Statistics features, fractal dimension features, entropy features, and time-frequency domain characteristics are four categories into which the returned features can be divided. The combination of temporal and frequency data has been utilized in many research to automatically identify nonstationary EEG at the onset of epilepsy. The bulk of studies has used a supervised learning paradigm to automatically detect EEG using machine learning.

Regardless of the classification, the input EEGs fall into, the EEGs used to train classifiers are labeled using prior data (Wang *et al*., 2019). To do the intended objective, feature extraction includes transforming input data into a set of features that extract information from the input data. Maximum, mean, variance, standard deviation, entropy, energy, RMS, kurtosis, and skewness are some of the qualities that were discovered in this work. The maximum value displays the rapidity with which patterns increase during seizure activity. The mean provides the absolute values of each channel signal. The variance of the signals demonstrates how data are spread out and close to the mean. A measure that deviates from the variance is the standard deviation. These alterations signify the change from a regular state to an epileptic one (Swami *et al*., 2016). A characteristic of an energy feature is nonlinearity. The brain is made up of linked neurons that cooperate to carry out specific tasks.

Neurons require a lot of energy to function properly. The start of epileptic convulsions causes energy imbalances (Swami *et al*., 2016). Entropy, another nonlinear characteristic, is characterized as the degree of uncertainty or randomness in the signal. Entropy level abnormalities are also brought on by the beginning of epileptic convulsions. The mathematical expression of the maximum value is given as:

$$\square_x = \square \; ax\left(x(n)\right) \qquad (3.4)$$

where M denotes the highest value. The mean value gives the absolute values for each channel signal. The mean is expressed mathematically as follows:

$$\square_t = \frac{1}{\square} \square \sum_{i=}^{1} x(t_i) \qquad (3.5)$$

the mean of the input signals is μ. The variance of the signals indicates how data are spread out

$$\square = \frac{\mathbf{1}}{n} \square \sqrt{\square_{i=1}\left(x(i) - \square\right)^2} \qquad (3.7)$$

where σ is the standard deviation. Energy it is said that energy is the work capacity. The brain is made up of linked neurons that cooperate to carry out specific tasks. For them to perform their usual function, neurons need a lot of energy. Energy levels become unbalanced when epileptic convulsions begin. Nonlinearity is a property of energy.

The following is the mathematical representation of energy:

and close to the mean. The following is the mathematical representation of variance: (Ajay & Lalit, 2015):

$$2 \quad 1 \,□ \quad □ \quad □_t = n \;□\; {}_{i=1}\left(x(i)\ -\ \right)$$

(3.6) where variance is σ_t. A signal's spread-out standard deviation from the variance It is a transition from the regular state to the epileptic state that takes place. The standard deviation is expressed mathematically as follows:

$$1 \quad □ \quad 2 \quad □_n = \;□□ \quad □ \quad {}_{n=1}x(n)$$

(3.8)

E_n represents the energy.

Entropy, a nonlinear characteristic, is the degree of uncertainty or unpredictability in the signal. Entropy level abnormalities are also brought on by the beginning of epileptic convulsions. Entropy is expressed mathematically as follows:

$$E = -\sum_{n=1}^{N}\left(\ \left(\ {}_{n}\right)\right)^2 * \quad {}_2\ \left(\ {}_{n}\right) p\ x\ \log\ p\ x\ x_2$$

(3.9) where E is the entropy, $(p(x_n)$ is the probability for the input signal (x_n), n represent each signal, and N represents the total number of records of the signal of the dataset. A statistic known as kurtosis is used to describe the distribution of observed data around the mean. It speaks of the height at which a dataset peaked.

It can be calculated as follows mathematically:

$$\frac{1 \quad N \quad 4}{N\ □1\ {}_{i=1}\left({}^{N}x(n) - {}^{2}□□\right.}$$

$$k_n\ □□□\ N\ □_{i=1}(□x)\ □□ \quad = \quad x_2 \quad)$$

(3.10)

Since data grow more symmetrical as their value approaches zero, skewness is defined as asymmetry from the normal distribution in a set of statistical data. However, data that is positively skewed, also known as right-sided skewed, has a positive value, whereas negatively skewed data, also known as left-sided skewed, has a negative value. By definition, regularly distributed data has minimal skewness. You can gauge skewness by using:

$$\varphi_t = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{x(t_i)-\mu_t}{\sigma_t}\right)^3 \qquad (3.11)$$

In this work, a signal strength estimator for EEG frequency bands is called root mean square (RMS). It is defined as: It gives a measurement of the size of the

variable quantity.

$$RMS = \sqrt{\frac{1}{n}\sum_{i=1}^{N} x(i)^2} \qquad (3.13)$$

RMS represent the time series signal can be determined as the square root of the mean of all the signal samples. Sample entropy can be used to calculate signal complexity, which is regarded as a crucial aspect (Fergus *et al.*, 2015). The suggested model for extracting epilepsy features from EEG recordings is shown in Figure 3.2.

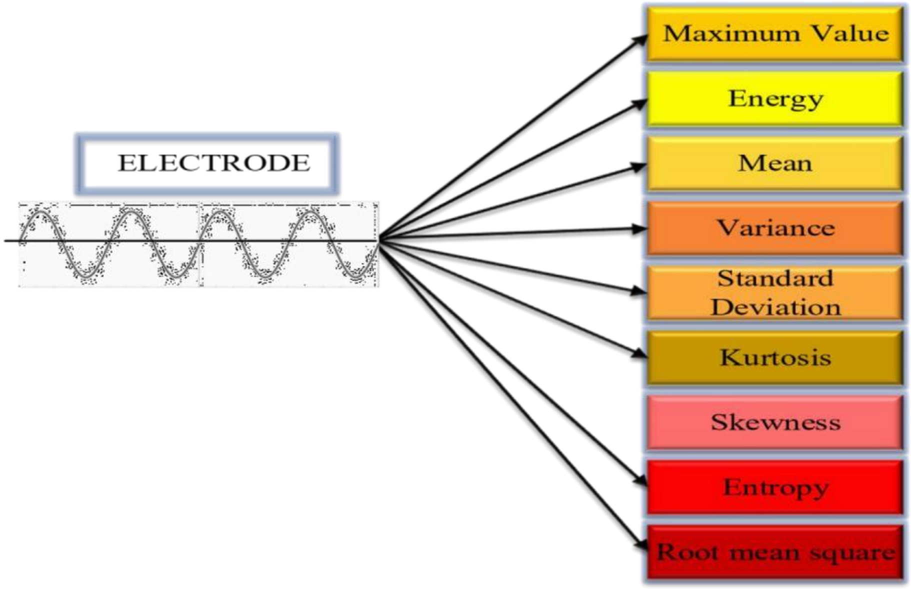

Figure 3.2: Proposed Model for Epilepsy Seizure Detection from EEG Signal

3.3 Methodology for the Improved Grasshopper Optimization Algorithm (IGOA) for Optimal Feature Selection

According to Hassanien and Emary (2018), GOA is a meta-heuristic technique that may be utilized to address optimization problems and yields excellent results. The GOA behaves in a way that is similar to grasshopper swarms in nature. The mathematical model used to replicate the swarming behavior of grasshoppers is as follows, according to Saremi et al.

(2017):

$$X_i = S_i + G_i + A_i \qquad (3.14)$$

where X_i defines the position of the i-th grasshopper, S_i is the social interaction, G_i is the gravity force on the i-th grasshopper, and A_i shows the wind advection. The S component is calculated as follows:

$$S_i = \square_{N\ j=1 j\ 1}\, s\left(d_{ij}\right) \hat{d}_{ij} \qquad (3.15)$$

where d_{ij} is the distance between the i-th and the j-th grasshopper, calculated as $d_{ij} =$

$|X_j - X_i|$, N is the number of grasshoppers, and $\hat{d}_{ij} = \frac{x_j - x_i}{d_{ij}}$ is a unit vector from the i-th grasshopper to the j-th grasshopper. The s function, which defines the social forces, is calculated as follows:

$$s(r) = fe^{\frac{-r}{t}} - e^{-r} \qquad (3.16)$$

f denotes the attractiveness's intensity. This is how the G component is determined:

$$G_i = -g\hat{e}_g \qquad (3.17)$$

where g is the gravitational constant and $\hat{e}_g$ shows a unity vector towards the centre of the earth. The A component is calculated as follows:

$$A_i = u\hat{e}_w$$

(3.18) where u is a constant drift and $\hat{e}_w$ is a unit vector in the direction of the wind.

Substituting *S, G,* and *A* in Equation 3.14, this equation can be expanded as follows:

$$X_i = \sum_{j=1, j\neq i}^{N} S\left(\left|X_j - X_i\right|\right)\frac{x_j - x_i}{d_{ij}} - g\hat{e}_g + u\hat{e}_w \qquad (3.19)$$

Because the grasshoppers quickly enter their comfort zone and the swarm does not converge to a predetermined place, this mathematical model cannot be used to directly resolve optimization problems. The updated form of this equation shown below can help with optimization problems:

$$X_j^d = c\left(\sum_{j=1, j\neq i}^{N} c\,\frac{ub_d - lb_d}{2}\, S\left(\left|X_j^d - X_i^d\right|\right)\frac{x_j - x_i}{d_{ij}}\right) + \hat{T}_d \qquad (3.20)$$

where ub_d is the upper bound in the Dth dimension, lb_d is the lower bound in the Dth dimension, $S(r) = fe^{\frac{-r}{t}} - e^{-r}$, $\hat{T}_d$ is the value of the Dth dimension in the target (best solution found so far), and *C* is a decreasing coefficient to shrink the comfort zone, repulsion zone, and attraction zone. Note that *S* is almost similar to the *S* component in Equation 3.14. However, do not consider gravity (no *G* component) and assume that the wind direction (*A* component) is always towards a target T_d. A grasshopper's next location is decided by its current location, the location of the target, and the locations of every other grasshopper, as shown by Equation 3.20.

Keep in mind that the present grasshopper's position with other grasshoppers is taken into account in the first portion of this equation. The coefficient c reduces the comfort zone according to the number of iterations, as shown by the following formula:

$$C_{max} - C_{min}$$

$$C = C_{\max} - L \qquad (3.21)\ L$$

where $C_{\max}$ is the maximum value, $C_{\min}$ is the minimum value, l indicates the current iteration, and L is the maximum number of iterations.

A few drawbacks of GOA include its lack of a theoretical convergence feature, premature convergence in some sophisticated optimization approaches, and difficulties in fully utilizing the search space. To overcome this drawback, this research introduces elite opposition-based learning and exponentially switching parameters to balance between exploitation and exploration problems and the premature convergence as shown in Equations 3.22 and 3.23. the GOA is enhanced by hybridizing with the Elite Opposition Base Learning (EOBL) technique to improve the capability to explore the search domain as well as the approach to an optimum at a fast speed. The EOBL methodology is used after completing the exploration phase of the GOA. Figure 3.3 presents the Flowchart of the IGOA. Elite opposition:

For solution, X_i^t its elite opposition-based solution $EX_i\ ex_{i,1}^t, ex_{i,2}^t, \ldots□□, ex_{i,D}^t$ □□ is calculated (Mirjalili, 2015; Seyedali, 2015):

$$ex_{i,j}^t = k\cdot\left(EA_j^t + EB_j^t\right) - x_{i,j}^t \qquad (3.22)$$

$$EA_j^t = \min\left(ex_{m,j}^t\right), EB_j^t = \max\left(ex_{m,j}^t\right), \qquad (3.23)$$

$$ex_{i,j}^t = rand\left(EB_j^t - EA_j^t\right) + EA_j^t, if\, ex_{i,j}^t\ □ \quad LB_i \,\|\, □\, ex_{i,j}^t\ UB_i, \qquad (3.24)$$

$$where\, i = 1,2,\ldots\ldots,SN;\ j = 1,2,\ldots.,D; M = 1,2,\ldots,EN;\ k = rand(0,1),$$

$$EX_{mt} = □□\, ex_{mt,1}, ex_{mt,2},\ldots\ldots..,ex_{mt,D}\ □□\ ,m = 1,2,\ldots\ldots, EN. And\, ex_{it,j}$$

is an elite opposition-based solution of $ex_{i,j}^t$,EN a number of the selected elite solutions, usually set to be SN □1, as recommended in the literature (Chickermane & Gea, 1996). EB_j^t and EA_j^t are the $j-th$ dimension maximum and minimum values of the selected elite solutions, respectively. The generalized coefficient k can be used to produce various elite opposing individuals and then to build an opposite population that is appropriate for looking at the neighborhood space of the elite opposing persons, according to the elite oppositionbased learning theory. This smart technique improves the algorithm's ability to be exploited locally. The step for the improved GOA is:

i. Generate the initial population of Grasshoppers P_i *(i=1, 2, ..., n)* randomly
ii. Initialize c_{min}, c_{max} and maximum number of iterations, t_{max} iii.
Evaluate the fitness f(Pi) of each grasshopper Pi iv. T = the best solution
v. while (t < tmax) do vi. Update c1 and c2 using equation 3.21
exponentially vii. for i = 1 to N (all N grasshoppers in the
population) do viii. Normalize the distance between grasshoppers in the
range [1,4] ix. Update the position of the current grasshopper using
equation 3.20

x. Applied EOBL to update the P^{th} element of the search agent xi. End if
xii. End for xiii. Bring the current grasshopper back if it goes outside the
boundaries xiv. End for xv. Update T if there is a better solution xvi. t = t +
1 xvii. End while xviii. Return the best solution T

3.3.1 Improved GOA and standard GOA performance evaluation

IGOA and GOA performance were benchmarked using the unimodal and multimodal benchmark test functions, respectively displayed in Tables 2.6 and 2.7 in chapter two. Because the algorithms only had one global optimal solution, unimodal functions were used to evaluate their usability. Multimodal test functions are used to evaluate the algorithm's exploration because they have a lot of local optima. The best, worst, mean values and standard deviation for each function were recorded across a total of 10 different runs, each with 200 iterations. The best values test if the algorithms were successful in finding the global optimum value while the consistency of the algorithms in getting the ideal fitness values was assessed using the mean value over ten runs.

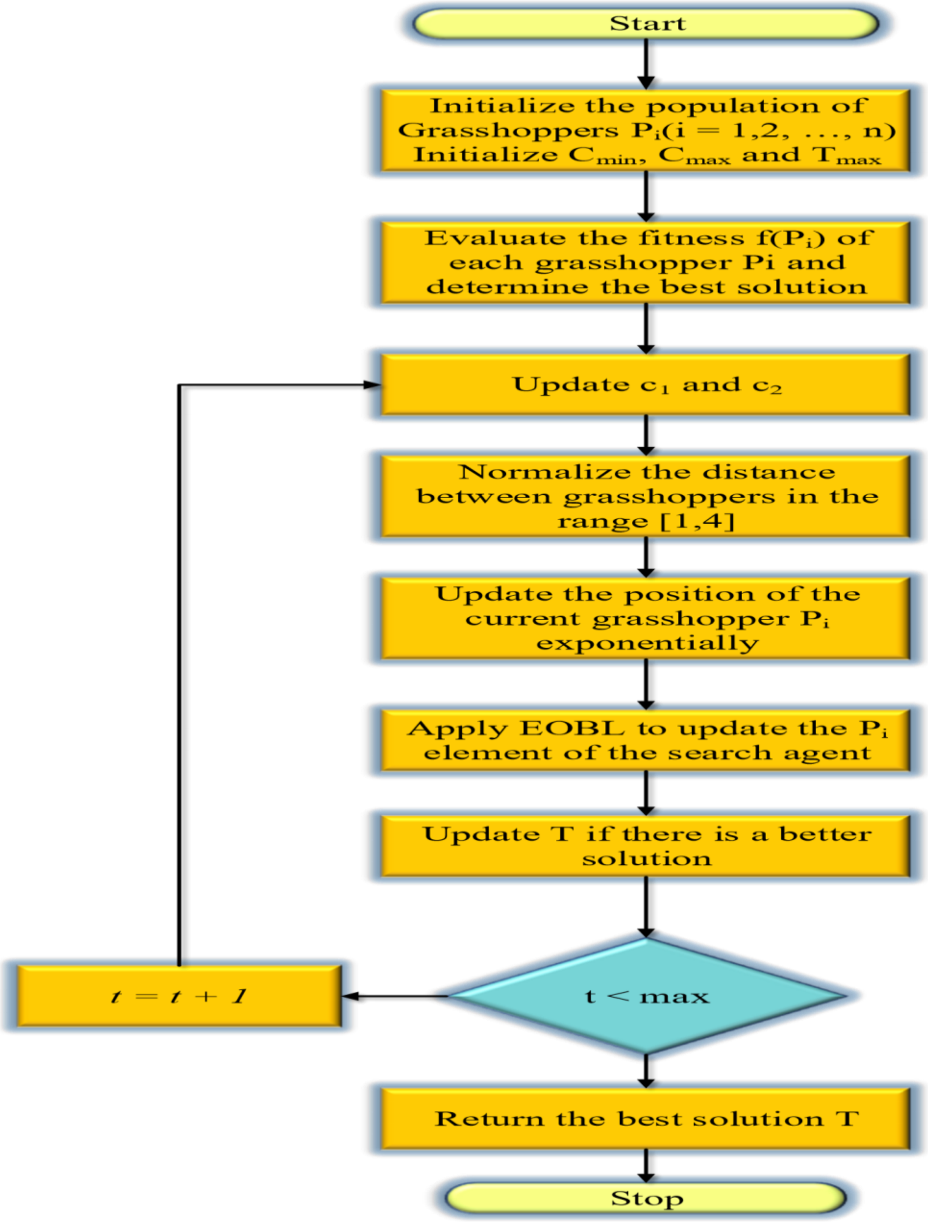
Start
Initialize the population of Grasshoppers $P_i(i = 1,2, \ldots, n)$ Initialize C_{min}, C_{max} and T_{max}
Evaluate the fitness $f(P_i)$ of each grasshopper Pi and determine the best solution
Update c_1 and c_2
Normalize the distance between grasshoppers in the range [1,4]
Update the position of the current grasshopper P_i exponentially
Apply EOBL to update the P_i element of the search agent
Update T if there is a better solution
t < max
$t = t + 1$
Return the best solution T
Stop

Figure 3.3: Flowchart of the Improved Grasshopper Optimization Algorithm

3.4 Methods for Epilepsy Feature Selection for Classification

This section presents the research methodology for epilepsy feature selection using IGOA, GOA, PSO, SSOA GWO, and BA for classification for research objective three. The stepby-step for each optimization algorithm is shown in tables 2.12, 2.13, 2.14, 2.15, 2.16 and

2.17 respectively.

3.5 Methods for Epilepsy Classification using Artificial Neural Network

The epilepsy-related variables that were obtained from EEG signals are fed into a machinelearning algorithm to train and test the efficacy of the suggested methodology. The two parameters of machine learning methods for artificial neural networks are optimized using IGOA, GOA, PSO, GWO, SSOA, and BA. Relevant features are selected from the full feature set for categorization. In section 3.5.1, machine learning algorithms for artificial neural networks and their parameters are briefly described.

3.5.1 Artificial neural network (ANN)

ANN is a pattern categorization tool that uses nonlinear data and is modeled after a biological neural network (Singh *et al*., 2021). It is one of the most commonly utilized classifiers for the classification of seizures in EEG signals and is made up of interconnected layers of neurons that are employed to carry out computation in pattern recognition (Kaur & Singh, 2017; Singh *et al*., 2015). The layers are the input layer, hidden layer, and output layer. The weights assigned to each neuron in the layer below serve as links between the various neurons in each layer. The neurons' inputs are multiplied by weights to create their inputs, and a transfer function creates their outputs (Singh *et al*., 2019). In this study, a multilayer feedforward network is used. The number of classes in an output layer defines the size of the output layer, while the number of input characteristics determines the size of an input layer. Ten hidden neurons have been identified through experimentation. The Bayesian gradient function is used to train an ANN. The hyperbolic tangent sigmoid and the SoftMax functions serve as the transfer functions between an output layer and a hidden layer, respectively. Classification is one of the neural network applications and research fields with the most activity. The difficulties with posterior probability estimates, how conventional and neural classifiers relate to one another, the trade-off between learning and generalization in classification, the choice of feature variables, and the impact of misclassification costs. For classification, a multilayer feed-forward network is employed. The usefulness depends on assumptions and conditions, even though classical statistical categorization offers a wide range of approaches. Neural networks are used to address the aforementioned issue since they are universal functional approximators, self-adaptive approaches, and non-linear models. Figure 3.4 present the neural network architecture used for this research, Figure 3.4 presents the flow process for selection and classification, and Figure 3.6 for the flowchart of the proposed feature classification-based GOA. Models for multi-layer feedforward neural networks were developed. A nine input features chosen by the various algorithms are used to train each model. The optimal number of features needed for a particular ANN model's training is determined by the EOBL-IGOA method. The following parameters were used in designing the ANN models:

i. Number of inputs (selected by EOBL-IGOA and other algorithms) ii. Multilayer Feedforward Neural Network was chosen because of its generalization and faulttolerant ability.

iii. The number of neurons was chosen based on the rule of thumbs which states that the number of hidden layer neurons is $2/3$ of the size of the input layer.

iv. The training function: Bayesian gradient was chosen because of its faster and its ability to attain low MSE on time as compared to scaled conjugate gradient and other ANN training algorithms.

v. Activation Function: hyperbolic tangent sigmoid at the output layer and SoftMax function at the hidden layer sizes. This is because the expected output of the classification is either 0 or 1.

The steps for epilepsy classification using an Artificial Neural Network are:

i. Input the epilepsy dataset ii. Denoise the dataset using low-pass filters iii. Compute first order, second-order statistical features for each data iv. Select the optimal features using IGOA, GOA, PSO, GWO, SSOA, and BA

v. Store the optimal features as the knowledge base vi. Train the feedforward neural network with the stored extracted features and save the network vii. Test the feedforward network with the testing data (i.e., data that are unknown to the network) to classify the data. viii. Repeat steps 1 to step 8 for all the input

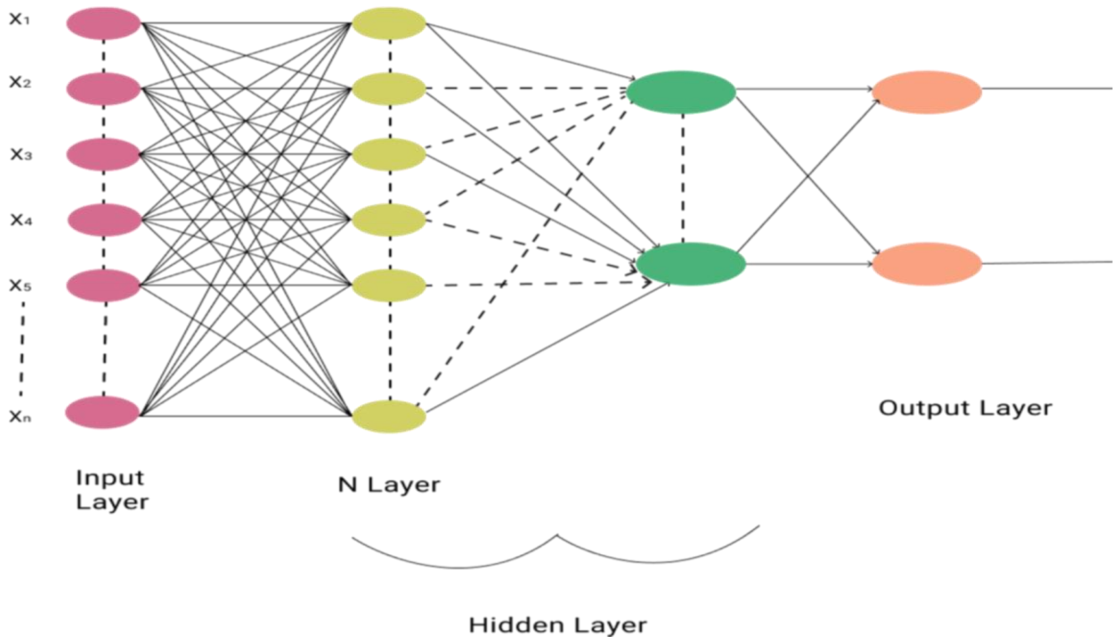

Figure 3.4: Neural Network Architecture

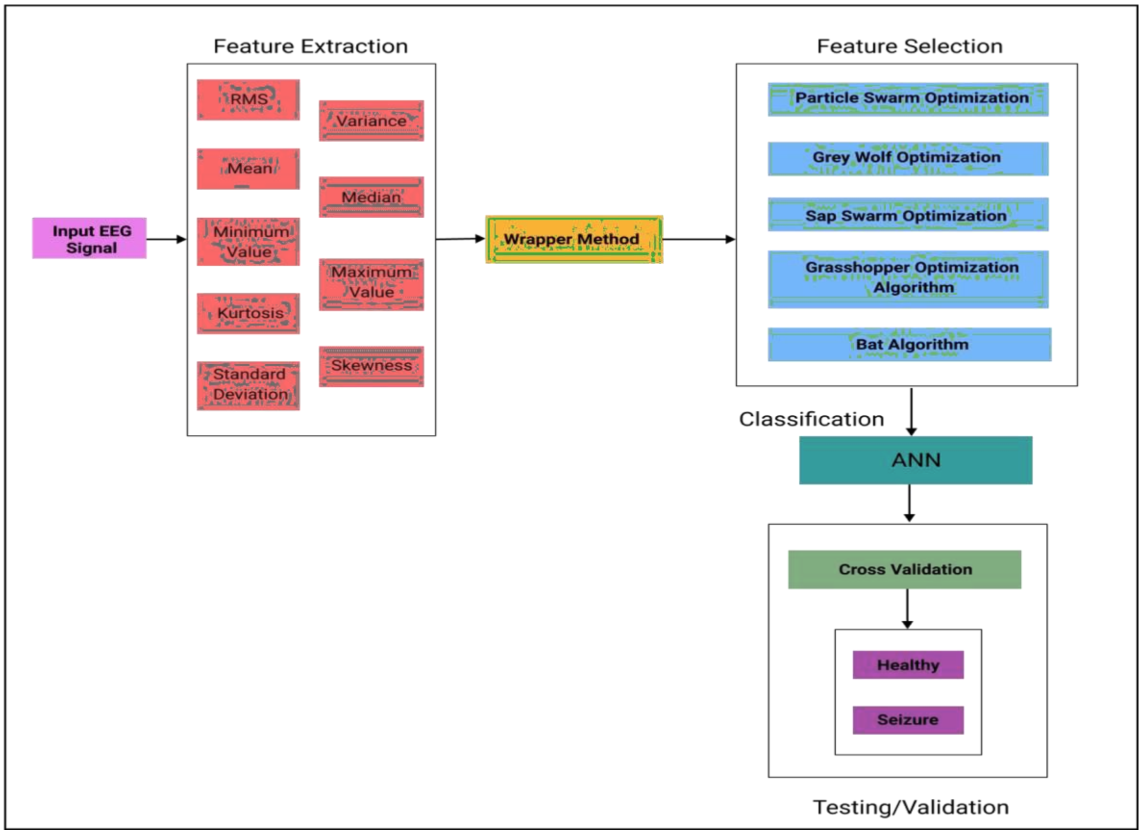

Figure 3.5: Flow Process for Selection and Classification

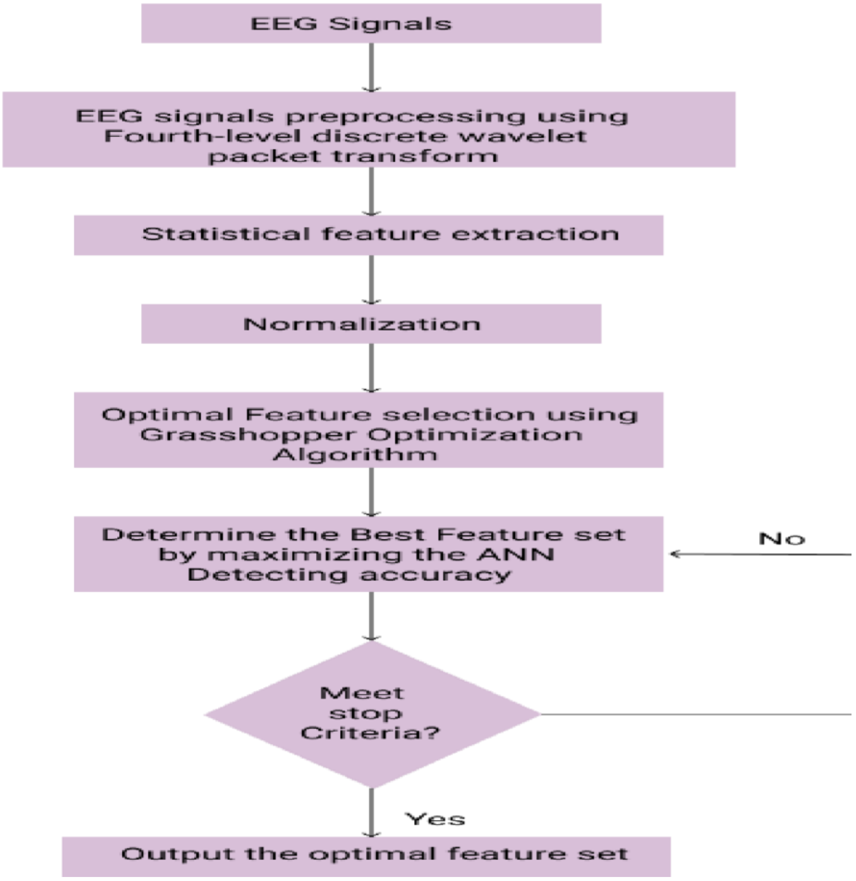
EEG Signals
EEG signals preprocessing using Fourth-level discrete wavelet packet transform
Statistical feature extraction
Normalization
Optimal Feature selection using Grasshopper Optimization Algorithm
Determine the Best Feature set by maximizing the ANN Detecting accuracy
No
Meet stop Criteria?
Yes
Output the optimal feature set

Figure 3.6: Flowchart of the Proposed Feature Selection Scheme Based IGOA

3.6 Performance Evaluation Measurement

Sets A, B, C, and D are considered the positive class in this study, while Sets E is considered the reverse. In this work, we employed five metrics—Accuracy (Acc), Sensitivity (Sens), Specificity (Spec), Precision (Prec), and F-Measure—to evaluate the classification performance for diverse test scenarios (F). In general, all of the performance

measurements described above have an impact on the four key metrics of a positive/negative binary classification result: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). The terms False Positive and True Positive denote the detection of seizure activity that was not a seizure, whereas True Negative denotes the identification of seizure activity that was actually not a seizure (Hamad *et al*., 2018; Swami *et al*., 2016).

i. True positive (TP): The proportion of EEG recordings that were determined to be ictal by both neurologists and artificial neural networks.

ii. True negative (TN): The proportion of EEG recordings that neurologists and artificial neural networks have determined to be seizure-free.

iii. False-positive (FP): The percentage of EEG recordings that the Artificial Neural

TP	True Positive	If someone has a seizure and it is appropriately identified as such by medical personnel.
TN	True Negative	if a person is healthy and the classifier detects no seizures.
FP	False Positive	when a seizure is detected in a normal person by the classifier.
FN	False Negative	erroneous detection when the seizure sufferer is identified by the classifier as normal.

Network classified as ictal but that neurologist classified as not seizure activity. iv. False-negative (FN): The percentage of EEG recordings that the artificial neural network classified as seizure-free but that neurologist classified as a seizure. Table 3.2 present the summary of the classification outcome with summary, while table

3.3 presents the formula for accuracy, sensitivity, specificity, precision, and F1 measure.

Table 3.2: Summary of Classification Outcome with Summary

Acronym	Detection Type	Real-World Scenario

Table 3.3: Formula for Accuracy, Sensitivity, Specificity, Precision, and F1 measure

	Class =1	Class = 0
Class = 1	True Positive (TP)	False positive (FP)
Class =0	False negative (FN)	True Negative (TN)

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + TN + FP} \times 100$$

$$\text{Sensitivity} = \frac{TP}{TP + FN} \times 100$$

$$\text{Specificity} = \frac{TN}{TN + FP} \times 100$$

$$\text{Precision} = \frac{TP}{TP + FP} \times 100$$

$$\text{Fmeasure} = 2 \times \frac{\text{Precision} \times \text{Sensitivity}}{\text{Precision} + \text{Sensitivity}} \times 100$$

CHAPTER FOUR

4.0 RESULTS AND DISCUSSION

4.1 Overview

The outcomes of the methodology carried out and discussed in chapter three to accomplish all the objectives of these research studies are reported in this chapter. The analysis of the results and how they answer the research questions are also presented.

This section presents the results of utilizing various classifiers to categorize the dataset of epileptic seizures as well as some modified parameters for a particular classifier. Working with a big dataset that has a lot of properties, 178 in all, is one of the implementation obstacles. Feature reduction can be used for the close classification of epileptic seizure cases with some selected features, as shown in the feature extraction in section 3.4. A sample of the waveform is shown in Figures 4.1, 4.2, 4.3, 4.4, and 4.5 for each of the classes. The analysis will take a binary form for cases that experienced an epilepsy seizure and those that did not, with classes in Figures 4.2, 4.3, 4.4, and 4.5, respectively. It is important to note that only samples linked with Figure 4.1 had an epilepsy seizure. Each class of the waveform has a distinct pattern and some similarities, except for that of figure 4.1 with a clear pattern. This is due to the disruptive nature of epilepsy seizures in EEG signals. Figure 4.6 present the combination of the waveform for the various classes.

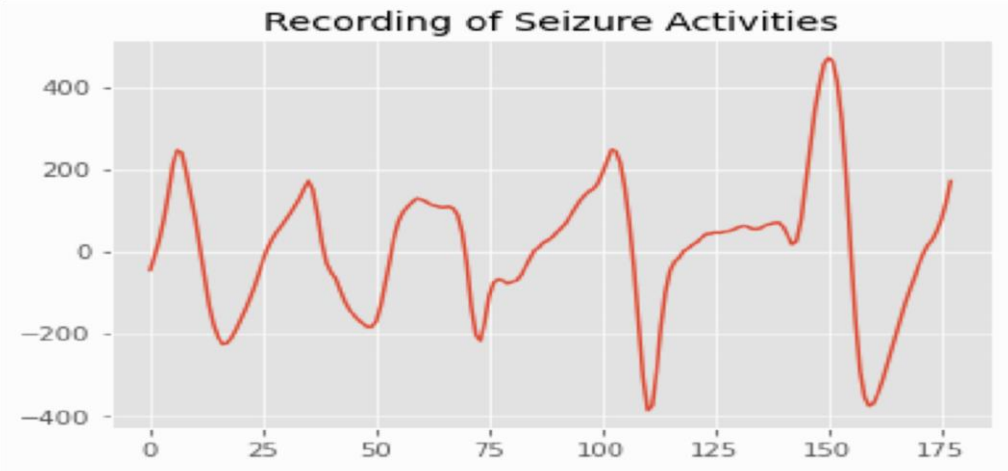

Figure 4.1: Waveform Recording for Epilepsy Seizure Activities

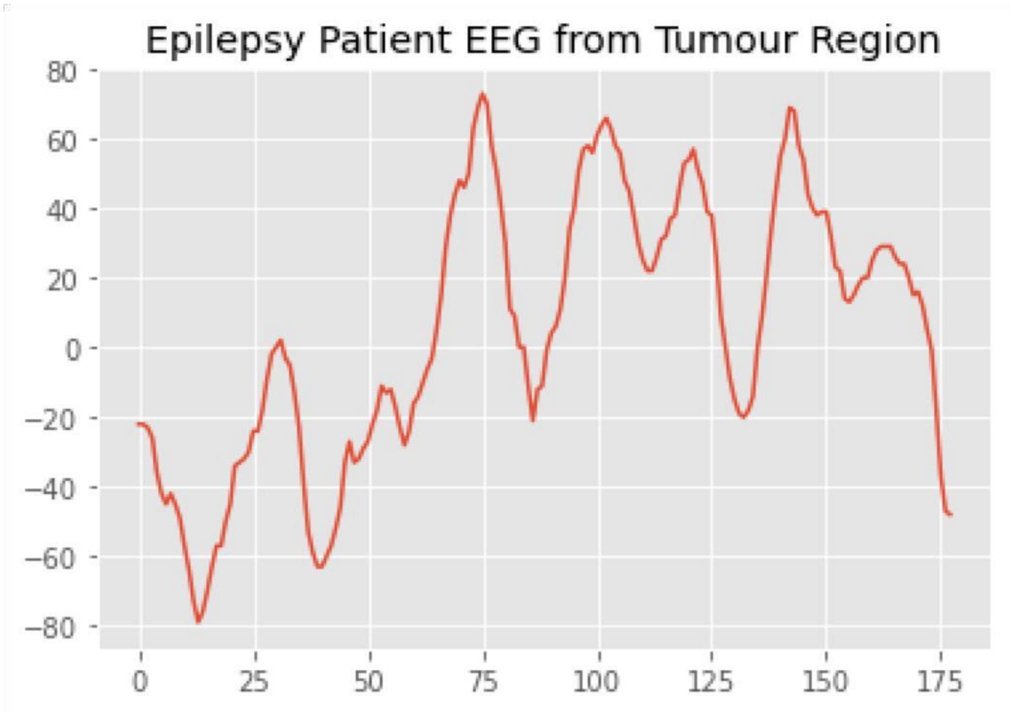

Figure 4.2: Waveform Recording of Epilepsy EEG Patient from Tumour Region

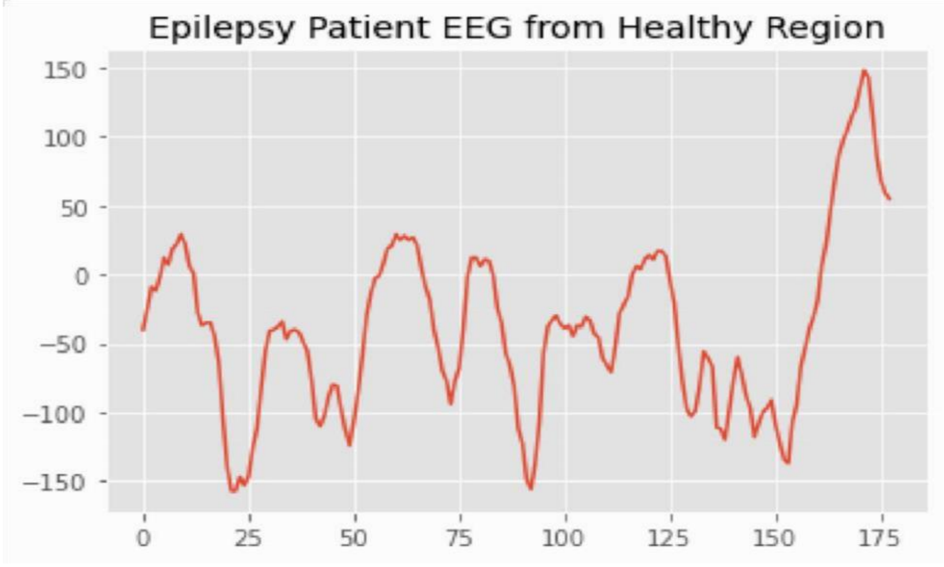

Figure 4.3: Waveform Recording of Epilepsy Patient EEG from Healthy Region

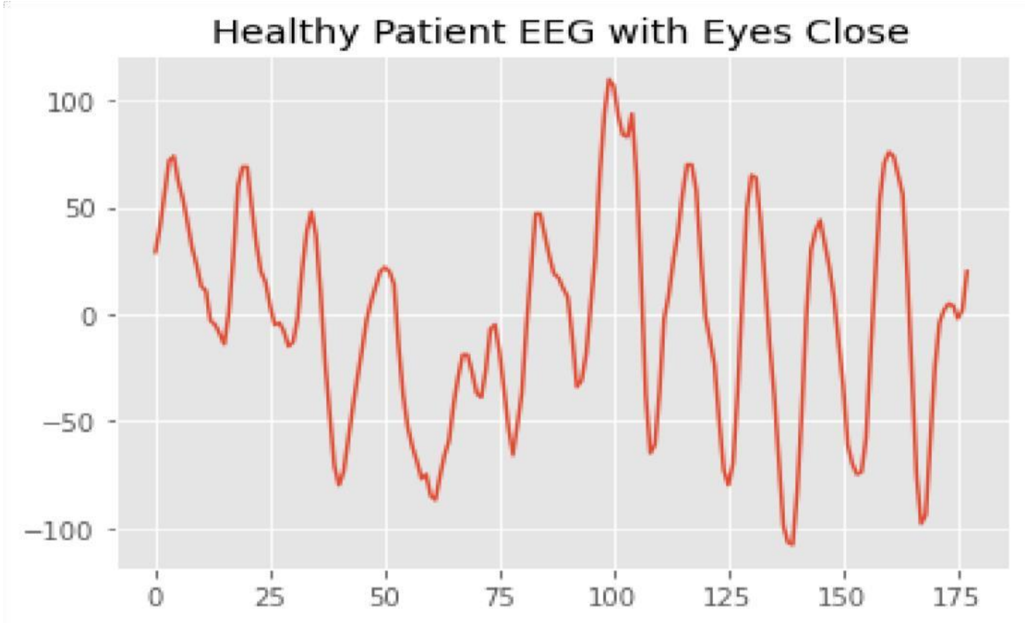

Figure 4.4: Waveform Recording of Healthy Patient EEG with Eyes Closed

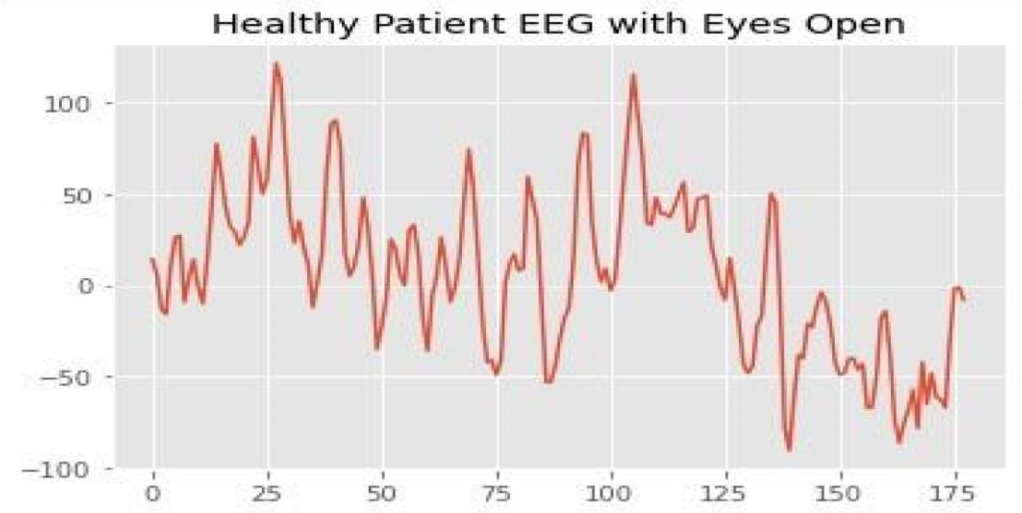

Figure 4.5: Waveform Recording of Healthy Patient EEG with Eyes Open

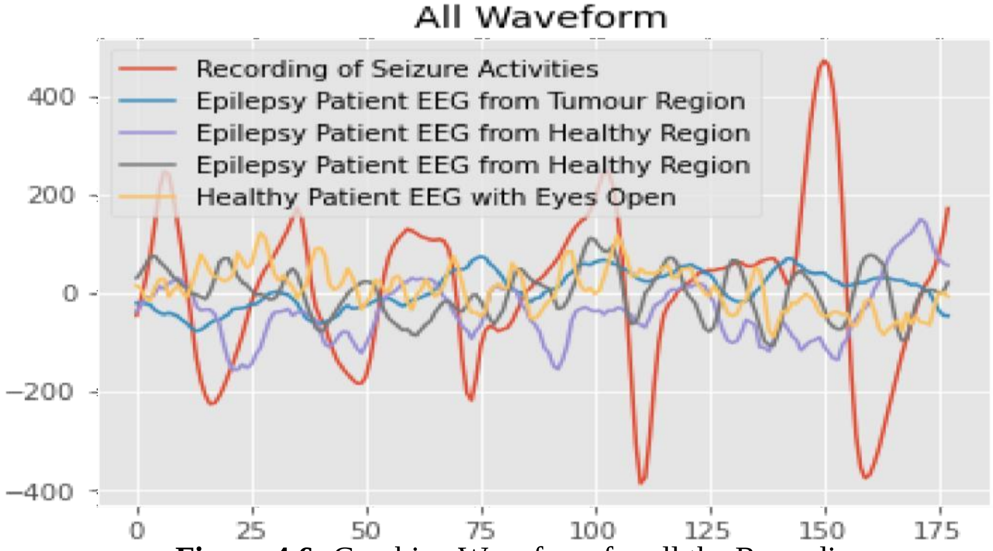

Figure 4.6: Combine Waveform for all the Recording

4.2. Results of Feature Selection and Extraction

The outcomes of the attributes chosen for usage in this research are presented in this section. Nine (9) features were compared for the diagnosis of seizures in EEG signals for IGOA, GOA, PSO, SSA, BA, and GWO with their diverse search agent, population, and generation. Each extracted feature is evaluated using accuracy, precision, F1 score, recall, and AUC; the same evaluation is utilized for feature selection and classification for all algorithms. Figures 4.7 through Figure 4.14 present the plot for each feature from the dataset, while Table 4.1 displays a summary of the features that were chosen. These answer the research objective one.

Table 4.1: Summary of the Features Selected

S/N	Features	Equations
1	Maximum Value	$M_x = Max(x(n))$
2	Mean Value	1 N □$_t$ = □x(t$_t$) N $i=1$
3	Variance	1 N 2 $_t$ = (x□(i)) □ −□ n $i=1$ —

—

n 2 —

4 Standard Deviation 1 $□ = □ \left(x(i) - □\right)$

n $i=1$

5 Energy 1 N

—

E_n $x(n)_2 = □□$

N $n=1$

N

$E = - p(x_n))$ 2 $\log_2(p(x_n))$ □(□

6 Entropy x^2

$n=1$

1 N 4 7 Kurtosis

— $x n$ □

$k = N$ $i=1$((

) $-□$ $x2$)

□

$1 \; N \quad 2$

□□ n□$_{i=1}(-$□$_x)$

□□

3

$1 \; ^{N}$□ $x(t$ — ————

$=$ □ $_t$ 8 Skewness □

□$_t =$ □$_{i\,1}$ □ $_t$□$) -$ □

□ □N

9 Root Mean Square $RMS = \sqrt{\frac{1}{n}}$ □ $^{N=1}_{i=1} x(i)^2$

□

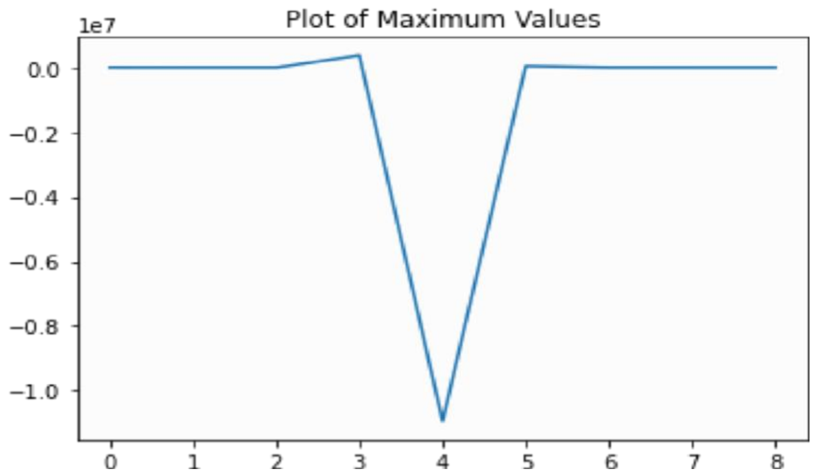

Figure 4.7: Maximum Values Plot from the Dataset

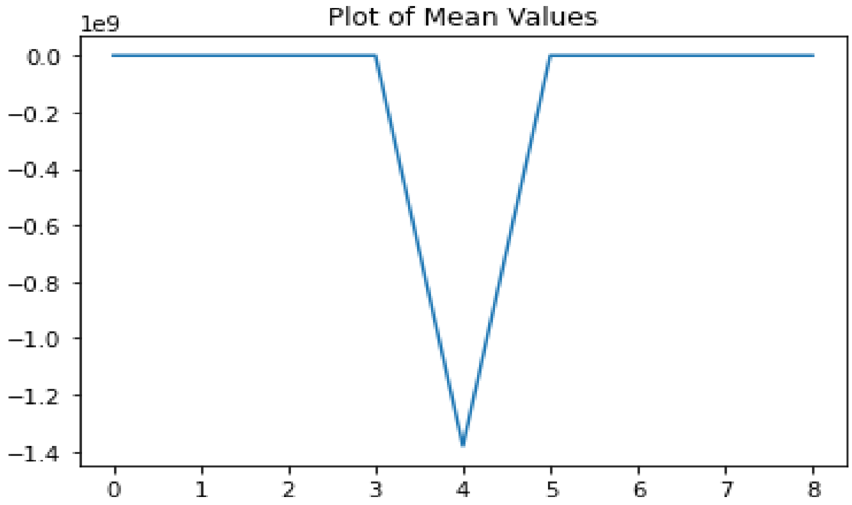

Figure 4.8: Mean Values Plot from the Dataset

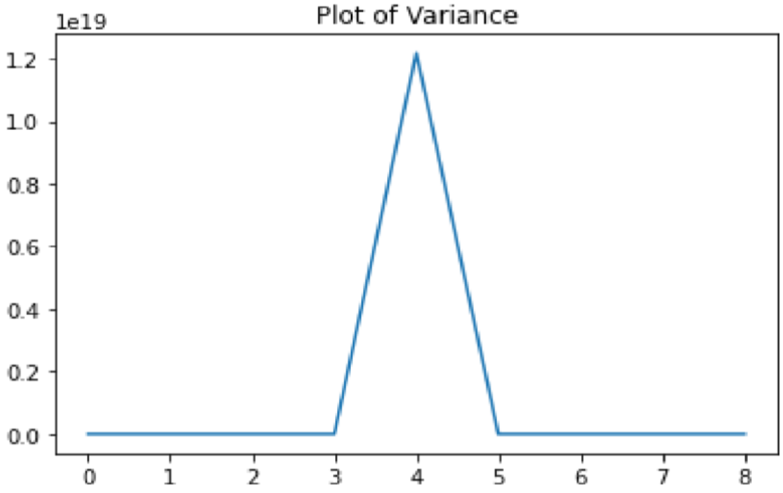

Figure 4.9: Variance Plot from the Dataset

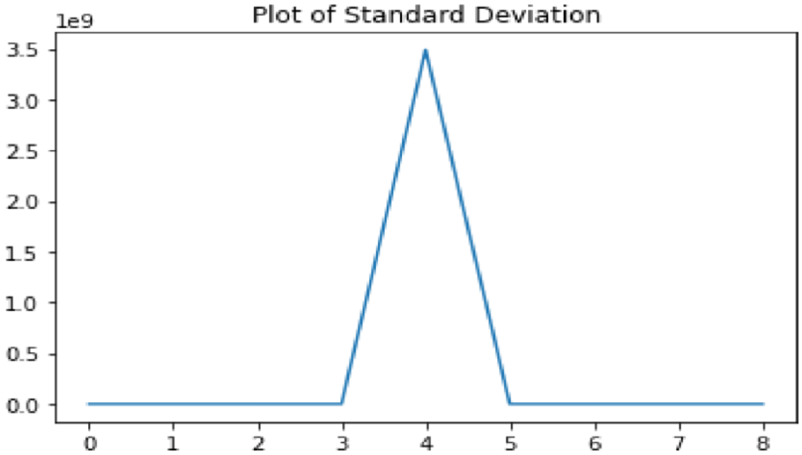

Figure 4.10: Standard Deviation Plot from the Dataset

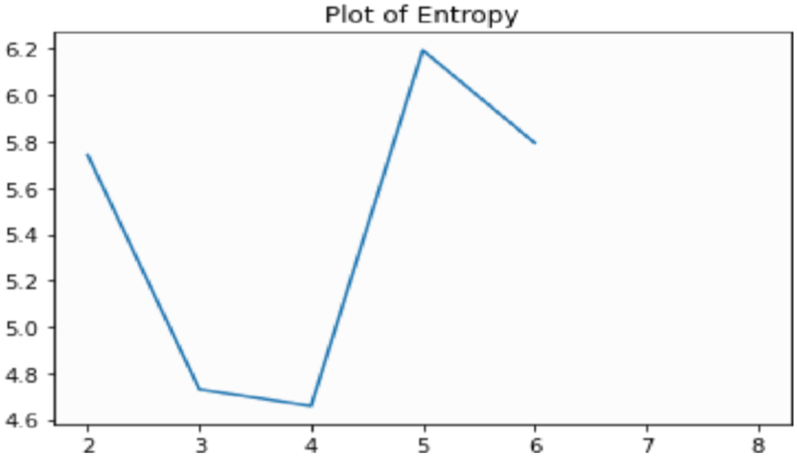

Figure 4.11: Entropy Plot from the Dataset

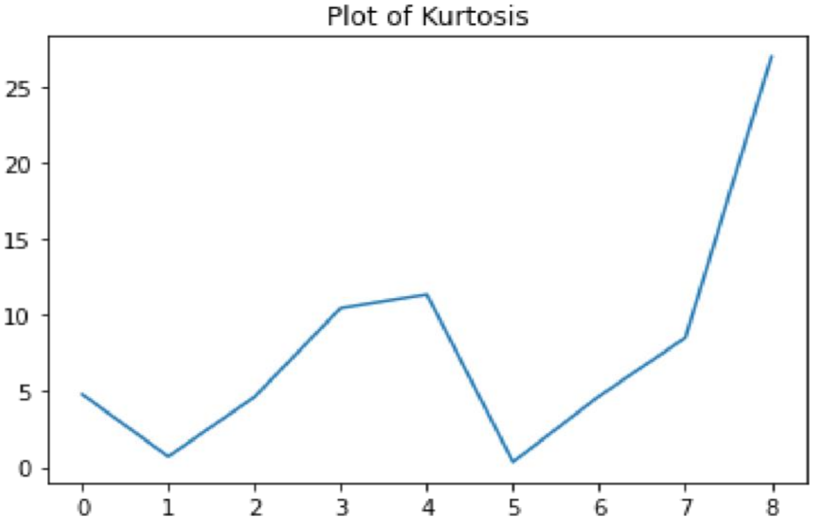

Figure 4.12: Kurtosis Plot from the Dataset

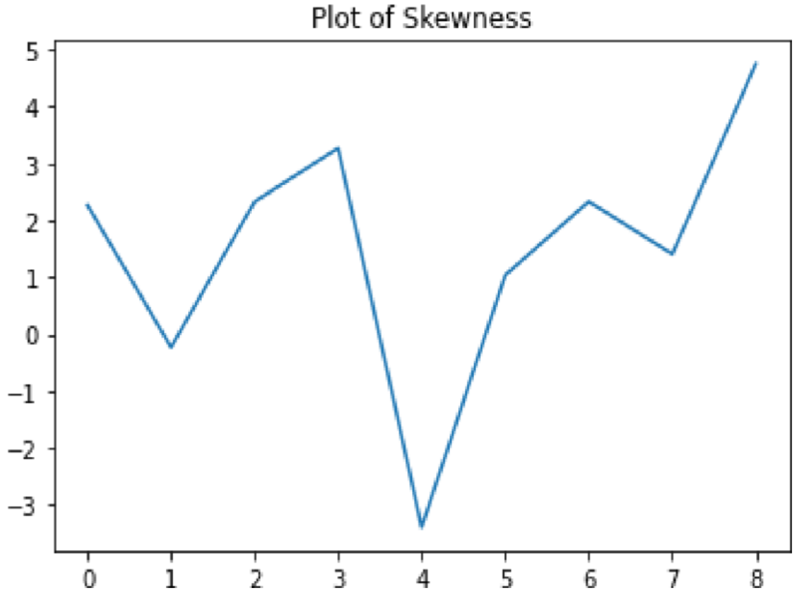

Figure 4.13: Skewness Plot from the Dataset

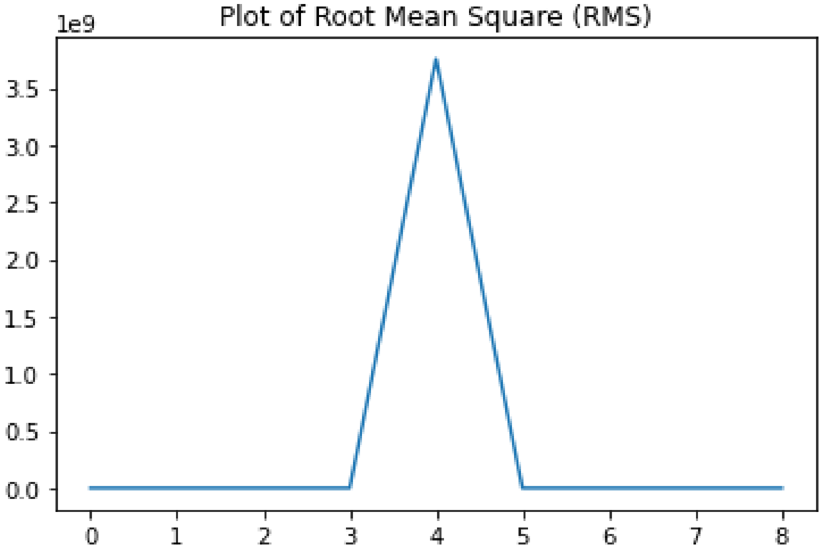

Figure 4.14: RMS Plot from the Dataset

4.3 Performance Result for the improved Grasshopper Optimization Algorithm

The experiments carried out and discussed in chapter three to meet this research study's objectives two and three are summarized in this section. The analysis of the results and how they answer the research questions are also presented. Using elite opposition-based learning and exponential switching parameters between local and random walks, the results for objectives two and three enhance the grasshopper optimization technique. Both unimodal and multimodal test function outcomes are included in the findings. The average fitness of grasshoppers and convergence curves are shown in Figures 4.15 and 4.16 to show how this behavior enhances grasshopper fitness. Each test function's curve demonstrates clear declining behavior. This shows that GOA improves the initial random population on the test functions and systematically improves the approximation accuracy with time.

Figures 4.15 and figure 4.16 are the deliverable for research objective two.

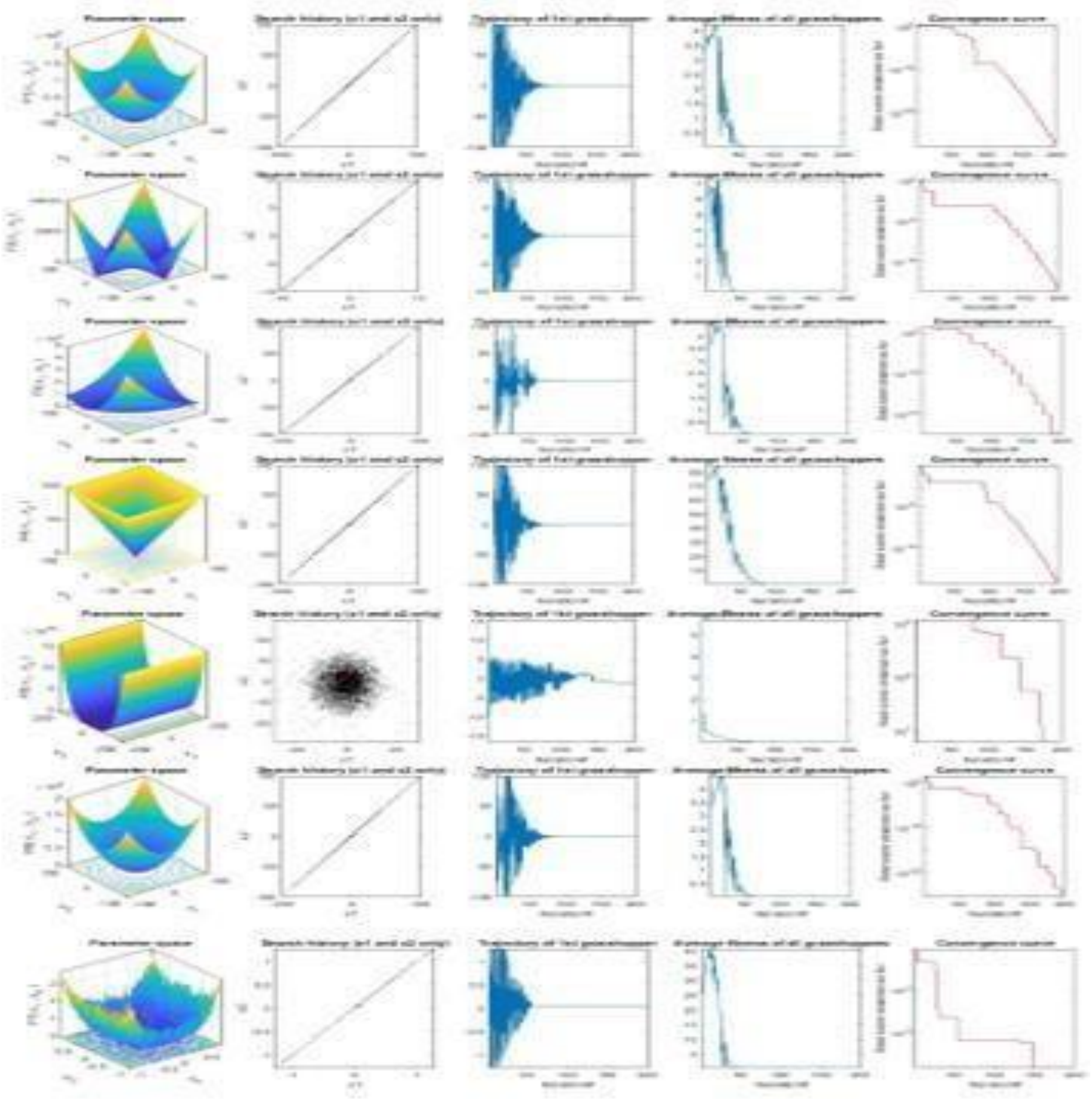

Figure 4.15: Behaviour of IGOA on the Unimodal Benchmark Function

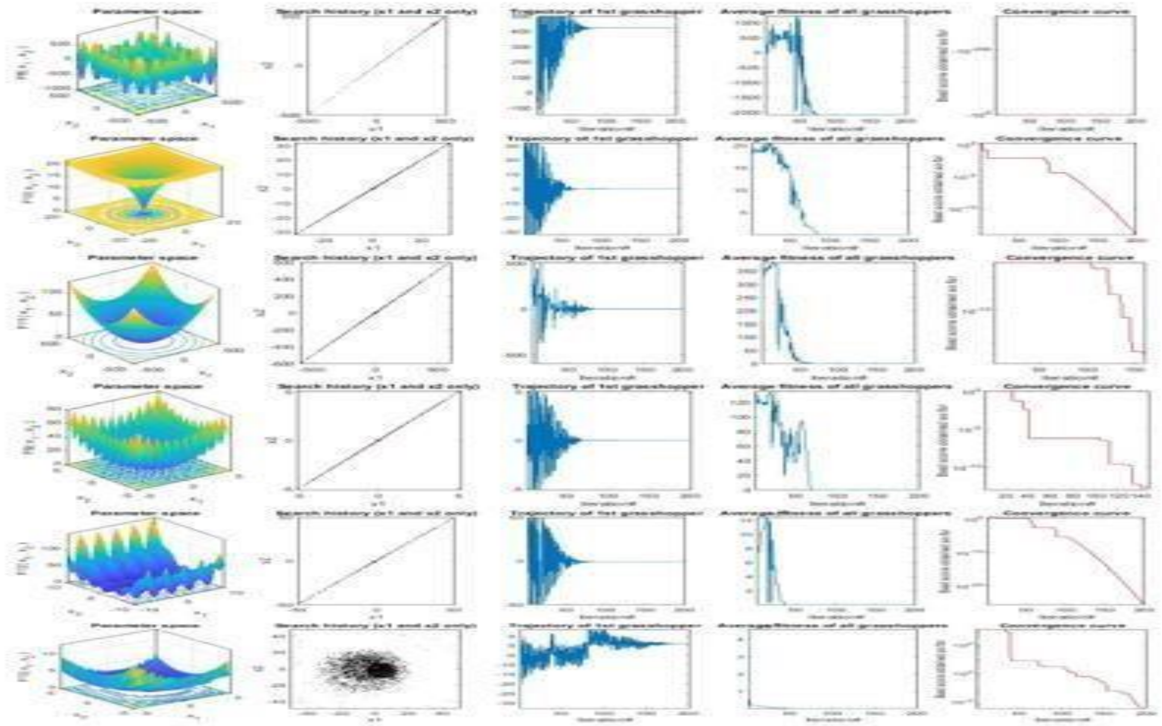

Figure 4.16: Behaviour of IGOA on the Multimodal Benchmark Function

4.3.1 IGOA and GOA performance on unimodal test functions

The result in Table 4.2 shows the optimal fitness values, worst, mean, and standard deviation fitness values over 10 runs obtained using IGOA compared to results obtained using GOA for unimodal test functions (F1 to F7). From Table 4.2, it is observed that IGOA obtained the global optima for all unimodal test functions. This indicates that IGOA has high exploitation capability due to the introduction of elite opposition-based learning and exponential switching parameters between local and random walks. Similarly, the convergence curves in Figure 4.17, show that IGOA converges faster for all cases compared to the GOA algorithm. This is due to the introduction of elite opposition-based learning and exponential switching parameters between local and random walks. Table 4.2 and Figure 4.17 are the deliverables for research objective three for the unimodal function.

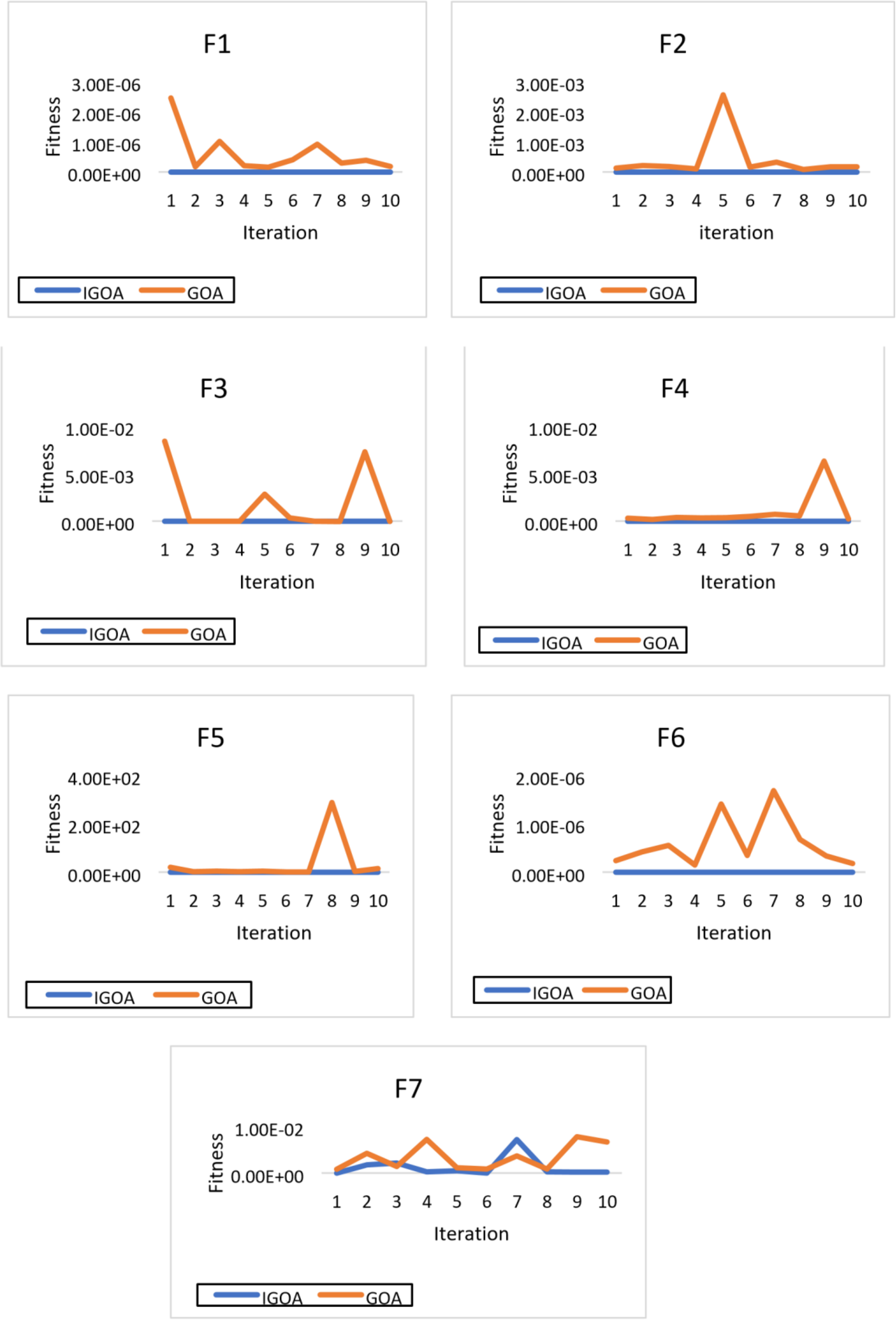

Figure 4.17: Convergence curves for Unimodal Test Functions

Table 4.2: Algorithm Performance on Unimodal Test Functions

Function	Performance	IGOA	GOA
F1	Best	8.01E-29	8.9364E-07
	Worst	4.00E-15	1.4542E-04
	Average	4.00E-15	-2.1607E-04
	STD	0	4.0627E-04
F2	Best	2.13E-14	0.00011426
	Worst	4.26E-15	3.7353E-05
	Average	4.26E-15	-7.4955E-06
	STD	0	2.7585E-05
F3	Best	1.27E-25	4.5428E-06
	Worst	4.80E-14	0.001800
	Average	4.80E-14	-4.7674E-05
	STD	7.06E-30	0.001400
F4	Best	5.68E-15	0.00037436
	Worst	-5.68E-15	3.7436E-04
	Average	-5.68E-15	-7.8352E-05
	STD	0	3.2830E-04
F5	Best	1.44E-27	0.612680
	Worst	1	0.882400
	Average	1	0.555300
	STD	1.24E-16	0.304000
F6	Best	4.03E-26	6.6087E-07
	Worst	-0.5	-0.499700
	Average	-0.5	-0.5000
	STD	6.21E-17	4.0314E-04
F7	Best	0.00014458	0.00120160
	Worst	0.0469	-0.021900
	Average	0.0469	-0.082600
	STD	0	0.043900

4.3.2 IGOA and GOA performance on multimodal test functions

Table 4.3 presents the result obtained by IGOA compared to results obtained by GOA for multimodal test functions. From Table 4.3, IGOA was able to obtain the optimal value for all the multimodal test functions. The results are an indication that the developed IGOA possesses better exploration ability and avoids getting stuck in local optima. The fitness curves in Figure 4.18 confirms the accurate convergence nature of IGOA for all the multimodal function as compared with GOA. This is attributed to the robust exploration strategy of IGOA due to the introduction of elite opposition-based learning and exponential switching parameters between local and random walks. Table 4.3 and Figure 4.18 are the deliverables for research objective three for the multimodal function.

Table 4.3: Algorithm Performance on Multimodal Test Functions

Function	Performance	IGOA	GOA
F8	Best	-2094.914	-1403.9075
	Worst	420.9687	421.0100
	Average	420.9687	222.4525
	STD	0	274.0962
F9	Best	0	6.964700
	Worst	7.85E-10	0.99500
	Average	7.85E-10	-0.59700
	STD	0	1.134400
F10	Best	7.99E-15	0.00034756
	Worst	1.71E-15	1.7756E-04
	Average	1.71E-15	6.29E-05
	STD	2.20E-31	6.69E-05
F11	Best	0	0.089061
	Worst	7.24E-09	16.3022
	Average	7.24E-09	6.2132
	STD	0	6.164
F12	Best	2.47E-26	7.44E-06
	Worst	-1	-0.9893
	Average	-1	-0.995700
	STD	0	0.00500
F13	Best	1.42E-26	0.0032146
	Worst	1	1.0933
	Average	1	1.0053
	STD	0	0.0778

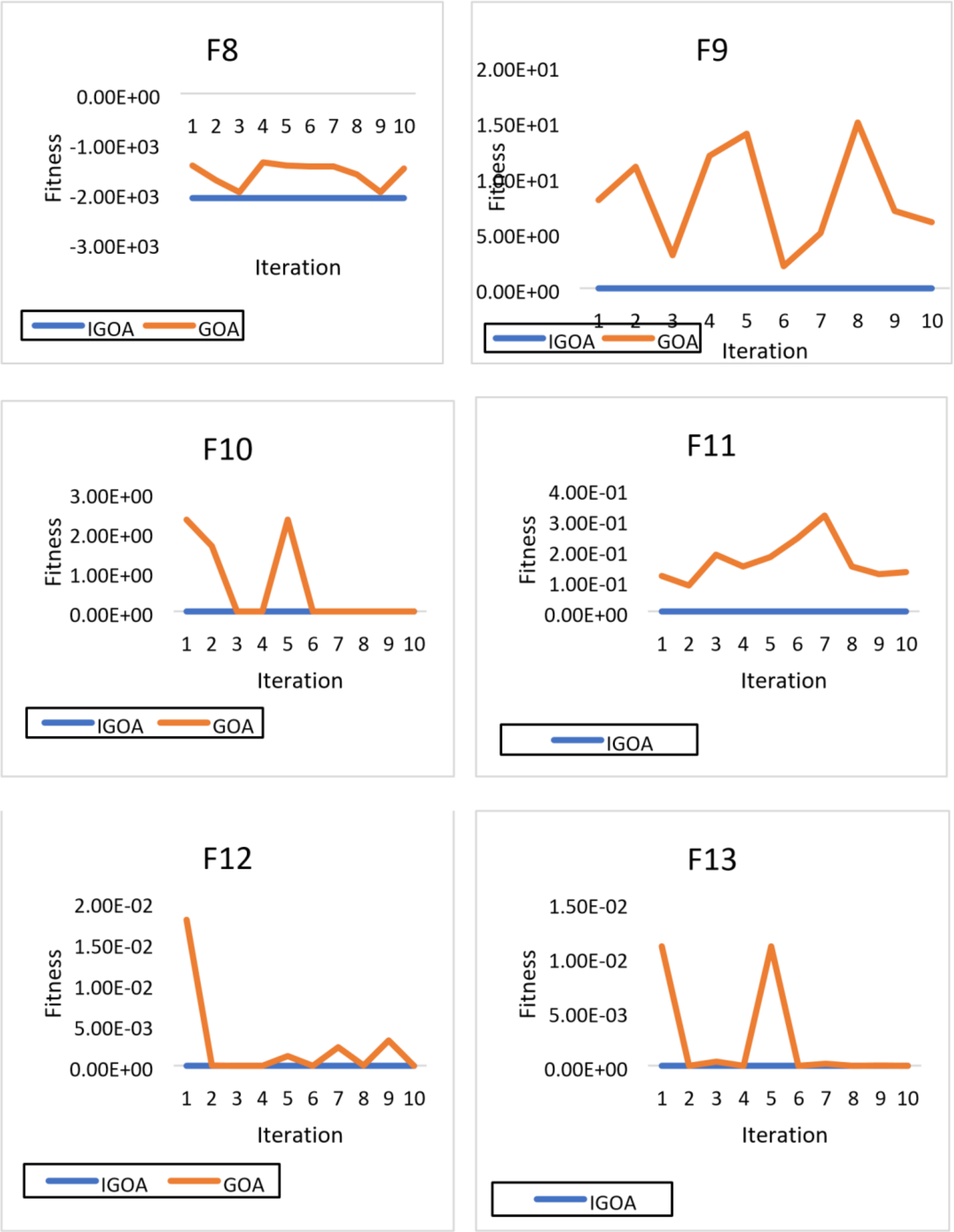

Figure 4.18: Convergence curves for Multimodal Test Functions

4.4 Result of Epilepsy Classification Using Improved Grasshopper Optimization Algorithm

This section displays the findings from categorizing epilepsy using IGOA-ANN and various feature extraction ranges from 1, 3,5, 7, and 9 with their respective search agents.

4.4.1 Epilepsy classification using one feature, and 5, 10, 15, 20, 25, 30, and 35 Search agents respectively

The effectiveness of a classifier can be assessed using several metrics, such as accuracy, precision, recall, F1 score, AUC, confusion matrices, and receiver operating characteristic

(ROC) curves. The evaluation techniques for IGOA-ANN utilizing one feature and 5, 10, 15, 20, 25, 30, and 35 search agents, respectively, are shown in Tables 4.4 and 4.5 in this subsection. For the search agents 5, 10, 15, 20, 25, 30, and 35, the classifier correctly predicted the true negative classes but wrongly predicted 2, 8, 5, 1, and 14 of the falsenegative classes. The F1 score, recall, accuracy, precision, and AUC are shown in Table 4.5. It demonstrates how the IGOA-ANN classifier's accuracy rises and falls as the number of search agents increases. The accuracy remains the same from 95.6% to 95.0% as search agent varies from 5 to 35, respectively except for 30 search agents where it decreases to 94.4%. The best accuracy is achieved at 95.6%, with 85.76% precision, 99.60 sensitivity, and an F1 score of 92.16% and 95.20% AUC respectively.

Table 4.4: IGOA-ANN Metrics Using One Feature Extraction

Number of Features	Number of Population	TP	FP	TN	FN	Time
[3]	5	398	20	80	2	52.250
[3]	10	398	14	86	8	107.911
[6]	15	395	19	81	5	198.124
[3]	20	395	13	87	5	59.412
[7]	25	399	24	76	1	273.683
[1]	30	386	14	86	14	397.021
[7]	35	396	15	85	4	55.747

Table 4.5: IGOA-ANN Performance Evaluation Using One Feature Extraction

Number of Features	Number of Population	Accuracy	Precision	Sensitivity	F1-Score	AUC

[3]	5	95.6	0.8576	0.9960	0.9216	0.9520
[3]	10	95.6	0.7788	0.9980	0.8749	0.9160
[6]	15	95.2	0.7610	0.9920	0.8613	0.9020
[3]	20	96.4	0.8645	0.9920	0.9239	0.9520
[7]	25	95.0	0.8353	0.9980	0.9095	0.9440
[1]	30	94.4	0.7522	0.9860	0.8534	0.8930
[7]	35	96.2	0.8545	0.9880	0.9117	0.9420

4.4.2 Epilepsy classification using three features, and 5, 10, 15, and 30 search agents respectively

Presented in this subsection in Tables 4.6 and 4.7 are the evaluation methods for IGOAANN using three features and 5, 10, 15, 20, 25, 30, and 35 search agents respectively. For the search agents 5, 10, 15, 20, 25, 30, and 35, the classifier correctly predicted the true negative classes but mistakenly predicted 0, 0, 2, and 0 of the false-negative classes. Table 4.7 presents the F1 score, recall, accuracy, precision, and AUC. It shows that the accuracy increases and decreases as the search agent increases for the IGOA-ANN classifier. The accuracy increases from 97.6% to 98.2% as search agents increase from 5 to 35, respectively. The best accuracy is achieved at 98.20%, with 96.68% precision, 98.20 sensitivity, F1 score of 97.43%, and 97.80% AUC respectively for 25 search agents for the IGOA algorithm.

Table 4.6: IGOA-ANN Metrics Using Three-Feature Extraction

Number of Features	Number of Population	TP	FP	TN	FN	Time
[9 1 6]	5	400	12	88	0	52.250
[4 1 2]	10	400	15	85	0	107.911
[9 7 3]	15	398	7	93	2	198.124
[4 7 6]	20	400	6	94	0	78.903
[8 7 1]	25	400	11	89	0	273.683
[1 8 7]	30	400	12	88	0	397.021
[9 7 6]	35	399	9	91	1	88.329

Table 4.7: IGOA-ANN Performance Evaluation Using Three-Feature Extraction

Number of Features	Number of Population	Accuracy	Precision	Sensitivity	F1-Score	AUC
[9 1 6]	5	97.6	0.9684	0.9760	0.9722	0.9740
[4 1 2]	10	97.0	0.8481	0.9840	0.9110	0.9400
[9 7 3]	15	98.2	0.8756	0.9960	0.9319	0.9590
[4 7 6]	20	98.8	0.9689	0.9880	0.9784	0.9830

[8 7 1]	25	97.8	0.9668	0.9820	0.9743	0.9780
[1 8 7]	30	97.6	0.9684	0.9760	0.9722	0.9740
[9 7 6]	35	98.0	0.8542	0.9980	0.9205	0.9520

4.4.3 Epilepsy classification using five features, and 5, 10, 15, 20, 25, 30, and 35 search agents respectively

The evaluation techniques for IGOA-ANN utilizing five characteristics and, correspondingly, 5, 10, 15, 20, 25, 30, and 35 search agents are shown in Tables 4.8 and 4.9. The classifier successfully predicted the true negative classes, but mistakenly anticipated the false-negative classes 0, 0, 1, and 0, as well as 0 false negatives for the search agents 5, 10, 15, 20, 25, and 35, respectively. The F1 score, recall, accuracy, precision, and AUC are shown in Table 4.9. It demonstrates that for the IGOA-ANN classifier, accuracy rises as the search agent does. As the number of search agents rises from 5 to 35, the accuracy increases from 98.2% to 99.7%. The best accuracy for the GOA algorithm is 99.6%, with 99.6% precision, 99.6% sensitivity, and 99.6% AUC for each of the 30 search agents. The performance of IGOA-ANN utilizing five features is therefore validated by the higher values of the ROC at 30 search agents.

Table 4.8: IGOA-ANN Metrics Using Five-Feature Extraction

Number of Features	Number of Population	TP	FP	TN	FN	Time
[6 1 3 9 7]	5	400	9	91	0	52.250
[6 6 2 7 9]	10	400	11	89	0	107.911
[6 4 5 3 7]	15	400	6	94	0	198.124
[3 1 7 9 8]	20	400	17	83	0	90.833
[9 7 6 2 1]	25	399	18	82	1	273.683
[7 5 9 3 1]	30	400	4	96	0	397.021
[8 5 9 1 2]	35	400	17	83	0	81.107

Table 4.9: GOA-ANN Performance Evaluation Using Five-Feature Extraction

Number Number of Accuracy Precision Sensitivity F1-Score AUC of Population Features

[6 1 3 9 7]	5	98.2	0.9763	0.9840	0.9801	0.9820
[6 6 2 7 9]	10	97.8	0.9033	0.9820	0.9410	0.9590
[6 4 5 3 7]	15	98.8	0.9822	0.9900	0.9861	0.9880
[3 1 7 9 8]	20	96.6	0.9605	0.9680	0.9642	0.9660
[9 7 6 2 1]	28	96.2	0.8606	0.9840	0.9182	0.9450
[7 5 9 3 1]	30	99.6	0.9960	0.9960	0.9960	0.9960

[8 5 9 1 2]	35	96.6	0.9587	0.9700	0.9643	0.9670

4.4.4 Epilepsy classification using seven features, and 5, 10, 15, 20, 25, 30, and 35 search agents respectively

The techniques for evaluating IGOA-ANN utilizing seven characteristics and 5, 10, 15, 20, 25, 30, and 35 search agents, respectively, are presented in Tables 4.10 and 4.11. For search agents 5, 10, 15, 20, 25, 30, and 35, the classifier accurately identified the true negative classes and correctly predicted 0,0,0,0 and 0 of the false-negative classes. The F1 score, recall, accuracy, precision, and AUC are shown in Table 4.11. It demonstrates that for the IGOA-ANN classifier, accuracy rises as the search agent does. As the number of search agents rises from 5 to 30, the accuracy increases from 97.2% to 99.4%. The best accuracy for the GOA algorithm is 99.4%, with 99.4% precision, 99.4% sensitivity, and 99.4% AUC for each of the 30 search agents. The performance of GOA-ANN using seven features is thus validated by the higher values of the ROC at 30 search agents.

Table 4.10: IGOA-ANN Metrics Using Seven Feature Extraction

Number of Features	Number of Population	TP	FP	TN	FN	Time
[4 1 6 7 9 3 2]	5	400	14	86	0	52.250
[9 3 1 6 5 7 8]	10	400	15	85	0	107.911
[1 6 7 2 9 4 8]	15	400	11	89	0	198.124
[8 2 1 4 3 9 6]	20	400	18	82	0	94.067
[3 7 8 5 6 1 4]	25	400	22	78	0	273.683
[4 7 8 3 6 1 5]	30	400	5	95	0	397.021
[6 8 2 1 4 3 9]	35	400	13	87	0	112.729

Table 4.11: IGOA-ANN Performance Evaluation Using Seven Feature Extraction

Number of Features	Number of Population	Accuracy	Precision	Sensitivity	F1-Score	AUC
[4 1 6 7 9 3 2]	5	97.2	0.9076	0.9800	0.9424	0.9590
[9 3 1 6 5 7 8]	10	97.0	0.9504	0.9800	0.9650	0.9720
[1 6 7 2 9 4 8]	15	97.8	0.9743	0.9820	0.9781	0.9800
[8 2 1 4 3 9 6]	20	96.4	0.9702	0.9741	0.9721	0.9730
[3 7 8 5 6 1 4]	25	95.6	0.9456	0.9640	0.9547	0.9590
[4 7 8 3 6 1 5]	30	99.4	0.9940	0.9940	0.9940	0.9940
[6 8 2 1 4 3 9]	35	97.4	0.9702	0.9740	0.9721	0.9730

4.4.5 Epilepsy classification using nine features, and 5, 10, 15, 20, 25, 30, and 35 search agents respectively

Tables 4.12 and 4.13 present the evaluation methods for IGOA-ANN using nine features and 5, 10, 15, 20, 25, 30, and 35 search agents respectively. The classifier accurately detected the genuine negative classes and forecasted 0, 0,0,0, and 0 of the false-negative classes correctly for 5, 10, 15, 20, 25, 30, and 35 search agents respectively. Table 4.13 shows the F1 score, recall, accuracy, precision, and AUC. It shows that the accuracy increases as the search agent increases for the IGOA-ANN classifier. The accuracy changes from 95.60% to 96.0% as search agents increase from 5 to 35, respectively. The best accuracy is achieved at 96.60%, with 94.00% precision, 98.00% sensitivity, and an F1 score of 95.96% and 96.90% AUC respectively for 10 search agents for the IGOA.

Table 4.12 IGOA-ANN Metrics Using Nine Features and 5,10, 15, 25,20, 30,35 Generation

Number of Features	Number of Population	TP	FP	TN	FN	Time
[1 6 3 8 2 9 4 7 5]	5	400	22	78	0	52.250
[7 3 6 1 9 5 2 8 4]	10	400	17	83	0	107.911
[6 5 2 9 8 1 3 7 4]	15	400	24	76	0	198.124
[2 7 4 8 3 1 9 6 5]	20	400	23	77	0	93.406
[1 6 4 8 9 7 5 2 3]	25	400	21	79	0	273.683
[7 9 4 5 8 2 1 6 3]	30	400	20	80	0	397.021
[3 2 1 9 5 6 8 4 7]	35	400	18	82	0	88.938

Table 4.13 IGOA-ANN Performance Evaluation Using Nine Features and 5,10, 15,20, 25, 30, 35 Generation

Number of Features	Number of Population	Accuracy	Precision	Sensitivity	F1-Score	AUC
[1 6 3 8 2 9 4 7 5]	5	95.6	0.9421	0.9640	0.9529	0.9580
[7 3 6 1 9 5 2 8 4]	10	96.6	0.9400	0.9800	0.9596	0.9690
[6 5 2 9 8 1 3 7 4]	15	95.2	0.9279	0.9600	0.9437	0.9510
[2 7 4 8 3 1 9 6 5]	20	95.4	0.9351	0.9640	0.9493	0.9560
[1 6 4 8 9 7 5 2 3]	25	95.8	0.9434	0.9580	0.9506	0.9540
[7 9 4 5 8 2 1 6 3]	30	96.0	0.9440	0.9660	0.9549	0.9600
[3 2 1 9 5 6 8 4 7]	35	96.4	0.9528	0.9640	0.9584	0.9610

Table 4.14 present the evaluation methods for IGOA-ANN using one, three, five, seven, and nine features respectively for 5, 10, 15, 20, 25, 30, and 35 search agents. The F1 score, recall, accuracy, precision, and AUC for various feature classes are also shown. It demonstrates that as the number of features rises from one to seven, accuracy increases, except for nine features with a decrease compared to the seven features of the IGOA-ANN classifier. The best accuracy is achieved at 99.60%, with 99.60% precision, 99.60% sensitivity, and an F1 score of 99.60% and 99.60% AUC respectively for 30 search agents, and five feature extractions for the IGOA-ANN. It is observed from Figure 4.19 that the accuracy is near 100%, indicating a strong performance of the IGOA-ANN classifier using five feature

extractions. In the real classes, the classifier performs well, and vice versa for five features when compared to other classes of the feature's extraction.

Table 4.14: Summary of Epilepsy Detection Using IGOA with Various Search Agents

Evaluation Metrics	1 Feature 5 Search Agent	3 Features 15 Search Agent	5 Features 30 Search Agent	7 Features 30 Search Agent	9 Features 10 Search Agent
Accuracy	0.956	0.982	**0.9960**	**0.9940**	0.9600
Precision	0.8576	0.8756	**0.9960**	**0.9940**	0.9400
Sensitivity	0.9960	0.9960	**0.9960**	**0.9940**	0.9800
F1 Measure	0.9216	0.9319	**0.9960**	**0.9940**	0.9596
AUC	0.9520	0.9590	**0.9960**	**0.9940**	0.9690

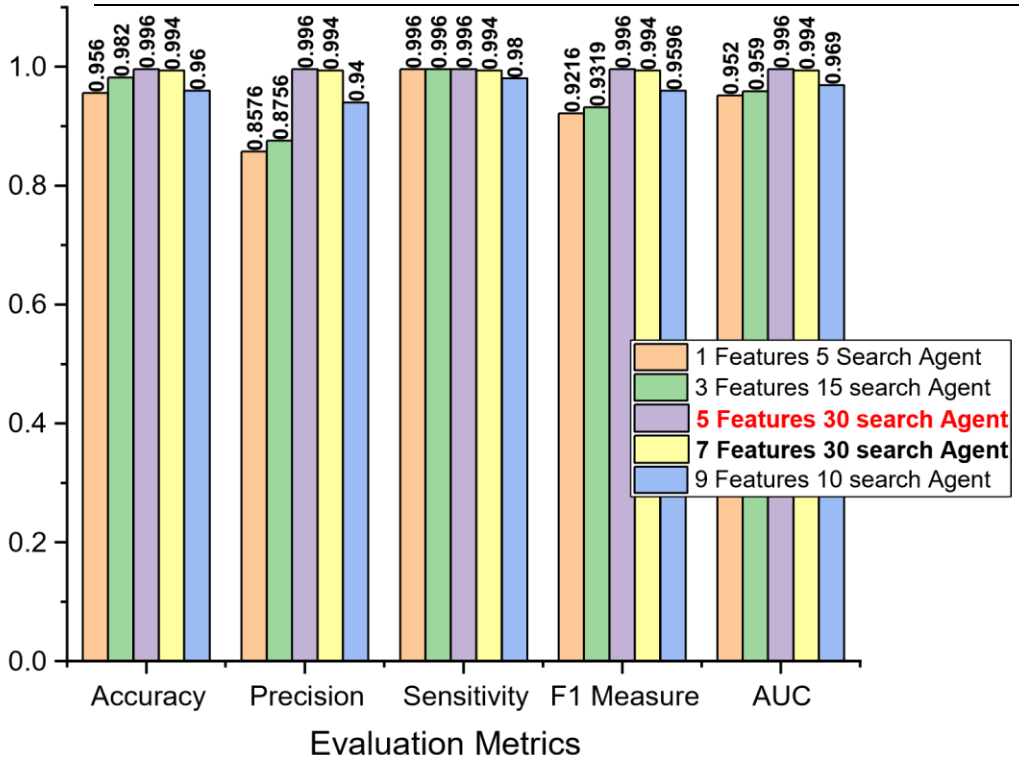

Figure 4.19: IGOA Algorithm Performance Evaluation with Various Search Agents

4.5 Epilepsy Classification Using Grasshopper Optimization Algorithm and Artificial Neural Network (GOA-ANN)

This section shows the outcomes of detecting and categorizing epilepsy using the Grasshopper optimization method and an artificial neural network, employing multiple feature extraction ranges from 1, 3,5, 7, and 9 with their respective search agents.

4.5.1 Epilepsy classification using one feature, and 5, 10, 15, 20, 30, and 30 search agents respectively

The effectiveness of a classifier can be assessed using several metrics, such as accuracy, precision, recall, F1 score, AUC, confusion matrices, and receiver operating characteristic (ROC) curves. The assessment techniques for GOA-ANN utilizing one feature and 5, 10, 15, 20, 25, 30, and 35 search agents, respectively, are shown in Tables 4.15 and 4.16 in this subsection. For search agents 5, 10, 15, 20, 25, 30, and 35, the classifier correctly predicted the true negative classes but incorrectly anticipated 11, 11, 21, and 14 of the false-negative classes. The F1 score, recall, accuracy, precision, and AUC are shown in Table 4.16. It demonstrates how the GOA-ANN classifier's accuracy rises and falls as the number of search agents increases. As the search agent increases from 5 to 35, the accuracy decreases from 96.00% to 94.40%. With an F1 score of 93.86%, a precision of 98.80%, a sensitivity of 89.38, and an AUC of 96.00%, the best accuracy is attained. By contrasting the values of SEN and SPE at each practical cut-off point, a receiver operating characteristic (ROC), or plot of sensitivity against 1-specificity, is produced. The area under the ROC curve (AROC), which is used to assess the diagnostic accuracy of any test, serves to quantify the overall efficacy of a clinical test. The area under the ROC curve (AUC) for the GOA-ANN classifier in Figure 4.20 is close to 1, indicating that it works well. For the depiction of the acquired results for 5, 10, 15, 20, 25, 30, and 35 search agents, respectively, the ROCs of various examples obtained using the GOA-ANN classifier are plotted as shown in Figures

4.20 to 4.26. In the real classes, the classifier does well, and vice versa.

Table 4.15: GOA-ANN Metrics Using One Feature Extraction

No of Feature	Number of GOA Search Agents	TP	FP	TN	FN	Time (s)
[3]	5	398	11	18	2	41.138
[2]	10	399	11	11	1	34.627
[1]	15	396	21	79	4	40.285
[5]	20	393	16	84	7	68.780
[4]	25	395	11	84	5	42.894
[1]	30	386	14	86	14	40.013
[3]	35	394	16	84	6	53.161

Table 4.16: GOA-ANN Performance Evaluation Using One Feature Extraction

No of	Number of	Accuracy	Precision	Sensitivity	F1	AUC

Feature	GOA Search Agents					
[3]	**5**	**0.9600**	**0.9880**	**0.8938**	**0.9386**	**0.9600**
[2]	10	0.8200	0.9340	0.5683	0.7061	0.6740
[1]	15	0.9500	0.9940	0.8323	0.9060	0.9400
[5]	20	0.9540	0.7714	1.000	0.8709	0.9130
[4]	25	0.9580	0.9980	0.7936	0.8842	0.9240
[1]	30	0.9440	0.9860	0.7539	0.8544	0.8940
[3]	35	0.9560	0.8115	0.9980	0.8951	0.9330

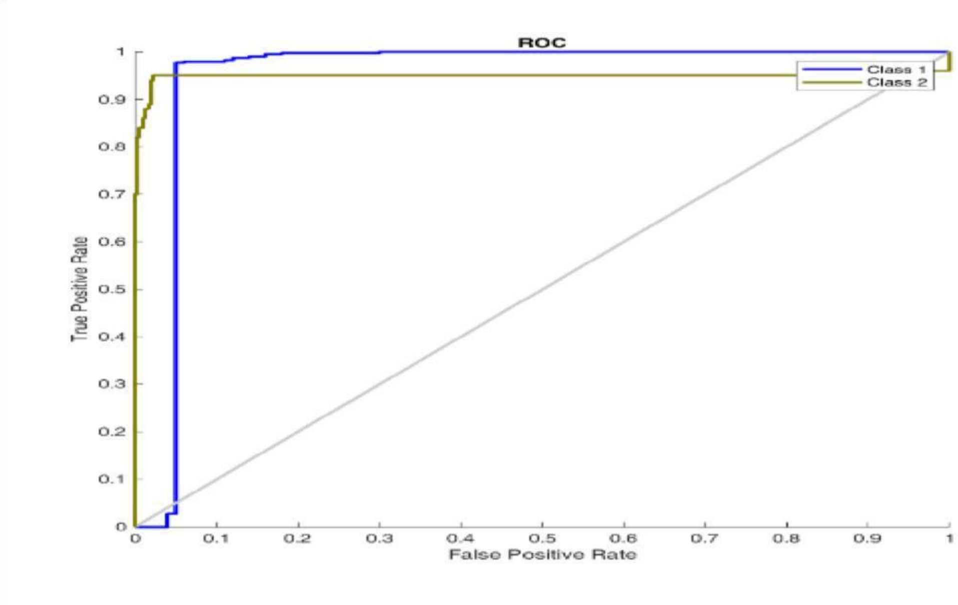

Figure 4.20: ROC Plot for GOA 1 Feature 5 Search Agent

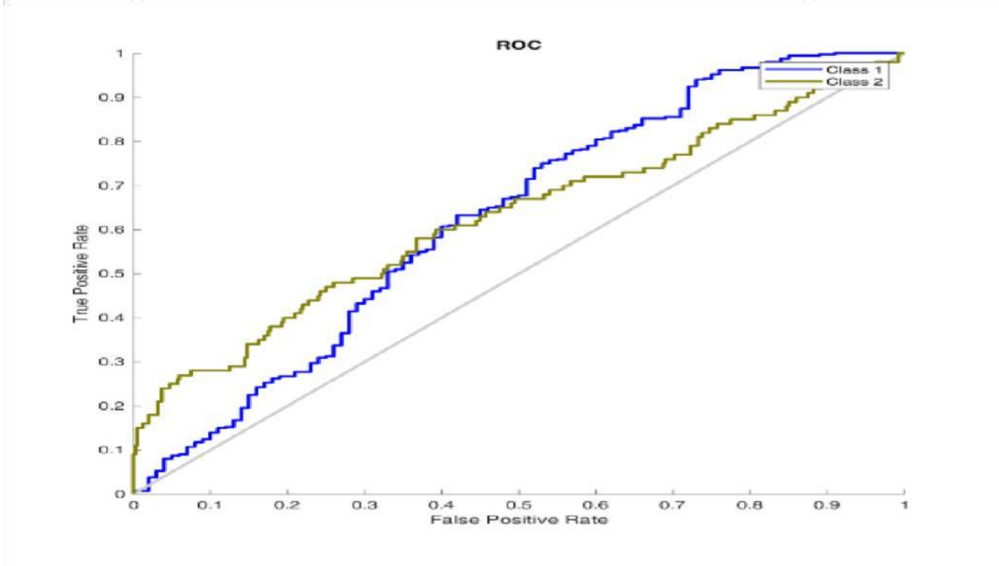

Figure 4.21: ROC Plot for GOA 1 Feature 10 Search Agent

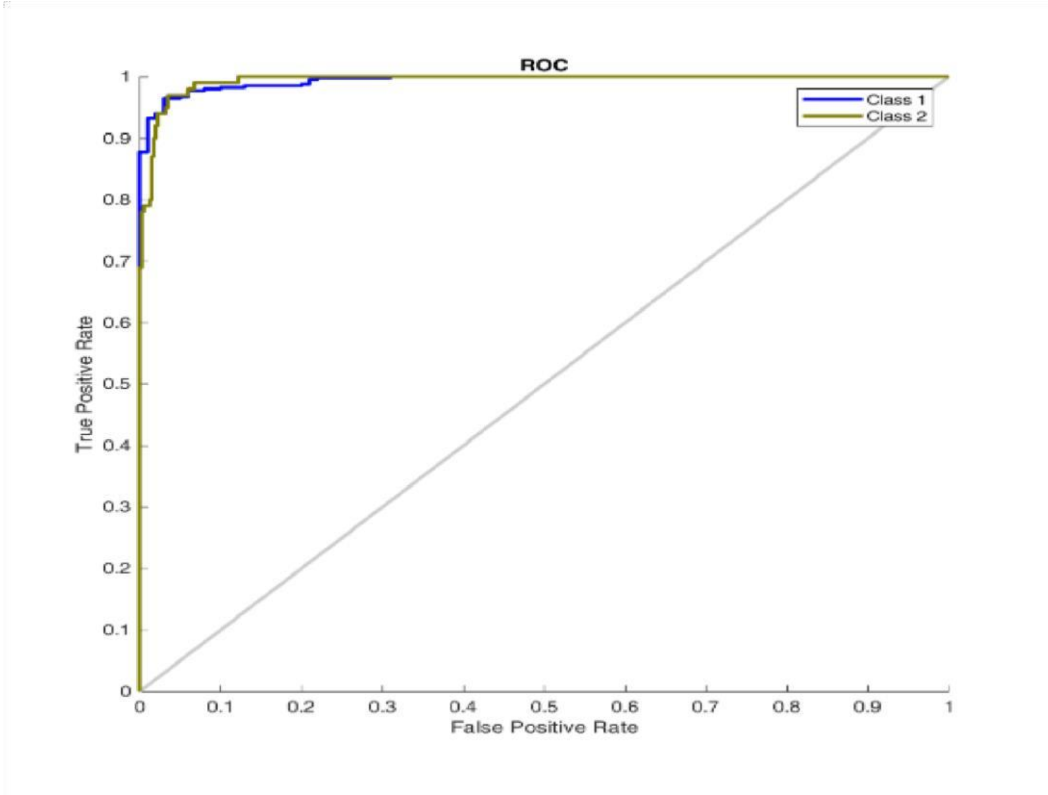

Figure 4.22: ROC Plot for GOA 1Feature 15 Search Agent

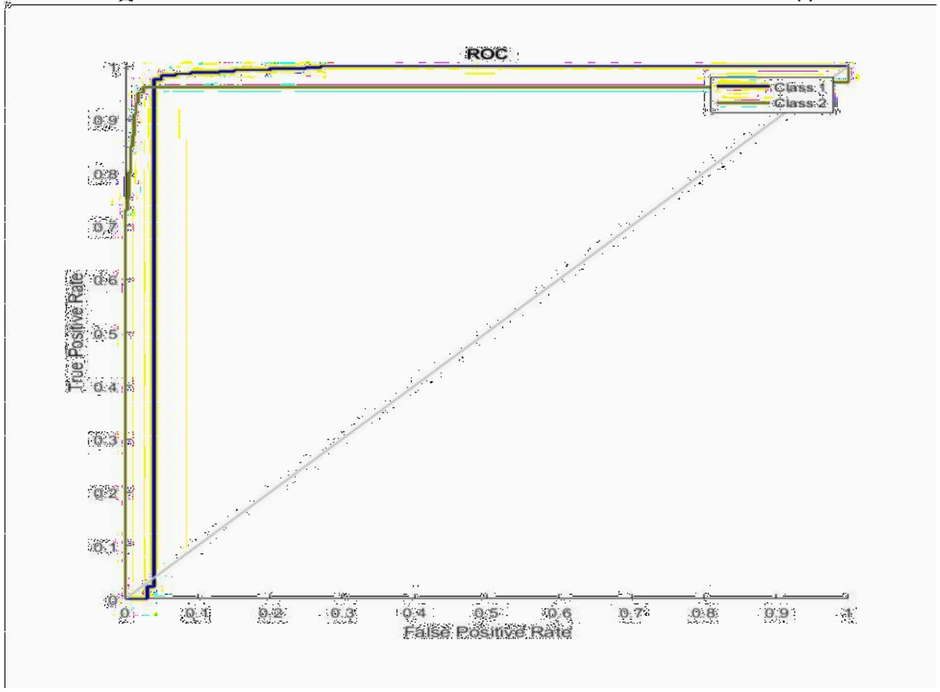

Figure 4.23: ROC Plot for GOA 1Feature 20 Search Agent

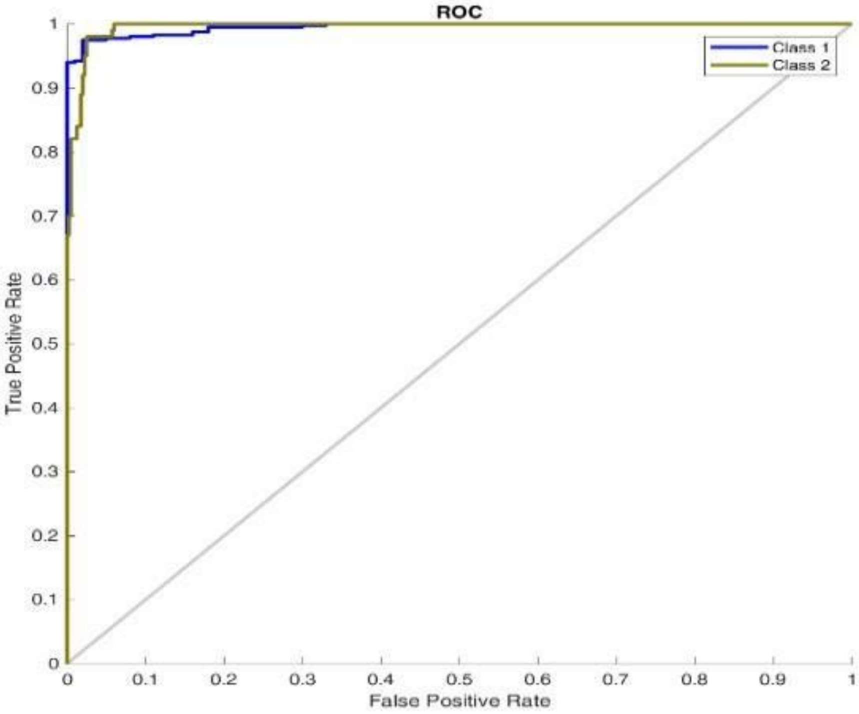

Figure 4.24: ROC Plot for 1 Feature 25 Search Agent

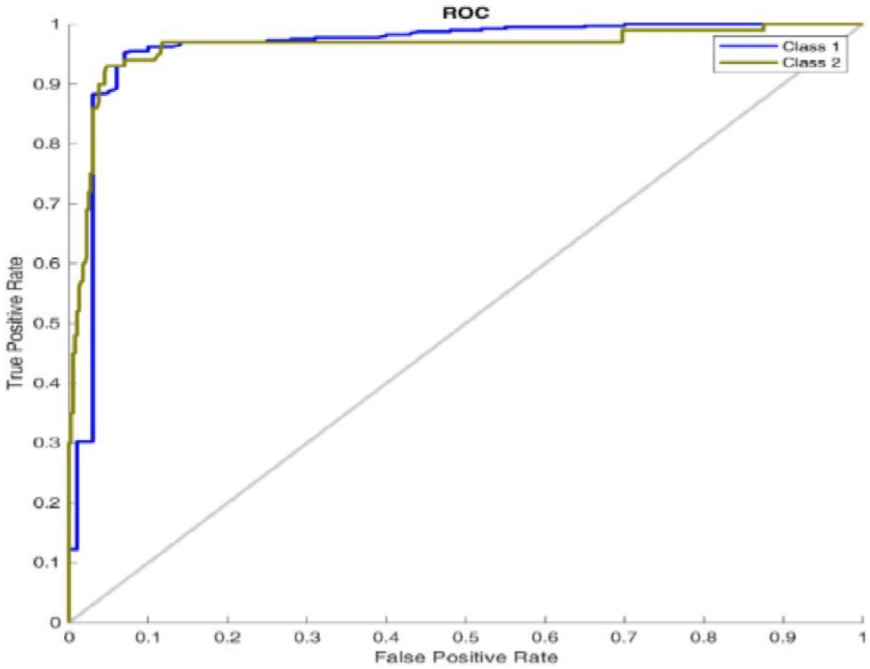

Figure 4.25: ROC Plot for GOA 30 Search Agent

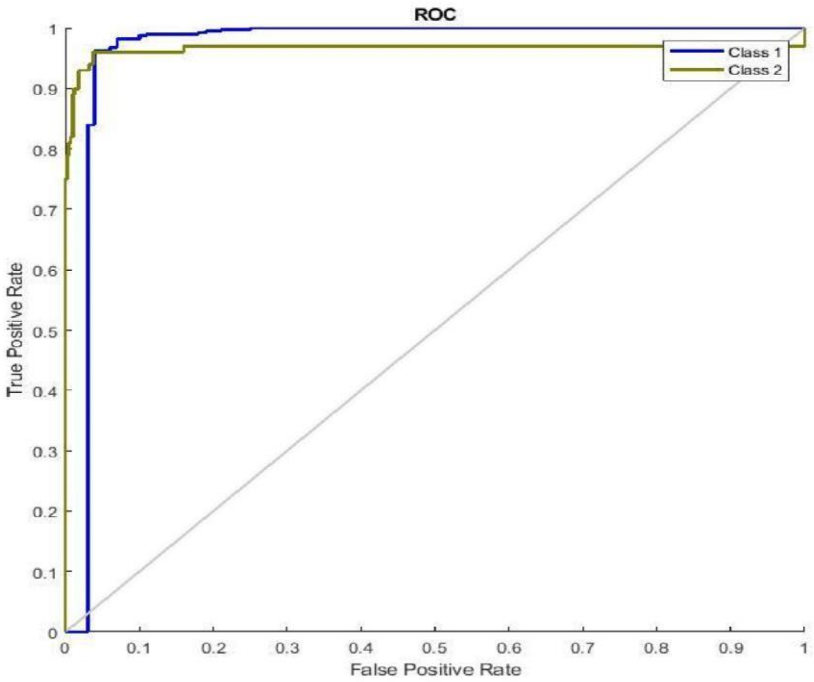

Figure 4.26: ROC Plot for GOA 35 Search Agent

4.5.2 Epilepsy classification using three features, and 5, 10, 15, 20, 25, 30, and 35 search agents respectively

This subsection presents the result of epilepsy detection using three feature selections for GOA-ANN classification with the various search agent of 5, 10, 15, 20, 25, 30, and 35 respectively.

Presented in this subsection in Tables 4.17 and 4.18 are the evaluation methods for GOAANN using three features and 5, 10, 15, 20, 25, 30, and 35 search agents respectively. For search agents 5, 10, 15, 20, 25, 30, and 35, the classifier correctly predicted the true negative classes but wrongly predicted 15, 12, 9, and 10 of the false-negative classes. The F1 score, recall, accuracy, precision, and AUC are shown in Table 4.18. It demonstrates how the GOA-ANN classifier's accuracy rises and falls as the number of search agents increases. The accuracy increases from 97.00% to 98.00% as search agents increase from 5 to 35, respectively. The best accuracy is achieved at 98.20%, with 93.45% precision, 99.00 sensitivity, and an F1 score of 96.04% and 97.40% AUC respectively for 15 search agents for the GOA algorithm. Figure 4.29 shows that the GOA-ANN classifier performs well because the area under the ROC curve (AUC) is close to 1. The representation of the acquired results for 5, 10, 15, 20, 25, 30, and 35 search agents, respectively, is demonstrated using the ROCs of various examples produced using the GOA-ANN classifier in Figures

4.27 to 4.33. In the real classes, the classifier does well, and vice versa.

Table 4.17: GOA-ANN Metrics Using Three-Feature Extraction

No of Feature	Number of GOA Search Agents	TP	FP	TN	FN	Time (s)
[1 4 2]	5	400	15	85	0	61.358
[1 4 8]	10	399	12	88	1	110.282
[4 5 6]	15	400	9	91	0	94.795
[7 1 9]	20	399	14	86	1	86.903
[1 2 4]	25	399	13	87	1	110.900
[1 3 4]	30	400	10	90	0	99.388
[2 9 6]	35	400	9	91	0	79.922

Table 4.18: GOA-ANN Performance Evaluation Using Three-Feature Extraction

No of Feature	Number of GOA Search Agents	Accuracy	Precision	Sensitivity	F1	AUC
[1 4 2]	5	97.0	0.8408	0.9900	0.9093	0.9410
[1 4 8]	10	97.4	0.9288	0.9860	0.9566	0.9700
[4 5 6]	**15**	**98.2**	**0.9345**	**0.9900**	**0.9604**	**0.9740**
[7 1 9]	20	97.0	0.8893	0.9760	0.9306	0.9500
[1 2 4]	25	97.2	0.9252	0.9820	0.9537	0.9660
[1 3 4]	30	98.0	0.9418	0.9820	0.9615	0.9710
[2 9 6]	35	98.2	0.9161	0.9860	0.9498	0.9660

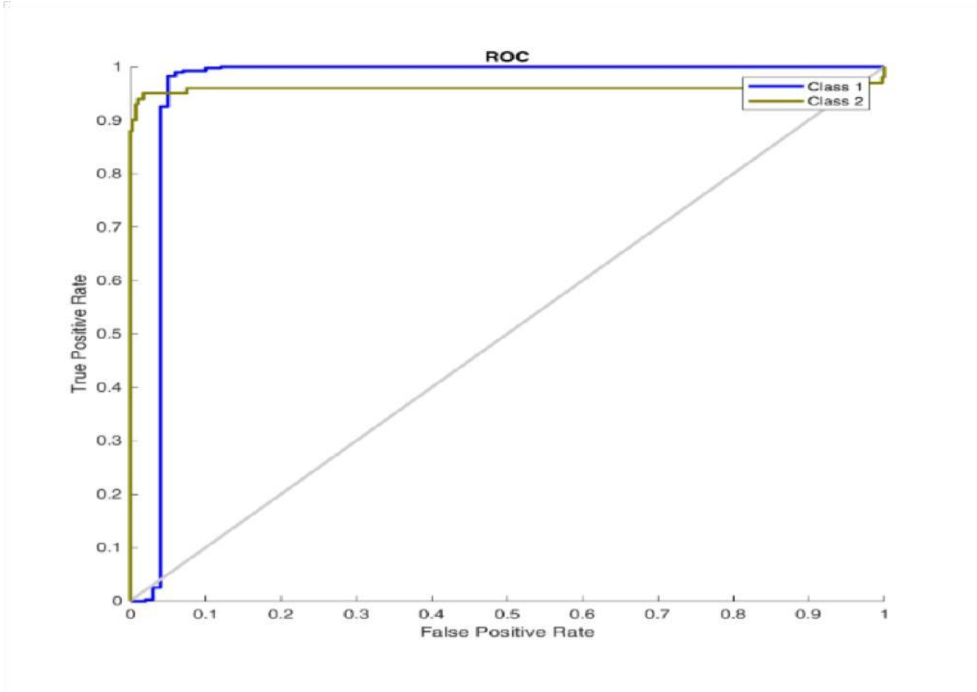

Figure 4.27: ROC Plot for GOA 3 Features 5 Search Agent

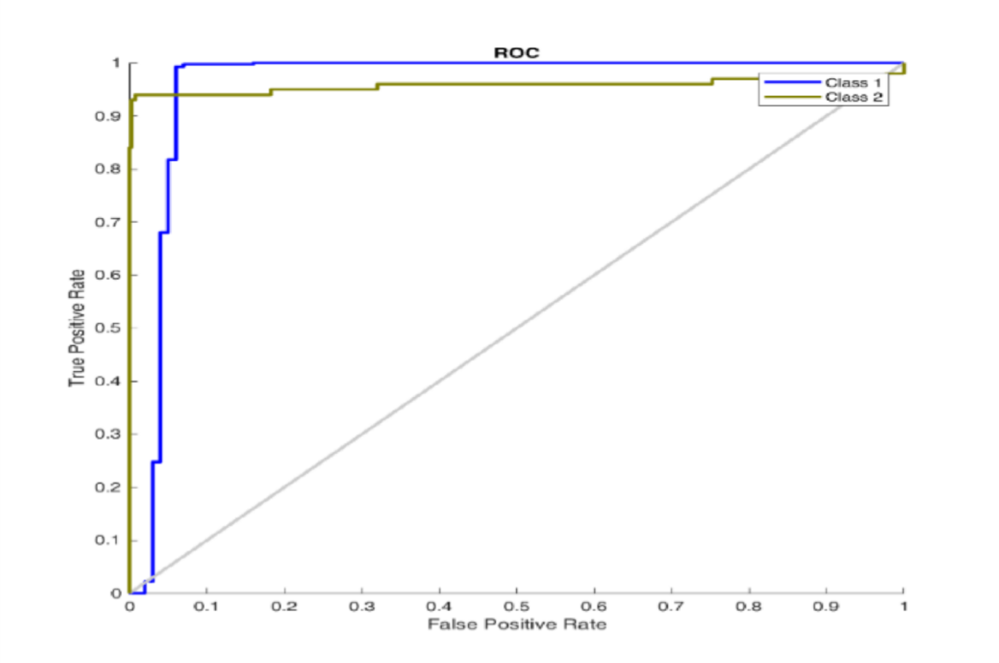

Figure 4.28: ROC Plot for GOA 3 Features 10 Search Agent

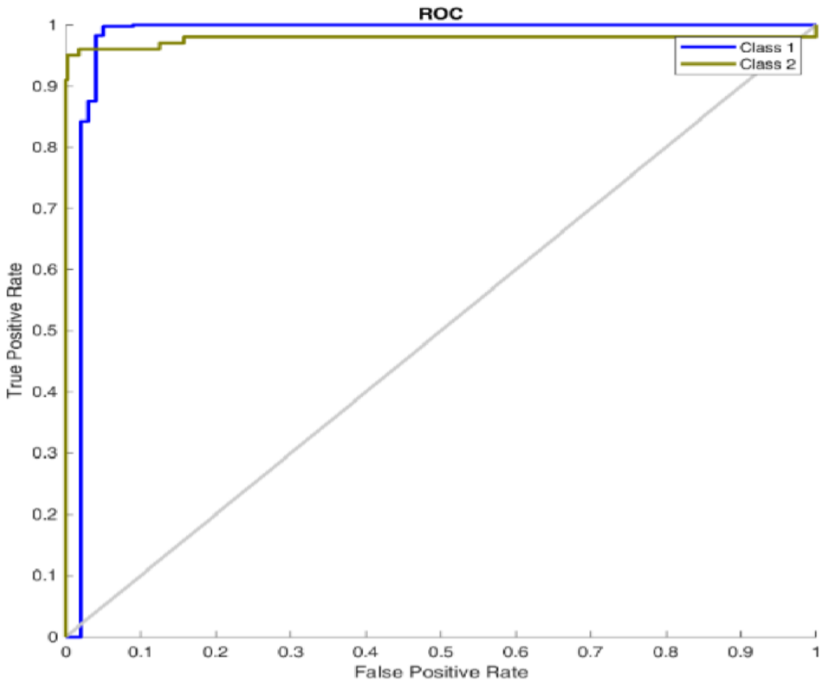

Figure 4.29: ROC Plot for GOA 3 Features 15 Search Agent

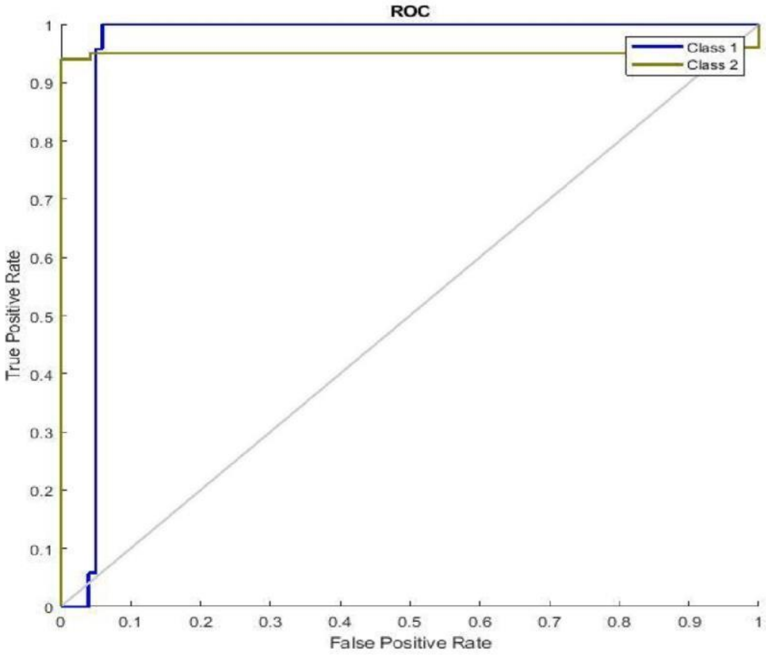

Figure 4.30: ROC Plot for GOA 3 Features 20 Search Agent

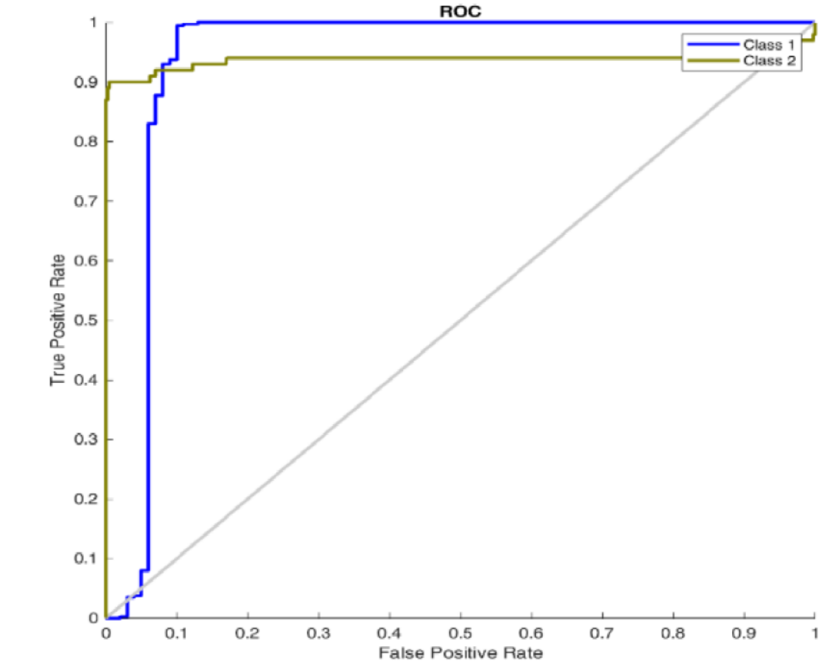

Figure 4.31: ROC Plot for GOA 3 Features 25 Search Agent

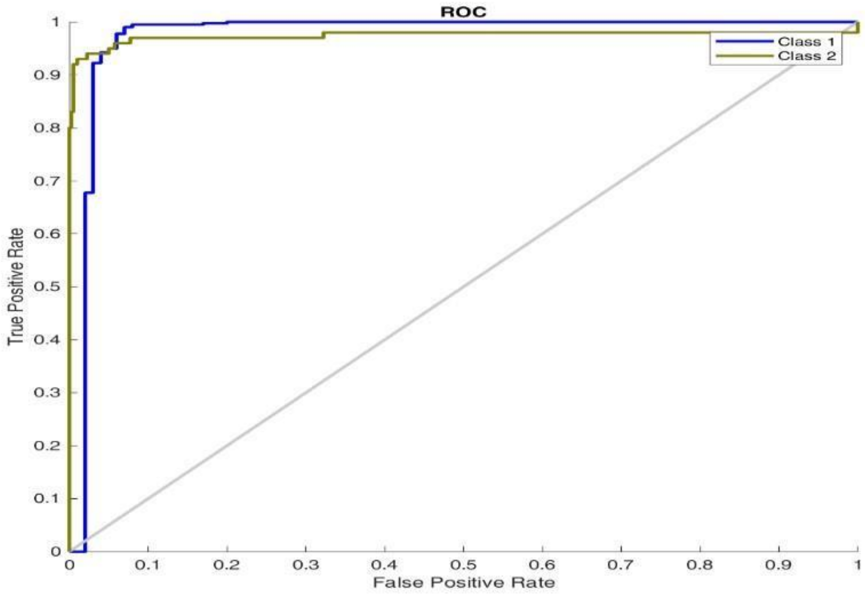

Figure 4.32: ROC Plot for GOA 3 Features 30 Search Agent

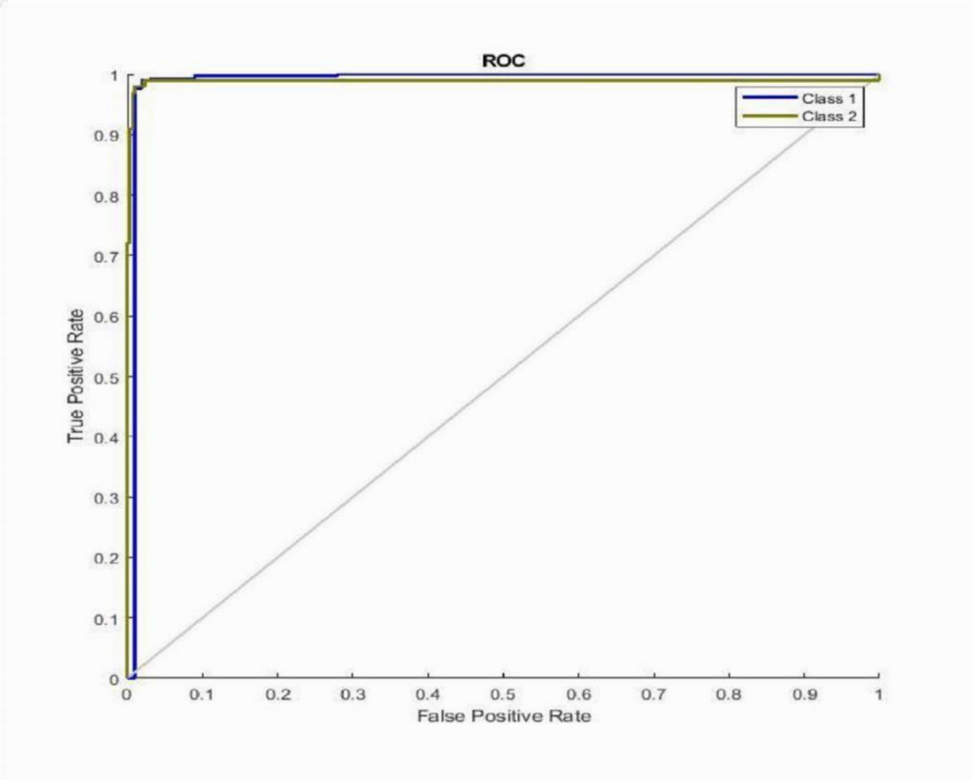

Figure 4.33: ROC Plot for GOA 3 Features 35 Search Agent

4.5.3 Epilepsy classification using five features, and 5, 10, 15, 20, 25, 30, and 35 search agents respectively

This subsection presents the result of epilepsy detection using five feature selections for GOA-ANN classification with the various search agent of 5, 10, 15, 20, 25, 30, and 35 respectively.

Tables 4.19 and 4.20 are the evaluation methods for GOA-ANN using five features and 5, 10, 15, 20, 25, 30, and 35 search agents respectively. The classifier properly predicted the true negative classes, mistakenly predicted 11, 7, 9, and 16 of the false-negative classes, and correctly predicted 0 false negatives for each of the following search agents: 5, 10, 15, 20, 25, 30, and 35. Table 4.20 displays the F1 score, accuracy, precision, recall, and AUC. It demonstrates how the GOA-ANN classifier's accuracy rises and falls as the number of search agents increases. The accuracy changes from 97.80% to 96.80% as search agents increase from 5 to 35, respectively. The best accuracy is achieved at 98.60%, with 98.02% precision, 98.80% sensitivity, F1 score of 98.41%, and 98.60% AUC respectively for 25 search agents for the GOA algorithm. It is observed from Figure 4.34 that the area under the ROC curve (AUC) is near 1, indicating a strong performance of the GOA-ANN classifier. The ROCs of various cases obtained using the GOA-ANN classifier are plotted as shown in figures 4.34 to 4.40 for the representation of the obtained results for 5, 10, 15, 20, 25, 30, and 35 search agents respectively. In the real classes, the classifier performed well, and vice versa. The higher values of the AROC at 15 search agents thus, validate the performance of GOA-ANN using five features.

Table 4.19: GOA-ANN Metrics Using Five-Feature Extraction

No of Feature	Number of GOA Search Agents	TP	FP	TN	FN	Time (s)
[3 9 1 2 4]	5	400	11	89	0	69.789
[9 1 2 3 4]	10	400	7	93	0	66.324
[1 5 6 7 9]	15	400	9	91	0	68.740
[5 9 4 6 3]	20	400	4	96	0	116.840
[1 4 2 3 5]	25	400	7	93	0	65.688
[1 4 5 8 9]	30	400	16	84	0	62.773
[6 3 4 5 2]	35	400	22	78	0	114.905

Table 4.20: GOA-ANN Performance Evaluation Using Five-Feature Extraction

No of Feature	Number of GOA Search Agents	Accuracy	Precision	Sensitivity	F1	AUC
[3 9 1 2 4]	5	97.8	0.9686	0.9800	0.9742	0.9770
[9 1 2 3 4]	10	98.6 98.2	0.9783	0.9860	0.9821	0.9840
[1 5 6 7 9]	15	**99.2**	0.9763	0.9840	0.9801	0.9820
[5 9 4 6 3]	**20**	**98.6**	**0.9785**	**0.9940**	**0.9862**	**0.9900**
[1 4 2 3 5]	**25**	96.8	**0.9802**	**0.9880**	**0.9841**	**0.9860**
[1 4 5 8 9]	30		0.9094	0.9820	0.9443	0.9610
[6 3 4 5 2]	35	95.6	0.8516	0.9760	0.9096	0.9360

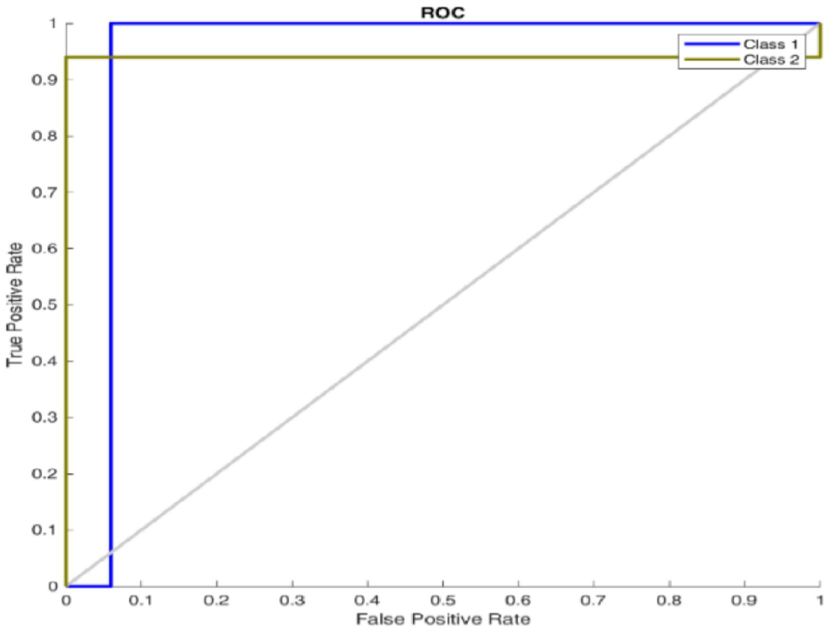

Figure 4.34: ROC Plot for GOA 5 Features 5 Search Agent

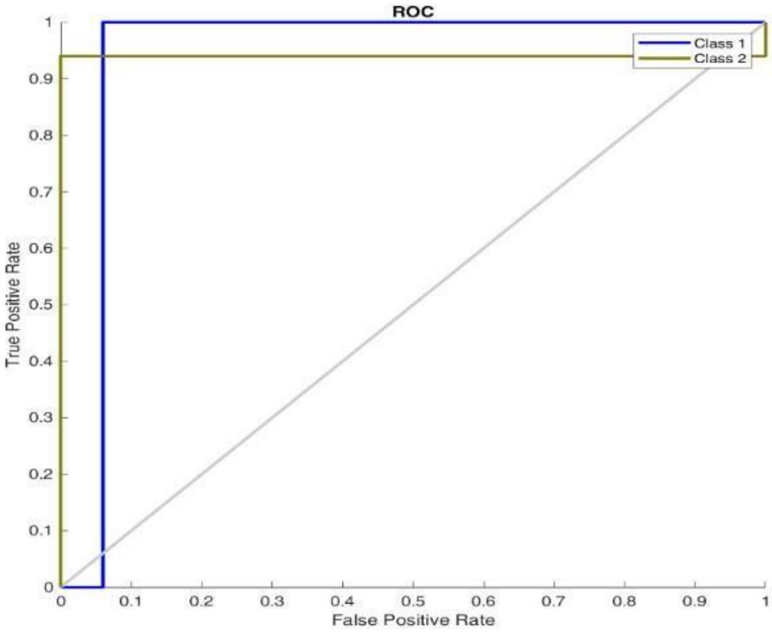

Figure 4.35: ROC Plot for GOA 5 Features 10 Search Agent

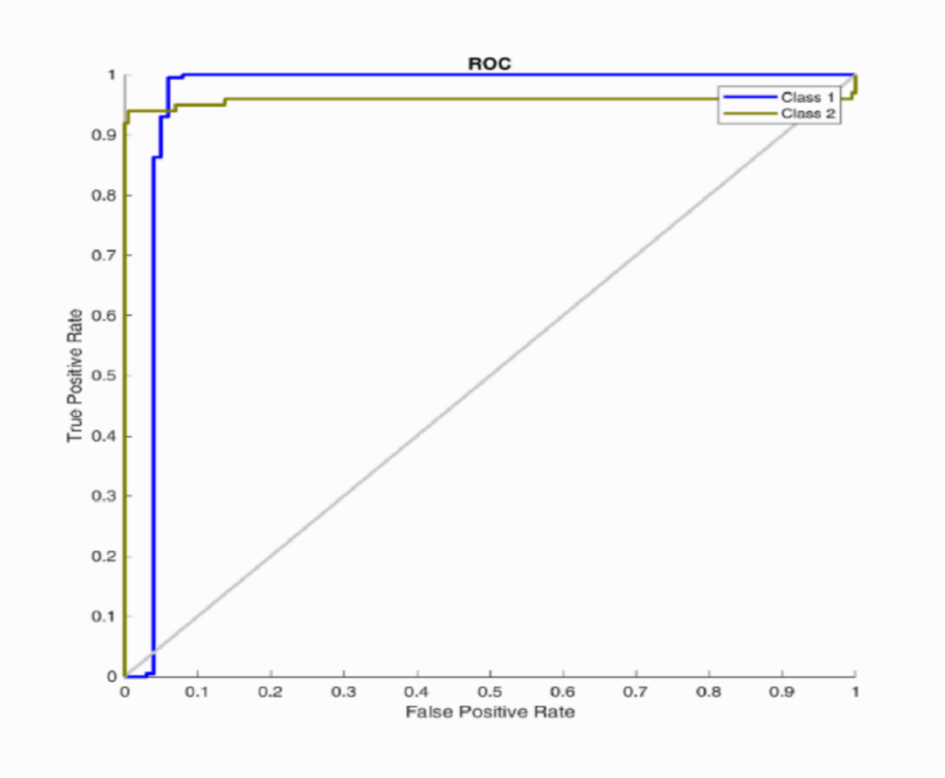

Figure 4.36: ROC Plot for GOA 5 Features 15 Search Agent

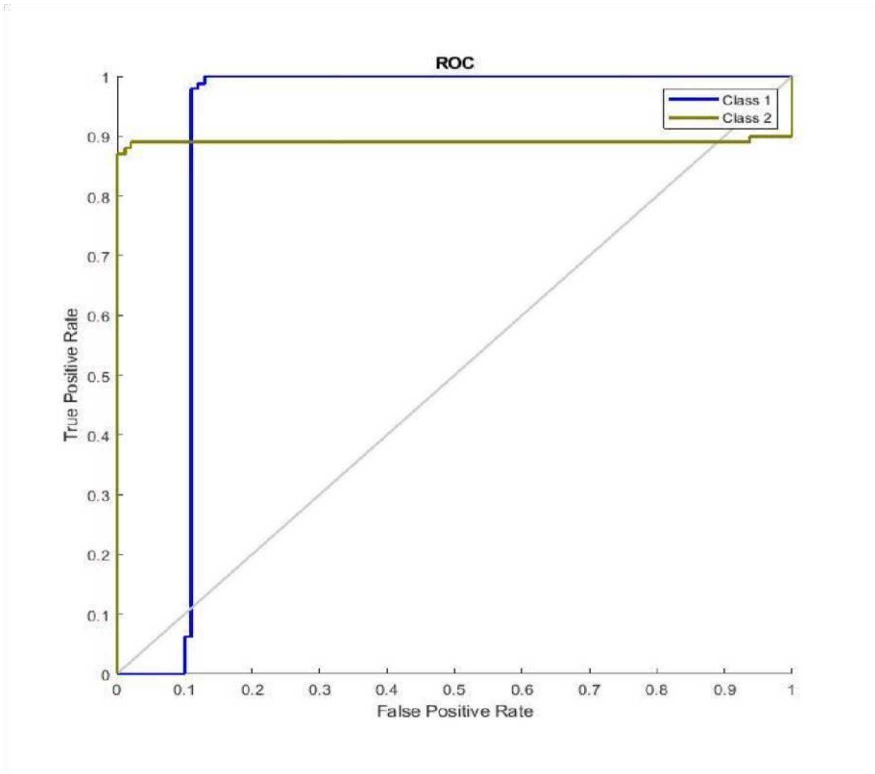

Figure 4.37: ROC Plot for GOA 5 Features 20 Search Agent

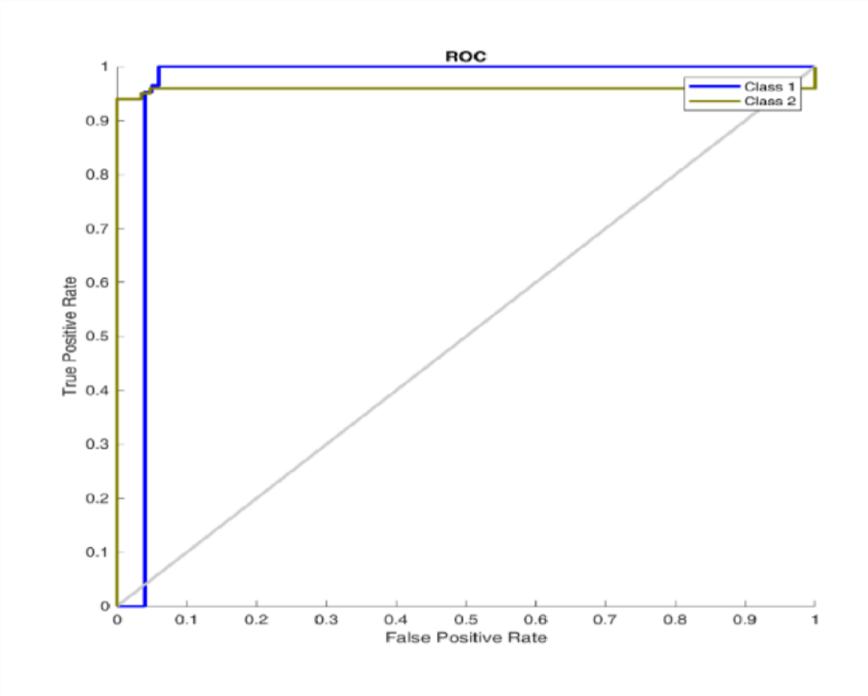

Figure 4.38: ROC Plot for GOA 5 Features 25 Search Agent

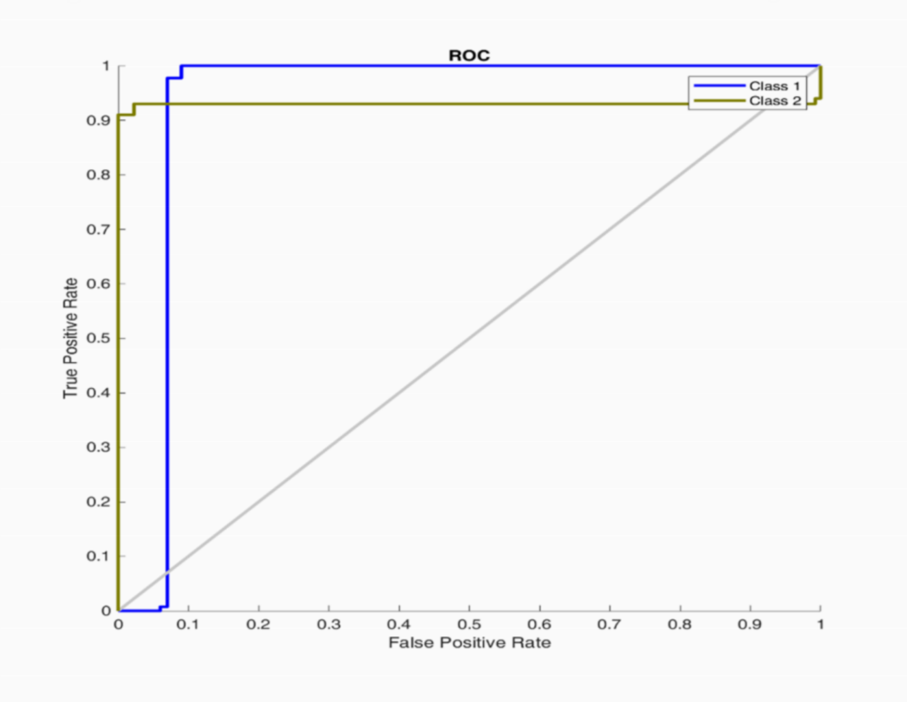

Figure 4.39: ROC Plot for GOA 5 Features 30 Search Agent

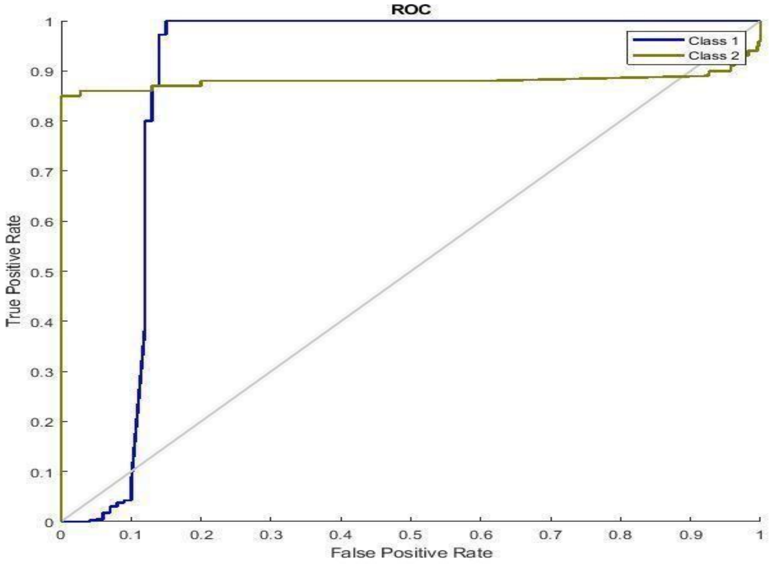

Figure 4.40: ROC Plot for GOA 5 Features 35 Search Agent

4.5.4 Epilepsy classification using seven features, and 5, 10, 15, 20, 25, 30, and 35 search agents respectively

This subsection presents the result of epilepsy detection using seven feature selections for GOA-ANN classification with the various search agent of 5, 10, 15, 20, 25, 30, and 35 respectively.

The evaluation techniques for GOA-ANN are shown in Tables 4.21 and 4.22, respectively, utilizing seven characteristics and 5, 10, 15, 20, 25, 30, and 35 search agents. True negative classes were correctly predicted by the classifier, while false negative classes were incorrectly predicted 12, 14, 13, 10, and 10, and correctly predicted 0 false negatives for search agents 5, 10, 15, 20, 25, 30, and 35, respectively. The F1 score, recall, accuracy, precision, and AUC are shown in Table 4.22. It demonstrates how the GOA-ANN

classifier's accuracy rises and falls as the number of search agents increases. The accuracy increases from 97.60% to 99.40% as search agents increase from 5 to 35, respectively. The best accuracy is achieved at 99.40%, with 98.81% precision, 99.60% sensitivity, and an F1 score of 99.21% and 99.40% AUC respectively for 30 search agents for the GOA algorithm. It is observed from Figure 4.46 that the area under the ROC curve (AUC) is almost 1, indicating a strong performance of the GOA-ANN classifier. The ROCs of various cases obtained using the GOA-ANN classifier are shown in figures 4.41 to 4.47 for the representation of the obtained results for 5, 10, 15, 20, 25, 30, and 35 search agents respectively. In the real classes, the classifier does well, and vice versa. The higher values of the ROC at 30 search agents thus, validate the performance of GOA-ANN using seven features.

Table 4.21: GOA-ANN Metrics Using Seven Feature Extraction

No of Feature	Number of GOA Search Agents	TP	FP	TN	FN	Time (s)
[1 2 5 7 3 4 6]	5	400	12	88	0	61.950
[2 3 1 4 5 6 7]	10	400	14	86	0	70.337
[2 3 4 5 1 6 7]	15	400	13	84	0	81.759
[5 3 7 4 2 6 8]	20	400	13	87	0	100.273
[5 6 8 9 1 2 3]	25	400	10	90	0	84.665
[1 6 7 2 3 4 5]	30	400	10	97	0	73.160
[5 8 3 1 6 7 9]	35	400	4	96	0	89.316

Table 4.22: GOA-ANN Performance Evaluation Using Seven Feature Extraction

No of Feature	Number of GOA Search Agents	Accuracy	Precision	Sensitivity	F1	AUC
[1 2 5 7 3 4 6]	5	97.6	0.9666	0.9780	0.9723	0.9750
[2 3 1 4 5 6 7]	10	97.2	0.9573	0.9760	0.9666	0.9710
[2 3 4 5 1 6 7]	15	97.4	0.9592	0.9750	0.9685	0.9730
[5 3 7 4 2 6 8]	20	97.4	0.9666	0.9780	0.9723	0.9750
[5 6 8 9 1 2 3]	25	98.0	0.9705	0.9820	0.9762	0.9762
[1 6 7 2 3 4 5]	**30**	**99.4**	**0.9881**	**0.9960**	**0.9921**	**0.9940**
[5 8 3 1 6 7 9]	35	99.2	0.9730	0.9960	0.9844	0.9900

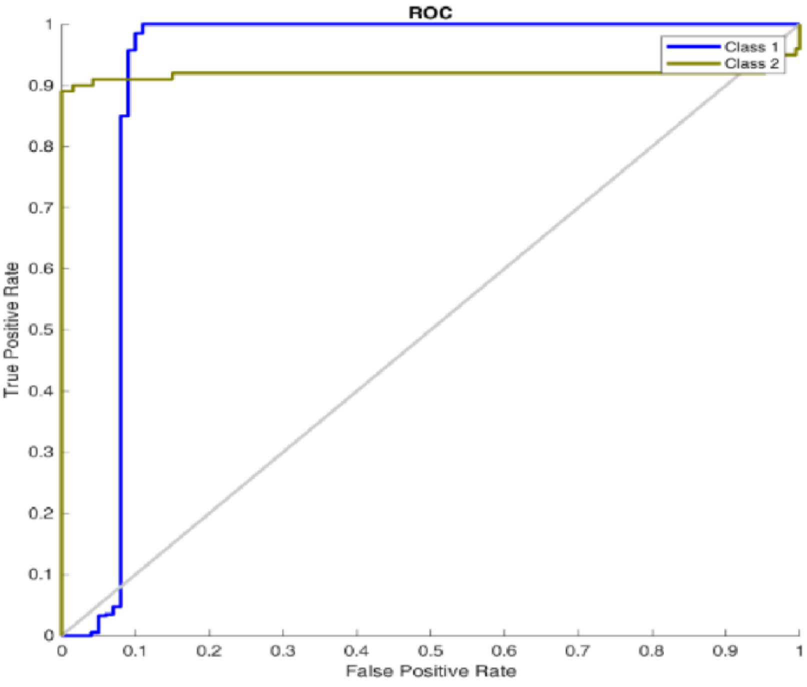

Figure 4.41: ROC Plot for GOA 7 Features 5 Search Agent

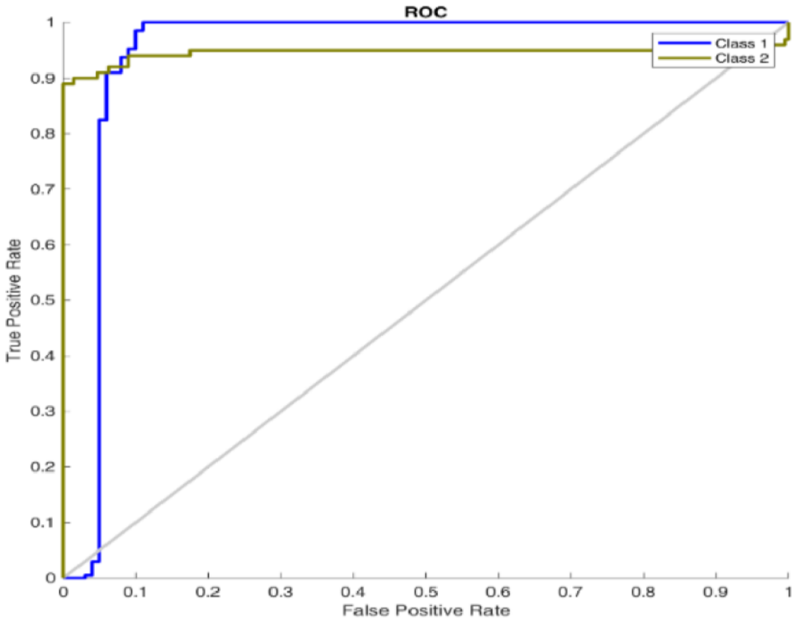

Figure 4.42: ROC Plot for GOA 7 Features 10 Search Agent

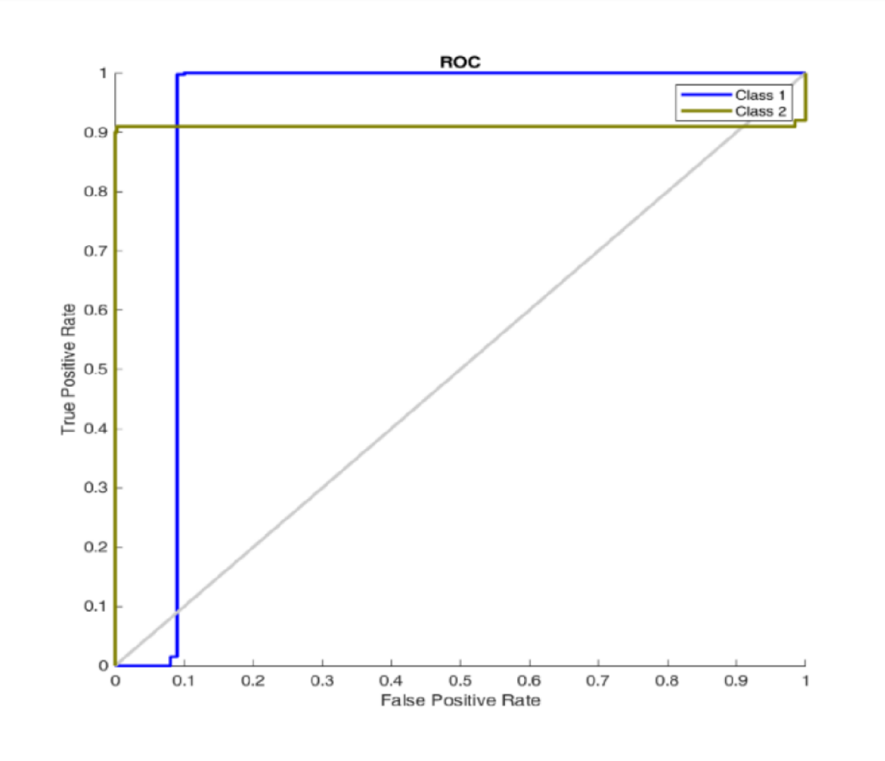

Figure 4.43: ROC Plot for GOA 7 Features 15 Search Agent

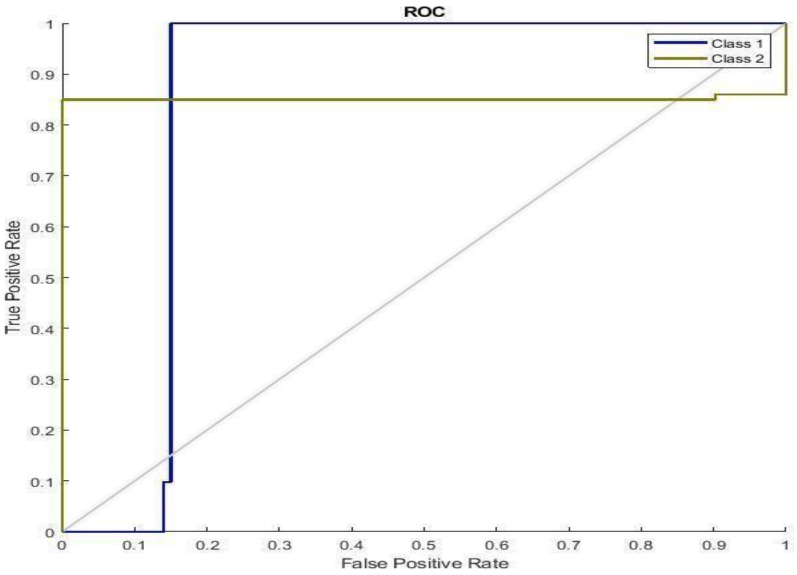

Figure 4.44: ROC Plot for GOA 7 Features 20 Search Agent

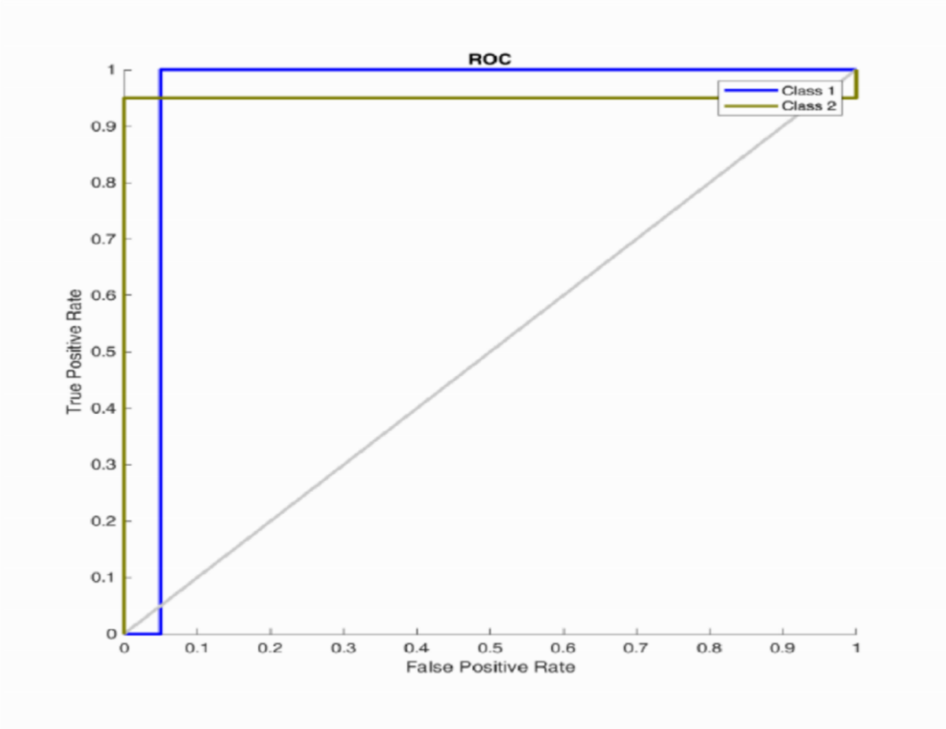

Figure 4.45: ROC Plot for GOA 7 Features 25 Search Agent

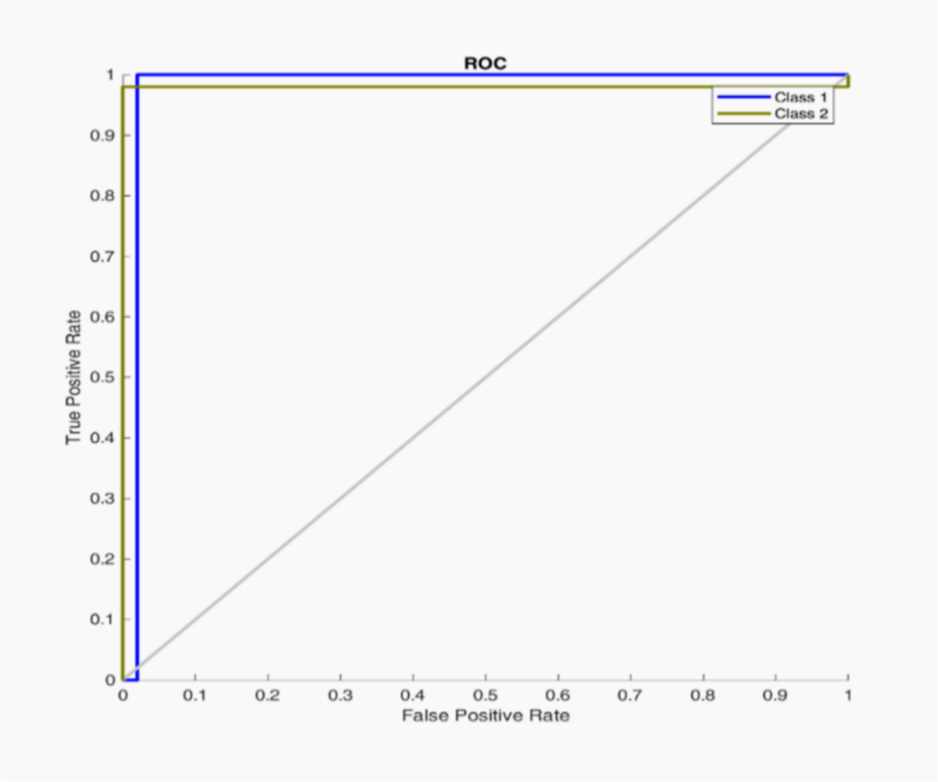

Figure 4.46: ROC Plot for GOA 7 Features 30 Search Agent

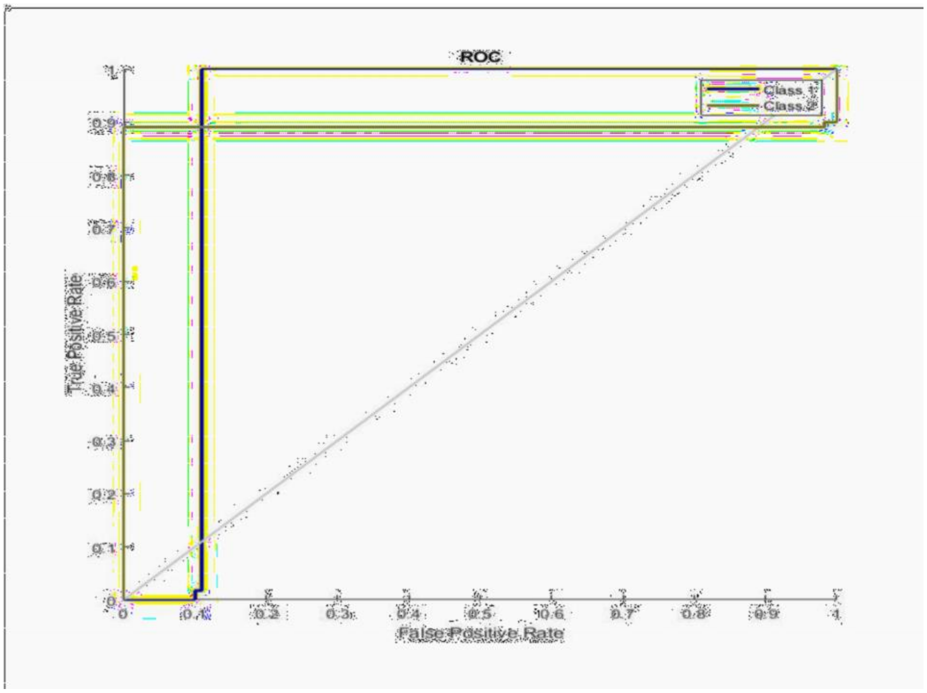

Figure 4.47: ROC Plot for GOA 7 Features 35 Search Agent

4.5.5 Epilepsy classification using nine features, and 5, 10, 15, 20, 25, 30 and 35 search agents respectively

This subsection presents the result of epilepsy detection using nine feature selections for GOA-ANN classification with the various search agent of 5, 10, 15, 20, 25, 30 and 35 respectively. The evaluation techniques for GOA-ANN are shown in Tables 4.23 and 4.24, respectively, utilizing nine characteristics and 5, 10, 15, 20, 25, 30 and 35 search agents. The classifier accurately identified the real negative classes, but it incorrectly predicted the fake negative classes 11, 9, 18, and 13. Additionally, it correctly predicted that there would be 0 false negatives for search agents 5, 10, 15, 20, 25, and 35. The F1 score, recall, accuracy, precision, and AUC are displayed in Table 4.24. It demonstrates how the GOAANN classifier's accuracy rises and falls as the number of search agents increases. The accuracy changes from 97.80% to 97.40% as search agents increase from 5 to 35, respectively. The best accuracy is achieved at 98.40%, with 97.07% precision, 98.60% sensitivity, F1 score of 97.83%, and 98.20% AUC respectively for 25 search agents for the GOA algorithm. It is observed from Figure 4.52 that the area under the ROC curve (AUC) is almost 1, indicating a strong performance of the GOA-ANN classifier. The ROCs of various cases obtained using the GOA-ANN classifier are shown in Figures 4.48 to 4.54 for the representation of the obtained results for 5, 10, 15, 20, 25, 30, and 35 search agents respectively. In the real classes, the classifier performs well, and vice versa. The higher values of the ROC at 25 search agents thus, validate the performance of GOA-ANN using nine features.

Table 4.23: GOA-ANN Metrics Using Nine-Feature Extraction

No of Feature	Number of GOA Search Agents	TP	FP	TN	FN	Time (s)

[1 2 5 7 3 4 6 7 9]	5	400	11 9	89	0	62.175
[1 3 6 7 8 9 2 4 5]	10	400	18	91	0	76.515
[4 5 7 9 1 2 3 6 8]	15	400	19	8 82	0	67.177
[9 3 8 2 5 1 7 4 6]	20	400	13	81	0	108.167
[3 5 7 1 2 4 6 8 9]	25	400	12	92	0	59.329
[1 2 6 7 3 4 5 8 9]	30	400		87	0	81.233
[6 3 4 1 8 7 2 9 5]	35	400		88	0	108.276

Table 4.24: GOA-ANN Performance Evaluation Using Nine-Feature Extraction

No of Feature	Number of GOA Search Agents	Accuracy	Precision	Sensitivity	F1	AUC
[1 2 5 7 3 4 6 7 9]	5	97.8	0.9542	0.9840	0.9781	0.9800
[1 3 6 7 8 9 2 4 5]	10	98.2	0.9764	0.9880	0.9822	0.9850
[4 5 7 9 1 2 3 6 8]	15	96.4	0.9362	0.9760	0.9557	0.9650
[9 3 8 2 5 1 7 4 6]	20	96.2	0.9466	0.9760	0.9611	0.9680
[3 5 7 1 2 4 6 8 9]	**25**	**98.4**	**0.9707**	**0.9860**	**0.9783**	**0.9820**
[1 2 6 7 3 4 5 8 9]	30	97.4	0.9664	0.9740	0.9702	0.9720
[6 3 4 1 8 7 2 9 5]	35	97.6	0.9666	0.9780	0.9723	0.9750

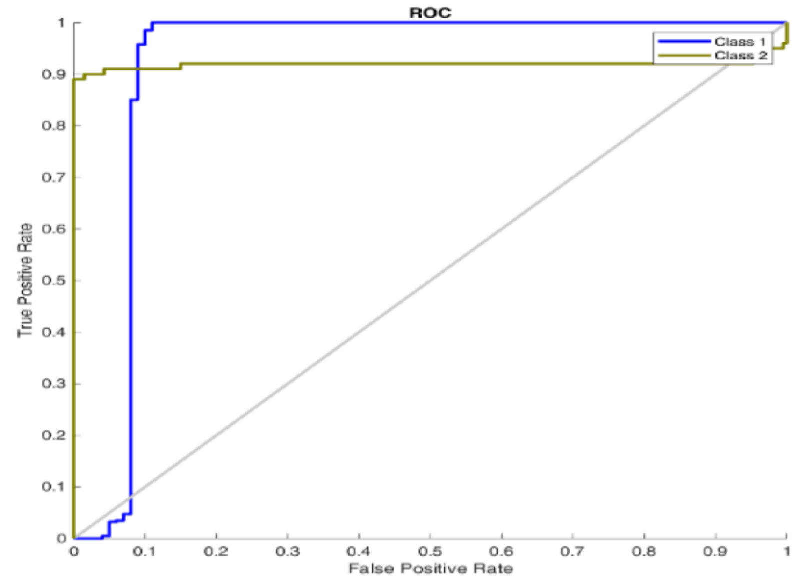

Figure 4.48: ROC Plot for GOA 9 Features 5 Search Agent

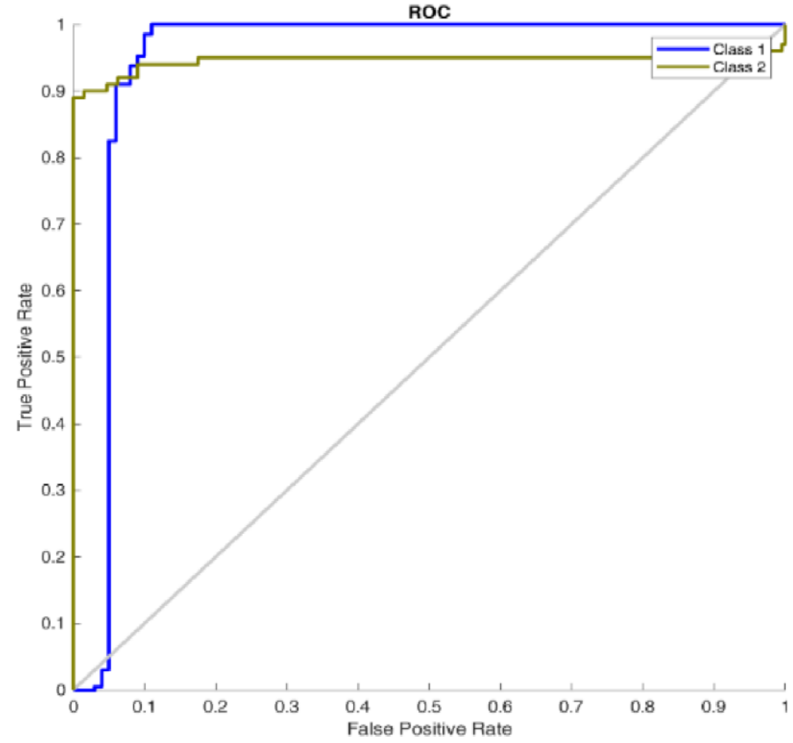

Figure 4.49: ROC Plot for GOA 9 Features 10 Search Agent

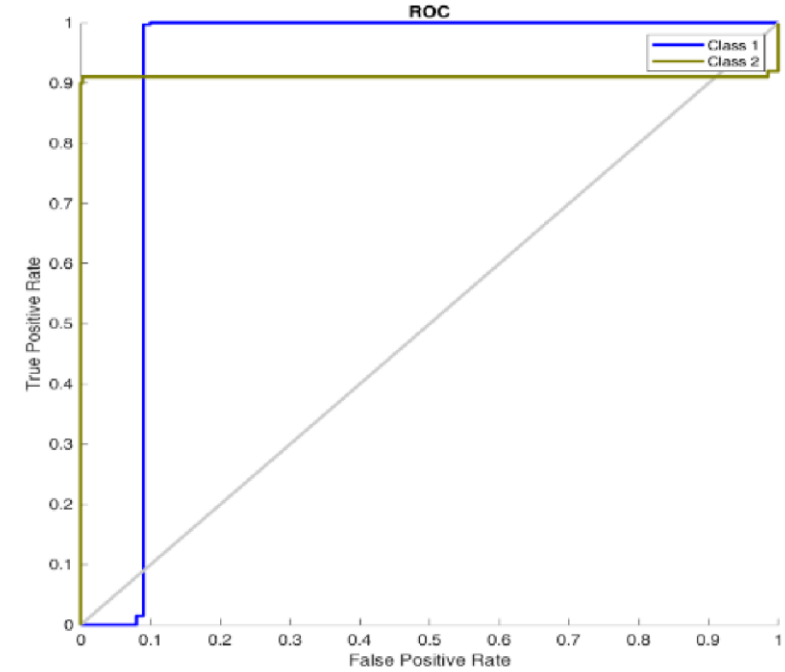

Figure 4.50: ROC Plot for GOA 9 Features 15 Search Agent

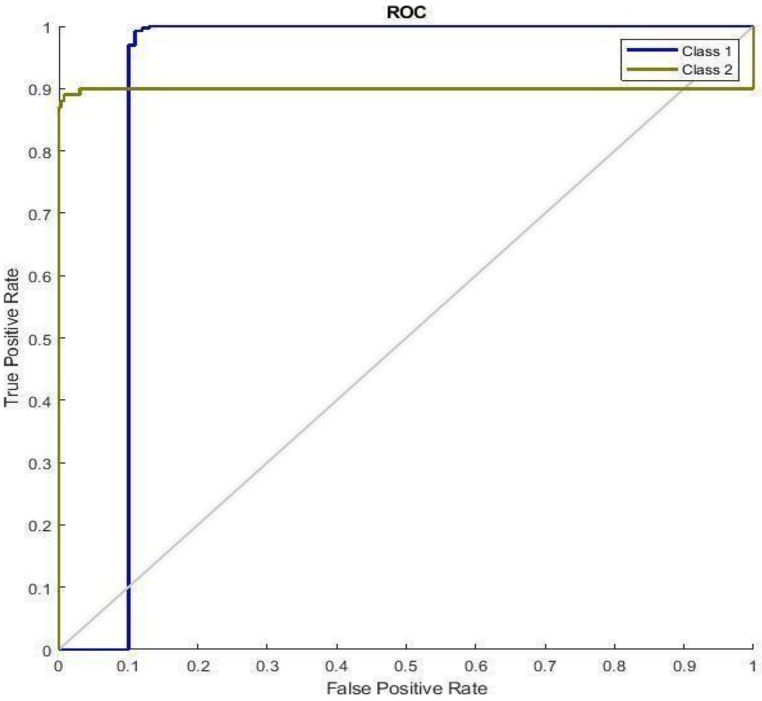

Figure 4.51: ROC Plot for GOA 9 Features 20 Search Agent

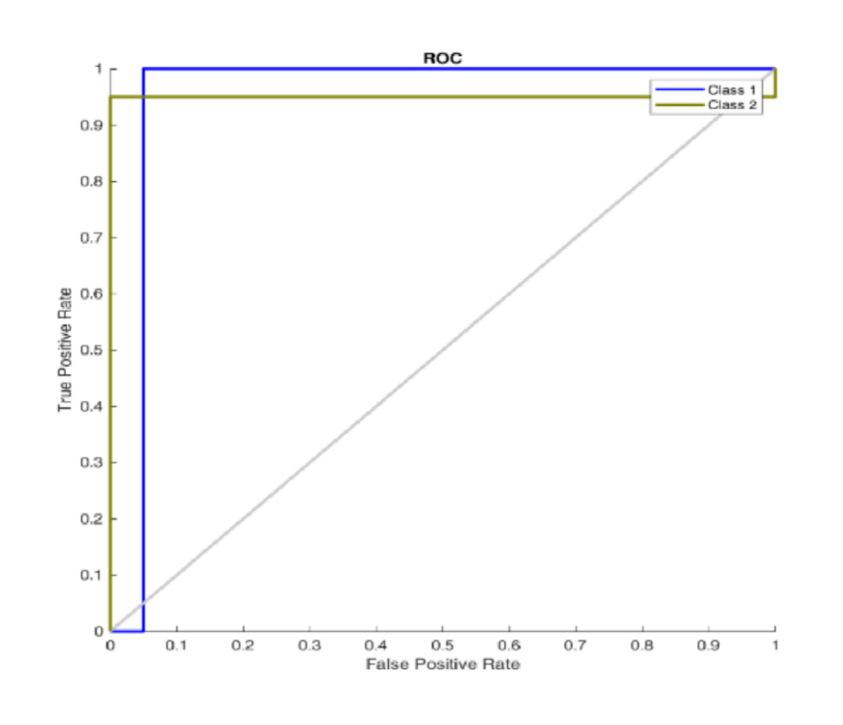

Figure 4.52: ROC Plot for GOA 9 Features 25 SA

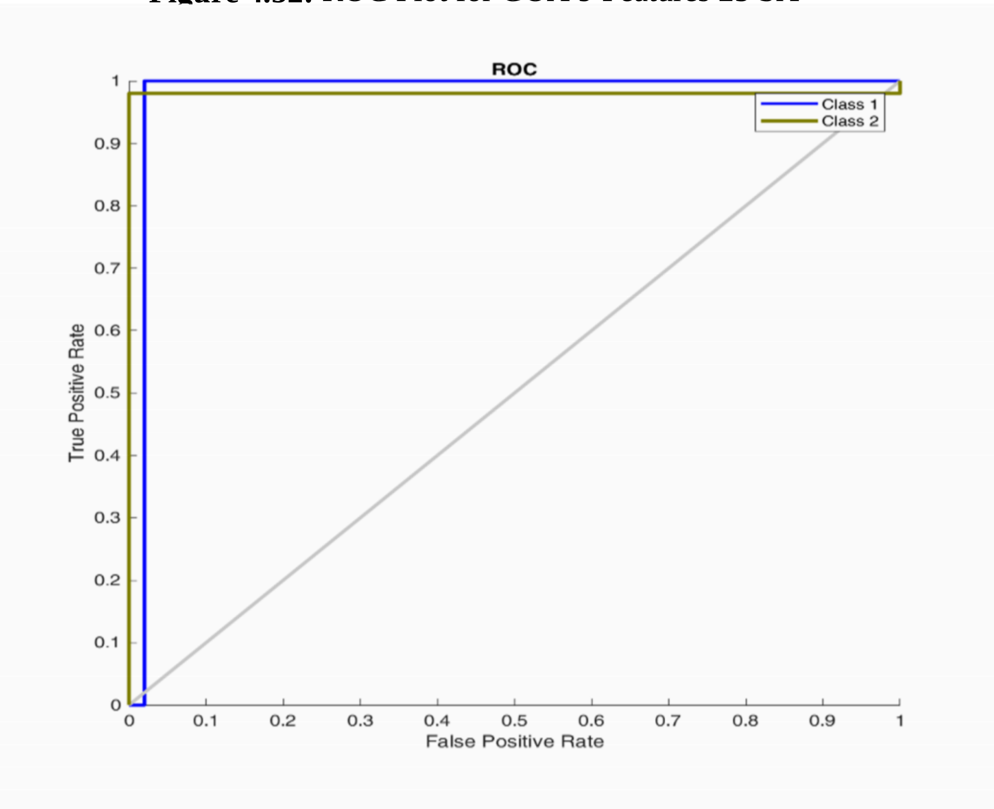

Figure 4.53: ROC Plot for GOA 9 Features 30 Search Agent

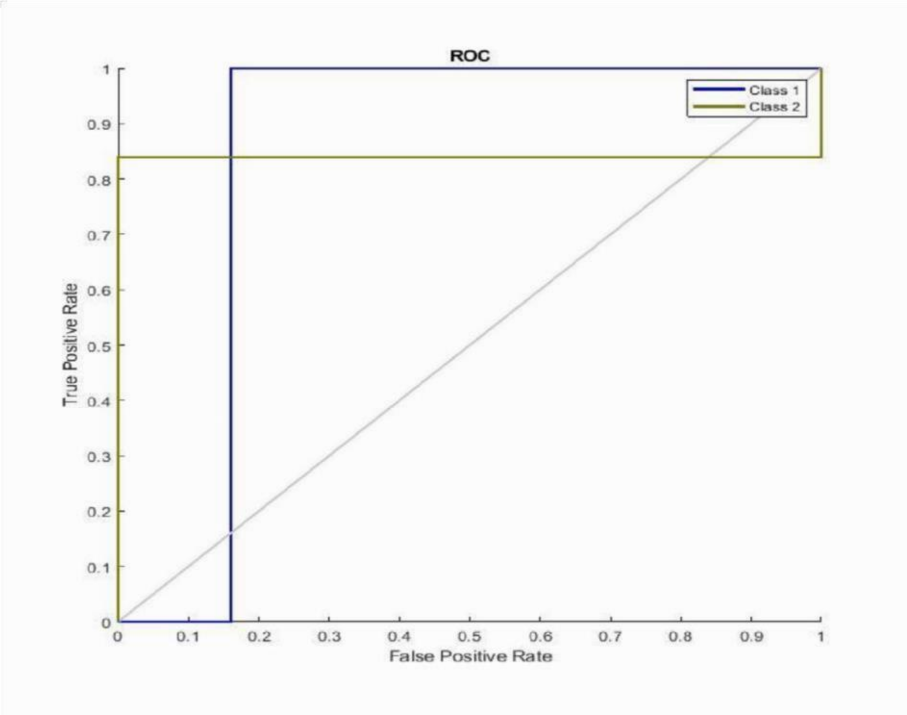

Figure 4.54: ROC Plot for GOA 9 Features 35 Search Agent

The evaluation techniques for GOA-ANN utilizing one, three, five, seven, and nine characteristics for 5, 10, 15, 20, 25, and 35 search agents, respectively, are presented in Table 4.25. For different feature classes, the F1 score, recall, accuracy, precision, and AUC are also displayed. It demonstrates that, except for nine features, which have a lower accuracy compared to seven features or the GOA-ANN classifier, the accuracy rises as the number of features grows from one to seven. With 98.81% precision, 99.60% sensitivity, an F1 score of 99.21% and 99.20% AUC for each of the 30 search agents, and seven feature extractions for the GOA-ANN algorithm, the greatest accuracy is attained at 99.40%. Figure 4.55 shows that the accuracy is almost 100%, demonstrating the GOA-ANN classifier's high performance when using seven feature extractions. When compared to other classes of the feature's extraction, The classifier performs effectively in the real classes and poorly in the false classes for seven features.

Table 4.25: Summary of Epilepsy Detection Using GOA-ANN with Various Search Agents

Evaluation Metrics	1 Feature 5 Search Agent	3 Features 15 Search Agent	5 Features 25 Search Agent	7 Features 30 Search Agent	9 Features 25 Search Agent
Accuracy	0.9600	0.9820	0.9860	**0.9940**	0.9840
Precision	0.9880	0.9345	0.9802	**0.9881**	0.9707
Sensitivity	0.8938	0.9900	0.9880	**0.9960**	0.9860
F1 Measure	0.9386	0.9604	0.9841	**0.9921**	0.9783
AUC	0.9600	0.9740	0.9860	**0.9940**	0.9820

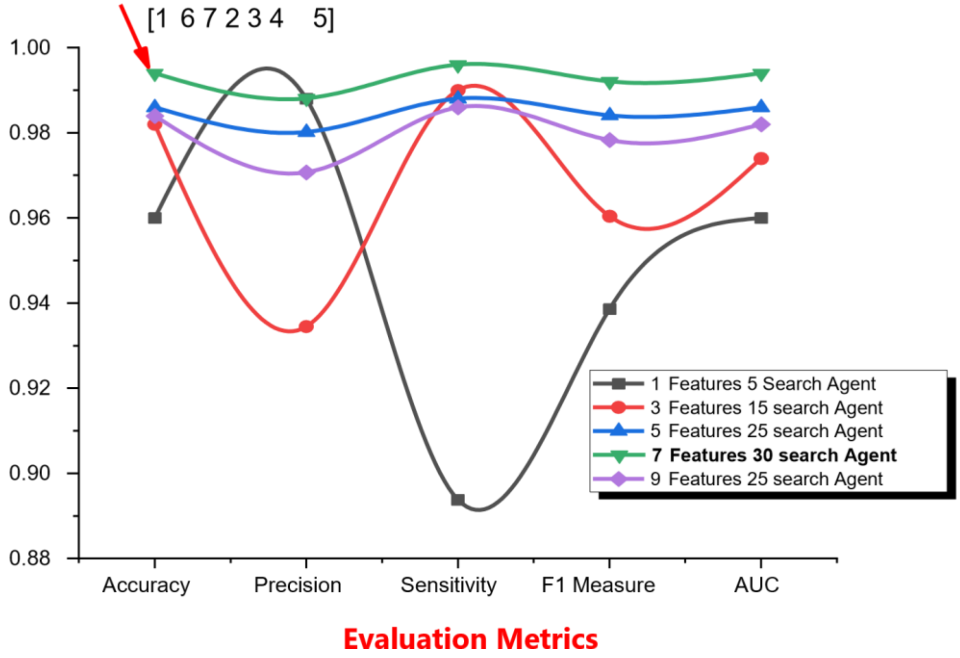

Figure 4.55: GOA-ANN Performance with Various Search Agents

4.6 Epilepsy Classification Using Grey Wolf Optimization Algorithm

In this section, epilepsy is detected and categorized using the Grey Wolf optimization technique and an artificial neural network. The various feature extraction ranges are 1, 3,5, 7, and 9, and their corresponding search agents are 5, 10, 15, 20, 25, and 35, respectively.

4.6.1. Epilepsy classification using one feature and 5, 10, 15, 20, 25, 30, and 35 search agents

The results of detecting and classifying epilepsy using the Grey Wolf optimization algorithm and an artificial neural network are shown in this subsection for one feature extraction and 5, 10, 15, 20, 25, 30, and 35 search agents, respectively. The evaluation techniques for Grey Wolf and ANN utilizing one feature and 5, 10, 15, 20, 25, 30, and 35 search agents, respectively, are presented in Tables 4.26 and 4.27. The classifier accurately predicted the true-negative classes for 14, 14, 14, and 14 whereas it wrongly predicted the false-negative classes for 14, 14, 14, and 14.

The F1 score, recall, accuracy, precision, and AUC are all displayed in Table 4.27. It demonstrates how the GOA-ANN classifier's accuracy rises and falls as the number of search agents increases. As the search agent changes from 5 to 35, the accuracy stays the same. With a 94.40% accuracy rate, 75.59% precision, 98.60 sensitivity, an F1 score of 85.44%, and an AUC of 89.40%, respectively, it is the best accuracy available. The overall effectiveness of a clinical test is determined by the area under the ROC curve (AROC), which is used to evaluate the diagnostic accuracy of any test. As shown in Figure 4.61, the Grey Wolf-ANN classifier, which uses thirty search agents, performs well, with an area under the ROC curve (AUC) of approximately 1, suggesting good performance. The ROCs of various examples acquired using the Grey Wolf-ANN classifier are represented for 5, 10, 15, 20, 25, 30, and 35 search agents, respectively, in Figures 4.56 to 4.62.

Table 4.26: Grey-Wolf Metrics Using One Feature Extraction

Feature Extraction	**Number of Search Agents**	**TP**	**FP**	**TN**	**FN**	**Time** (s)
[1]	5	386	14	86	14	24.917
[1]	10	386	14	86	14	44.426
[1]	15	386	14	86	14	62.084
[1]	20	386	14	86	14	116.090
[1]	25	386	14	86	14	110.580
[1]	30	386	14	86	14	120.286
[1]	35	386	14	86	14	192.042

Table 4.27: Grey-Wolf Performance Evaluation using One Feature Extraction

Feature Extraction	**Number of Search Agents**	**Accuracy**	**Precision**	**Sensitivity**	**F1**	**AUC**
[1]	5	94.4	0.7522	0.9860	0.8534	0.8930
[1]	10	94.4	0.7522	0.9860	0.8534	0.8930
[1]	15	94.4	0.7522	0.9860	0.8534	0.8930
[1]	20	94.4	0.7522	0.9860	0.8534	0.8930
[1]	25	94.4	0.7522	0.9860	0.8534	0.8930
[1]	**30**	**94.4**	**0.7539**	**0.9860**	**0.8544**	**0.8940**
[1]	35	94.4	0.7522	0.9860	0.8534	0.8930

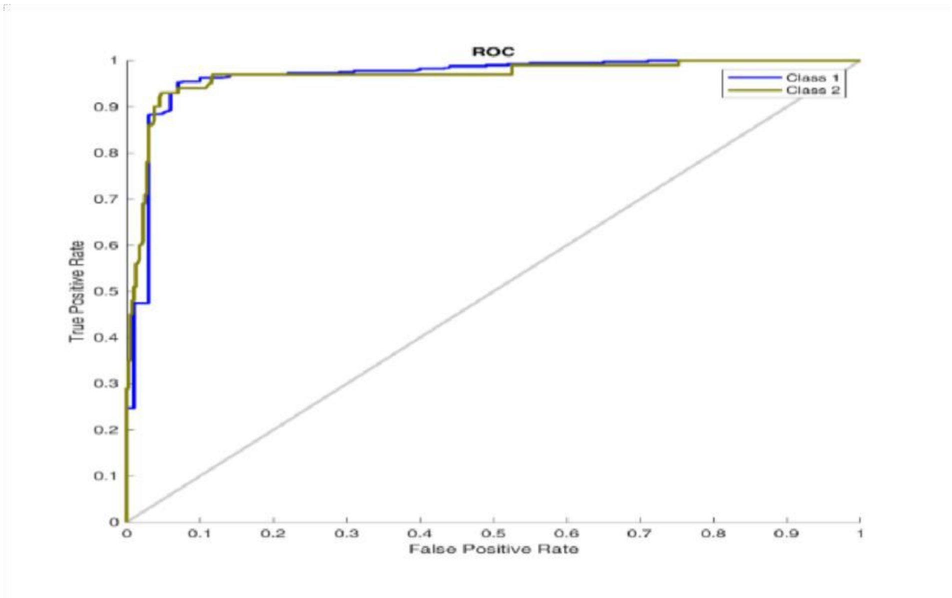

Figure 4.56: ROC Plot for Grey Wolf 1 Features 5 Search Agent

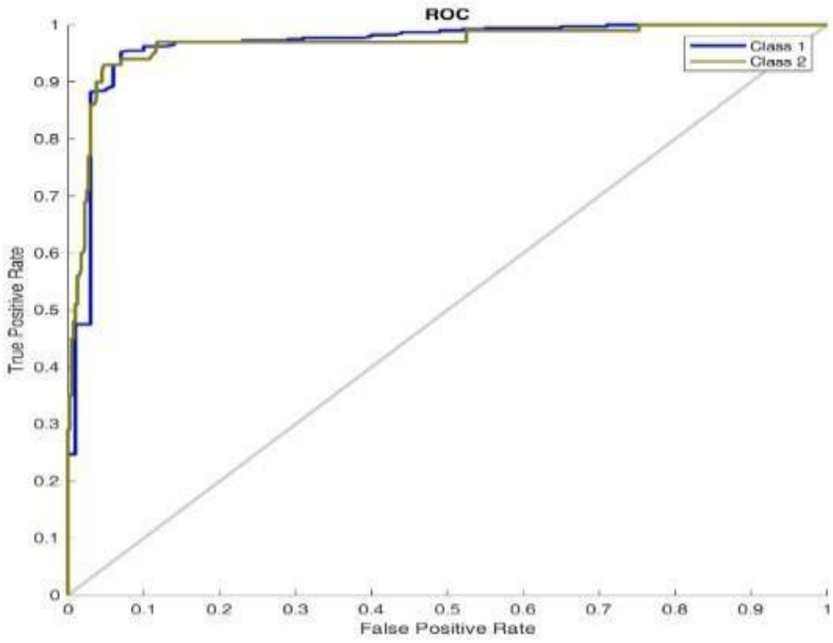

Figure 4.57: ROC Plot for Grey Wolf 1Feature 10 Search Agent

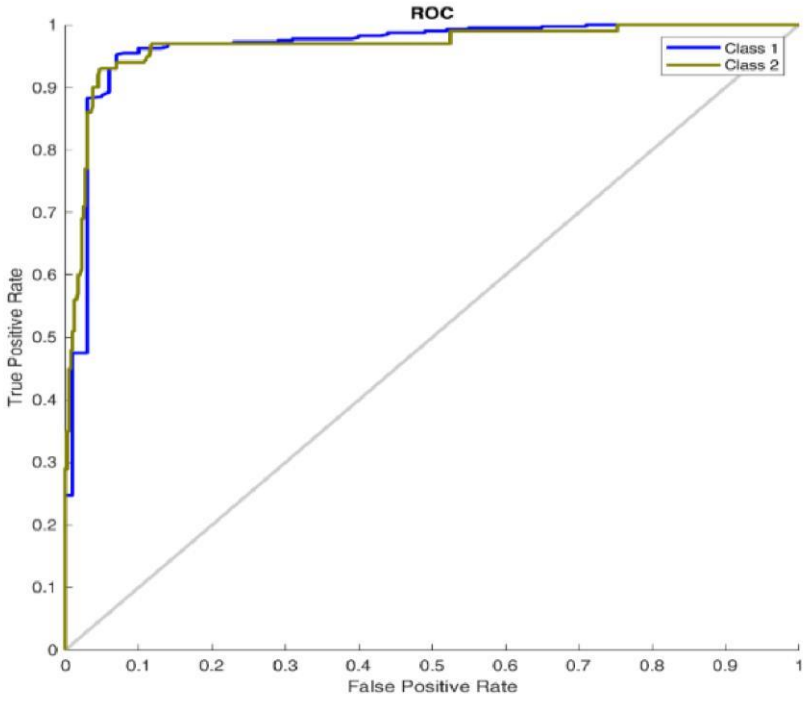

Figure 4.58: ROC Plot for Grey Wolf 1 Features 15 Search Agent

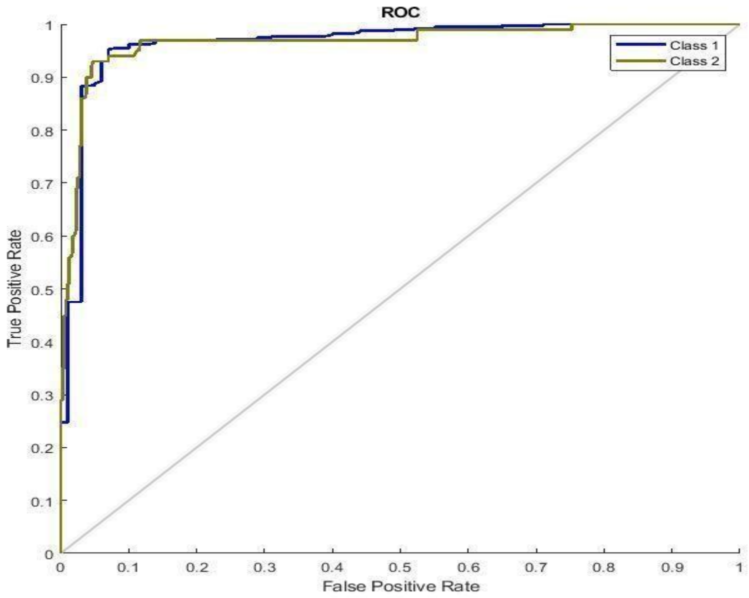

Figure 4.59: ROC Plot for Grey Wolf 1 Features 20 Search Agent

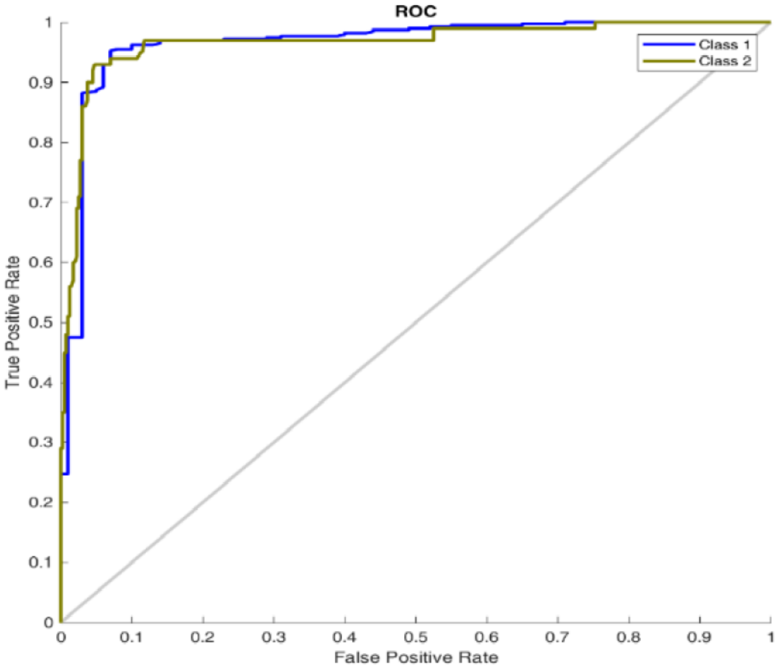

Figure 4.60: ROC Plot for Grey Wolf 1 Features 25 Search Agent

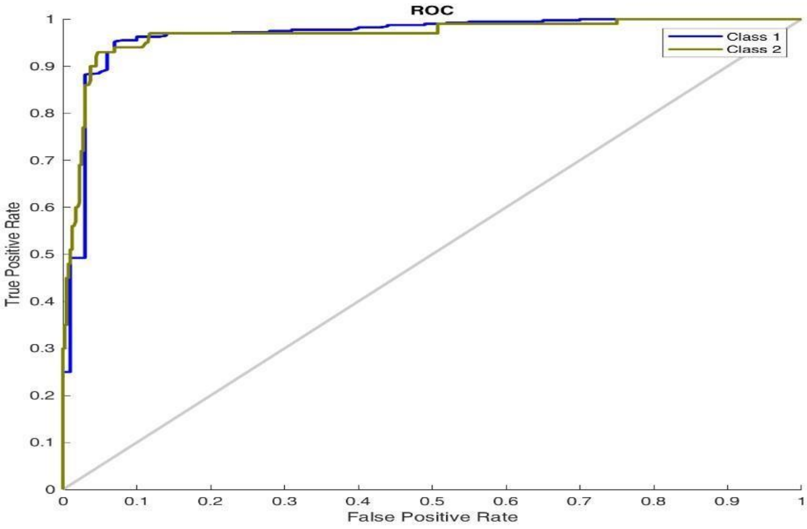

Figure 4.61: ROC Plot for Grey Wolf 1 Features 30 Search Agent

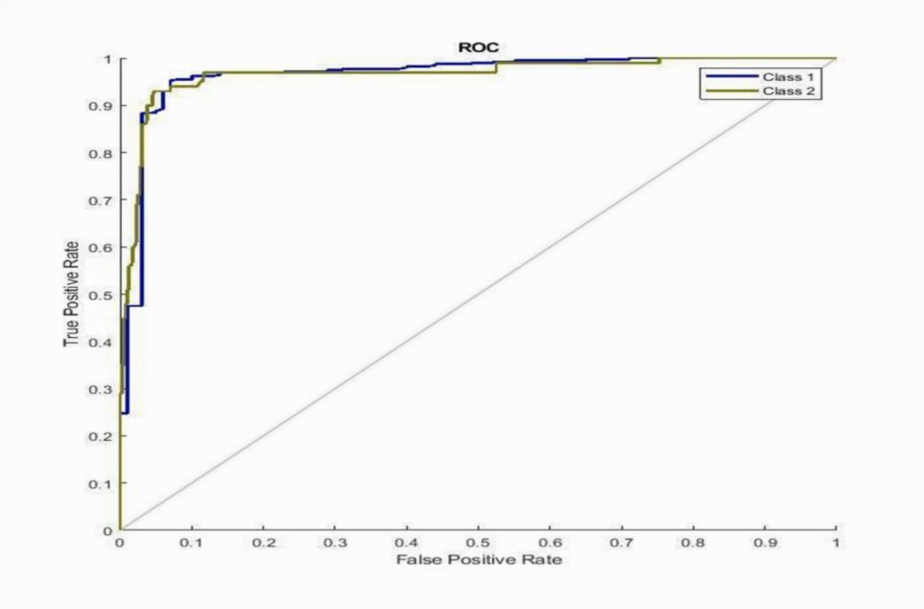

Figure 4.62: ROC Plot for Grey Wolf 1 Features 35 Search Agent

4.6.2 Epilepsy classification using three features and 5, 10, 15, 20, 25, 30, and 35 search agents

This subsection presents the result of epilepsy detection and classification using the Grey

Wolf optimization algorithm with an artificial neural network for three feature extraction and 5, 10, 15, 20, 25, 30, and 35 search agents respectively. Tables 4.28 and 4.29 present the evaluation metrics for Grey Wolf and ANN using three features and 5, 10, 15, 20, 25, 30, and 35 search agents respectively. The classifier accurately identified the real negative classes, but mistakenly anticipated the 14, 14, 13, 11, and 16 false-negative classes, and 0 true-negative classes.

Table 4.29 presents the F1 score, recall, accuracy, precision, and AUC. It shows that the accuracy increases as the search agent increases, except for the 30-search agent. The best accuracy is achieved at 97.80%, with 84.97% precision, 98.60 sensitivity, F1 score of 91.28%, and 94.20% AUC respectively. It is observed from Figure 4.67 that the area under the ROC curve (AUC) for 25 search agents perform better than the rest search agent, indicating a strong performance of the Grey Wolf-ANN classifier using twenty-five search agents. The ROCs of various cases obtained using the Grey Wolf-ANN classifier are shown in Figures 4.63 to 4.69 for the representation of the obtained results for 5, 10, 15, 20, 25, 30, and 35 search agents respectively.

Table 4.28: Grey-Wolf Metrics Using Three-Feature Extraction

Feature Extraction	**Number of Search Agents**	**TP**	**FP**	**TN**	**FN**	**Time** (s)
[1 2 3]	5	400	14	86	0	45.806
[1 2 3]	10	400	14	86	0	149.664
[1 2 3]	15	400	13	87	0	174.822
[1 2 3]	20	400	18	82	0	237.378
[1 2 3]	25	400	11	89	0	269.345
[1 2 3]	30	400	16	84	0	453.034
[1 2 3]	35	400	13	87	0	368.939

Table 4.29: Grey-Wolf Performance Evaluation using Three Feature Extraction

Feature Extraction	**Number of Search Agents**	**Accuracy**	**Precision**	**Sensitivity**	**F1-Score**	**AUC**
[1 2 3]	5	97.2	0.9336	0.9336	0.9581	0.9700
[1 2 3]	10	97.2	0.9336	0.9336	0.9581	0.9700
[1 2 3]	15	97.4	0.9520	0.9780	0.9648	0.9710
[1 2 3]	20	96.4	0.9121	0.9920	0.9504	0.9690
[1 2 3]	**25**	**97.8**	**0.8497**	**0.9860**	**0.9128**	**0.9420**

[1 2 3]	35	97.4	0.9504	0.9800	0.9650	0.9720

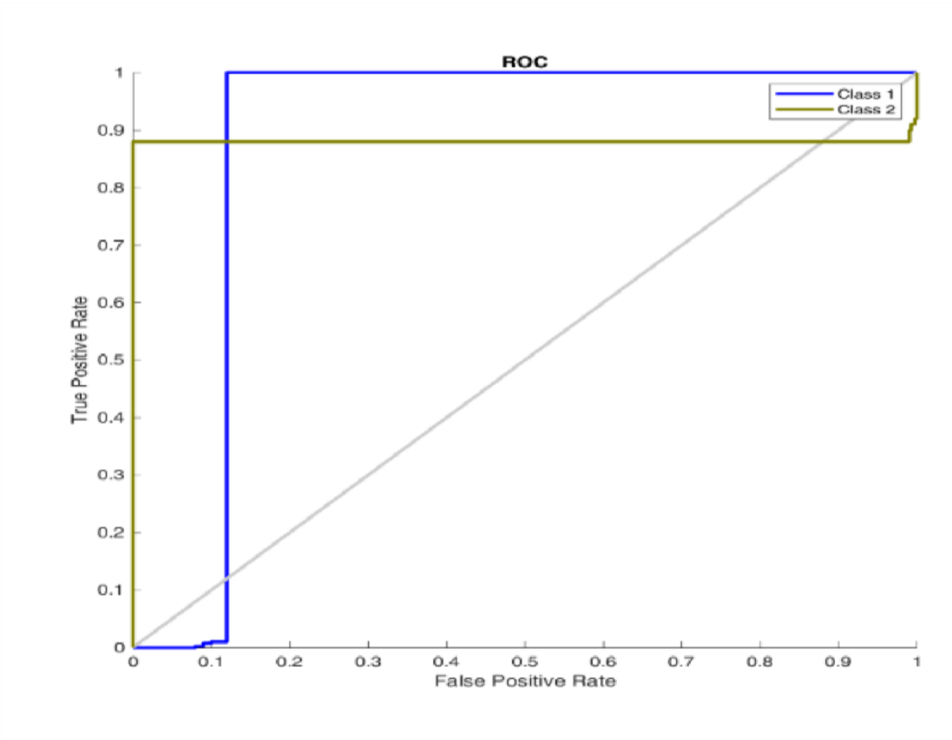

Figure 4.63: ROC Plot for Grey Wolf 3 Features 5 Search Agent

[1 2 3]	30	96.8	0.8392	0.9940	0.9101	0.9430

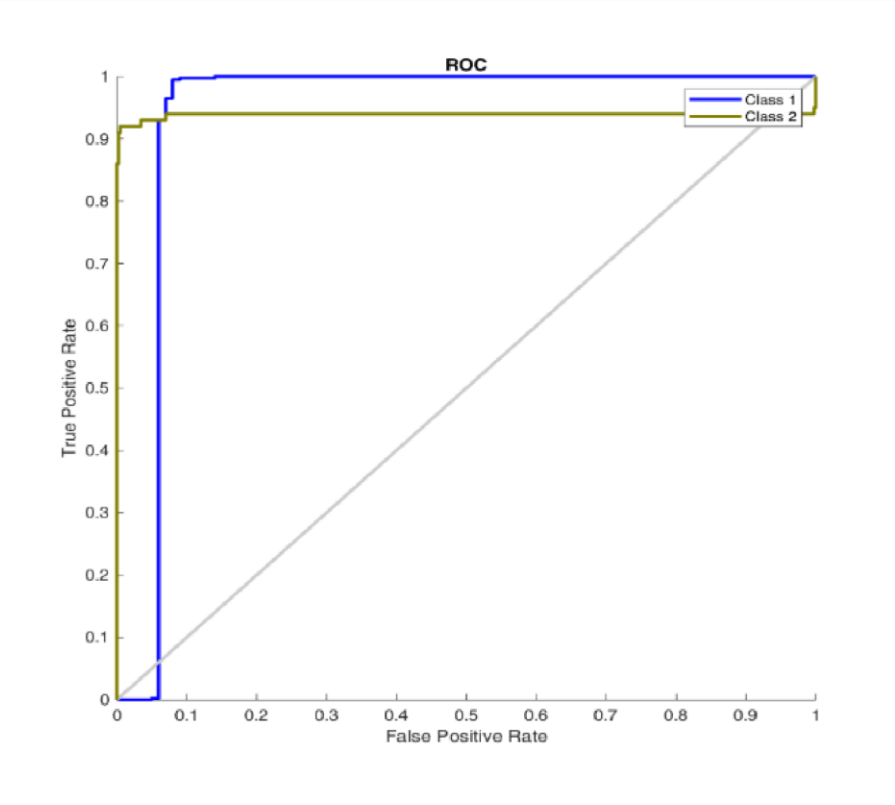

Figure 4.64: Plot for Grey Wolf 3 Feature 10 Search Agent

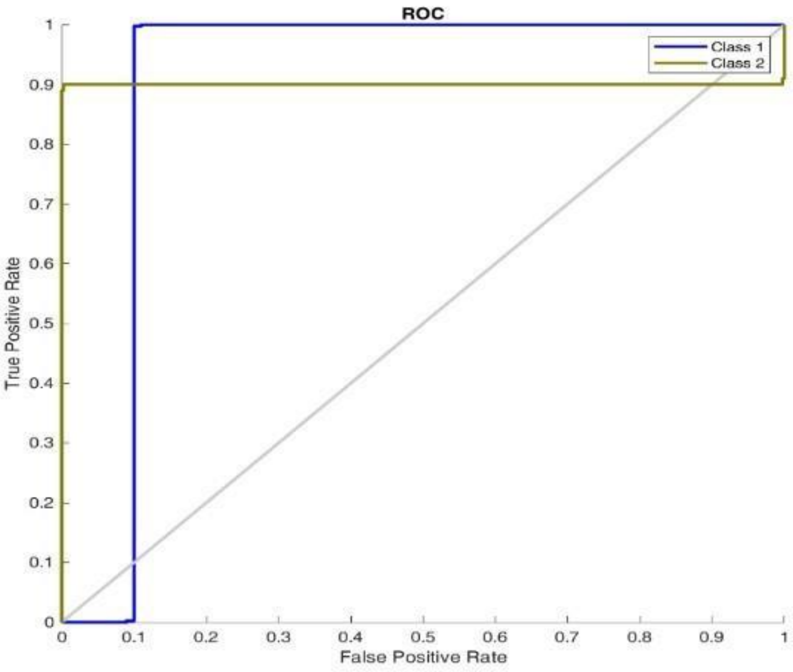

Figure 4.65: ROC Plot for Grey Wolf 3 Features 15 Search Agent

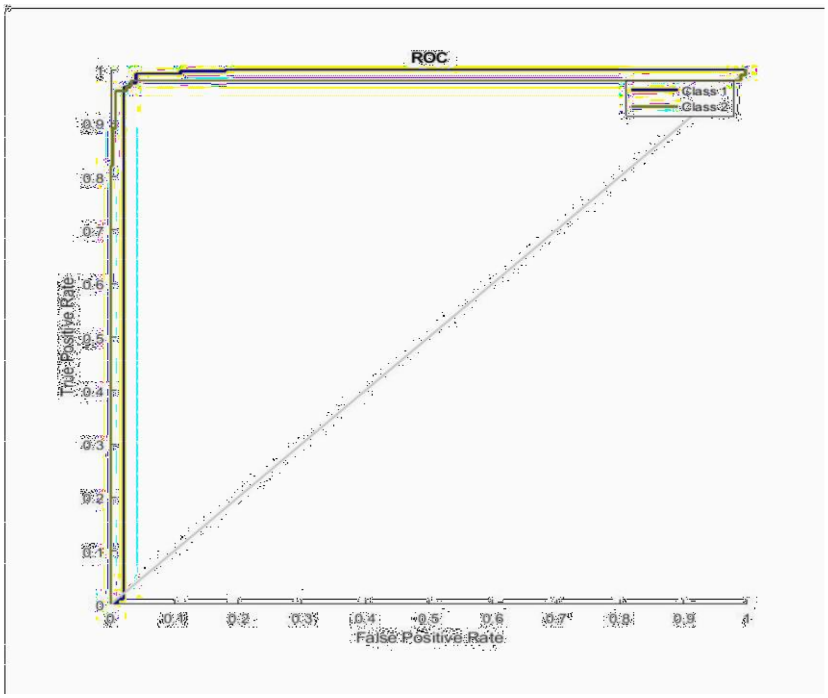

Figure 4.66: ROC Plot for Grey Wolf 3 Features 20 Search Agent

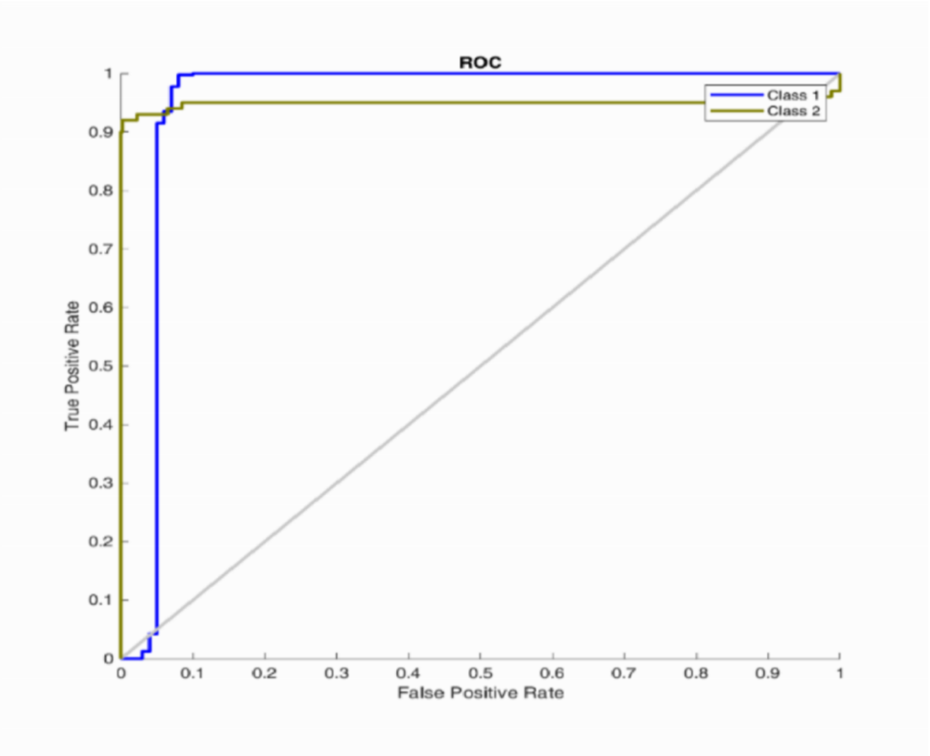

Figure 4.67: ROC Plot for Grey Wolf 3 Features 25 Search Agent

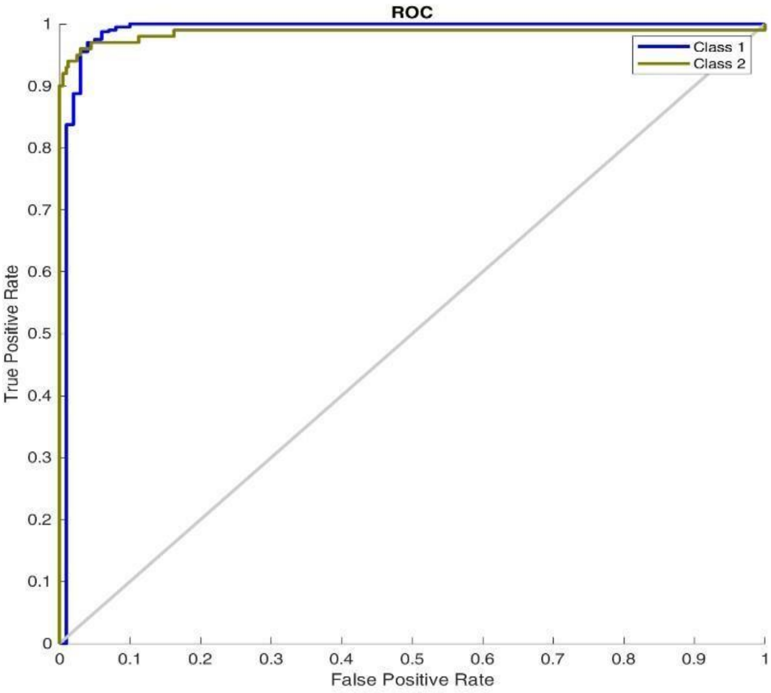

Figure 4.68: ROC Plot for Grey Wolf 3 Features 30 Search Agent

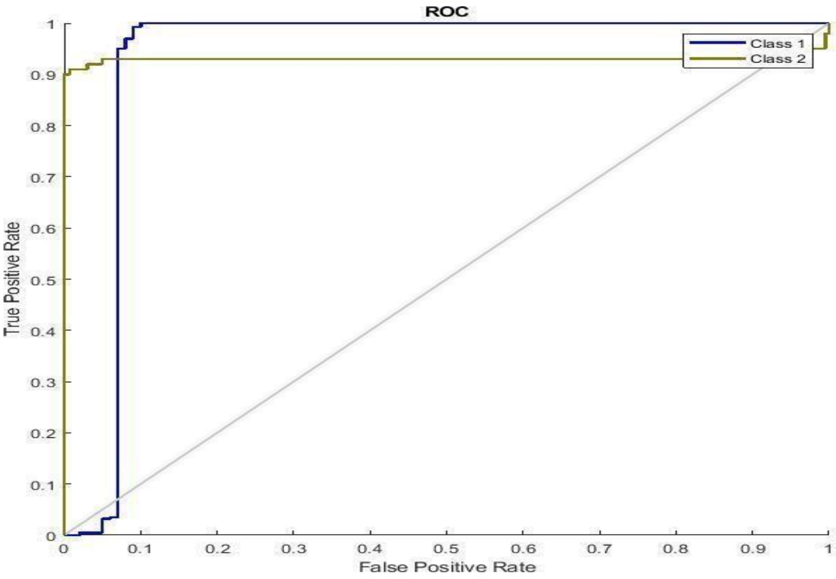

Figure 4.69: ROC Plot for Grey Wolf 3 Features 35 Search Agent

4.6.3 Epilepsy classification using five features and 5, 10, 15, 20, 25, 30, and 35 search agents Tables 4.30 and 4.31 present the evaluation metrics for Grey Wolf and ANN using five features and 5, 10, 15, 20, 25, 30, and 35 search agents respectively. The classifier accurately identified the real negative classes, but mistakenly anticipated the 12, 12, 15, 23, and 19 false-negative classes and 0 genuine negative classes.

Table 4.31 presents the accuracy, precision, recall, F1 score, and AUC. It shows that the accuracy varies between 97.60% and 96.20% as the search agent increases. The best accuracy is achieved at 97.60%, with 96.86% precision, 98.00 sensitivity, and an F1 score of 97.42% and 97.70% AUC respectively. It is observed from Figure 4.71 that the area under the ROC curve (AUC) for 10 search agents perform better than the rest search agent, indicating a strong performance of the Grey Wolf-ANN classifier using ten search agents.

The ROCs of various cases obtained using the Grey Wolf-ANN classifier are shown in Figures 4.70 to 4.76 for the representation of the obtained results for 5, 10, 15, 20, 25, 30, and 35 search agents respectively.

Table 4.30: Grey-Wolf Metrics Using Five-Feature Extraction

Feature Extraction	**Number of Search Agents**	**TP**	**FP**	**TN**	**FN**	**Time** (s)
[1 2 3 4 5]	5	400	12	88	0	79.803
[1 2 3 4 5]	10	400	12	88	0	78.006
[1 2 3 4 5]	15	400	15	85	0	221.104
[1 2 3 4 5]	20	400	10	90	0	300.374
[1 2 3 4 5]	25	400	23	77	0	494.584
[1 2 3 4 5]	30	400	19	81	0	509.171
[1 2 3 4 5]	35	400	14	86	0	507.870

Table 4.31: Grey-Wolf Performance Evaluation using Five Feature Extraction

Feature Extraction	**Number of Search Agents**	**Accuracy**	**Precision**	**Sensitivity**	**F1-Score**	**AUC**
[1 2 3 4 5]	5	97.6	0.9686	0.9800	0.9742	0.9770
[1 2 3 4 5]	**10**	**97.6**	**0.9686**	**0.9800**	**0.9742**	**0.9770**
[1 2 3 4 5]	15	97.0	0.9629	0.9780	0.9704	0.9740
[1 2 3 4 5]	20	97.0	0.9073	0.9900	0.9469	0.9660
[1 2 3 4 5]	25	95.4	0.9043	0.9660	0.9341	0.9480
[1 2 3 4 5]	30	96.2	0.9495	0.9680	0.9387	0.9630

[1 2 3 4 5]	35	97.2	0.9256	0.9860	0.9548	0.9690

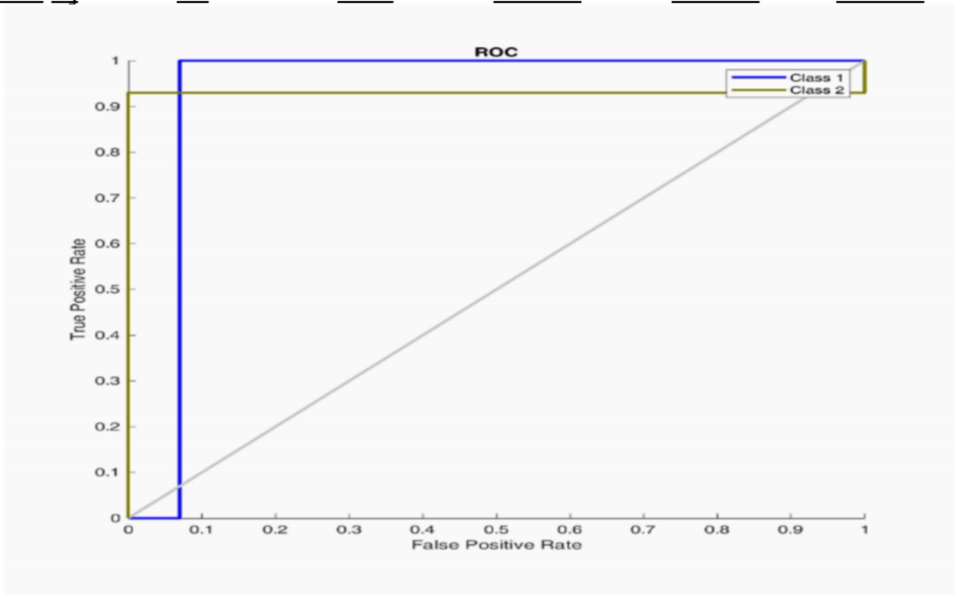

Figure 4.70: ROC Plot for Grey Wolf 5 Features 5 Search Agent

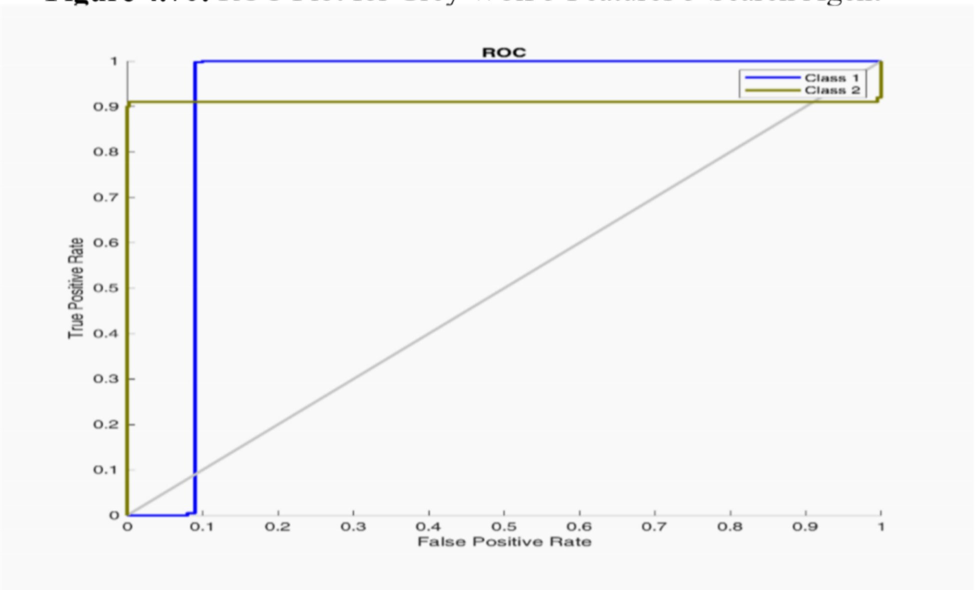

Figure 4.71: ROC Plot for Grey Wolf 5 Features 10 Search Agent

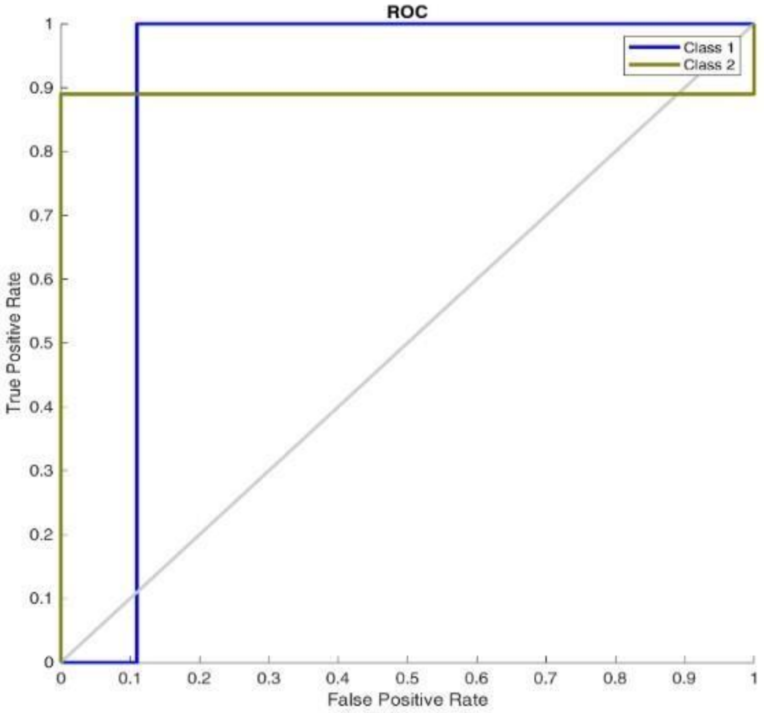

Figure 4.72: ROC Plot for Grey Wolf 5 Features 15 Search Agent

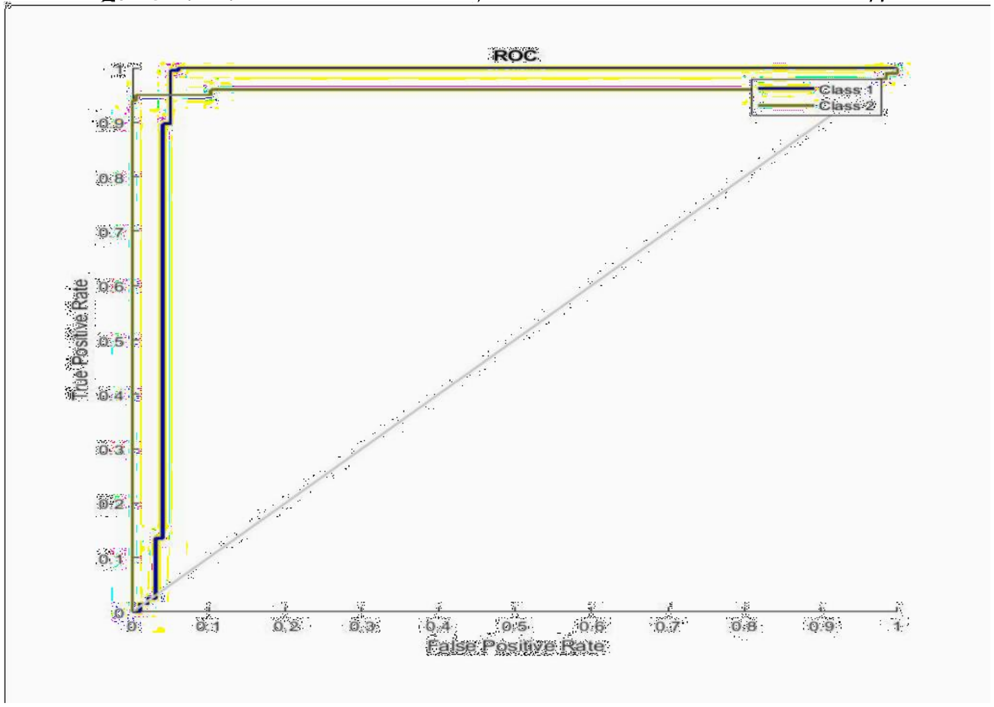

Figure 4.73: ROC Plot for Grey Wolf 5 Features 20 Search Agent

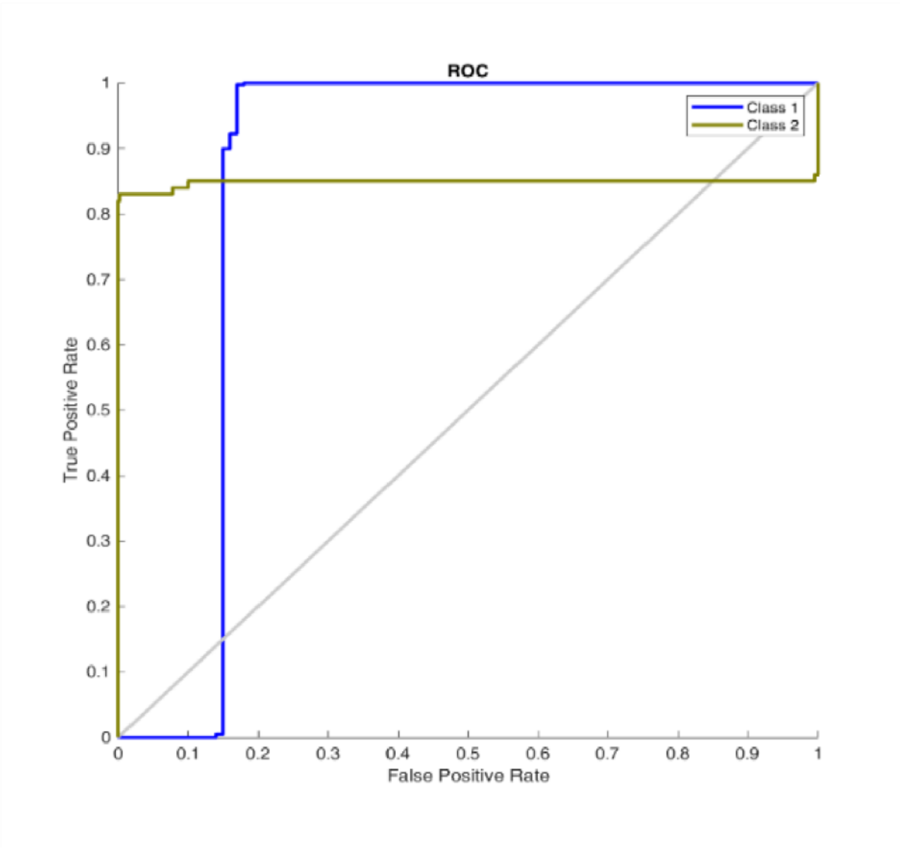

Figure 4.74: ROC Plot for Grey Wolf 5 Features 25 Search Agent

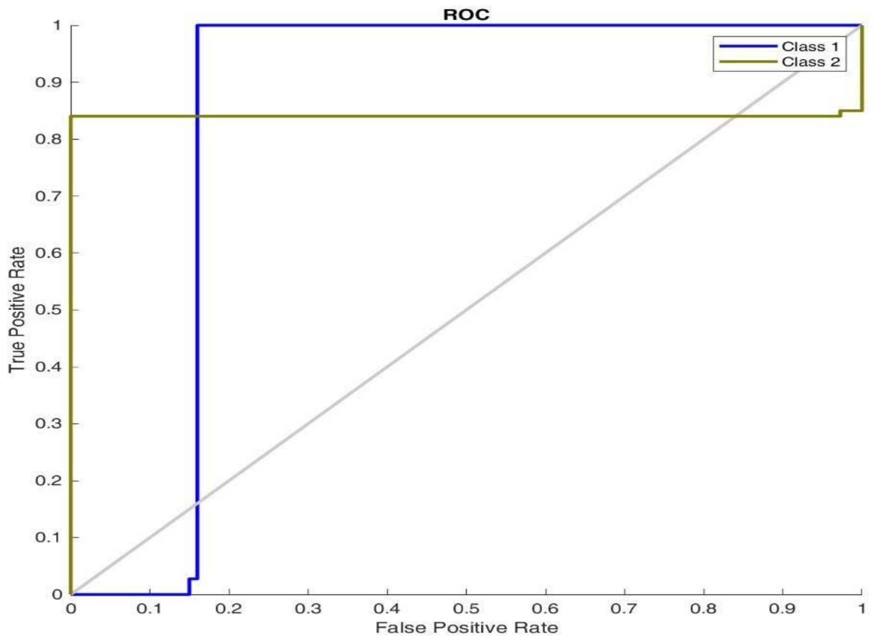

Figure 4.75: ROC Plot for Grey Wolf 5 Features 30 Search Agent

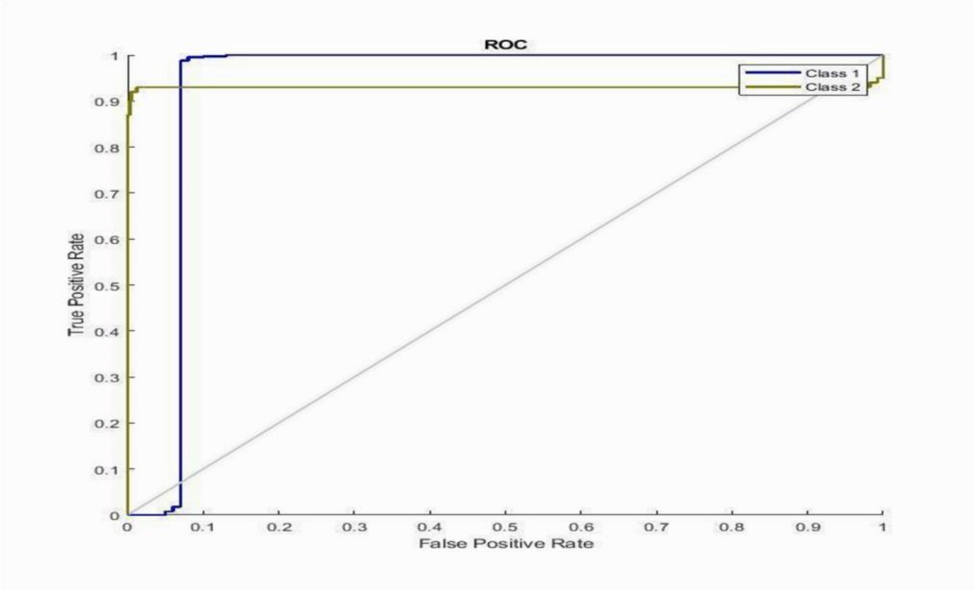

Figure 4.76: ROC Plot for Grey Wolf 5 Features 35 Search Agent

4.6.4 Epilepsy classification using seven features and 5, 10, 15, 20, 25, 30, and 35 search agents

Tables 4.32 and 4.33 present the evaluation metrics for Grey Wolf and ANN using seven features and 5, 10, 15, 20, 25, 30, and 35 search agents respectively. The classifier accurately identified the true negative classes, but mistakenly predicted the false-negative classes 8, 10, 9, 11, and 17 and 0 false negatives, respectively.

Table 4.33 presents the F1 score, recall, accuracy, precision, and AUC. It shows that the accuracy changes from 98.40% to 96.6% as the search agent increases from 5 to 35 respectively. The best accuracy is achieved at 98.40%, with 97.45% precision, 98.60 sensitivity, and F1 score of 98.02% and 98.30% AUC respectively for the five-search agent. It is observed from Figure 4.77 that the area under the ROC curve (AUC) for 5 search agents perform better than the rest search agent, indicating a strong performance of the Grey WolfANN classifier using five search agents. The ROCs of various cases obtained using the Grey Wolf-ANN classifier are shown in figures 4.77 to 4.83 for the representation of the obtained results for 5, 10, 15, 20, 25, 30, and 35 search agents respectively

Table 4.32: Grey-Wolf Metrics Using Seven-Feature Extraction

Feature Extraction	**Number of Search Agents**	**TP**	**FP**	**TN**	**FN**	**Time** (s)
[1 2 3 4 5 6 7]	5	400	8	92	0	104.322
[1 2 3 4 5 6 7]	10	400	10	90	0	148.772

[1 2 3 4 5 6 7]	15	400	9	91	0	266.121
[1 2 3 4 5 6 7]	20	400	8	92	0	323.882
[1 2 3 4 5 6 7]	25	400	11	89	0	465.337
[1 2 3 4 5 6 7]	30	400	17	83	0	577.265
[1 2 3 4 5 6 7]	35	400	20	80	0	553.670

Table 4.33: Grey-Wolf Performance Evaluation using Seven Feature Extraction

Feature Extraction	Number of Search Agents	Accuracy	Precision	Sensitivity	F1-Score	AUC
[1 2 3 4 5 6 7]	**5**	**98.4**	**0.9745**	**0.9860**	**0.9802**	**0.9830**
[1 2 3 4 5 6 7]	10	98.0	0.9705	0.9820	0.9762	0.9790
[1 2 3 4 5 6 7]	15	98.2	0.9783	0.9860	0.9821	0.9840
[1 2 3 4 5 6 7]	20	98.4	0.9763	0.9860	0.9801	0.9820
[1 2 3 4 5 6 7]	25	97.8	0.9666	0.9780	0.9723	0.9750
[1 2 3 4 5 6 7]	30	96.6	0.9570	0.9720	0.9645	0.9680
[1 2 3 4 5 6 7]	35	96.0	0.9528	0.9640	0.9584	0.9610

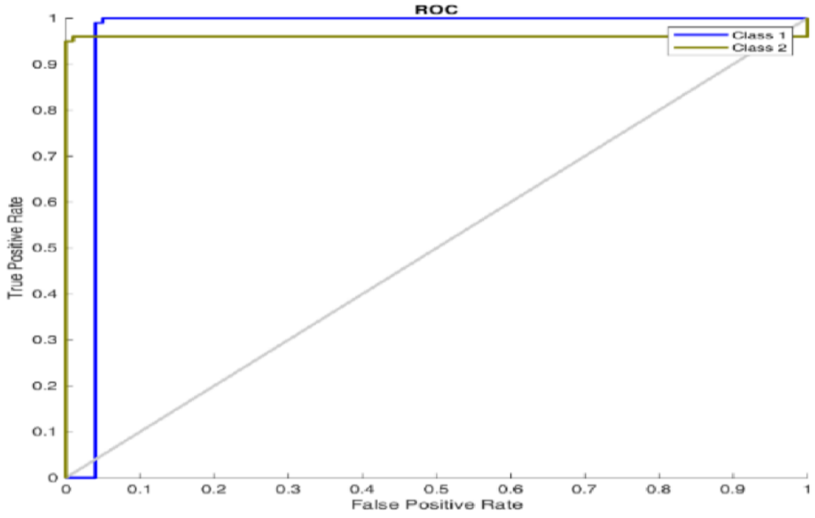

Figure 4.77: ROC Plot for Grey Wolf 7 Features 5 Search Agent

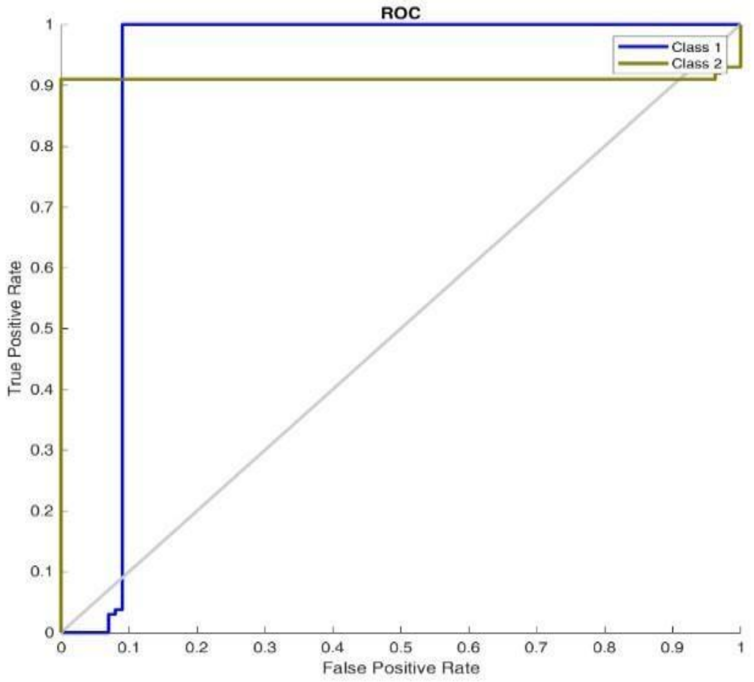

Figure 4.78: ROC Plot for Grey Wolf 7 Features 10 Search Agent

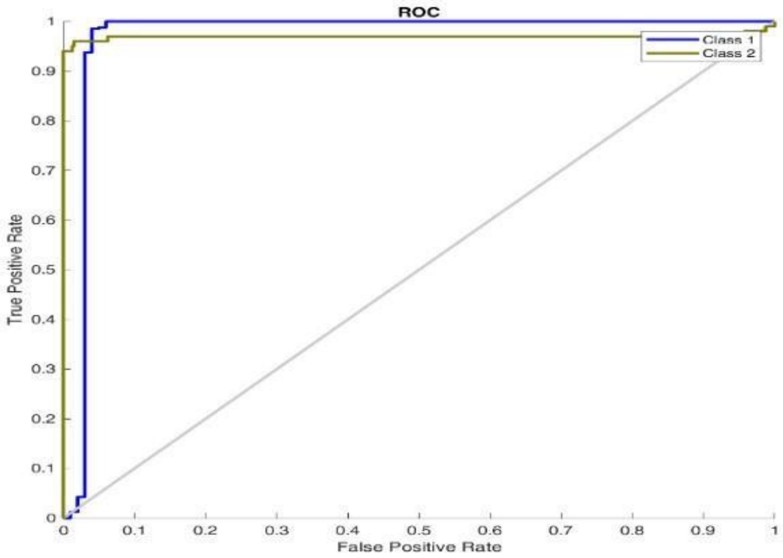

Figure 4.79: ROC Plot for Grey Wolf 7 Features 15 Search Agent

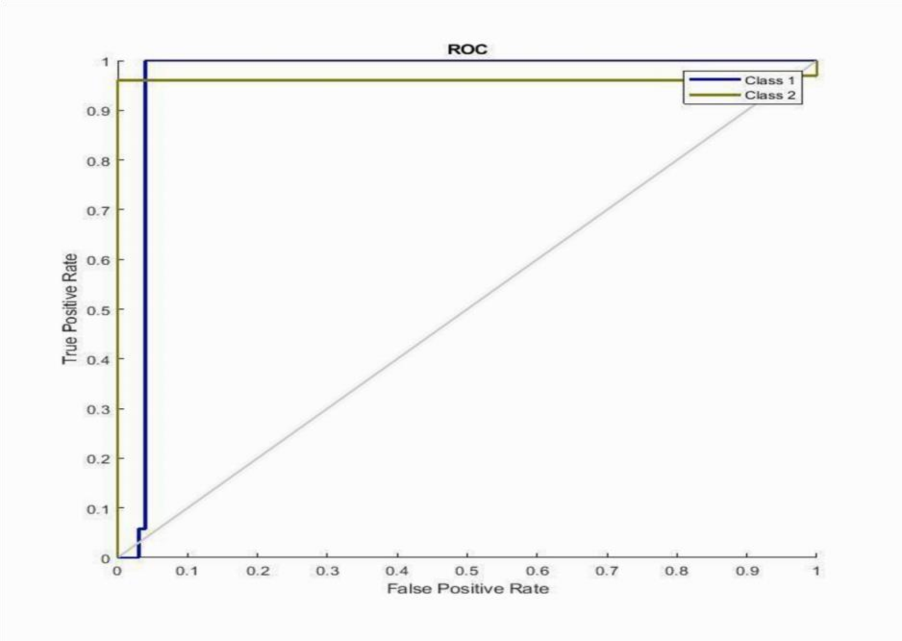

Figure 4.80: ROC Plot for Grey Wolf 7 Features 20 Search Agent

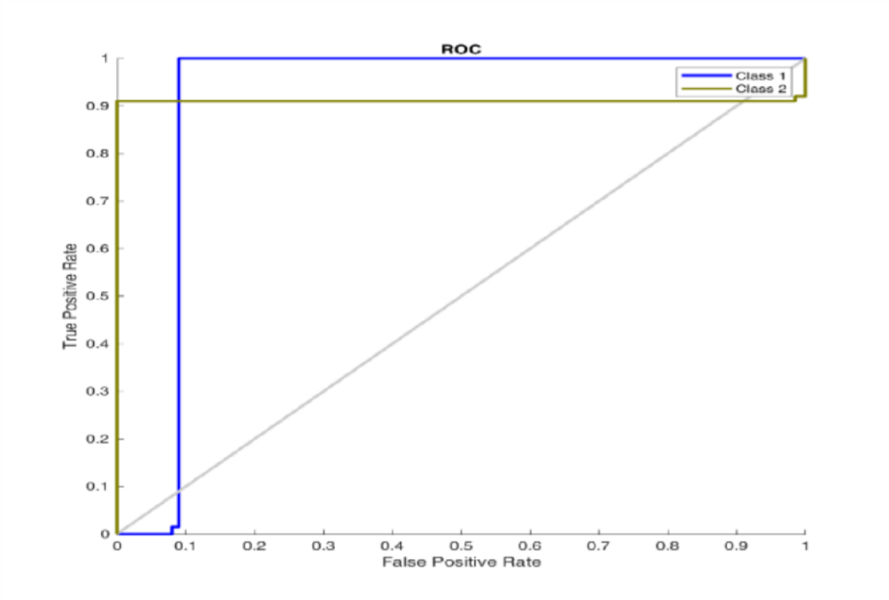

Figure 4.81: ROC Plot for Grey Wolf 7 Features 25 Search Agent

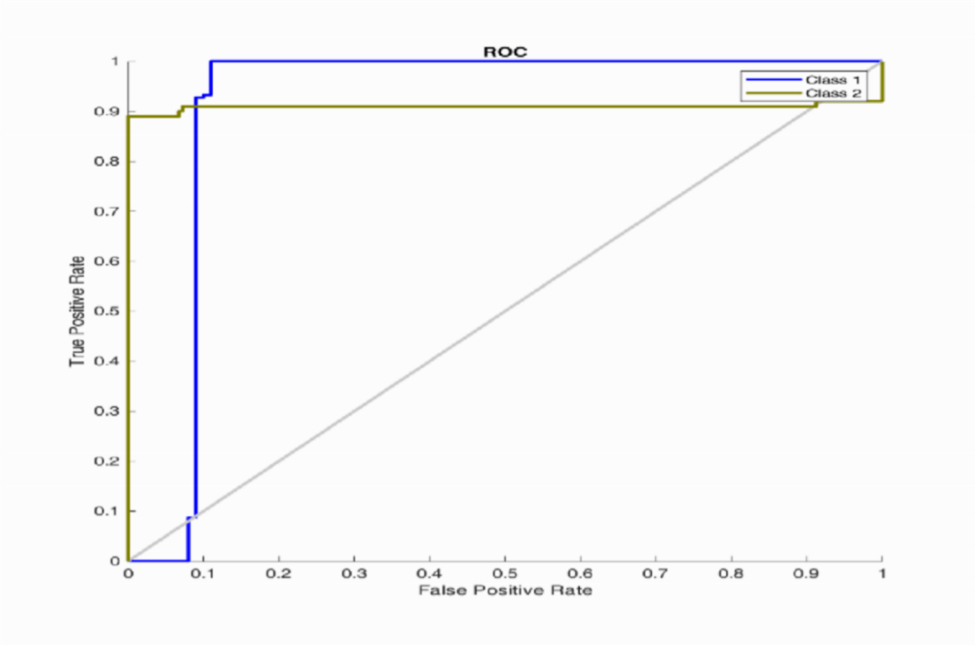

Figure 4.82: ROC Plot for Grey Wolf 7 Features 30 Search Agent

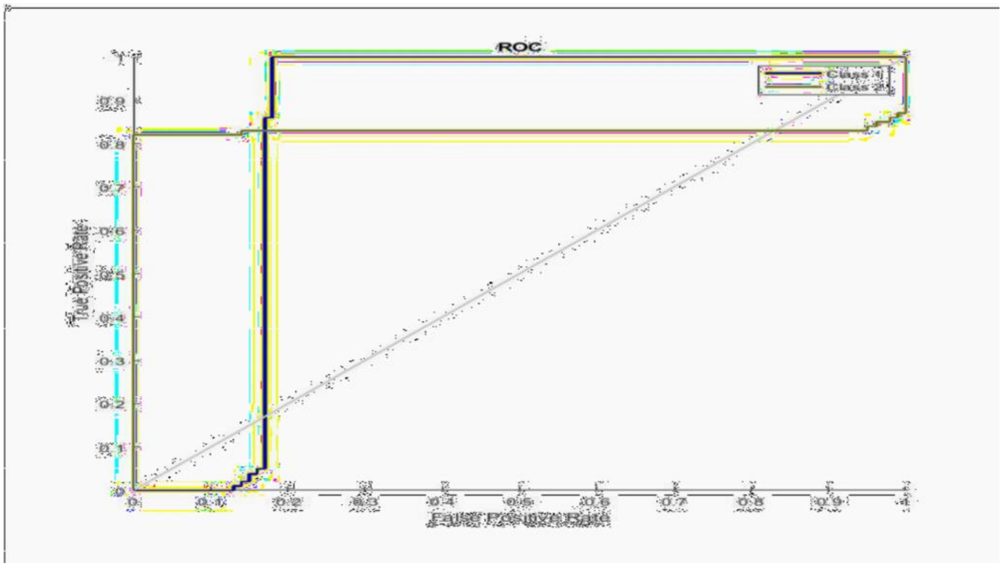

Figure 4.83: ROC Plot for Grey Wolf 7 Features 35 Search Agent

4.6.5 Epilepsy classification using nine features and 5, 10, 15, 20, 25, 30, and 35 search agents

Tables 4.34 and 4.35 present the evaluation metrics for Grey Wolf and ANN using nine features and 5, 10, 15, 20, 25, 30, and 35 search agents respectively. The classifier accurately identified the real negative classes, but mistakenly anticipated the 8, 12, 16, 16, and 12 false-negative classes, and 0 true-negative classes. Table 4.35 presents the F1 score, recall, accuracy, precision, and AUC. It shows that the accuracy varies between 98.40% and 96.60% as the search agent increases. The best accuracy is achieved at 98.40%, with 96.89% precision, 98.80 sensitivity, and an F1 score of 97.84% and 98.30% AUC respectively. It is observed from Figure 4.84 that the ROC curve for 5 search agents performs better than the rest search agent, indicating a strong performance of the Grey Wolf-ANN classifier using five search agents. The ROCs of various cases obtained using the Grey Wolf-ANN classifier are shown in Figures 4.84 to 4.90 for the representation of the obtained results for 5, 10, 15, 20, 25, 30, and 35 search agents respectively.

Table 4.34: Grey-Wolf Metrics Using Nine-Feature Extraction

Feature Extraction	**Number of Search Agents**	**TP**	**FP**	**TN**	**FN**	**Time** (s)
[1 2 3 4 5 6 7 8 9]	5	400	8	92	0	74.954
[1 2 3 4 5 6 7 8 9]	10	400	12	88	0	85.578
[1 2 3 4 5 6 7 8 9]	15	400	16	84	0	156.995
[1 2 3 4 5 6 7 8 9]	20	400	13	87	0	280.301
[1 2 3 4 5 6 7 8 9]	25	400	16	84	0	217.940
[1 2 3 4 5 6 7 8 9]	30	400	12	88	0	346.818
[1 2 3 4 5 6 7 8 9]	35	400	9	91	0	467.473

Table 4.35: Grey-Wolf Performance Evaluation using Nine Feature Extraction

Feature Extraction	**Number of Search Agents**	**Accuracy**	**Precision**	**Sensitivity**	**F1-Score**	**AUC**
[1 2 3 4 5 6 7 8 9]	**5**	**98.4**	**0.9689**	**0.9880**	**0.9784**	**0.9830**
[1 2 3 4 5 6 7 8 9]	10	97.6	0.9800	0.9724	0.9724	0.9760
[1 2 3 4 5 6 7 8 9]	15	96.8	0.9469	0.9800	0.9631	0.9710
[1 2 3 4 5 6 7 8 9]	20	97.4	0.9612	0.9800	0.9705	0.9750
[1 2 3 4 5 6 7 8 9]	25	96.8	0.9482	0.9740	0.9609	0.9670
[1 2 3 4 5 6 7 8 9]	30	96.6	0.9722	0.9760	0.9741	0.9750
[1 2 3 4 5 6 7 8 9]	35	98.2	0.9763	0.9840	0.9801	0.9820

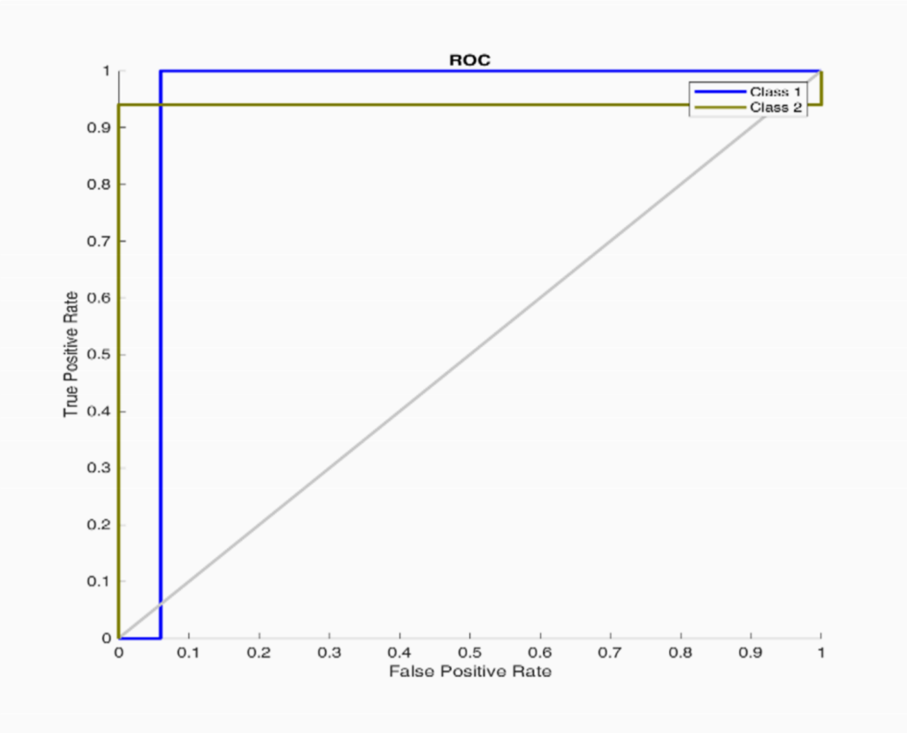

Figure 4.84: ROC Plot for Grey Wolf 9 Features 5 Search Agent

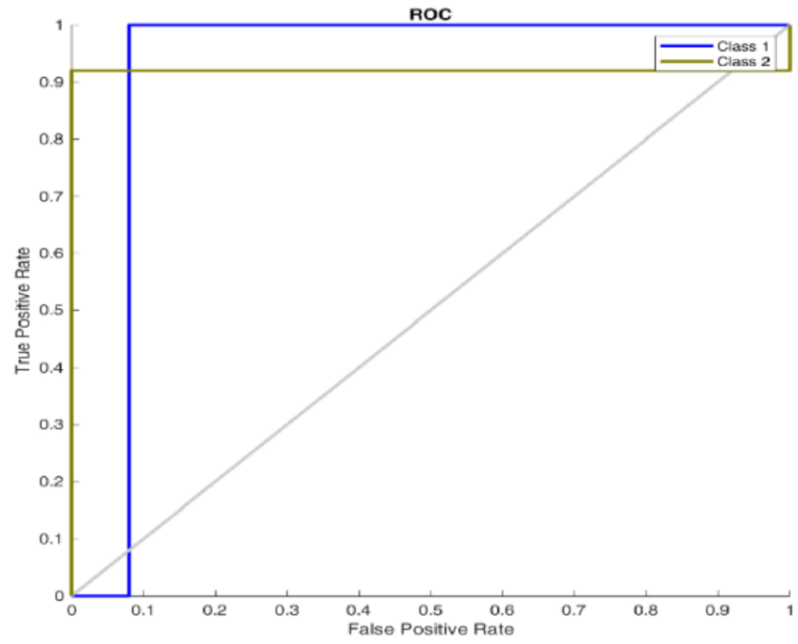

Figure 4.85: ROC Plot for Grey Wolf 9 Features 10 Search Agent

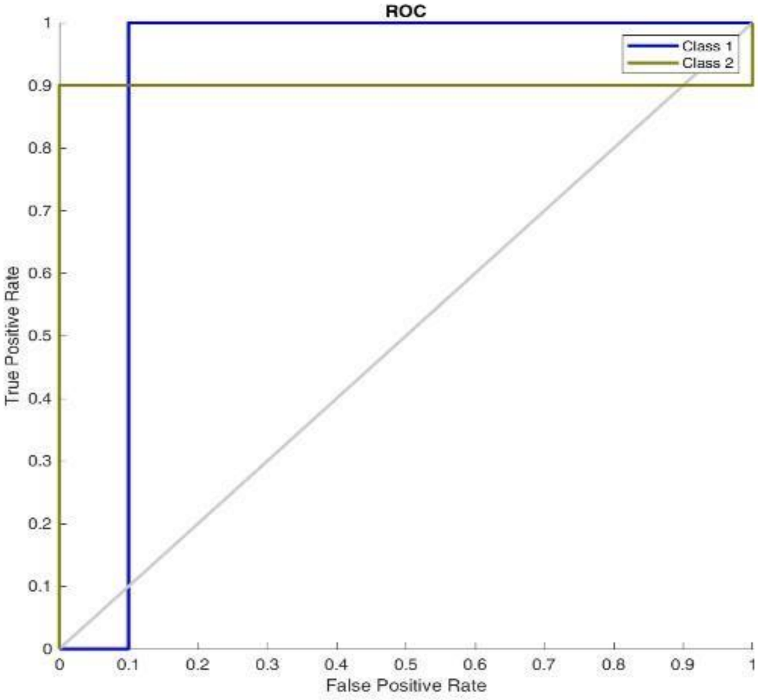

Figure 4.86: ROC Plot for Grey Wolf 9 Features 15 Search Agent

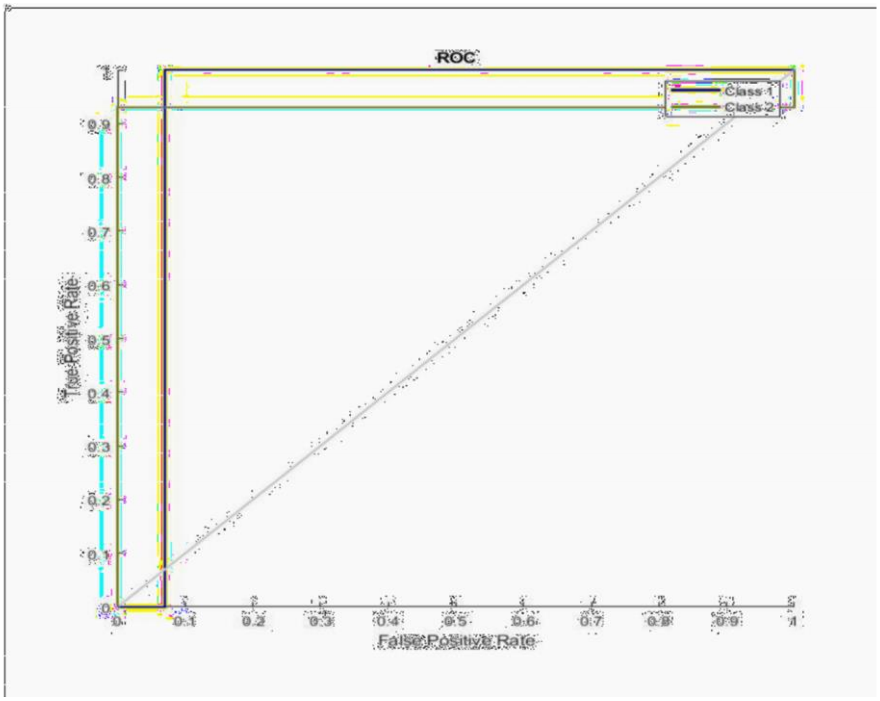

Figure 4.87: ROC Plot for Grey Wolf 9 Features 20 Search Agent

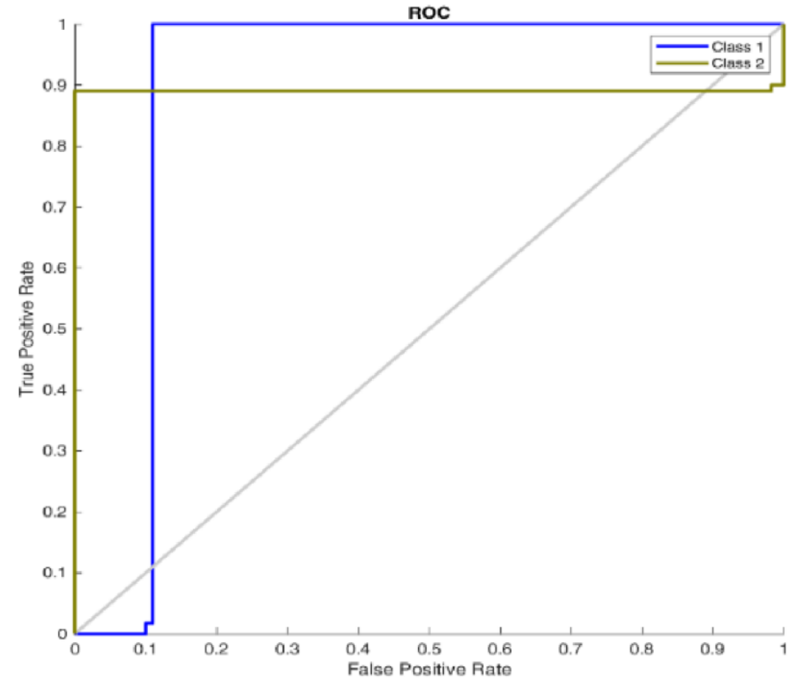

Figure 4.88: ROC Plot for Grey Wolf 9 Features 25 Search Agent

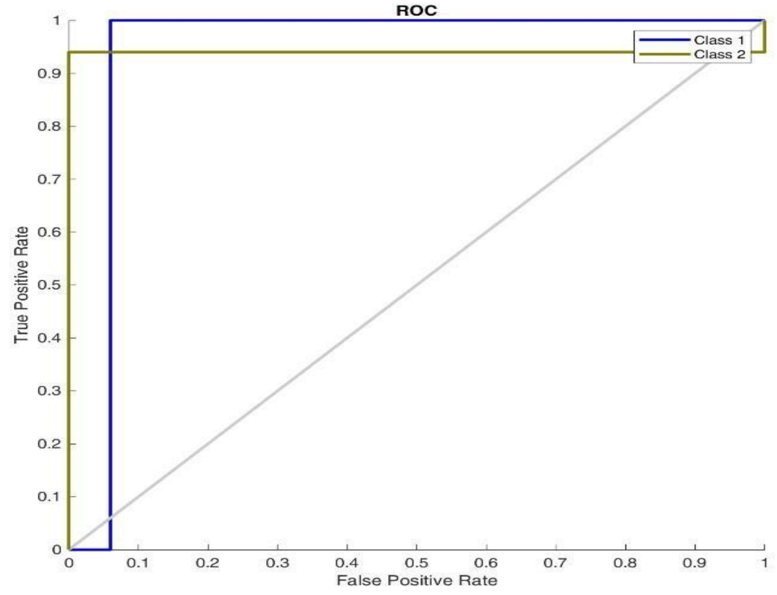

Figure 4.89: ROC Plot for Grey Wolf 9 Features 30 Search Agent

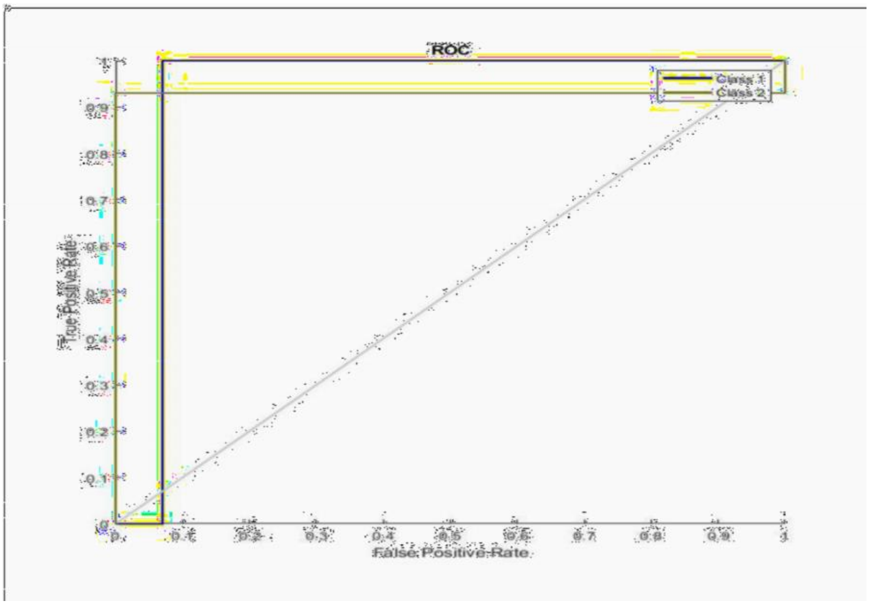

Figure 4.90: ROC Plot for Grey Wolf 9 Features 35 Search Agent

Table 4.36 present the evaluation methods for GWO-ANN using one, three, five, seven, and nine features respectively for 5, 10, 15, 20, 25, 30, and 35 search agents. The F1 score, recall, accuracy, precision, and AUC. for various feature classes are shown as well. It shows that the accuracy increases as the number of features increase from one to nine of the GWOANN classifiers. The best accuracy is achieved at 98.40%, with 97.45% precision, 98.60% sensitivity, and an F1 score of 98.02% and 98.30% AUC respectively for 30 search agents, and seven feature extractions for the GWO-ANN algorithm. It is observed from Figure 4.91 that the accuracy is near 100%, indicating a strong performance of the GWOANN classifier using seven feature extractions. In the real classes, the classifier performs well, and vice versa for seven features when compared to other classes of the feature's extraction.

Table 4.36: Summary of Epilepsy Detection Using Grey Wolf with Various Search Agents

Evaluation Metrics	**1 Feature 30 Search Agent**	**3 Features 15 Search Agent**	**5 Features 25 Search Agent**	**7 Features 30 Search Agent**	**9 Features 25 Search Agent**
Accuracy	0.944	0.978	0.976	**0.984**	**0.984**
Precision	0.7539	0.8497	0.9686	**0.9745**	**0.9689**
Sensitivity	0.986	0.986	0.98	**0.986**	**0.988**

F1 Measure	0.8544	0.9128	0.9742	**0.9802**	**0.9784**
AUC	0.894	0.942	0.977	**0.983**	**0.983**

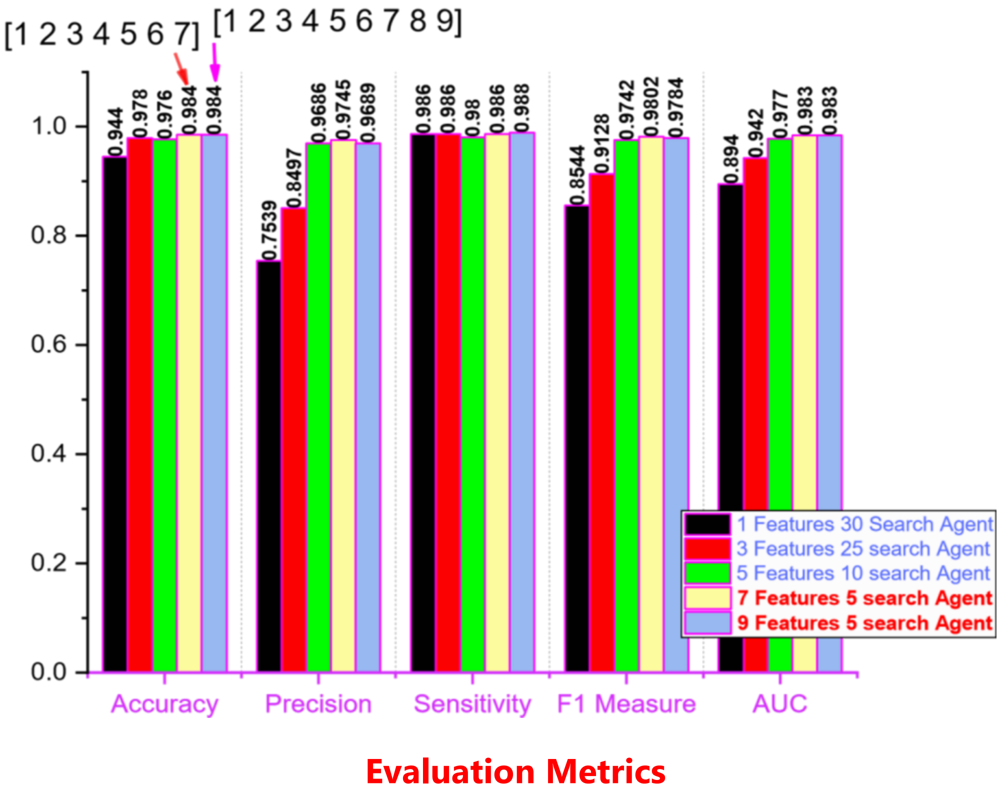

Figure 4.91: Grey Wolf Performance with Various Search Agents

4.7 Epilepsy Classification Using Salp Swarm Optimization Algorithm (SSOA)

This section highlights the results from the detection and classification of epilepsy using the SSOA-ANN method. The different feature extraction ranges used were 1, 3,5, 7, and

9, with corresponding search agents of 5, 10, 15, 20, 25, and 35.

4.7.1. Epilepsy classification using one feature and 5, 10, 15, 20, 25, 30, and 35 search agents

Tables 4.37 and 4.38 are the evaluation methods for SSOA-ANN using one feature and 5, 10, 15, 20, 25, 30, and 35 search agents respectively. In contrast, the classifier inaccurately predicted 53, 83, 16, 17, and 16 of the false-negative classes, as well as 11, 1, 5, and 6 falsenegative classes for search agents 5, 10, 15, 20, 25, and 35, respectively.

Table 4.38 presents the F1 score, recall, accuracy, precision, and AUC. It shows that the accuracy varies between 87.2% to 95.60% as the search agent increases from 5 to 35 for the SSOA-ANN classifier. The best accuracy is achieved at 95.80%, with 79.75% precision, 99.80 sensitivity, and an F1 score of 88.65% and 92.60% AUC respectively. It is observed from Figure 4.94 that the ROC curve for 15 search agents performs better than 5, 10, 15, 20, 25, 30, and 35 respectively, indicating a better performance of the SSOAANN classifier. The ROCs of various cases obtained using the SSOA-ANN classifier are plotted as shown in Figures 4.92 to 4.98 for the representation of the obtained results for 5, 10, 15, 20, 25, 30, and 35 search agents respectively. The classifier seems to function well in the real classes and vice versa.

Table 4.37: Salp-Swarm Metrics Using One Feature Extraction

Feature Extraction	Number of Search Agents	TP	FP	TN	FN	Time (s)
[8]	5	389	53	47	11	36.076
[2]	10	399	17	83	1	65.596
[4]	15	395	16	84	5	73.724
[7]	20	399	24	76	1	63.573
[6]	25	395	17	83	5	73.724
[4]	30	394	16	84	6	73.881
[8]	35	390	54	46	10	51.925

Table 4.38: Salp-Swarm Performance Evaluation Using One Feature Extraction

Feature Extraction	Number of Search Agents	Accuracy	Precision	Sensitivity	F1-Score	AUC
[8]	5	87.2	0.6808	0.9560	0.7953	0.8240
[2]	10	82.0	0.5694	0.9320	0.7069	0.6760
[4]	**15**	**95.8**	**0.7975**	**0.9980**	**0.8865**	**0.9260**
[7]	20	95.0	0.8353	0.9980	0.9095	0.9440
[6]	25	95.6	0.7627	0.9920	0.8624	0.9030
[4]	30	95.6	0.7975	0.9980	0.8865	0.9260
[8]	35	87.2	0.6808	0.9580	0.7959	0.8250

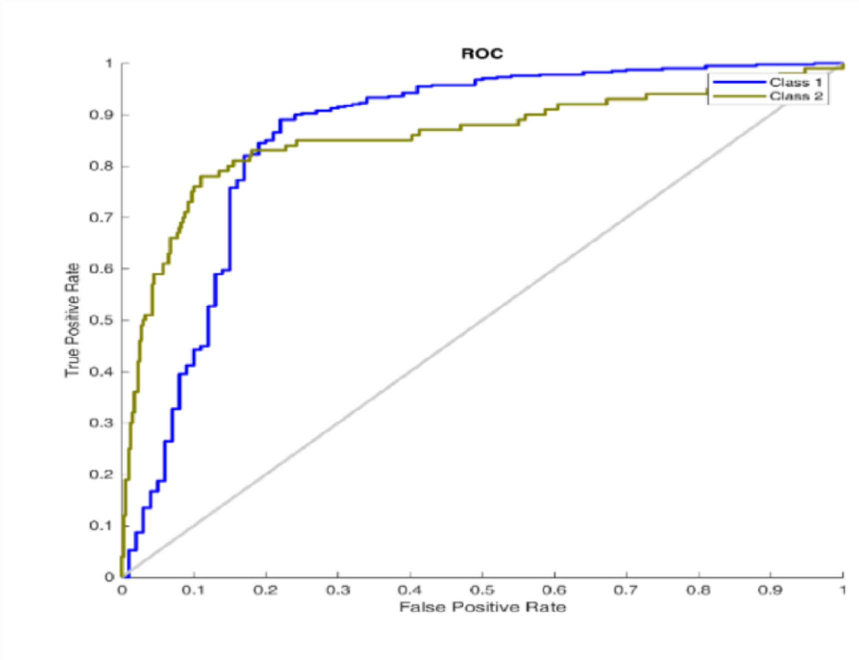

Figure 4.92: ROC Plot for Salp Swarm 1 Features 5 Search Agent

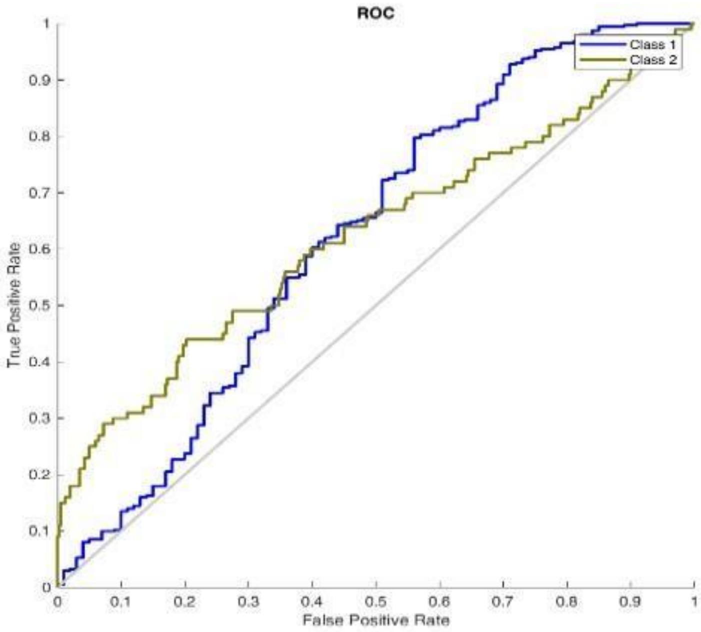

Figure 4.93: ROC Plot for Salp Swarm 1 Features 10 Search Agent

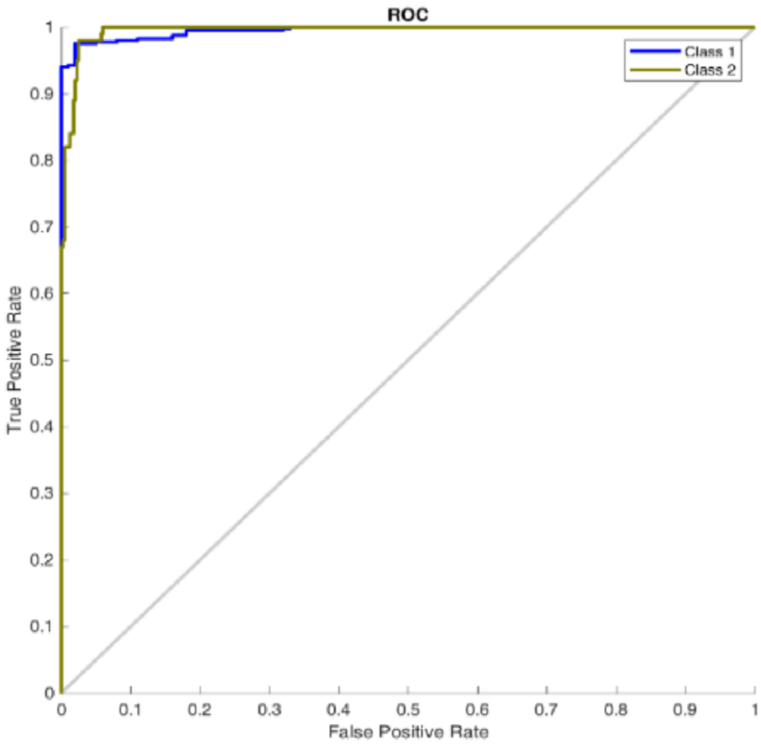

Figure 4.94: ROC Plot for Salp Swarm 1 Features 15 Search Agent

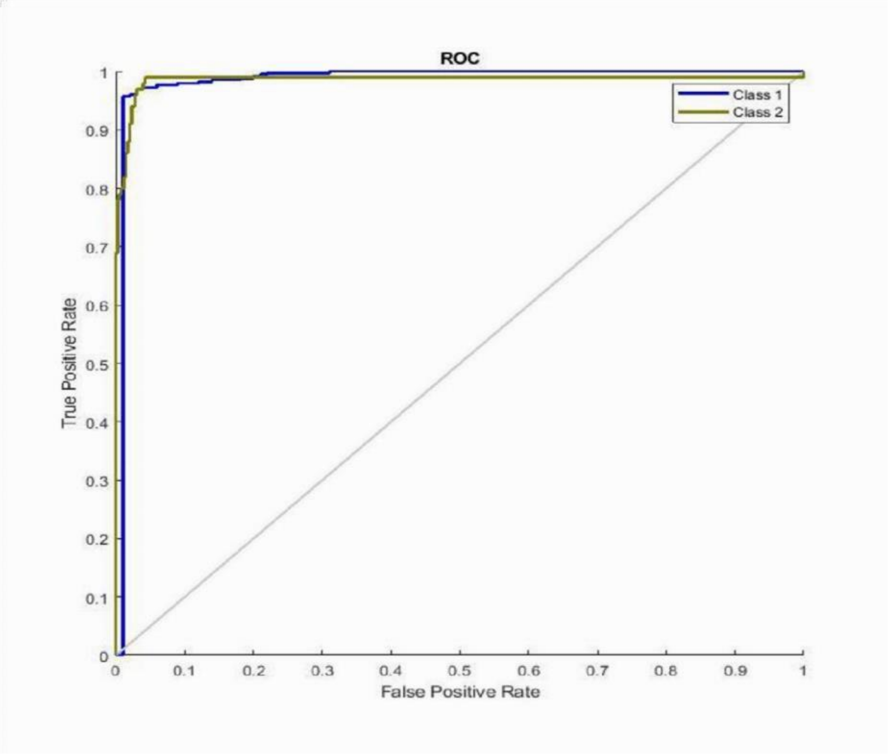

Figure 4.95: ROC Plot for Salp Swarm 1 Features 20 Search Agent

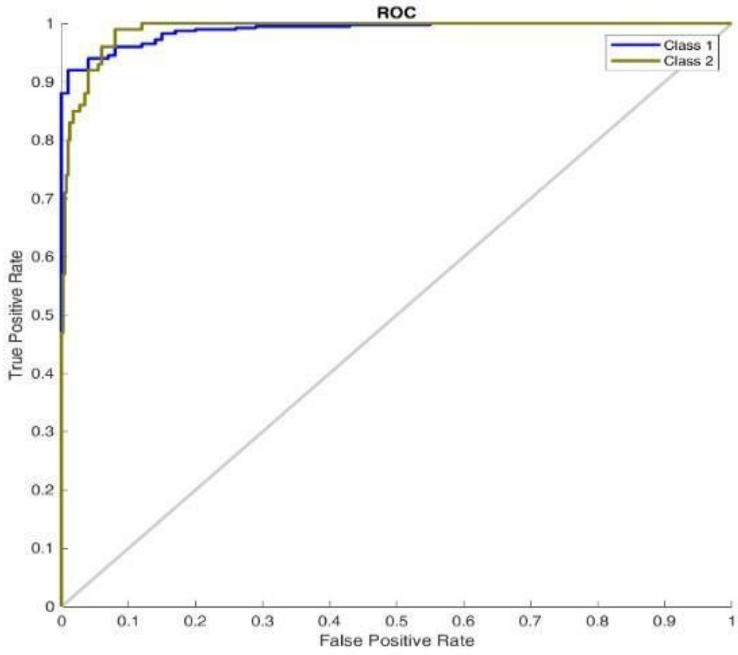

Figure 4.96: ROC Plot for Salp Swarm 1 Features 25 Search Agent

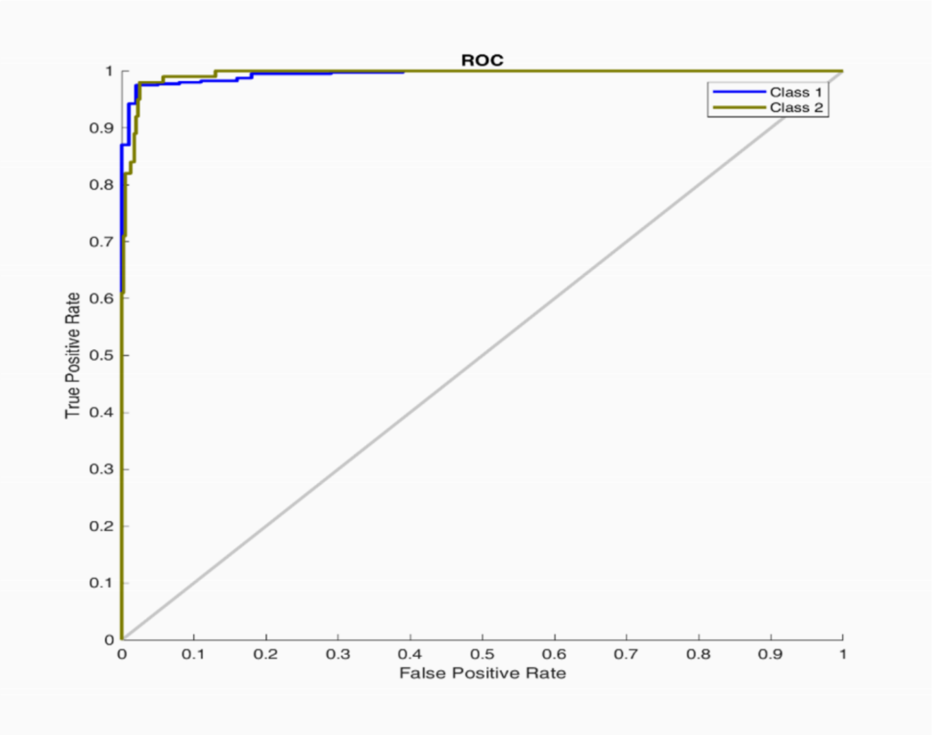

Figure 4.97: ROC Plot for Salp Swarm 1 Features 30 Search Agent

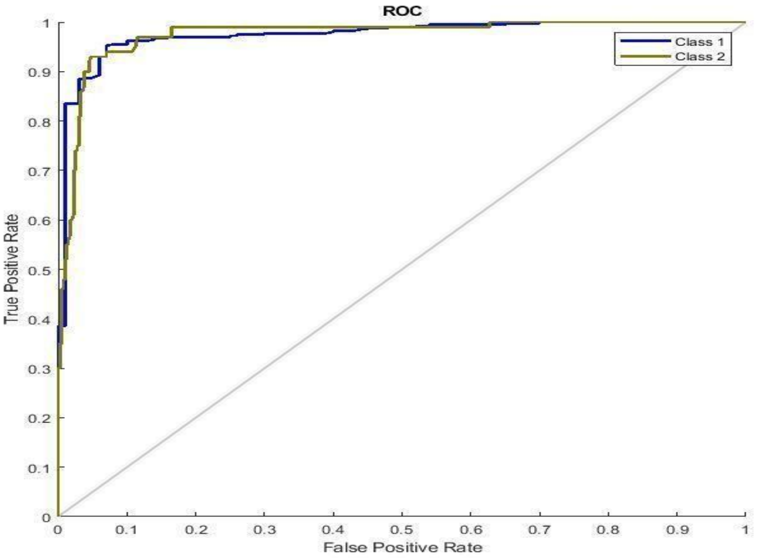

Figure 4.98: ROC Plot for Salp Swarm 1 Features 35 Search Agent

4.7.2 Epilepsy detection and classification using three feature and 5, 10, 15, 20, 25, 30, and 35 search agents

Table 4.40 presents the F1 score, recall, accuracy, precision, and AUC. It shows that the accuracy varies between 97.20% to 98.40% as the search agent increases from 5 to 35 for the SSOA-ANN classifier. The best accuracy is achieved at 98.40%, with 98.40% precision, 98.40 sensitivity, and an F1 score of 98.40% and 98.40% AUC respectively. It is observed from Figure 4.104 that the ROC curve for 30 search agents performs better than 5, 10, 15, 25, and 35 respectively, indicating a better performance of the SSOA-ANN classifier. The ROCs of various cases obtained using the SSOA-ANN classifier are shown in Figures 4.99 to 4.105 for the representation of the obtained results for 5, 10, 15, 20, 25, 30, and 35 search agents respectively.

Table 4.39: Salp-Swarm Metrics Using Three-Feature Extraction

Feature Extraction	Number of Search Agents	TP	FP	TN	FN	Time (s)
[5 3 2]	5	400	14	86	0	80.975
[3 6 9]	10	399	10	90	1	172.672
[1 7 8]	15	400	14	86	0	156.501
[1 4 9]	20	400	13	87	0	80.683
[9 7 0]	25	398	7	93	2	95.250
[8 7 3]	30	400	8	92	0	103.604
[3 7 6]	35	400	10	90	0	93.338

Table 4.40: Salp-Swarm Performance Evaluation Using Three-Feature Extraction

Feature Extraction	Number of Search Agents	Accuracy	Precision	Sensitivity	F1-Score	AUC
[5 3 2]	5	97.2	0.8234	0.9880	0.8982	0.9320
[3 6 9]	10	97.8	0.9112	0.9840	0.9462	0.9630
[1 7 8]	15	97.2	0.9026	0.9880	0.9434	0.9630
[1 4 9]	20	97.4	0.9702	0.9740	0.9721	0.9730
[9 7 0]	25	98.2	0.8601	0.9960	0.9231	0.9530
[8 7 3]	**30**	**98.4**	**0.9840**	**0.9840**	**0.9840**	**0.9840**
[3 7 6]	35	98.0	0.9361	0.9940	0.9642	0.9780

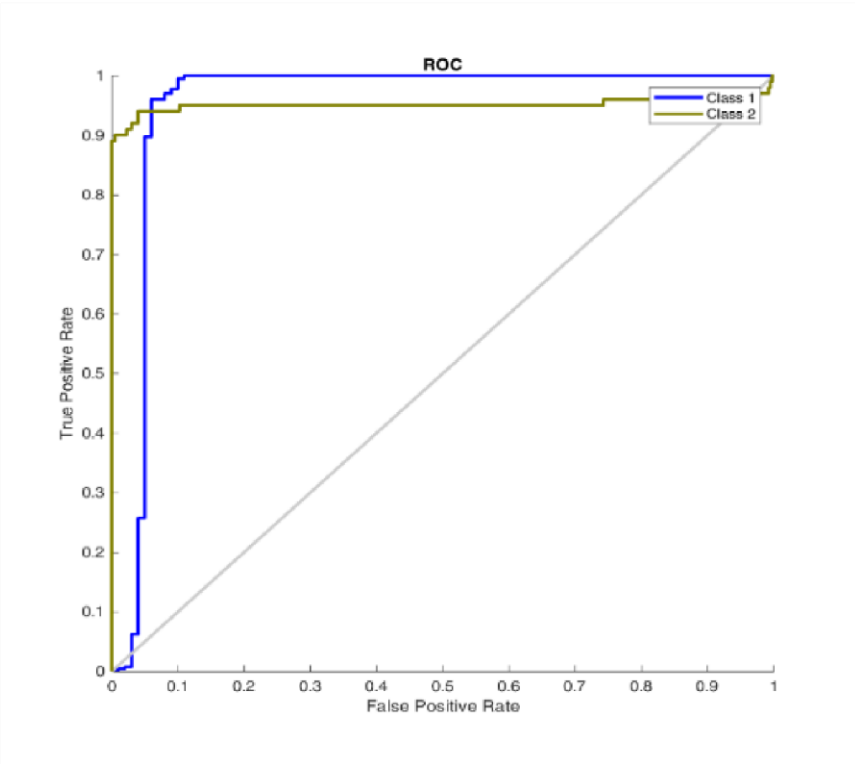

Figure 4.99: ROC Plot for Salp Swarm 3 Features 5 Search Agent

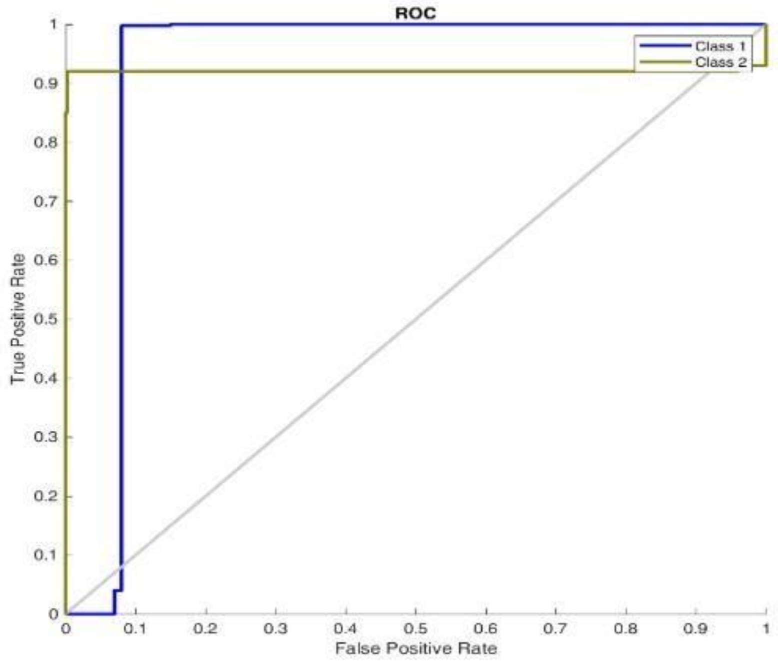

Figure 4.100: ROC Plot for Salp Swarm 3 Features 10 Search Agent

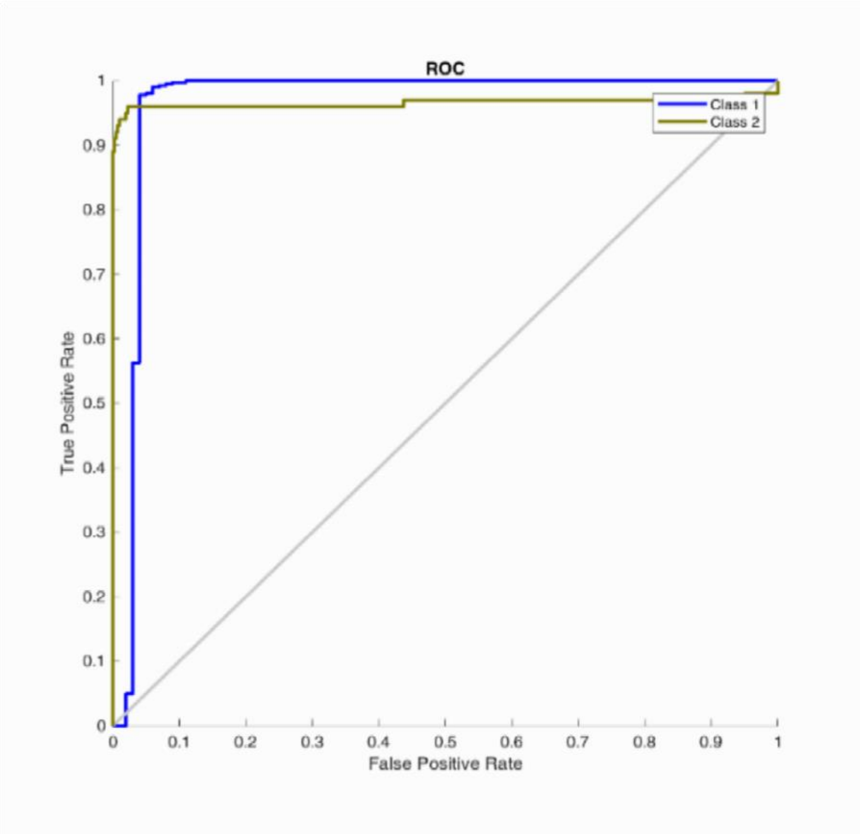

Figure 4.101: ROC Plot for Salp Swarm 3 Features 15 Search Agent

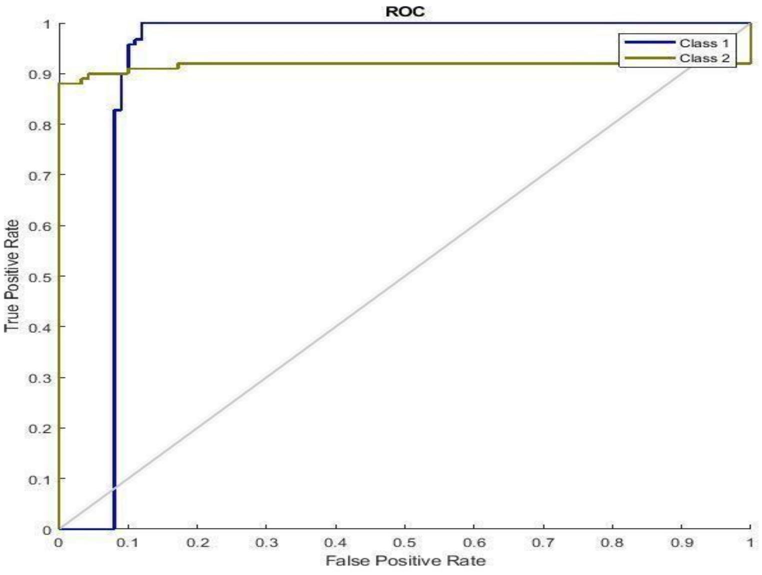

Figure 4.102: ROC Plot for Salp Swarm 3 Features 20 Search Agent

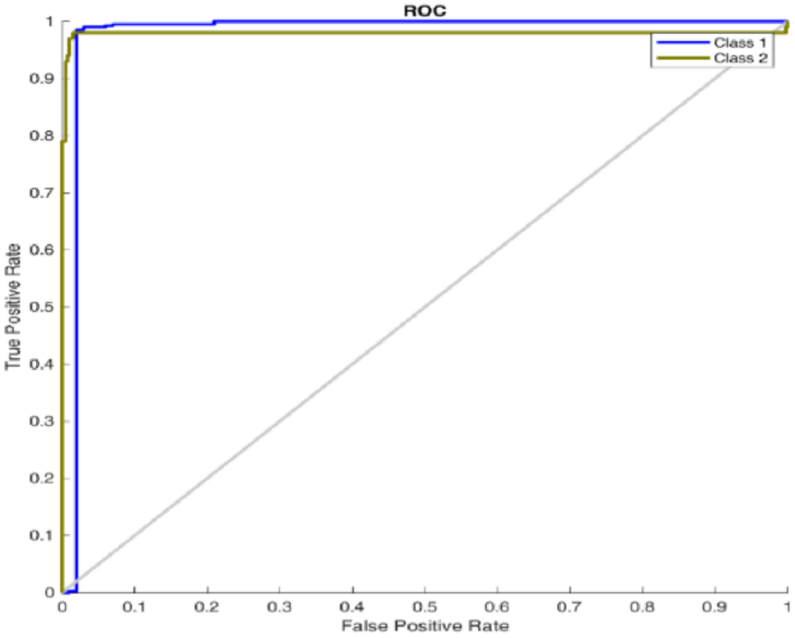

Figure 4.103: ROC Plot for Salp Swarm 3 Features 25 Search Agent

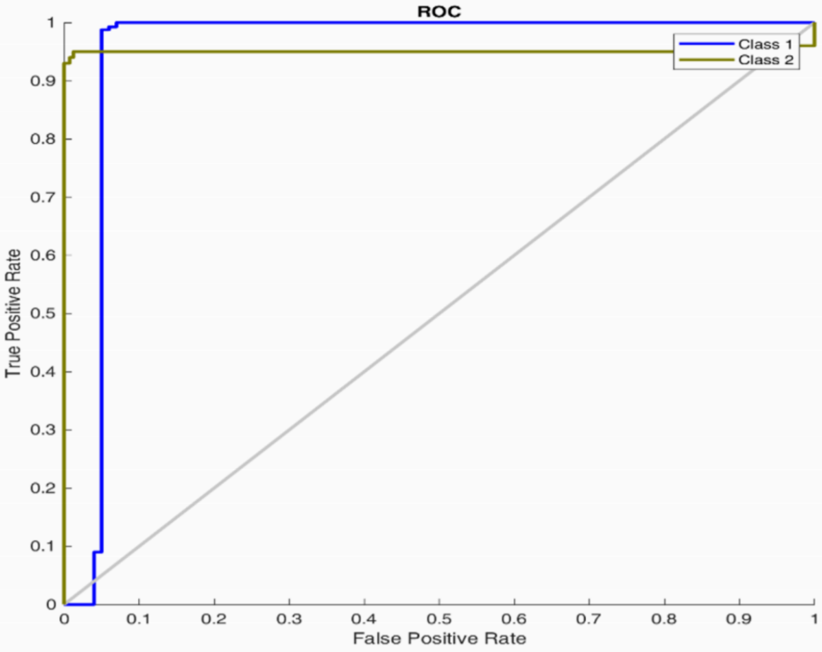

Figure 4.104: ROC Plot for Salp Swarm 3 Features 30 Search Agent

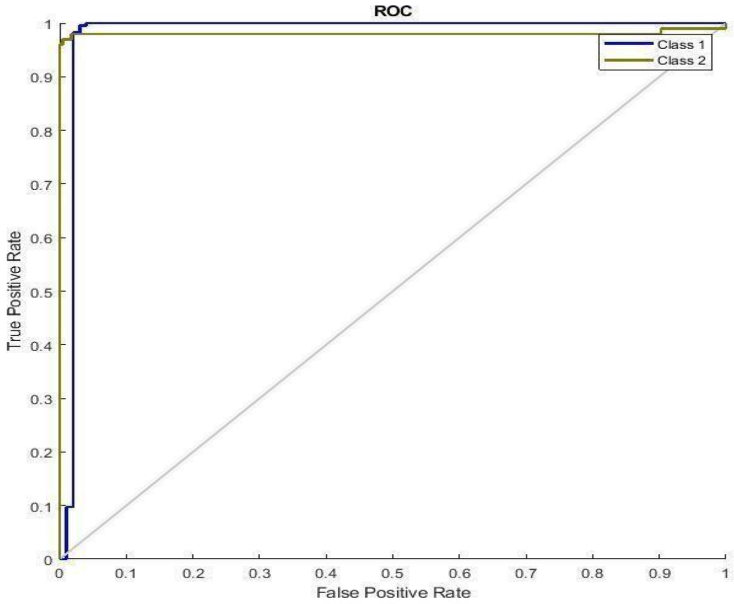

Figure 4.105: ROC Plot for Salp Swarm 3 Features 35 Search Agent

4.7.3 Epilepsy classification using five features and 5, 10, 15, 20, 25, 30, and 35 search agents

Table 4.42 presents the F1 score, recall, accuracy, precision, and AUC. It shows that the accuracy varies between 95.40% to 98.40% as the search agent increases from 5 to 35 for the SSOA-ANN classifier using three features. The best accuracy is achieved at 98.40%, with 97.83% precision, 98.60 sensitivity, F1 score of 98.21%, and 98.40% AUC respectively. It is observed from Figure 4.108 that the ROC curve for 15 search agents performs better than 5, 10, 25, 30, and 35 respectively, indicating a better performance of the SSOA-ANN classifier using three features. The ROCs of various cases obtained using the SSOA-ANN classifier are shown in Figures 4.106 to 4.112 for the representation of the obtained results for 5, 10, 15, 20, 25, 30, and 35 search agents respectively.

Table 4.41: Salp-Swarm Metrics Using Five-Feature Extraction

Feature Extraction	Number of Search Agents	TP	FP	TN	FN	Time (s)
[4 1 5 6 7]	5	400	12	88	0	129.111
[4 6 9 7 8]	10	400	11	89	0	101.543
[8 9 1 5 6]	15	400	8	92	0	101.369
[8 5 9 6 4]	20	400	16	84	0	106.846
[8 5 7 3 9]	25	400	11	89	0	136.340
[5 7 2 3 8]	30	400	23	77	0	128.828

[3 8 6 2 7]	35	400	7	93	0	103.053

Table 4.42: Salp-Swarm Performance Evaluation Using Five-Feature Extraction

Feature Extraction	Number of Search Agents	Accuracy	Precision	Sensitivity	F1-Score	AUC
[4 1 5 6 7]	5	97.6	0.9033	0.9820	0.9410	0.9590
[4 6 9 7 8]	10	97.8	0.9704	0.9780	0.9742	0.9760
[8 9 1 5 6]	**15**	**98.4**	**0.9783**	**0.9860**	**0.9821**	**0.9840**
[8 5 9 6 4]	20	97.6	0.9702	0.9780	0.9811	0.9850
[8 5 7 3 9]	25	97.8	0.9723	0.9800	0.9762	0.9780
[5 7 2 3 8]	30	95.4	0.8952	0.9660	0.9293	0.9450
[3 8 6 2 7]	35	97.4	0.9784	0.9800	0.9848	0.9870

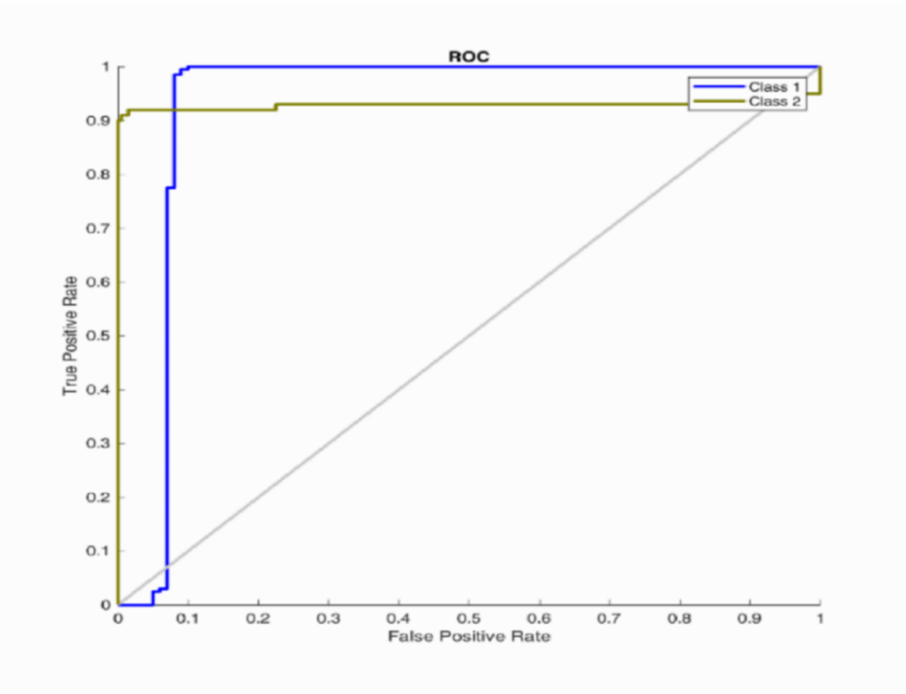

Figure 4.106: ROC Plot for Salp Swarm 5 Features 5 Search Agent

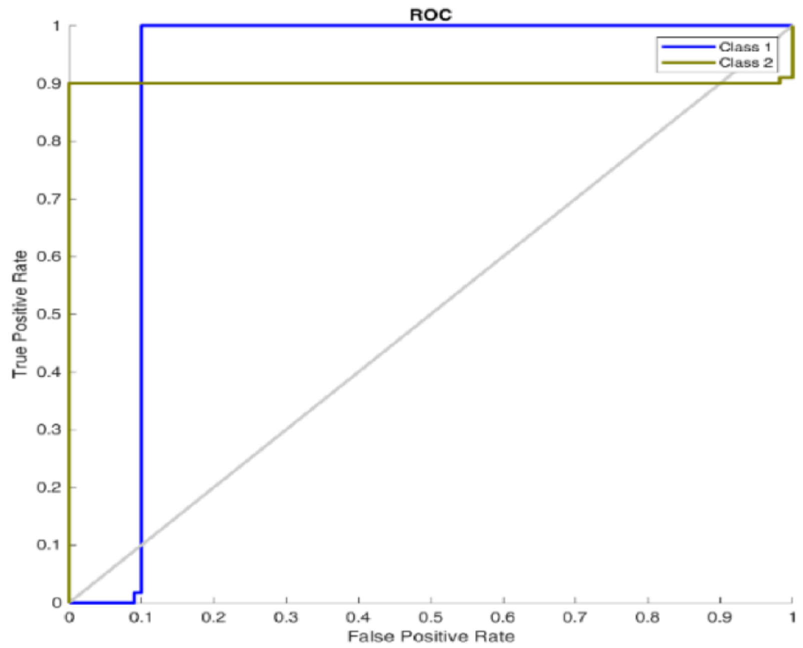

Figure 4.107: ROC Plot for Salp Swarm 5 Features 10 Search Agent

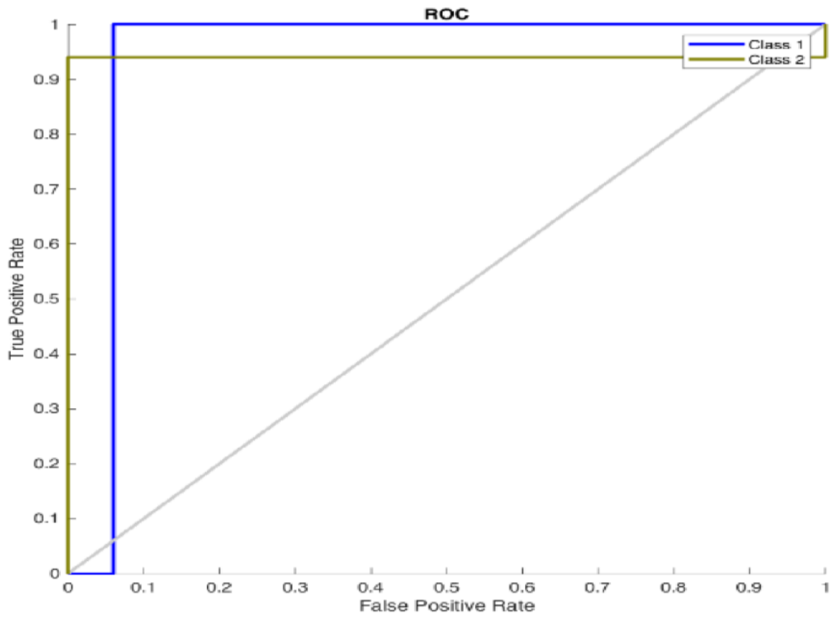

Figure 4.108: ROC Plot for Salp Swarm 5 Features 15 Search Agent

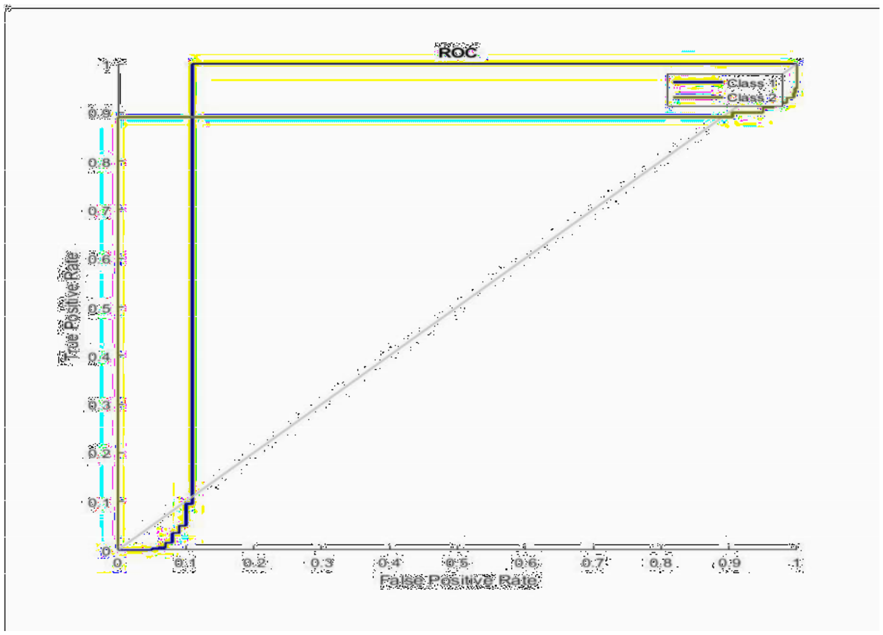

Figure 4.109: ROC Plot for Salp Swarm 5 Features 20 Search Agent

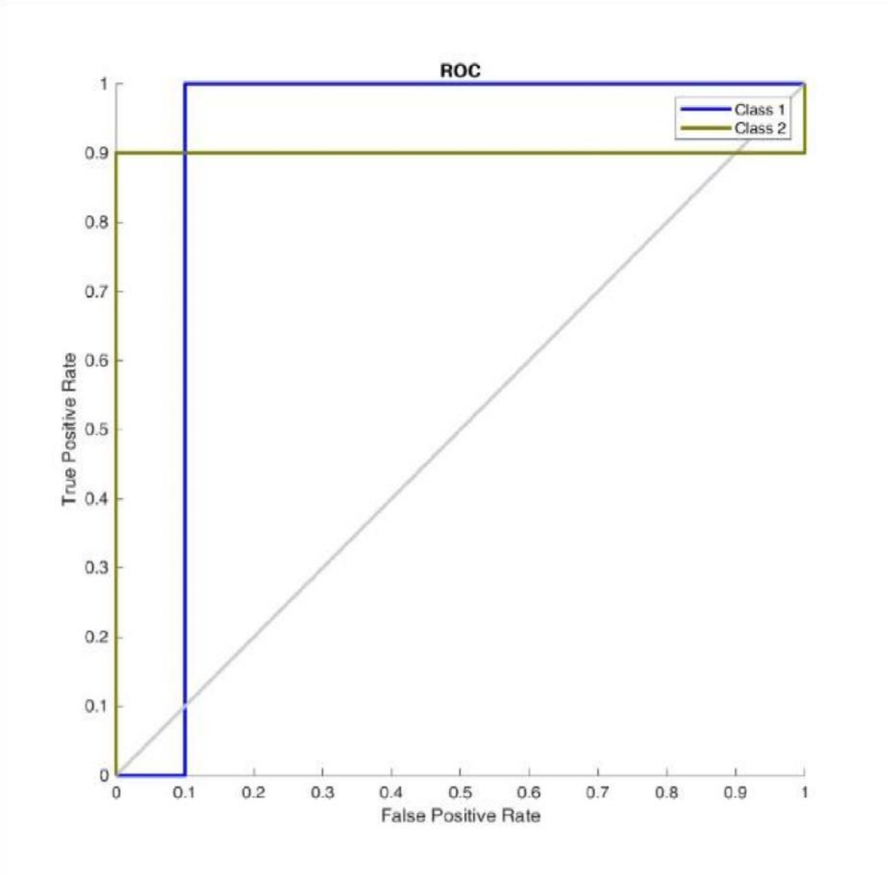

Figure 4.110: ROC Plot for Salp Swarm 5 Features 25 Search Agent

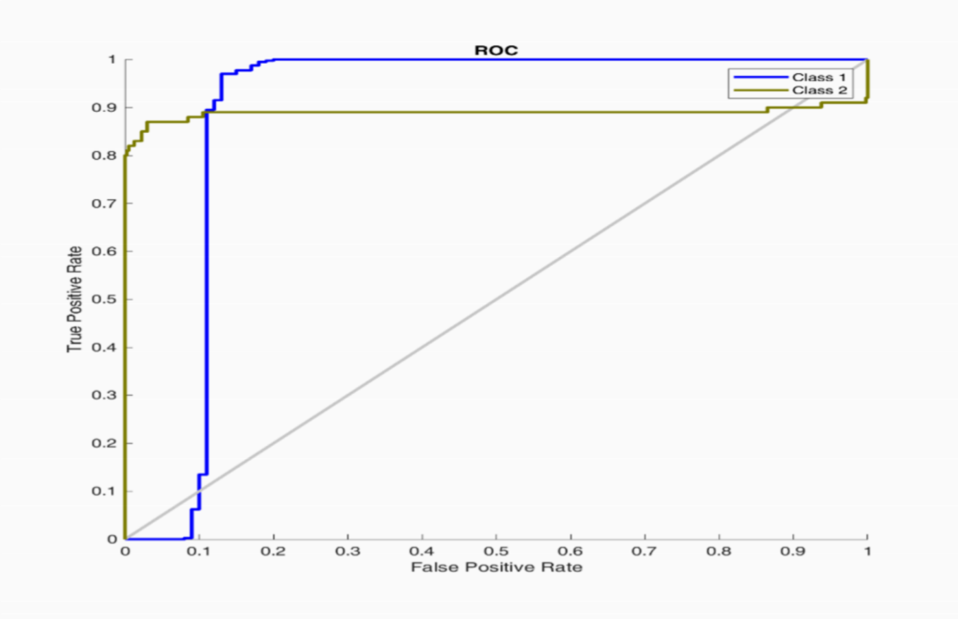

Figure 4.111: ROC Plot for Salp Swarm 5 Features 30 Search Agent

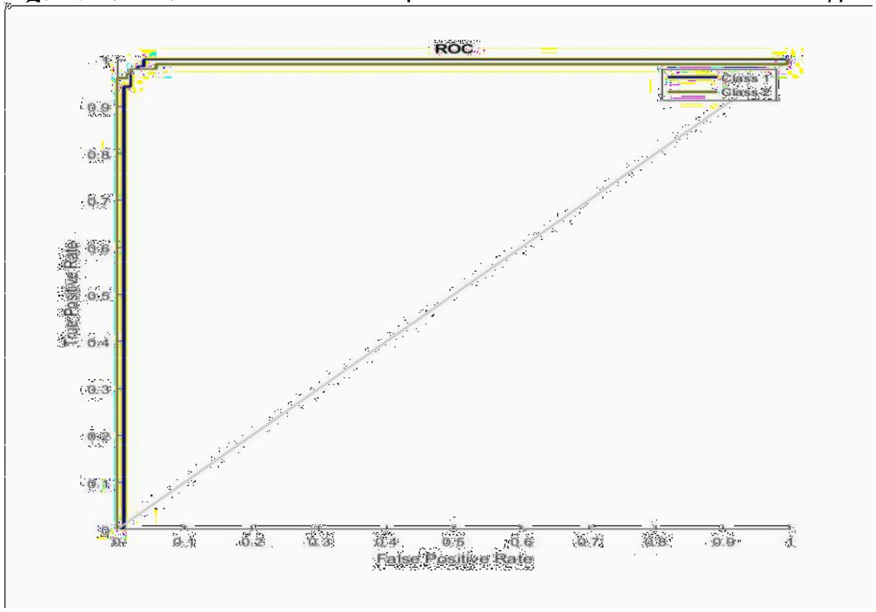

Figure 4.112: ROC Plot for Salp Swarm 5 Features 35 Search Agent

4.7.4 Epilepsy classification using seven features and 5, 10, 15, 20, 25, 30, and 35 search agents

Table 4.44 presents the F1 score, recall, accuracy, precision, and AUC. It is observed that the accuracy varies between 95.40% to 97.20% as the search agent increases from 5 to 35 for the SSOA-ANN classifier using seven features. The best accuracy is achieved at 97.20%, with 95.70% precision, 97.20 sensitivity, F1 score of 96.45%, and 96.80% AUC respectively. Shows in Figure 4.114 that the ROC curve for 10 search agents perform better than 5, 15, 20, 25, 30, and 35 respectively, indicating a better performance of the SSOAANN classifier using seven features. The ROCs of various cases obtained using the SSOAANN classifier are shown in Figures 4.113 to 4.119 for the representation of the obtained results for 5, 10, 15, 20, 25, 30, and 35 search agents respectively.

Table 4.43: Salp-Swarm Metrics Using Seven-Feature Extraction

Feature Extraction	Number of Search Agents	TP	FP	TN	FN	Time (s)
[5 3 4 7 1 6 2]	5	400	16	84	0	114.141
[7 4 9 2 5 8 6]	10	400	14	86	0	138.144
[8 5 7 4 2 3 6]	15	400	23	77	0	146.531
[1 3 4 7 5 9 2]	20	400	8	92	0	113.173
[8 2 1 5 9 7 6]	25	400	18	82	0	121.243
[1 8 9 5 7 6 3]	30	400	16	84	0	104.419
[7 4 8 9 3 1 6]	35	400	11	89	0	114.518

Table 4.44: Salp-Swarm Performance Evaluation Using Seven-Feature Extraction

Feature Extraction	Number of Search Agents	Accuracy	Precision	Sensitivity	F1- Score	AUC
[5 3 4 7 1 6 2]	5	96.8	0.9625	0.9700	0.9662	0.9680
[7 4 9 2 5 8 6]	**10**	**97.2**	**0.9570**	**0.9720**	**0.9645**	**0.9680**
[8 5 7 4 2 3 6]	15	95.4	0.9179	0.9600	0.9385	0.9480
[1 3 4 7 5 9 2]	20	96.4	0.9563	0.9740	0.9601	0.9620
[8 2 1 5 9 7 6]	25	96.4	0.9440	0.9660	0.9549	0.9100
[1 8 9 5 7 6 3]	30	96.8	0.9447	0.9740	0.9591	0.9660
[7 4 8 9 3 1 6]	35	96.8	0.9668	0.9720	0.9643	0.9680

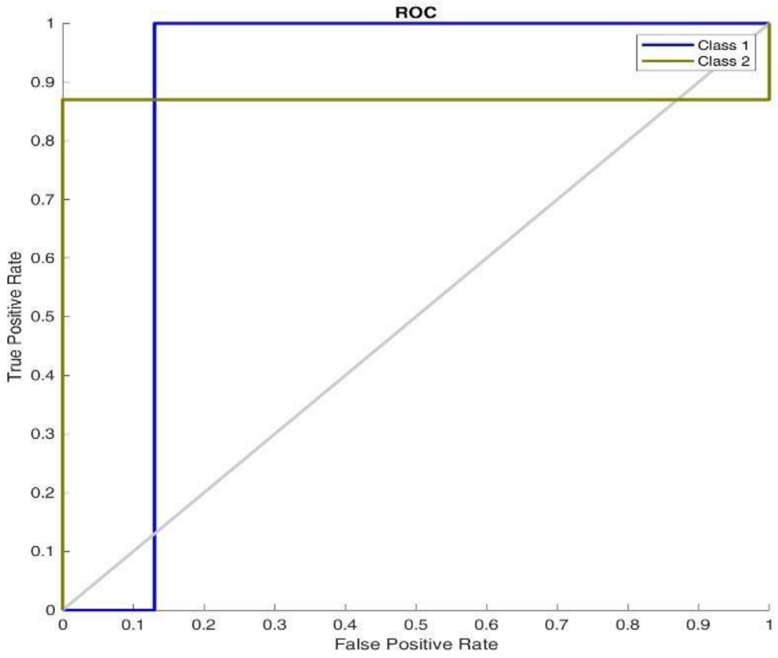

Figure 4.113: ROC Plot for Salp Swarm 7 Features 5 Search Agent

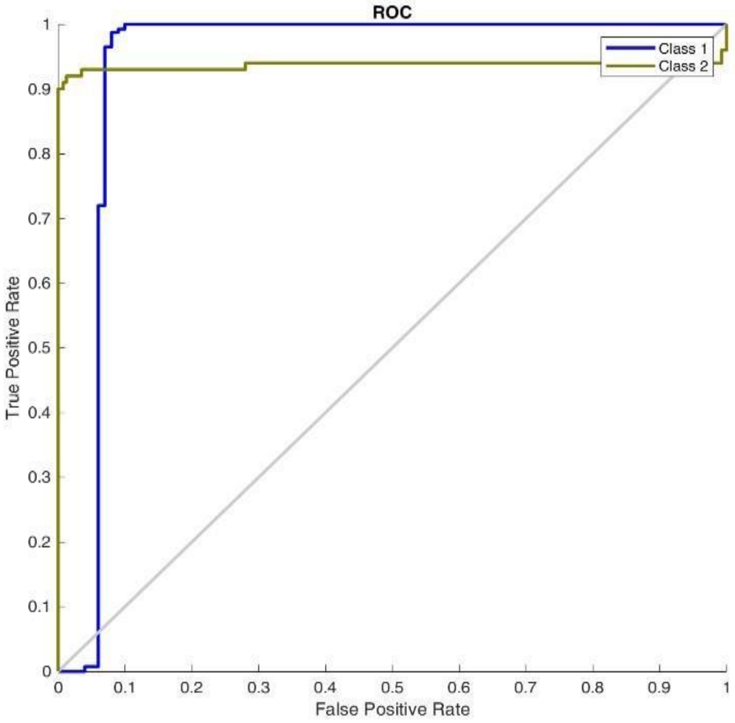

Figure 4.114: ROC Plot for Salp Swarm 7 Features 10 Search Agent

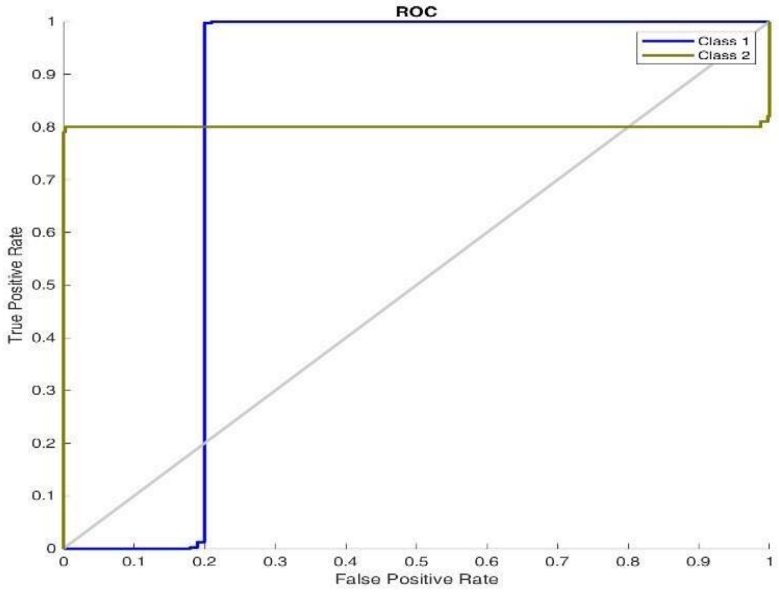

Figure 4.115: ROC Plot for Salp Swarm 7 Features 15 Search Agent

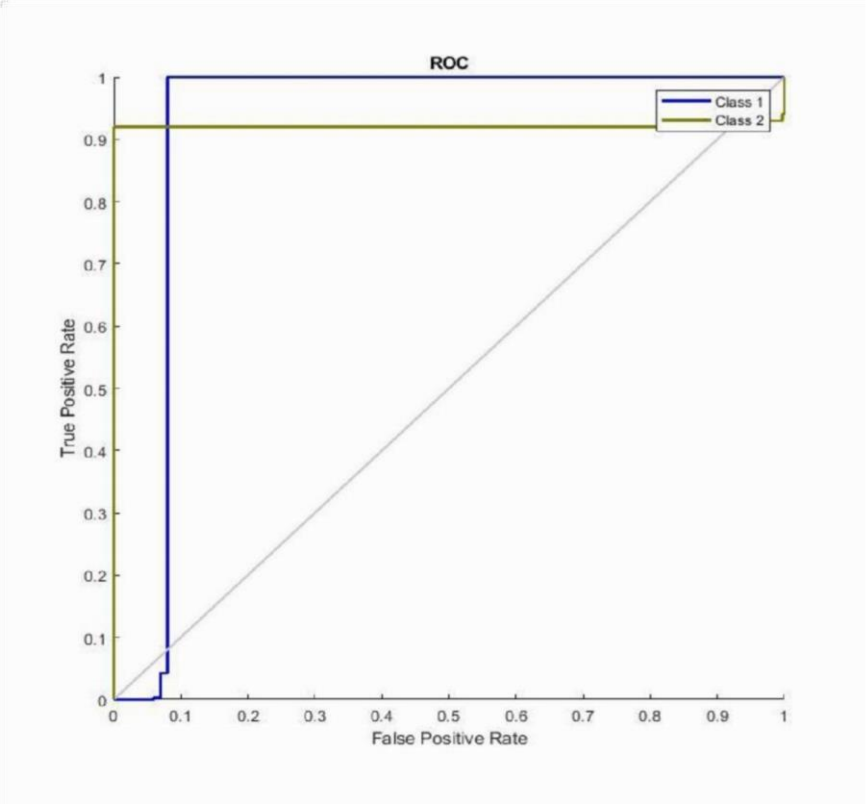

Figure 4.116: ROC Plot for Salp Swarm 7 Features 20 SA

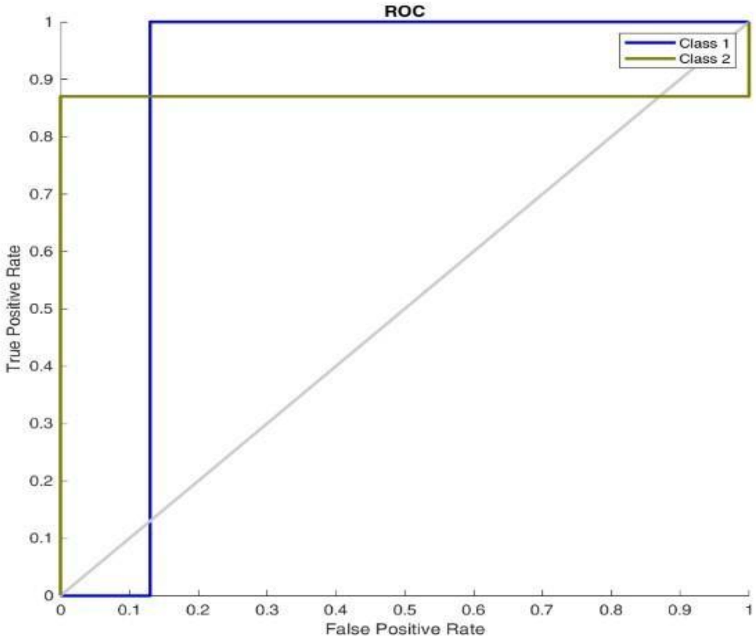

Figure 4.117: ROC Plot for Salp Swarm 7 Features 25 Search Agent

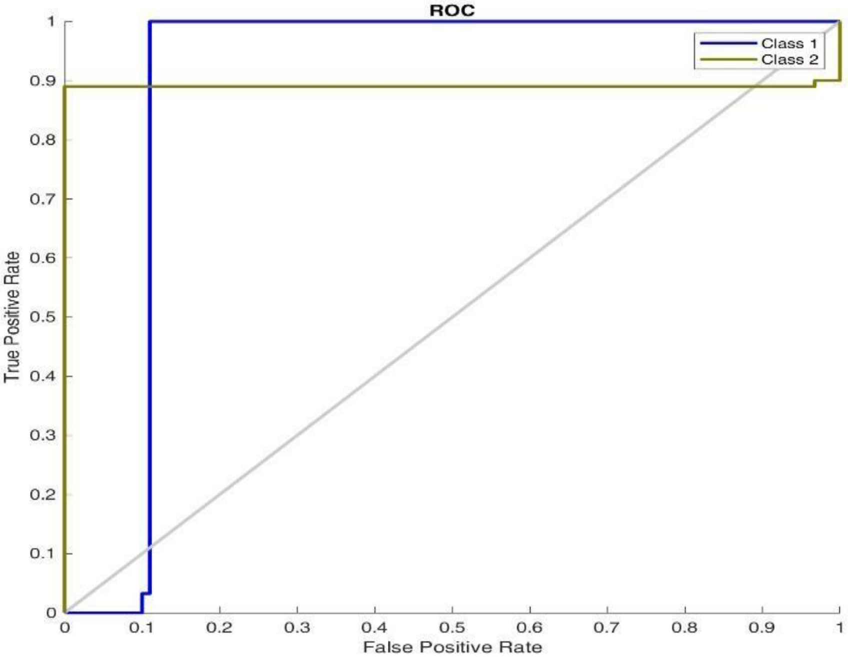

Figure 4.118: ROC Plot for Salp Swarm 7 Features 30 Search Agent

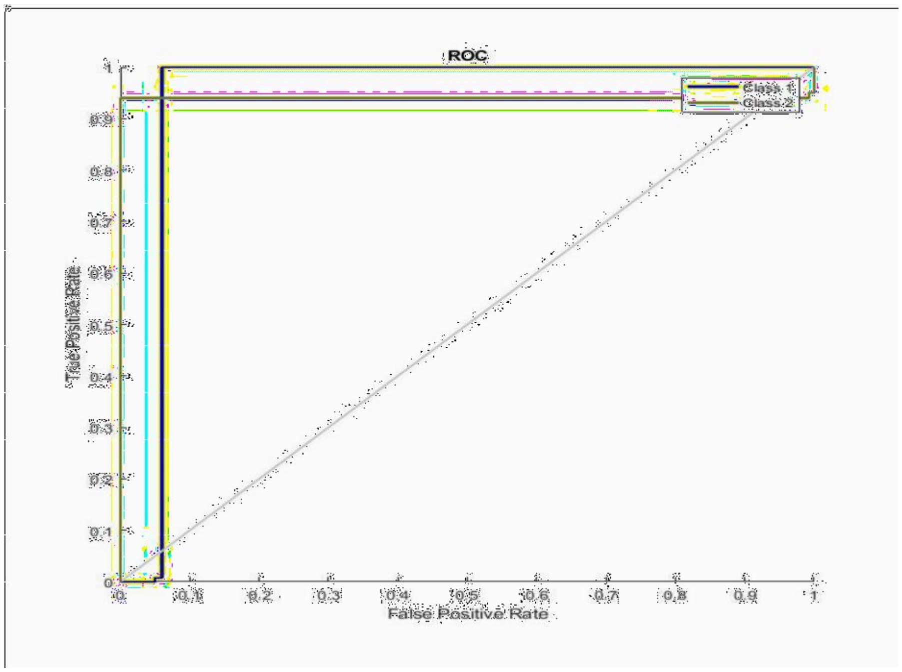

Figure 4.119: ROC Plot for Salp Swarm 7 Features 35 Search Agent

4.7.5 Epilepsy classification using nine feature and 5, 10, 15, 20, 25, 30, and 35 search agents

Table 4.46 presents the F1 score, recall, accuracy, precision, and AUC. It shows that the accuracy varies between 96.00% to 97.60% as the search agent increases from 5 to 35 for the SSOA-ANN classifier using nine features. The best accuracy is achieved at 97.60%, with 96.84% precision, 97.60 sensitivity, and an F1 score of 97.22% and 97.40% AUC respectively. It is observed from Figure 4.122 that the ROC curve for 15 search agents performs better than 5, 10, 25, 30, and 35 respectively, indicating a better performance of the SSOA-ANN classifier using nine features. The ROCs of various cases obtained using the SSOA-ANN classifier are shown in Figures 4.120 to 4.126 for the representation of the obtained results for 5, 10, 15, 20, 25, 30, and 35 search agents respectively.

Table 4.45: Salp-Swarm Metrics Using Nine-Feature Extraction

Feature Extraction	Number of Search Agents	TP	FP	TN	FN	Time (s)
[4 8 6 2 1 3 9 5 7]	5	400	13	87	0	129.088
[3 8 4 7 9 6 2 1 5]	10	400	13	87	0	85.841
[4 8 3 1 6 7 9 5 2]	15	400	12	88	0	66.588
[6 5 3 2 8 9 7 1 4]	20	400	19	81	0	111.800
[5 3 7 4 1 2 6 9 8]	25	400	20	80	0	61.906

[5 6 2 4 7 1 3 8 9]	30	400	12	88	0	63.126
[2 8 1 3 4 5 9 7 6]	35	400	15	85	0	108.987

Table 4.46: Salp-Swarm Performance Evaluation Using Nine-Feature Extraction

Feature Extraction	Number of Search Agents	Accuracy	Precision	Sensitivity	F1-Score	AUC
[4 8 6 2 1 3 9 5 7]	5	97.4	0.9629	0.9780	0.9704	0.9740
[3 8 4 7 9 6 2 1 5]	10	97.4	0.9592	0.9780	0.9685	0.9730
[4 **8 3 1 6 7 9 5 2]**	**15**	**97.6**	**0.9684**	**0.9760**	**0.9722**	**0.9740**
[6 5 3 2 8 9 7 1 4]	20	96.2	0.9405	0.9660	0.9531	0.9590
[5 3 7 4 1 2 6 9 8]	25	96.0	0.9239	0.9700	0.9464	0.9570
[5 6 2 4 7 1 3 8 9]	30	97.6	0.9612	0.9800	0.9705	0.9750
[2 8 1 3 4 5 9 7 6]	35	97.0	0.9607	0.9720	0.9663	0.9690

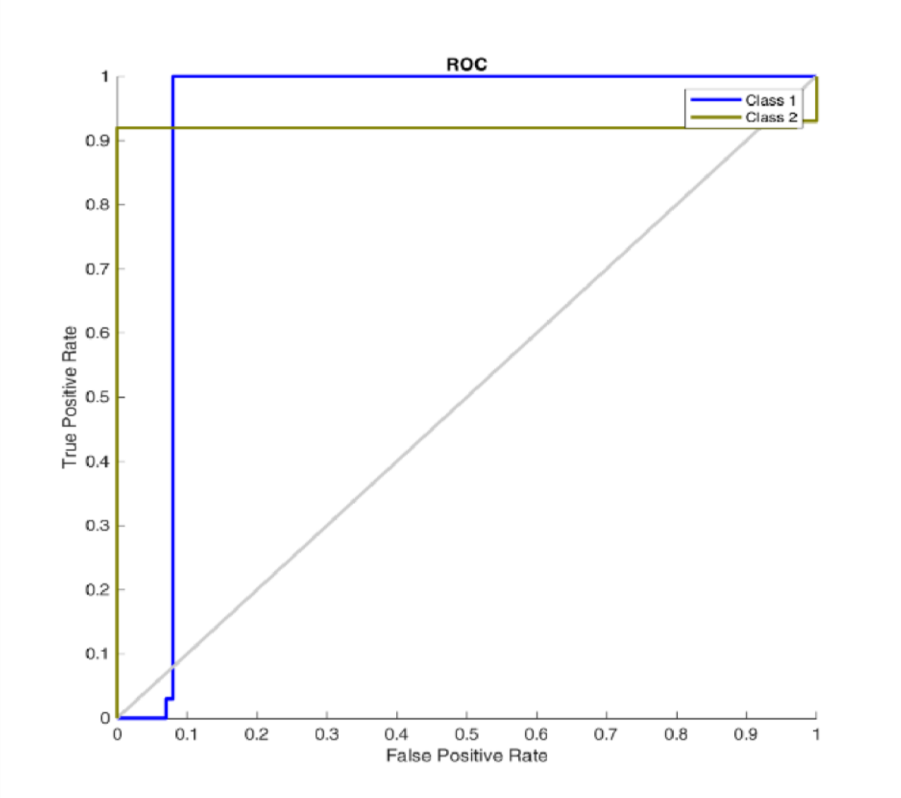

Figure 4.120: ROC Plot for Salp Swarm 9 Features 5 Search Agent

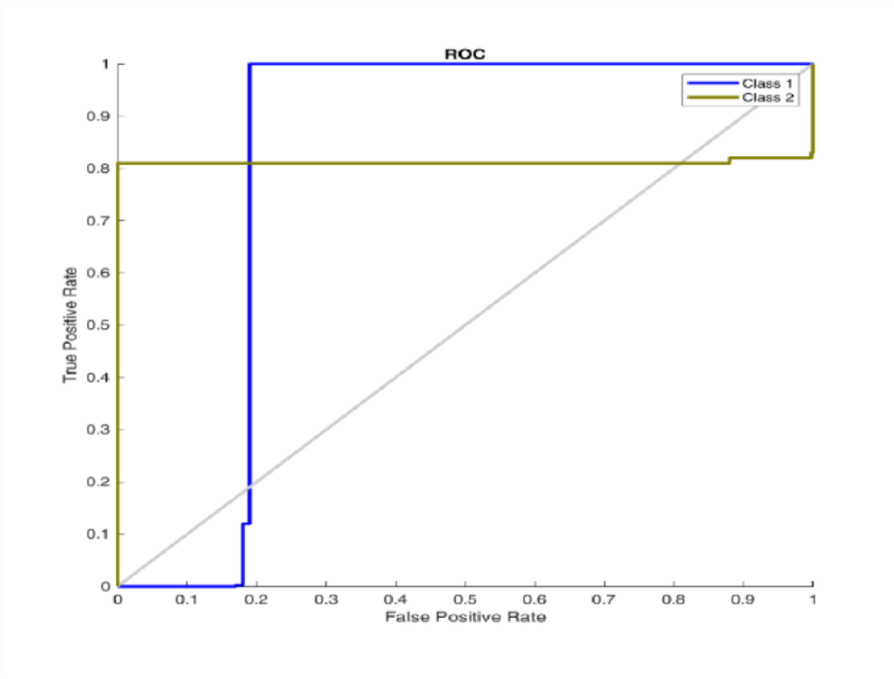

Figure 4.121: ROC Plot for Salp Swarm 9 Features 10 Search Agent

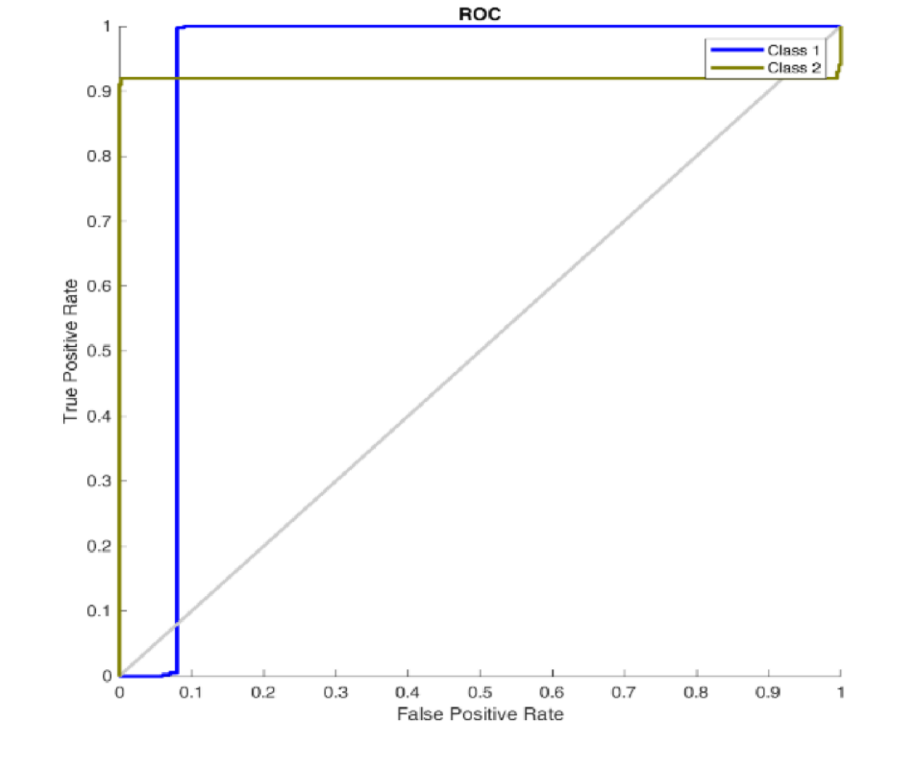

Figure 4.122: ROC Plot for Salp Swarm 9 Features 15 Search Agent

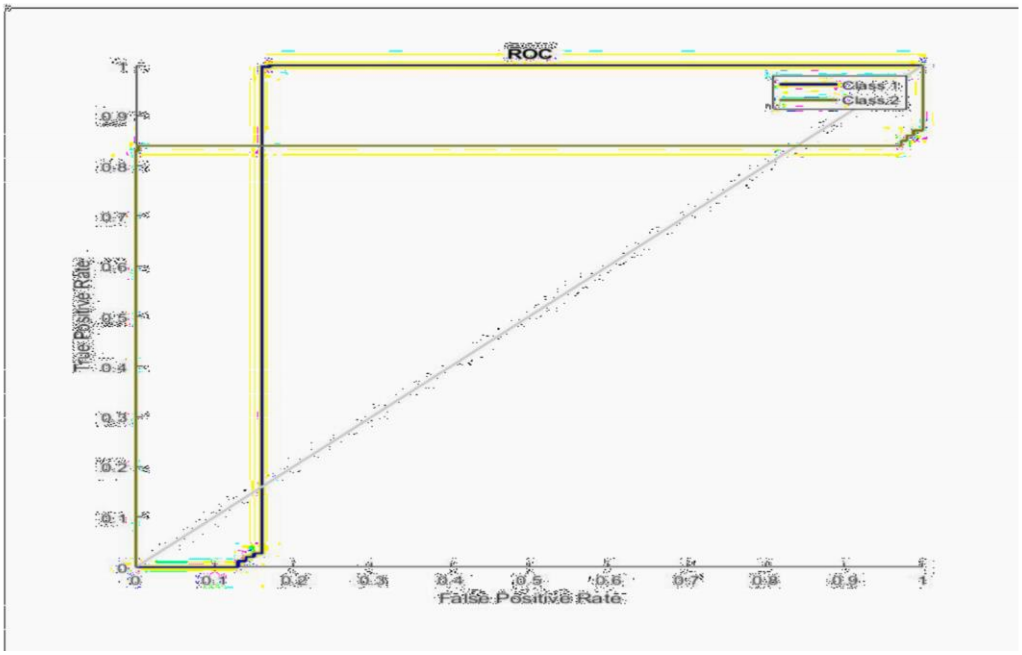

Figure 4.123: ROC Plot for Salp Swarm 9 Features 20 Search Agent

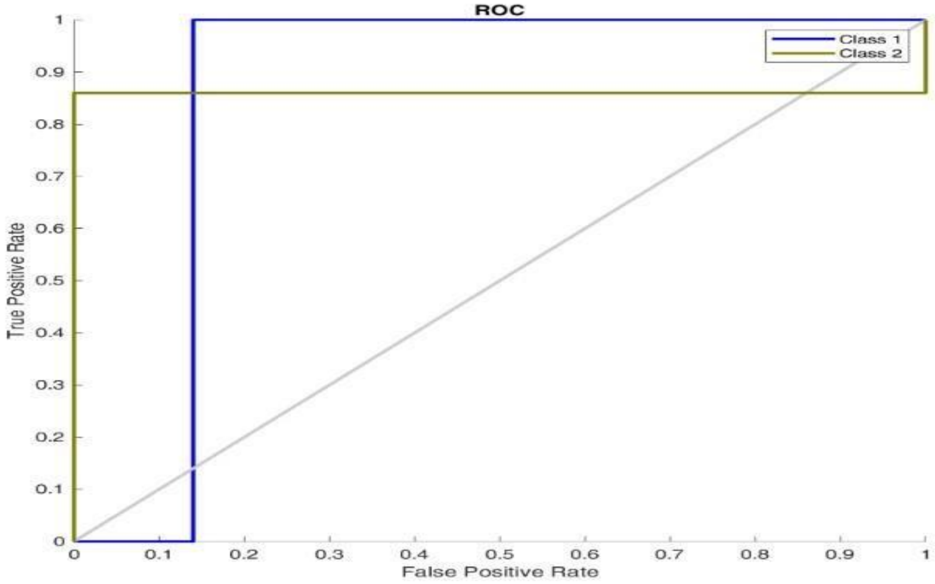

Figure 4.124: ROC Plot for Salp Swarm 9 Features 25 Search Agent

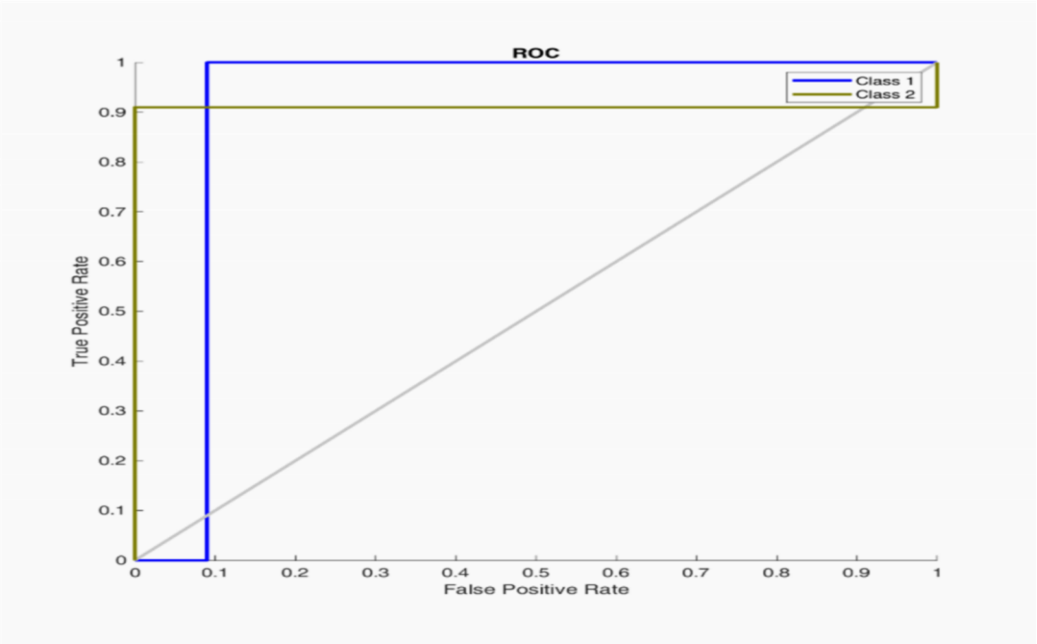

Figure 4.125: ROC Plot for Salp Swarm 9 Features 30 Search Agent

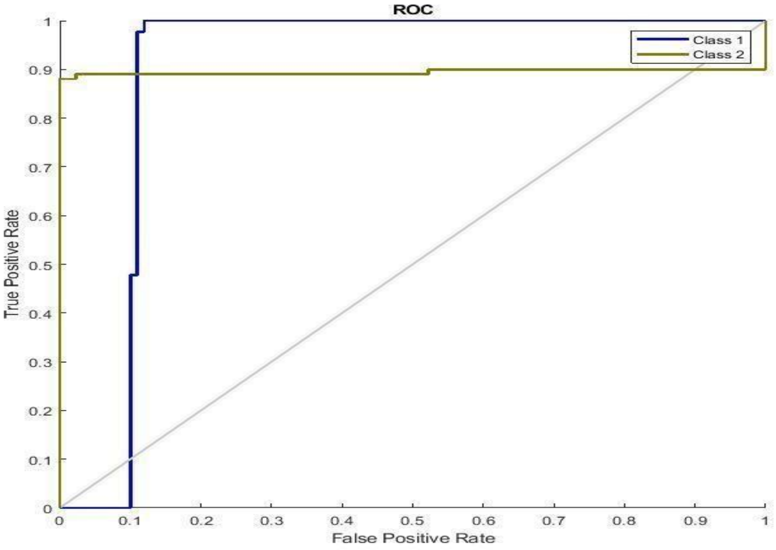

Figure 4.126: ROC Plot for Salp Swarm 9 Features 35 Search Agent

Table 4.47 present the evaluation methods for SSOA-ANN using one, three, five, seven, and nine features respectively for 5, 10. 15, 25, and 30 search agents respectively. The F1 score, recall, accuracy, precision, and AUC for various feature classes are shown as well.

It shows that the best accuracy is achieved by using three features and five features respectively. The best accuracy is achieved at 98.40%, with 98.40% precision, 98.40% sensitivity, and F1 score of

98.40% and 98.40% AUC respectively for 30 search agents, and three feature extractions for the SSOA-ANN algorithm. It is observed from Figure 4.127 that the accuracy is near 100%, indicating a strong performance of the SSOA-ANN classifier using three feature extractions. The classifier performs well in the true classes and vice versa for three features when compared to other classes of the feature's extraction.

Table 4.47: Summary of Epilepsy Detection Using Salp Swarm Algorithm with Various Search Agents

Evaluation Metrics	1 Feature 15 Search Agent	3 Features 30 Search Agent	5 Features 15 Search Agent	7 Features 10 Search Agent	9 Features 15 Search Agent
Accuracy	0.958	**0.984**	**0.984**	0.972	0.976
Precision	0.7975	**0.984**	**0.9783**	0.957	0.9684
Sensitivity	0.998	**0.984**	**0.986**	0.972	0.976
F1 Measure	0.8865	**0.984**	**0.9821**	0.9645	0.9722
AUC	0.926	**0.984**	**0.984**	0.968	0.974

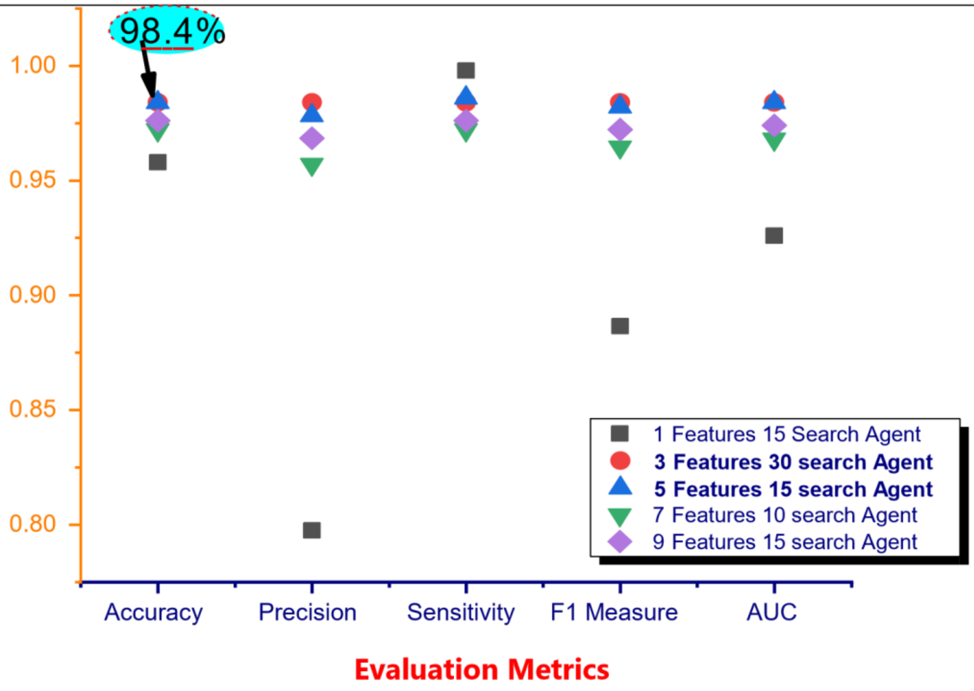

Figure 4.127: Salp Swarm Algorithm Performance with Various Search Agents

4.8 Epilepsy Classification Using Bat Optimization Algorithm

The results of epilepsy detection and classification using the Bat optimization algorithm and an artificial neural network are shown in this section. The different feature extraction ranges from 1, 3,5, 7, and 9 with 5, 10, 15, 20, 25, 30, and 35 generations, respectively.

4.8.1. Epilepsy classification using one population and 5, 10, 15, 20, 25, 30, and 35 generation

The evaluation procedures for BA-ANN utilizing one feature and 5, 10, 15, 20, 25, 30, and 35 search agents are shown in Tables 4.48 and 4.49, respectively. For 5, 10, 15, 20, 25, 30, and 35 generations, the classifier positively identified the true negative classes, falsely predicted 16, 16, 24, 17, and 16 of the false-negative classes, and correctly predicted 5, 6, 5, 1 and 7 false-negative classes. The F1 score, recall, accuracy, precision, and AUC are shown in Table 4.49. It demonstrates that as the generation for the BA-ANN classifier rises from 5 to 35, the accuracy fluctuates between 95.6% and 98.80%. The greatest accuracy is attained at 98.80%, with precision and sensitivity of 76.10% and 99.20%, an F1 score of 86.13%, and an AUC of 90.20%, respectively. Figure 4.130 shows that the Bat-ANN classifier performs better than the 5, 10, 15, 20, 25, 30, and 35 generations, respectively, in terms of the ROC curve. Figures 4.128 to 4.134 demonstrate the ROCs of several examples produced using the

Bat-ANN classifier for the representation of the acquired findings for 5, 10, 15, 20, 25, and 35 generations, respectively. In the real classes, the classifier performs well, and vice versa.

Table 4.48: Bat Algorithm Metrics Using One Population and 5, 10, 15, 20, 25, 30, and 35 Generation

Number of Populations	Number of Generation	TP	FP	TN	FN	Time (s)
[4]	5	395	16	84	5	30.361
[4]	10	394	16	84	6	52.941
[6]	15	395	17	83	5	57.6685
[1]	20	393	14	86	7	114.607
[7]	25	399	24	76	1	91.777
[5]	30	393	16	84	7	130.233
[5]	35	400	19	81	0	240.699

Table 4.49: Bat Algorithm Performance Evaluation Using One Population and 5, 10, 15, 20, 25, 30, and 35 Generation

Number of	Number of	Accuracy	Precision	Sensitivity	F1-	AUC
Populations	Generation	Score				
[4]	5	95.8	0.7936	0.9980	0.8842	0.9240
[4]	10	95.6	0.7975	0.9980	0.8865	0.9260
[6]	**15**	**98.8**	**0.7610**	**0.9920**	**0.8613**	**0.9020**
[1]	20	94.4	0.7522	0.9860	0.8534	0.8930

[7]	25	95.0	0.8353	0.9980	0.9095	0.944
[5]	30	95.4	0.7714	1.0000	0.8769	0.9130
[5]	35	95.4	0.7714	1.0000	0.8709	0.9130

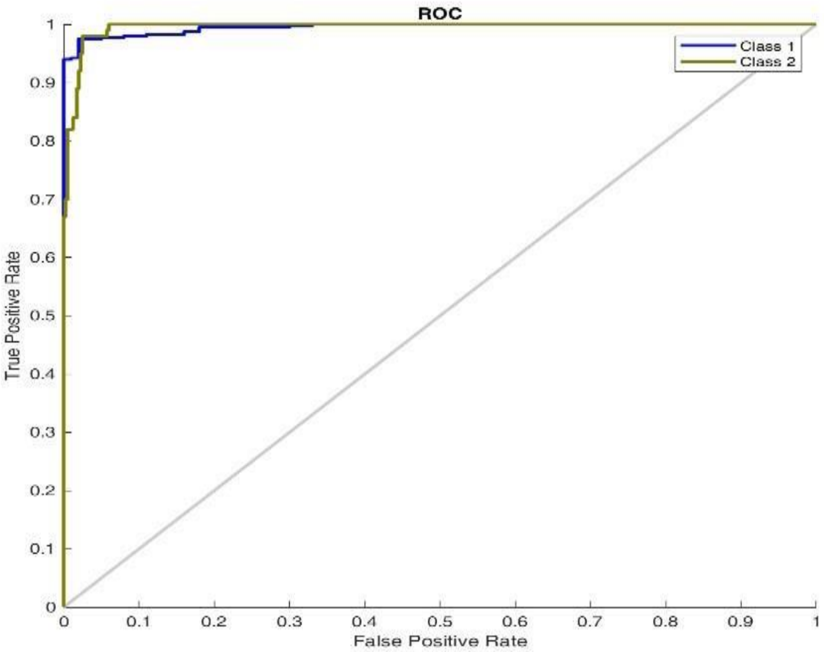

Figure 4.128: ROC Plot for Bat Algorithm 1 Population 5 Generation

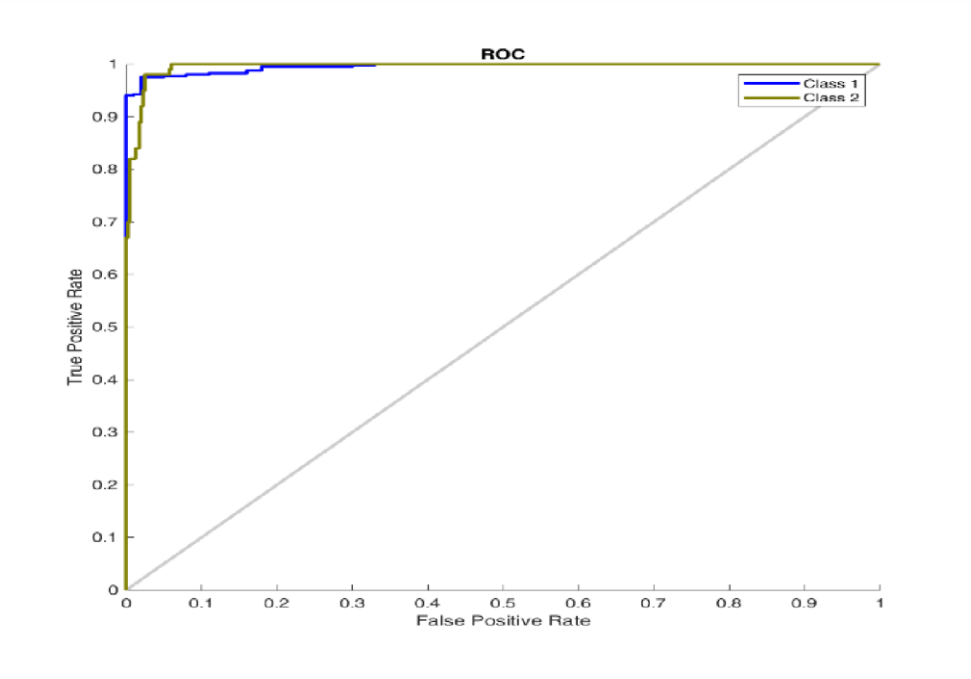

Figure 4.129: ROC Plot for Bat Algorithm 1 Population 10 Generation

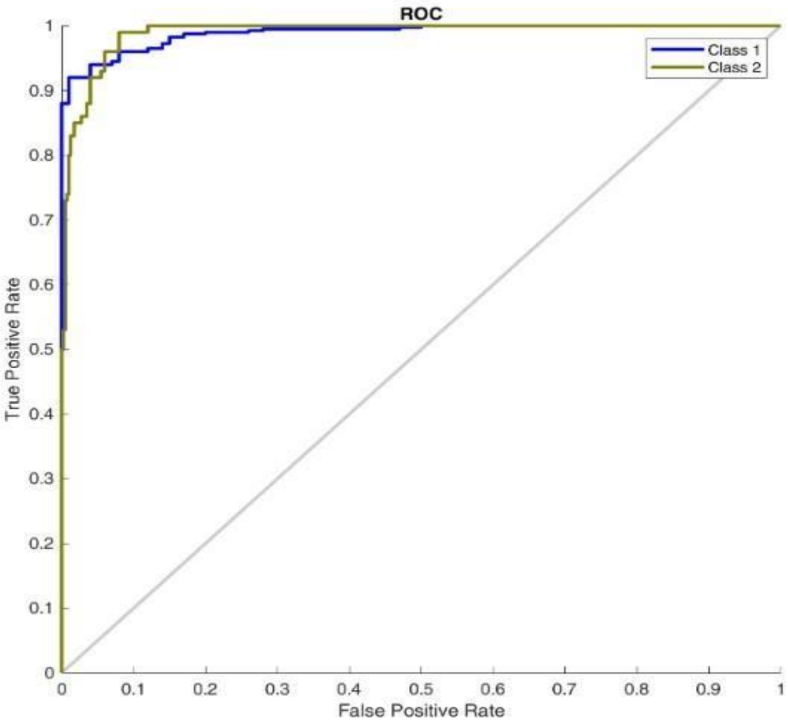

Figure 4.130: ROC Plot for Bat Algorithm 1 Population 15 Generation

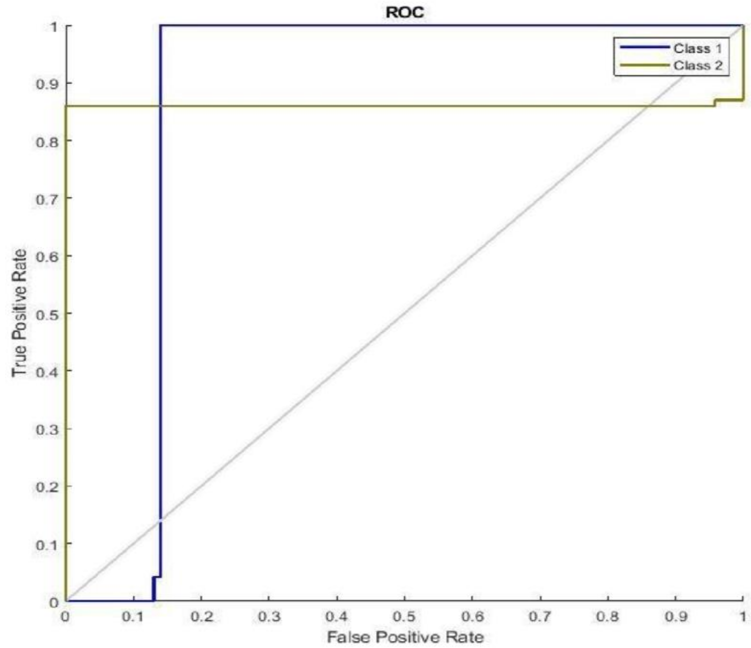

Figure 4.131: ROC Plot for Bat Algorithm 1 Population 20 Generation

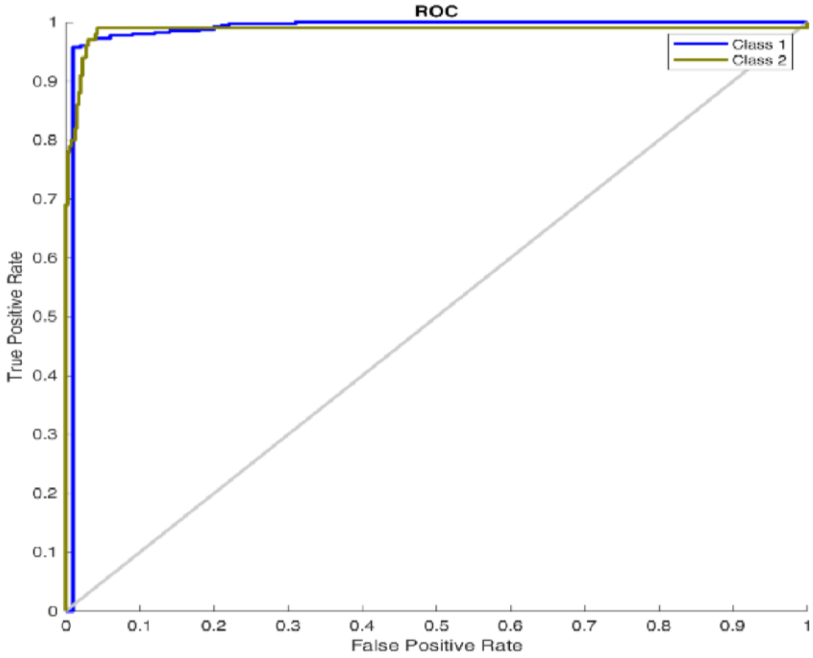

Figure 4.132: ROC Plot for Bat Algorithm 1 Population 25 Generation

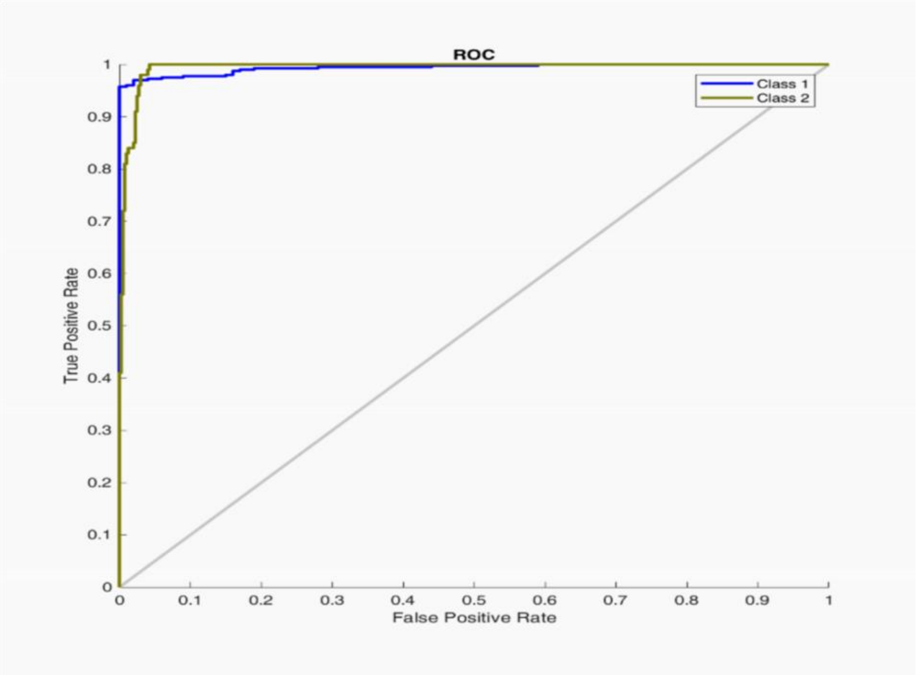

Figure 4.133: ROC Plot for Bat Algorithm 1 Population 30 Generation

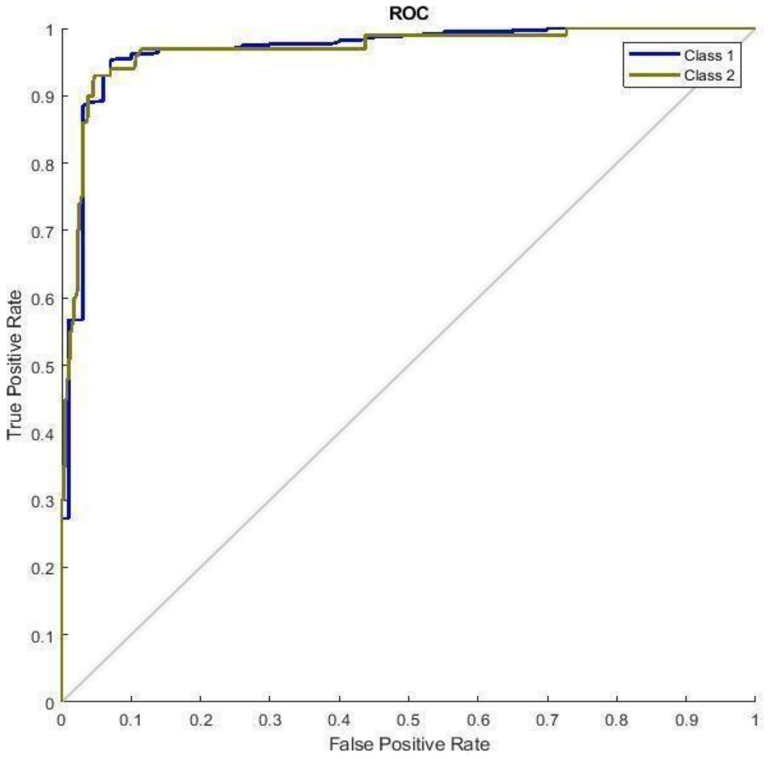

Figure 4.134: ROC Plot for Bat Algorithm 1 Population 35 Generation

4.8.2 Epilepsy classification using three populations and 5, 10, 15, 20, 25, 30, and 35 generation

Table 4.51 presents the F1 score, recall, accuracy, precision, and AUC. It shows that the accuracy varies between 95.80% to 98.20% as the generation increases from 5 to 35 for the BA-ANN classifier. The best accuracy is achieved at 98.20%, with 90.14% precision,

99.00 sensitivity, F1 score of 94.26%, and 96.40% AUC respectively. It is observed from Figure 4.137 that the ROC curve for 15 generations performs better than 5, 10, 25, 30, and 35 respectively, indicating a better performance of the BA-ANN classifier. The ROCs of various cases obtained using the BA-ANN classifier are plotted as shown in Figures 4.135 to 4.141 for the representation of the obtained results for 5, 10, 15, 20, 25, 30, and 35 generations respectively. The classifier seems to function well in the real classes and vice versa.

Table 4.50: Bat Algorithm Metrics Using Three Populations and 5, 10, 15, 20, 25, 30, and 35 Generation

Number of Populations	Number of Generation	TP	FP	TN	FN	Time
[6 9 4]	5	399	9	91	1	41.194
[4 9 5]	10	398	19	81	2	75.398
[3 4 6]	15	399	8	92	1	101.805
[6 1 3]	20	400	9	91	0	194.887
[4 3 9]	25	399	9	91	1	169.602
[6 4 7]	30	399	10	90	1	178.404
[6 1 3]	35	400	4	96	0	354.307

Table 4.51: Bat Algorithm Performance Evaluation Using Three Populations and 5, 10, 15, 20, 25, 30, and 35 Generation

Number of Populations	Number of Generation	Accuracy	Precision	Sensitivity	F1 Score	AUC
[6 9 4]	5	98.0	0.9743	0.9820	0.9781	0.9800
[4 9 5]	10	95.8	0.8742	0.9780	0.9232	0.9460
[3 4 6]	**15**	**98.2**	**0.9014**	**0.9900**	**0.9436**	**0.9640**

[6 1 3]	20	98.2	0.8464	0.9880	0.9117	0.9420
[4 3 9]	25	98.0	0.9369	0.9840	0.9599	0.9710
[6 4 7]	30	97.8	0.9161	0.9861	0.9498	0.9660
[6 1 3]	35	99.2	0.9881	0.9920	0.9900	0.9910

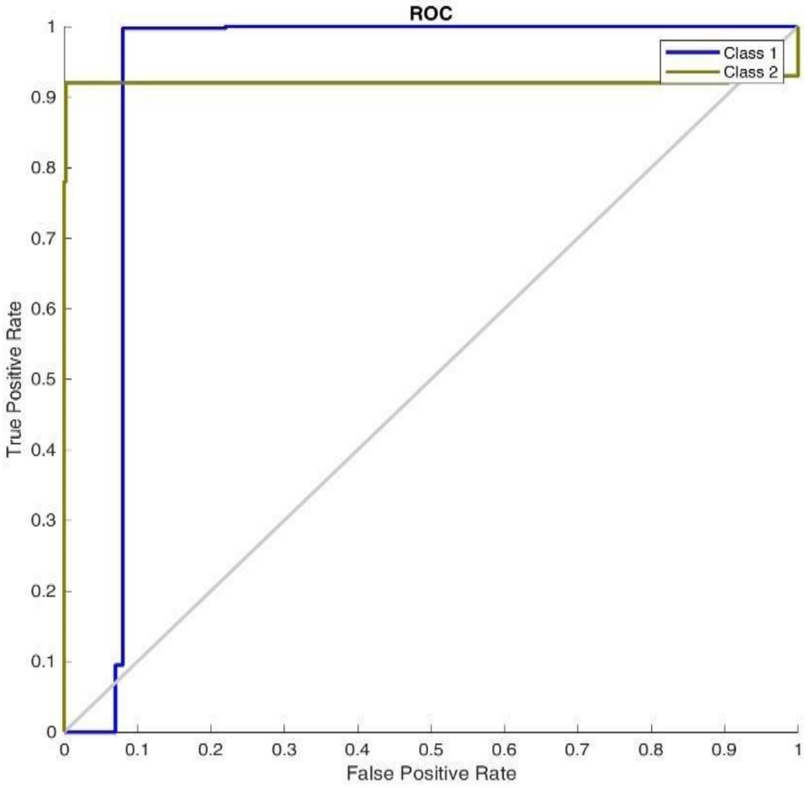

Figure 4.135: ROC Plot for Bat Algorithm 3 Population 5 Generation

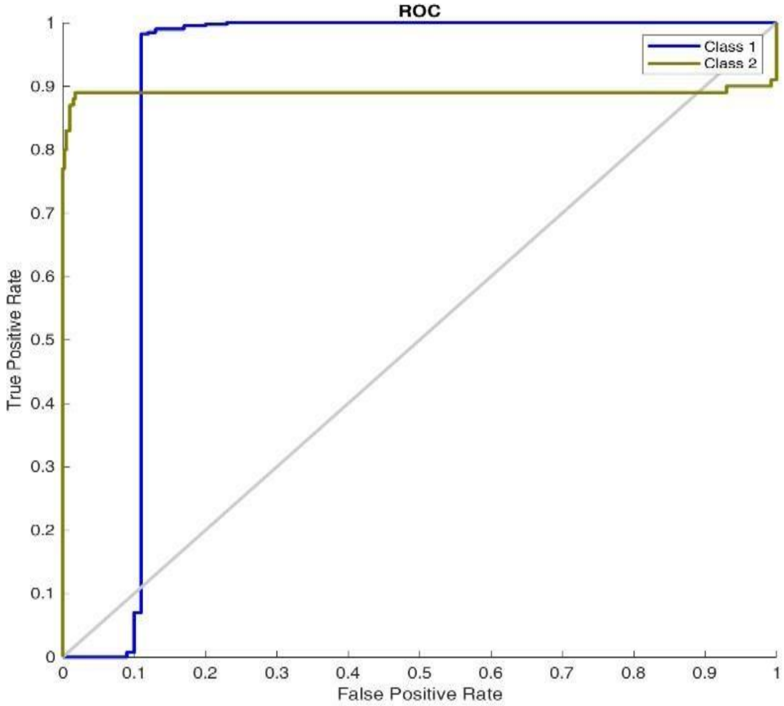

Figure 4.136: ROC Plot for Bat Algorithm 3 Population 10 Generation

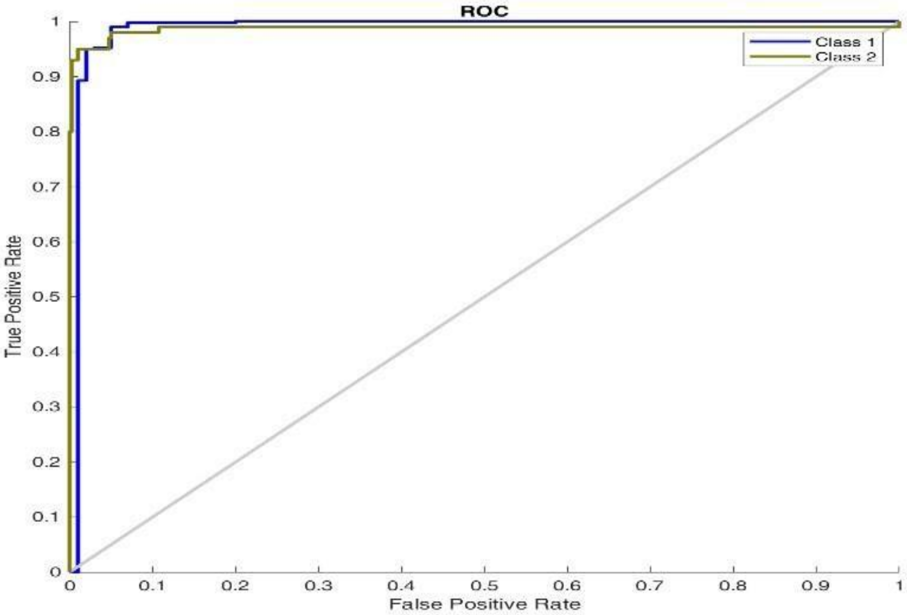

Figure 4.137: ROC Plot for Bat Algorithm 3 Population 15 Generation

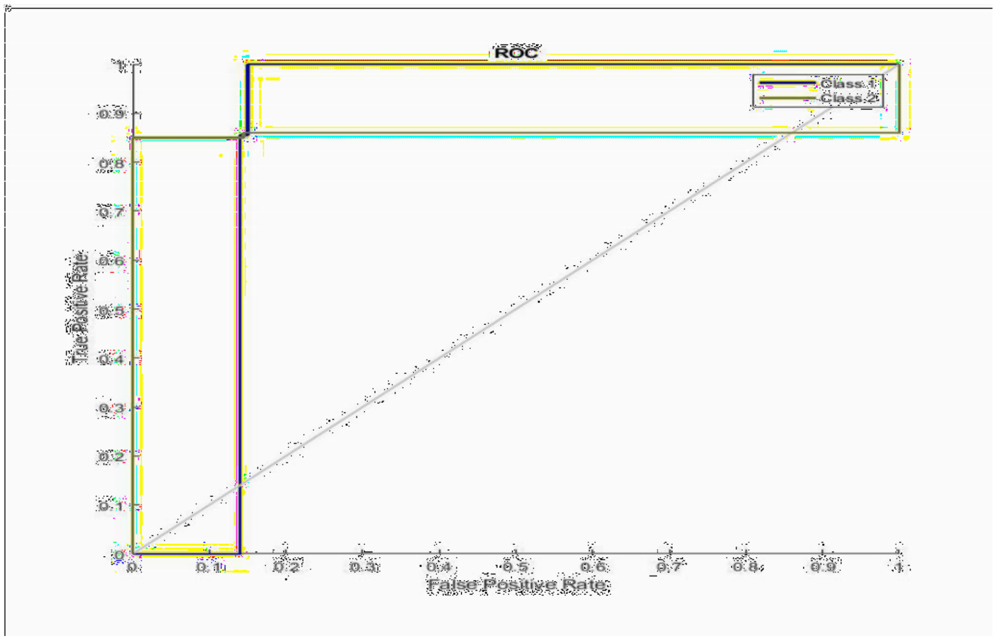

Figure 4.138: ROC Plot for Bat Algorithm 3 Population 20 Generation

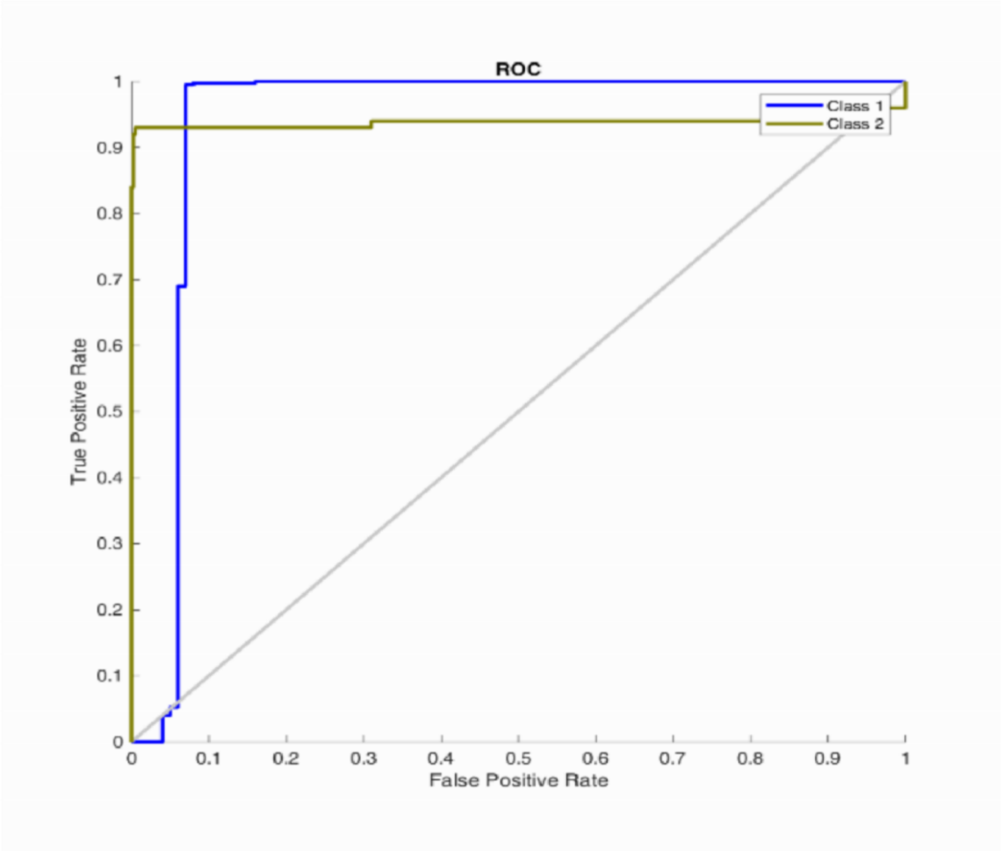

Figure 4.139: ROC Plot for Bat Algorithm 3 Population 25 Generation

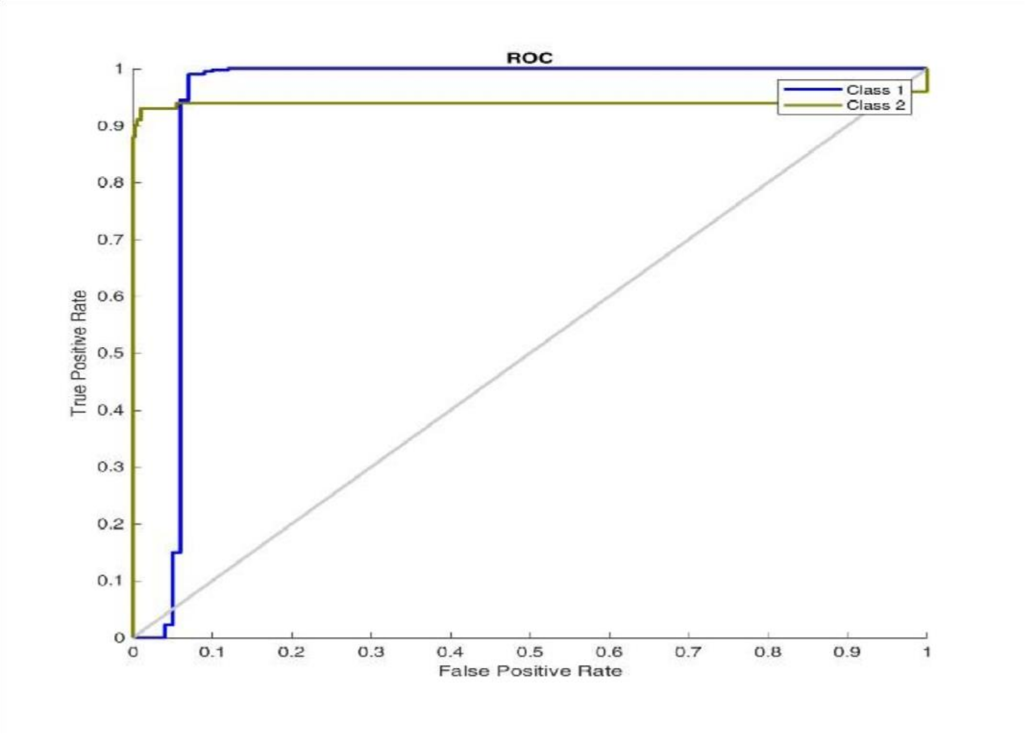

Figure 4.140: ROC Plot for Bat Algorithm 3 Population 30 Generation

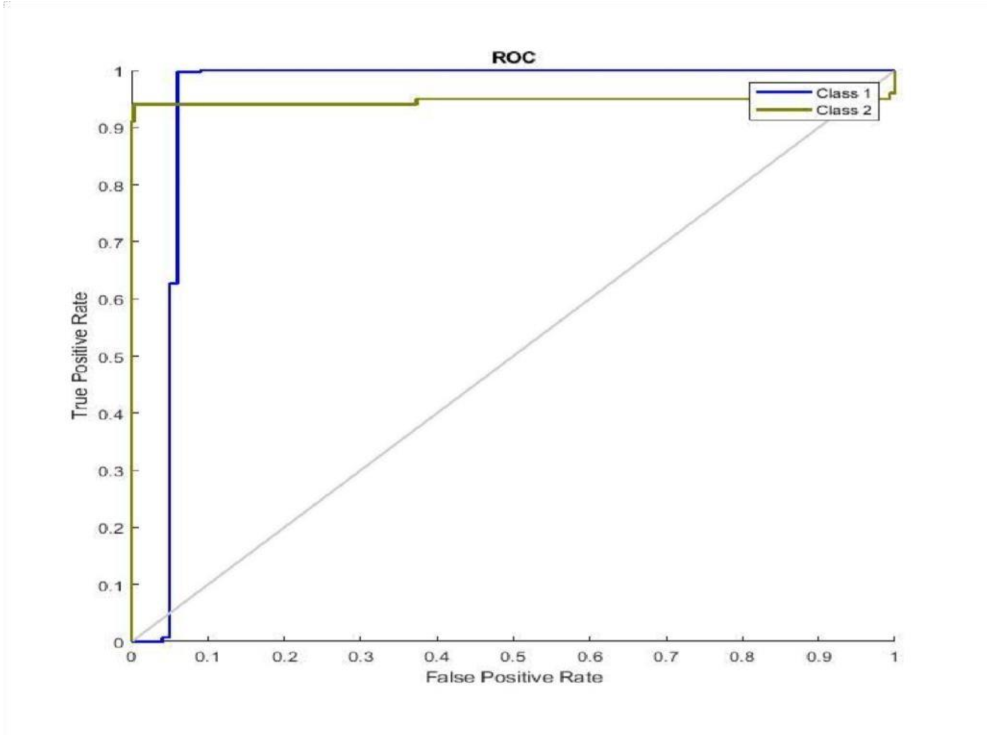

Figure 4.141: ROC Plot for Bat Algorithm 3 Population 35 Generation

4.8.3 Epilepsy classification using five populations and 5, 10, 15, 20, 25, 30, and 35 generation Table 4.53 presents the F1 score, recall, accuracy, precision, and AUC. It shows that the accuracy varies between 95.40% to 98.40% as the generation increases from 5 to 35 for the BA-ANN classifier. The best accuracy is achieved at 98.40%, with 76.70% precision, 98.60 sensitivity, F1 score of 74.64%, and 98.10% AUC respectively. It is observed from Figure 4.144 that the ROC curve for 15 generations performs better than 5, 10, 15, 20, 25, 30, and 35 respectively, indicating a better performance of the Bat-ANN classifier. The ROCs of various cases obtained using the BA-ANN classifier are plotted as shown in Figures 4.142 to 4.148 for the representation of the obtained results for 5, 10, 15, 20, 25, 30, and 35 generations respectively. It demonstrates that the classifier functions effectively in the real classes and vice versa.

Table 4.52: Bat Algorithm Metrics Using Five Populations and 5, 10, 15, 20, 25, 30, and 35 Generation

Number of Populations	Number of Generation	TP	FP	TN	FN	Time (s)
[4 2 1 9 6]	5	399	20	80	1	34.795
[9 8 6 3 7]	10	400	20	80	0	83.062
[8 5 1 3 9]	15	400	8	92	0	116.840
[1 3 6 2 7]	20	400	17	83	0	251.232
[8 6 1 5 2]	25	400	23	77	0	170.436
[6 3 8 9 4]	30	400	10	90	0	251.997
[3 7 6 2 8]	35	400	12	88	0	400.084

Table 4.53: Bat Algorithm Performance Evaluation Using Five Populations and 5, 10, 15, 20, 25, 30, and 35 Generation

Number of Populations	Number of Generation	Accuracy Score	Precision	Sensitivity	F1-	AUC

[4 2 1 9 6]	5	95.8	0.8748	0.9820	0.9253	0.9490
[9 8 6 3 7]	10	96.0	0.9492	0.9640	0.9566	0.9600
[8 5 1 3 9]	**15**	**98.4**	**0.9670**	**0.9860**	**0.9764**	**0.9810**
[1 3 6 2 7]	20	97.6	0.8938	0.9880	0.9386	0.9600
[8 6 1 5 2]	25	95.4	0.9332	0.9620	0.9474	0.9540
[6 3 8 9 4]	30	98.0	0.9670	0.9860	0.9764	0.9810
[3 7 6 2 8]	35	96.6	0.9585	0.9660	0.9622	0.9640

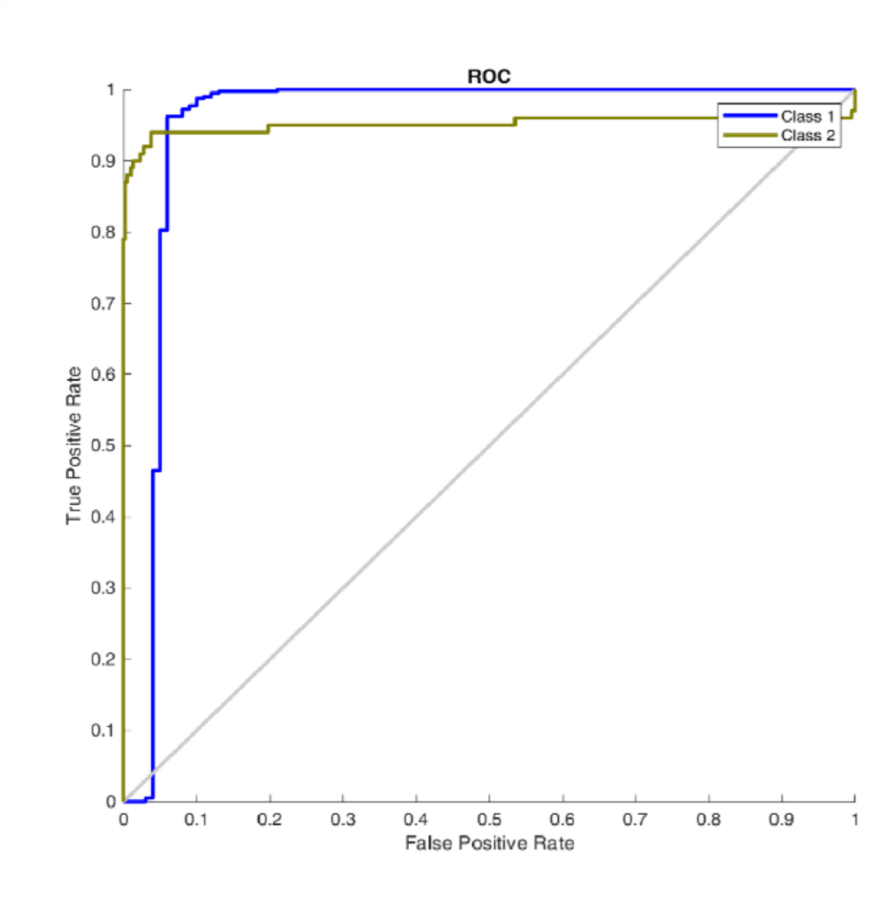

Figure 4.142: ROC Plot for Bat Algorithm 5 Population 5 Generation

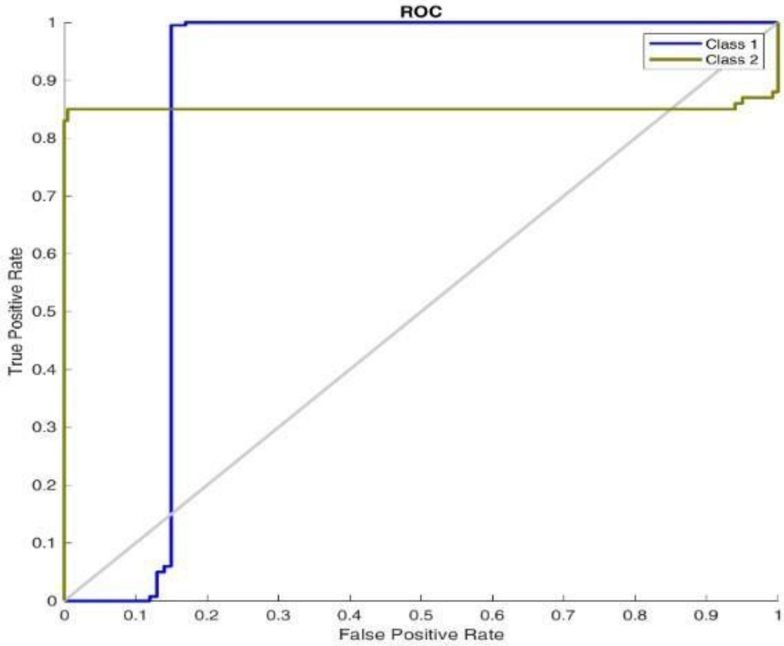

Figure 4.143: ROC Plot for Bat Algorithm 5 Population 10 Generation

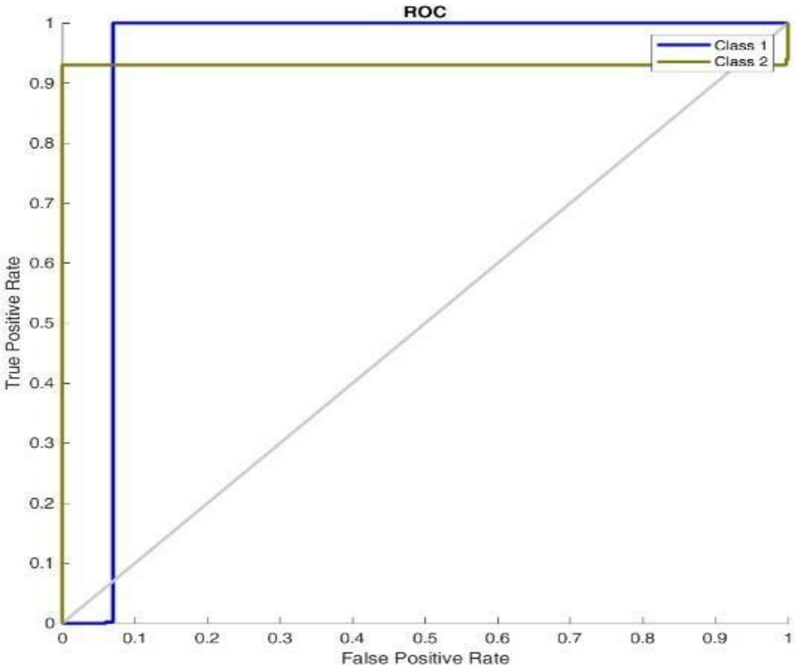

Figure 4.144: ROC Plot for Bat Algorithm 5 Population 15G

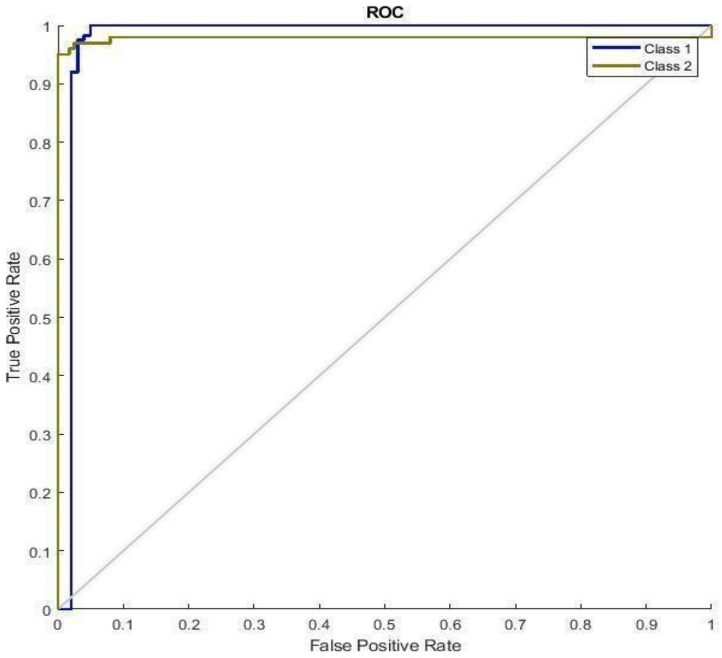

Figure 4.145: ROC Plot for Bat Algorithm 5 Population 20 Generation

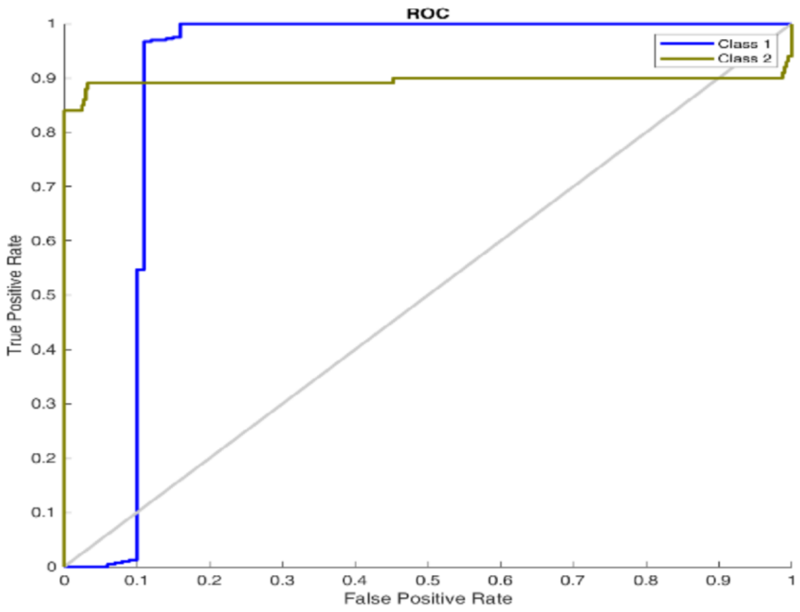

Figure 4.146: ROC Plot for Bat Algorithm 5 Population 25 Generation

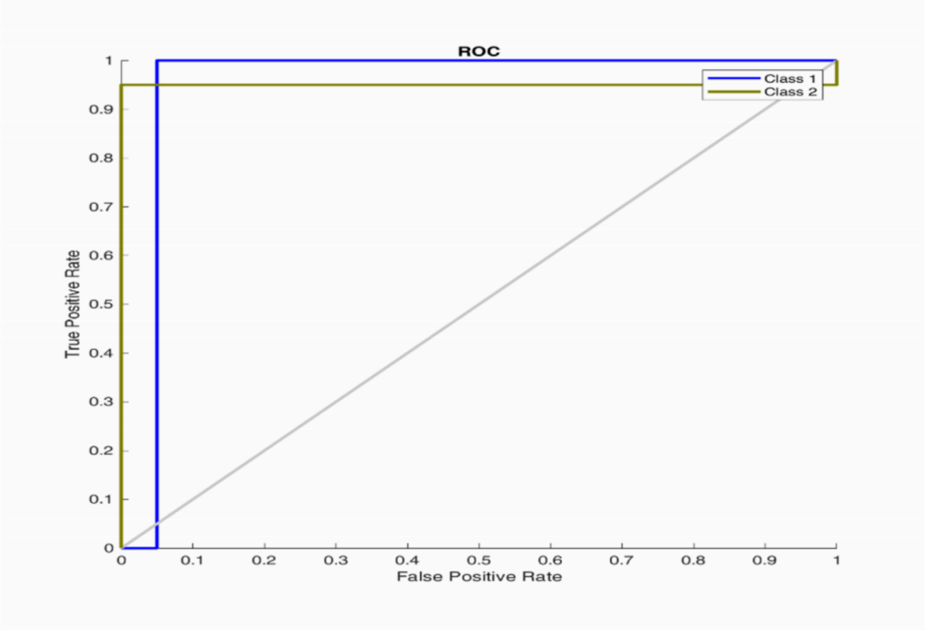

Figure 4.147: ROC Plot for Bat Algorithm 5 Population 30 Generation

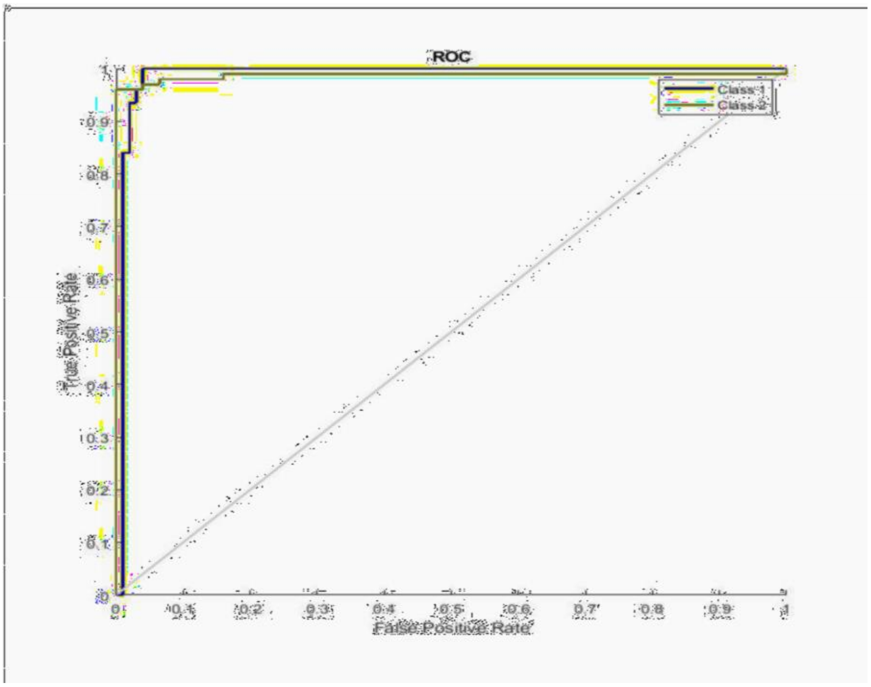

Figure 4.148: ROC Plot for Bat Algorithm 5 Population 35 Generation

4.8.4 Epilepsy classification using seven population and 5, 10, 15, 20, 25, 30, and 35 generation

Table 4.55 presents the F1 score, recall, accuracy, precision, and AUC. It shows that the accuracy varies between 95.6% to 97.80% as the generation increases from 5 to 35 for the BA-ANN classifier. The best accuracy is achieved at 97.80%, with 77.05% precision, 98.20 sensitivity, and F1 score of 97.62% and 97.90% AUC respectively. It is observed from Figure 4.154 that the ROC curve for 30 generations performs better than 5, 10, 15, 25, and 35 respectively, indicating a better performance of the BA-ANN classifier. The ROCs of various cases obtained using the BA-ANN classifier are plotted as shown in figures 4.149 to 4.155 for the representation of the obtained results for 5, 10, 15, 20, 25, 30, and 35 generations respectively. The classifier seems to function well in the real classes and vice versa.

Table 4.54: Bat Algorithm Metrics Using Seven Population and 5, 10, 15, 20, 25, 30, and 35 Generation

Number of Populations	Number of Generation	TP	FP	TN	FN	Time
[2 5 4 3 7 6 8]	5	400	22	78	0	51.731
[8 5 3 9 6 7 1]	10	400	22	78	0	77.790
[8 4 6 2 7 1 9]	15	400	13	87	0	110.359
[5 7 3 8 1 4 2]	20	400	9	91	0	307.944
[4 2 1 6 8 9 3]	25	400	16	84	0	190.649
[1 6 4 2 5 7 3]	30	400	11	89	0	266.320
[5 1 2 7 3 4 9]	35	395	15	85	5	511.739

Table 4.55: Bat Algorithm Performance Evaluation Using Seven Populations and 5, 10, 15, 20, 25, 30, and 35 Generation

Number of Populations	Number of Generation	Accuracy Score	Precision	Sensitivity	F1-	AUC
[2 5 4 3 7 6 8]	5	95.6	0.8969	0.9780	0.9357	0.9540
[8 5 3 9 6 7 1]	10	95.6	0.9421	0.9640	0.9529	0.9580
[8 4 6 2 7 1 9]	15	97.4	0.9684	0.9760	0.9722	0.9740
[5 7 3 8 1 4 2]	20	97.2	0.9578	0.9840	0.9707	0.9770
[4 2 1 6 8 9 3]	25	96.8	0.9415	0.9780	0.9594	0.9680
[1 6 4 2 5 7 3]	**30**	**97.8**	**0.9705**	**0.9820**	**0.9762**	**0.9790**
[5 1 2 7 3 4 9]	35	96.2	0.9609	0.9760	0.9684	0.9720

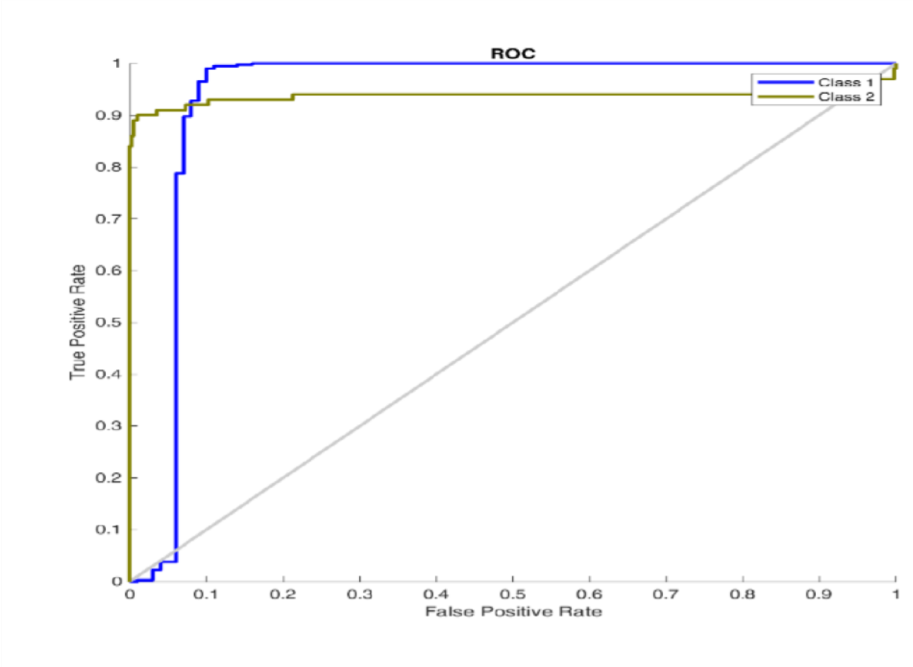

Figure 4.149: ROC Plot for Bat Algorithm 7 Population 5 Generation

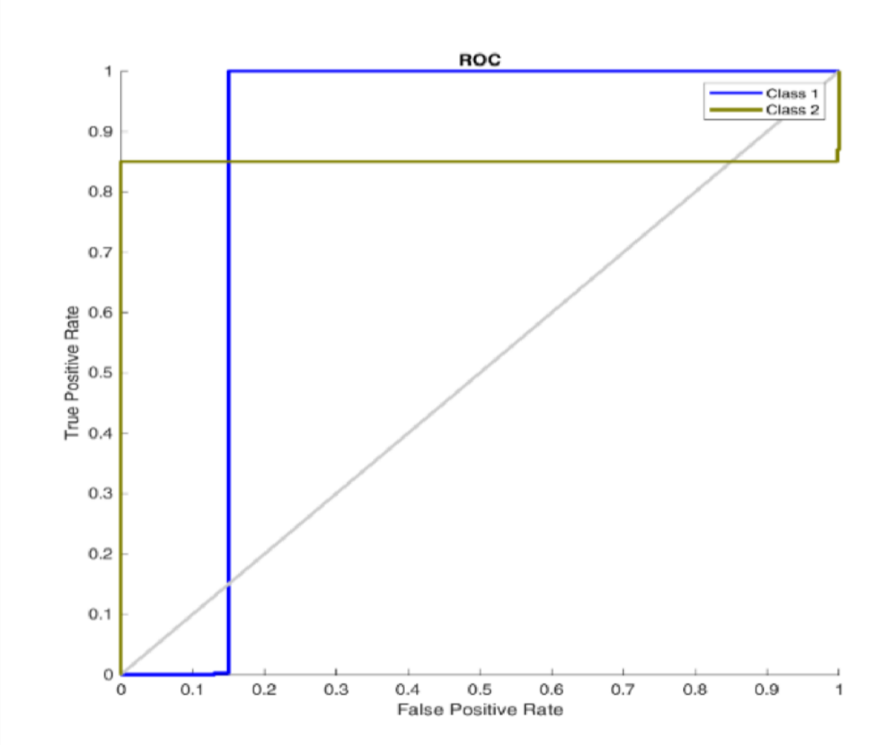

Figure 4.150: ROC Plot for Bat Algorithm 7 Population 10 Generation

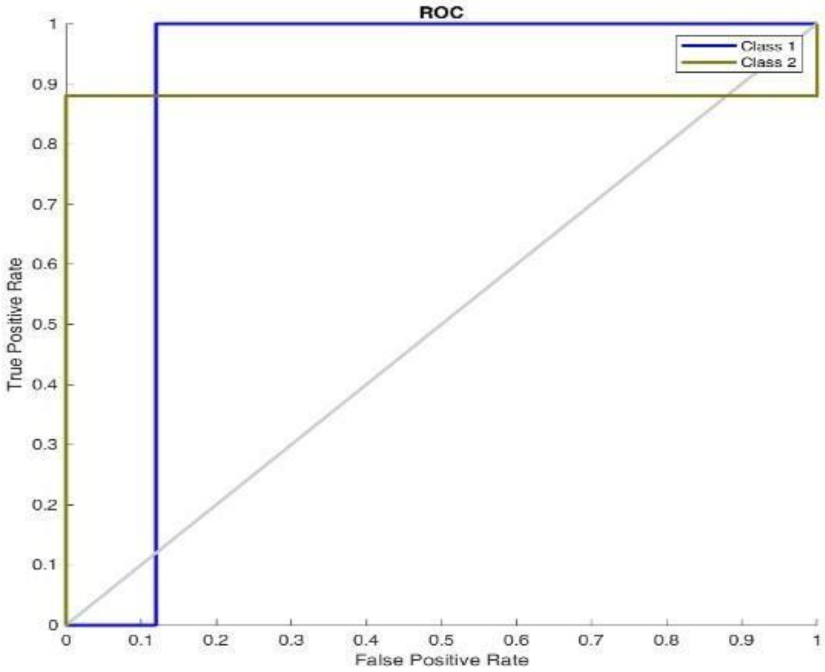

Figure 4.151: ROC Plot for Bat Algorithm 7 Population 15 Generation

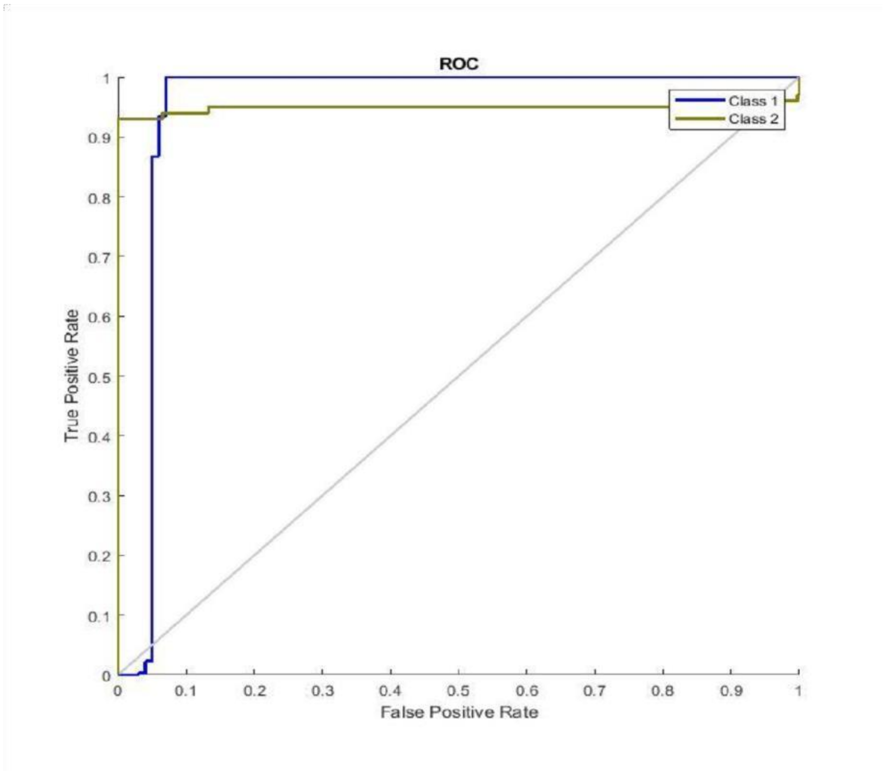

Figure 4.152: ROC Plot for Bat Algorithm 7 Population 20 Generation

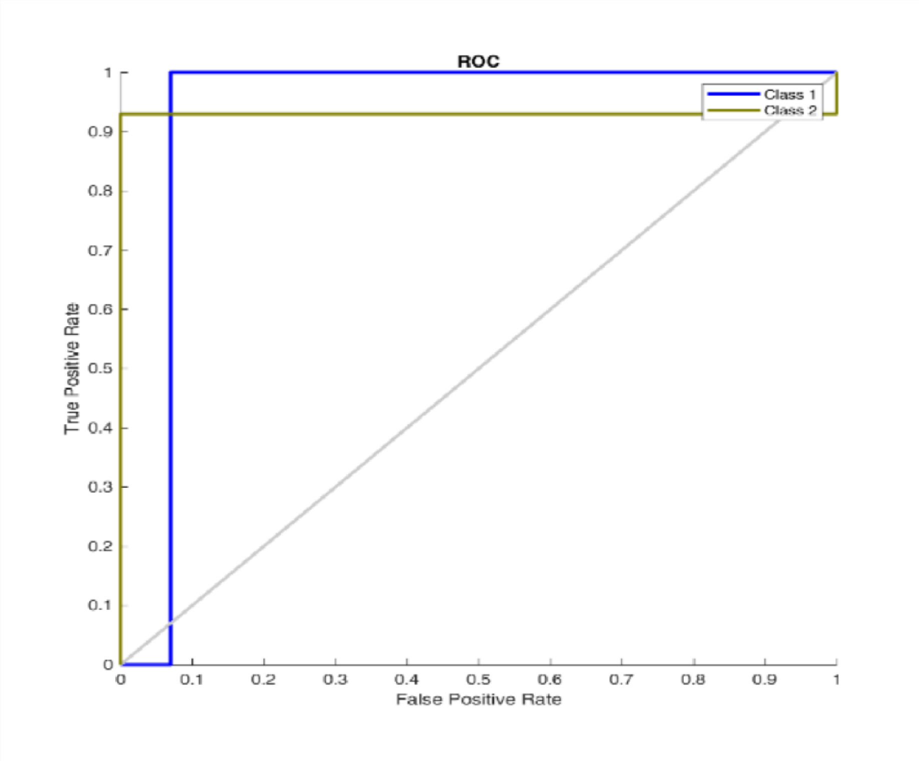

Figure 4.153: ROC Plot for Bat Algorithm 7 Population 25 Generation

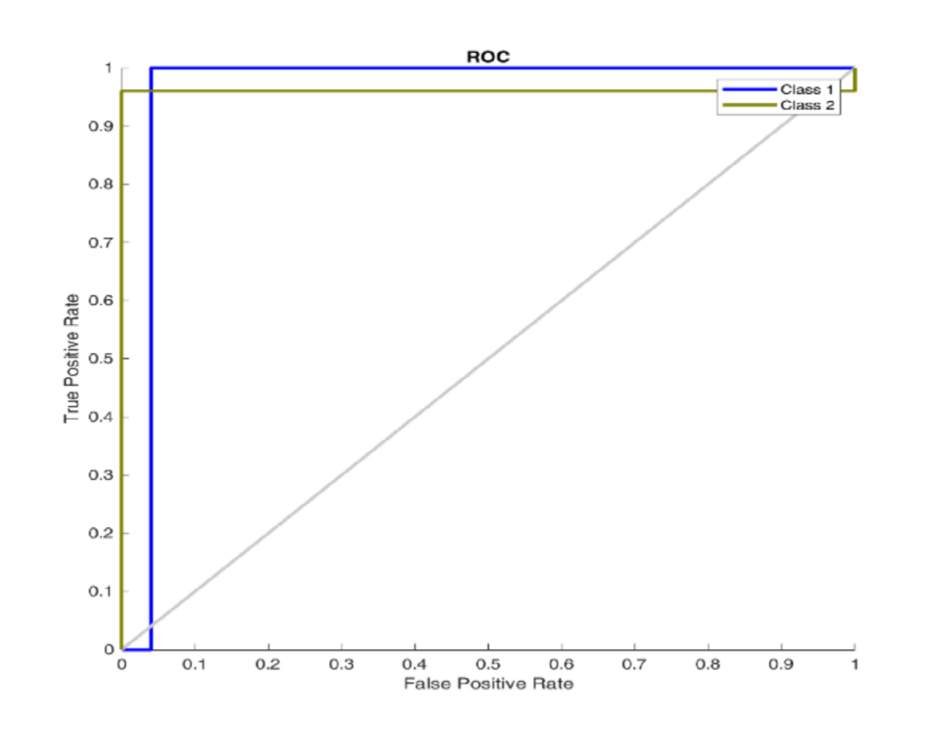

Figure 4.154: ROC Plot for Bat Algorithm 7 Population 30 Generation

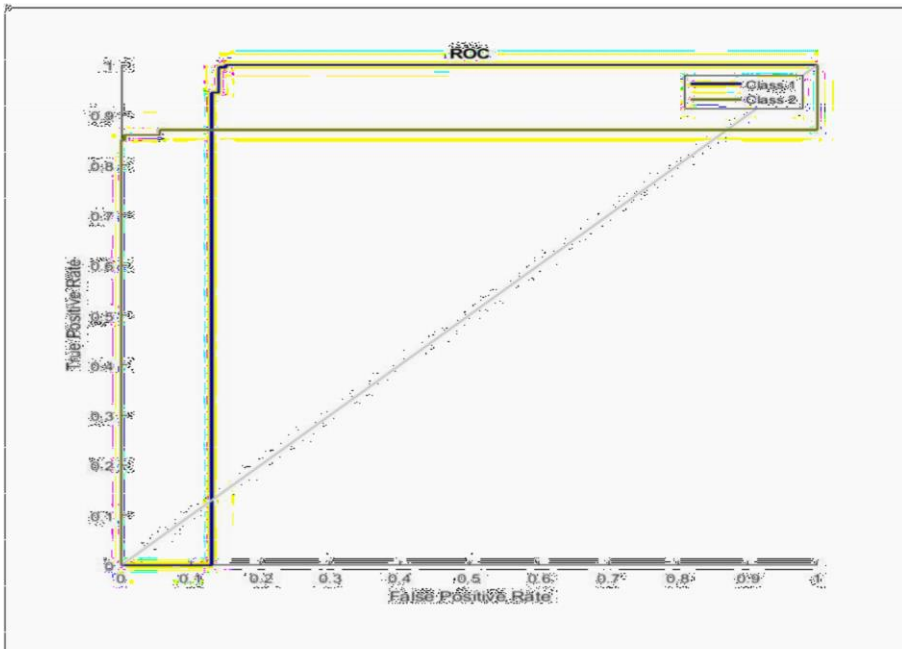

Figure 4.155: ROC Plot for Bat Algorithm 7 Population 35 Generation

4.8.5 Epilepsy classification using nine population and 5, 10, 15, 20, 25, 30, and 35 generation

Table 4.57 presents the F1 score, recall, accuracy, precision, and AUC. It shows that the accuracy varies between 95.20% to 98.00% as the generation increases from 5 to 35 for the BA-ANN classifier using nine features. The best accuracy is achieved at 98.00%, with 97.43% precision, 98.20 sensitivity, and an F1 score of 97.81% and 98.00% AUC respectively. It is observed from Figure 4.156 that the ROC curve for 5 generations performs better than 10, 15, 25, 30, and 35 respectively, indicating a better performance of the BA-ANN classifier. The ROCs of various cases obtained using the BA-ANN classifier are plotted as shown in Figures 4.156 to 4.162 for the representation of the obtained results for 5, 10, 15, 20, 25, 30, and 35 generations respectively.

Table 4.56: Bat Algorithm Metrics Using Nine Population and 5, 10, 15, 20, 25, 30, and 35 Generation

Number of Populations	Number of Generation	TP	FP	TN	FN	Time
[3 8 9 7 6 1 2 4 5]	5	400	10	90	0	47.742
[4 8 3 6 2 1 9 5 7]	10	400	17	83	0	77.244
[9 6 8 4 3 7 1 5 2]	15	400	16	84	0	111.205
[6 2 3 7 4 5 1 8 9]	20	400	17	83	0	267.117
[8 5 4 1 7 6 9 3 2]	25	400	24	76	0	162.168

[8 2 4 9 6 7 1 5 3]	30	400	15	85	0	403.347
[2 4 5 6 8 7 3 1 9]	35	400	8	92	0	465.755

Table 4.57: Bat Algorithm Performance Evaluation Using Nine Populations and 5, 10, 15, 20, 25, 30, and 35 Generation

Number of Populations	Number of Generation	Accuracy	Precision	Sensitivity	F1-Score	AUC
[3 8 9 7 6 1 2 4 5]	**5**	**98.0**	**0.9743**	**0.9820**	**0.9781**	**0.9800**
[4 8 3 6 2 1 9 5 7]	10	96.6	0.9427	0.9720	0.9572	0.9640
[9 6 8 4 3 7 1 5 2]	15	96.8	0.9551	0.9700	0.9625	0.9660
[6 2 3 7 4 5 1 8 9]	20	96.6	0.9551	0.9700	0.9625	0.9660
[8 5 4 1 7 6 9 3 2]	25	95.2	0.9340	0.9520	0.9429	0.9470
[8 2 4 9 6 7 1 5 3]	30	97.0	0.9517	0.9740	0.9627	0.9680
[2 4 5 6 8 7 3 1 9]	35	97.4	0.9701	0.9740	0.9721	0.9730

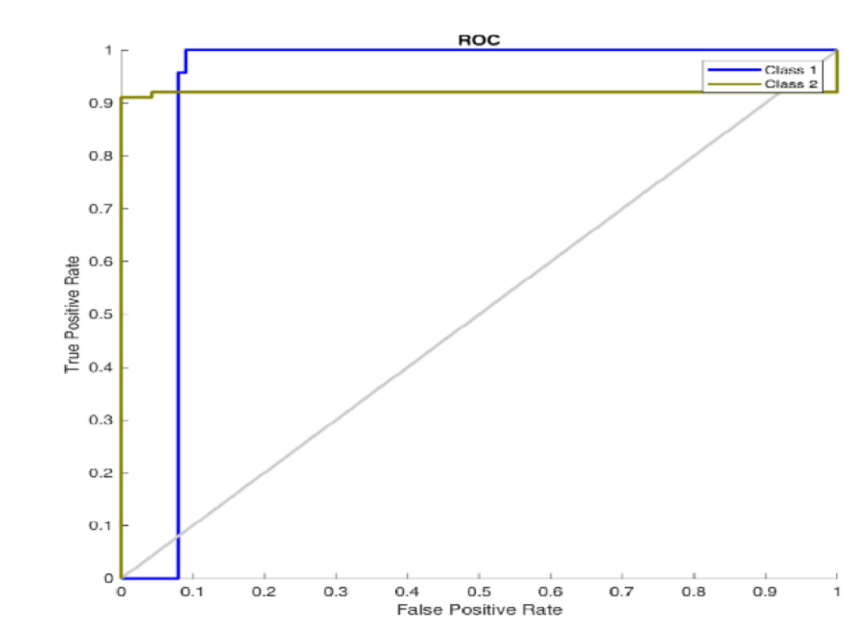

Figure 4.156: ROC Plot for Bat Algorithm 9 Population 5 Generation

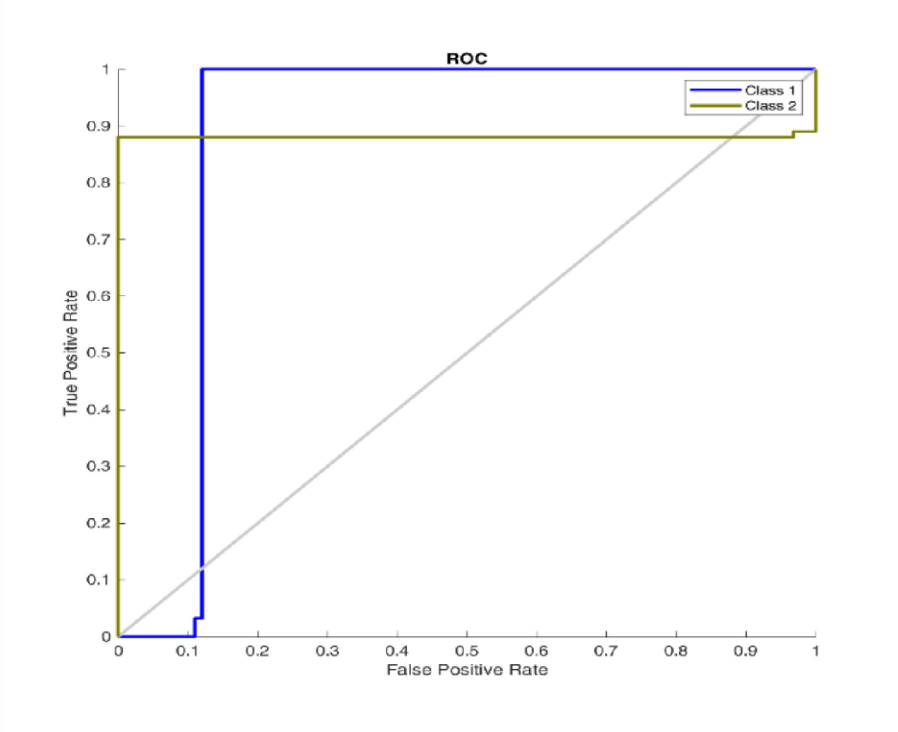

Figure 4.157: ROC Plot for Bat Algorithm 9 Population 10 Generation

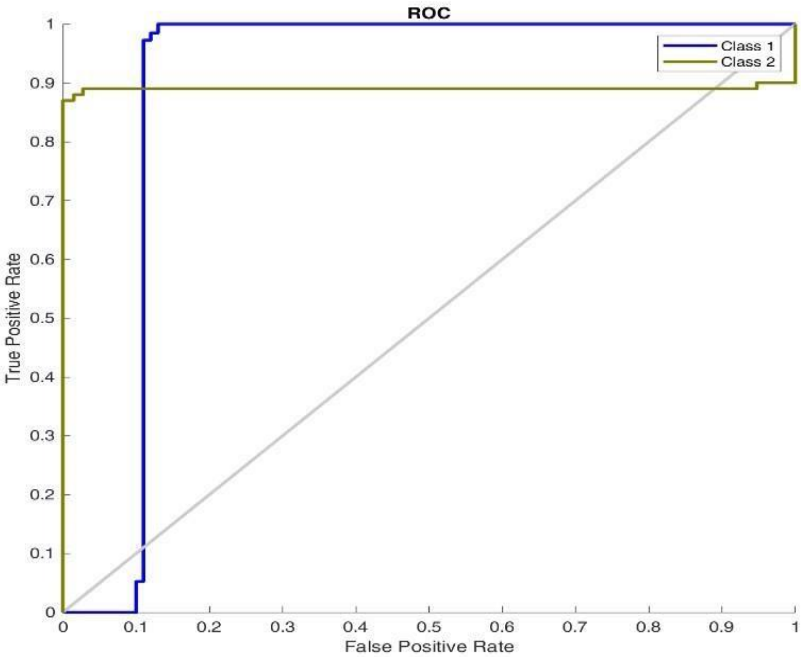

Figure 4.158: ROC Plot for Bat Algorithm 9 Population 15 Generation

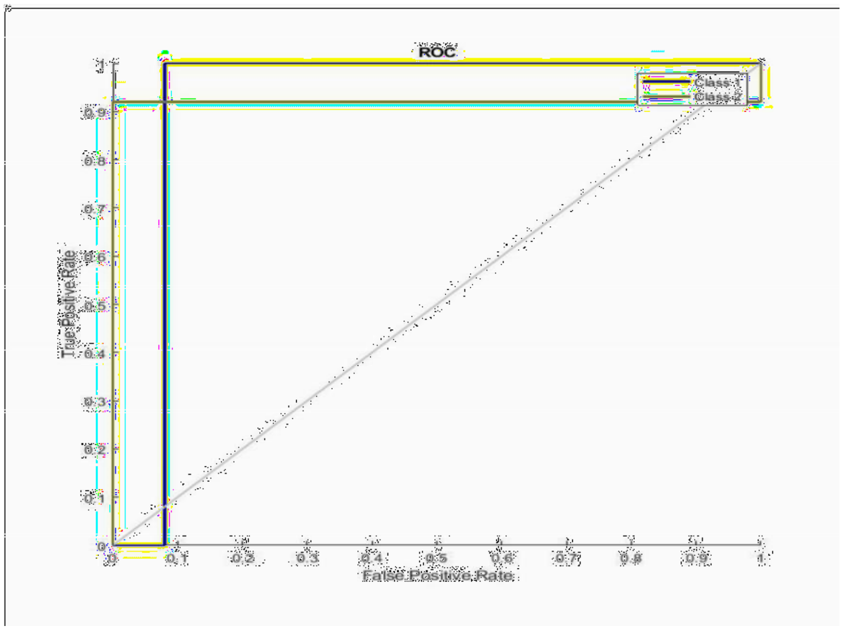

Figure 4.159: ROC Plot for Bat Algorithm 9 Population 20 Generation

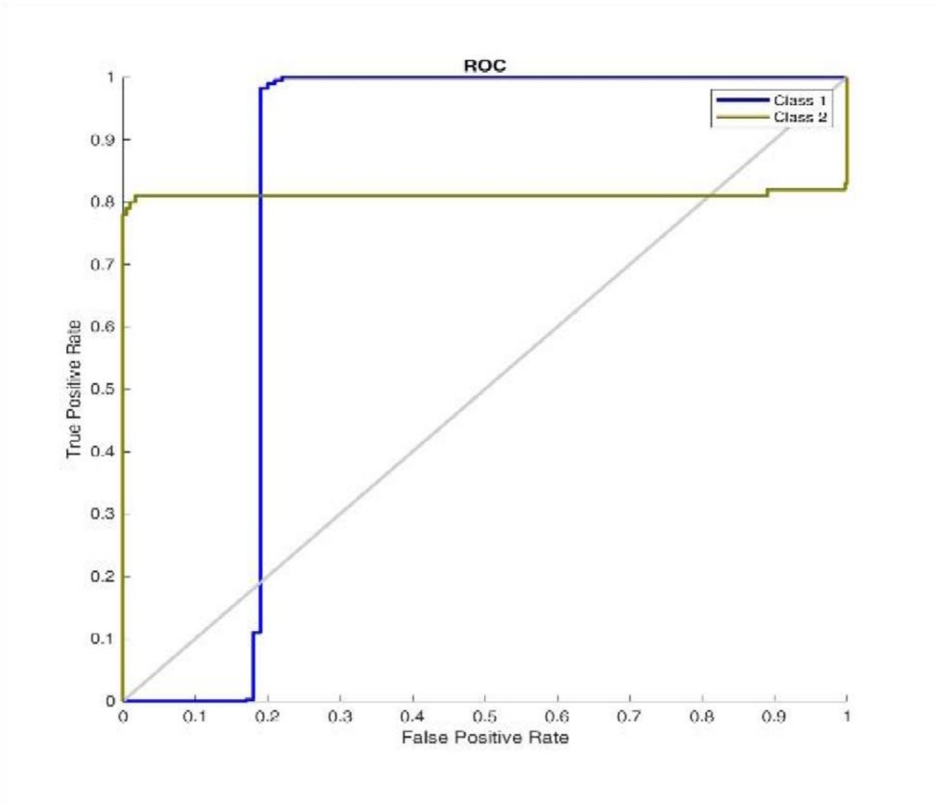

Figure 4.160: ROC Plot for Bat Algorithm 9 Population 25 Generation

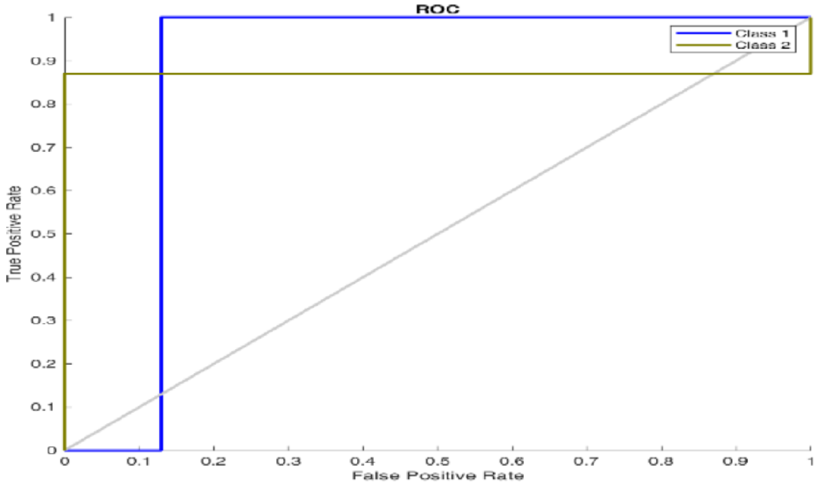

Figure 4.161: ROC Plot for Bat Algorithm 9 Population 30 Generation

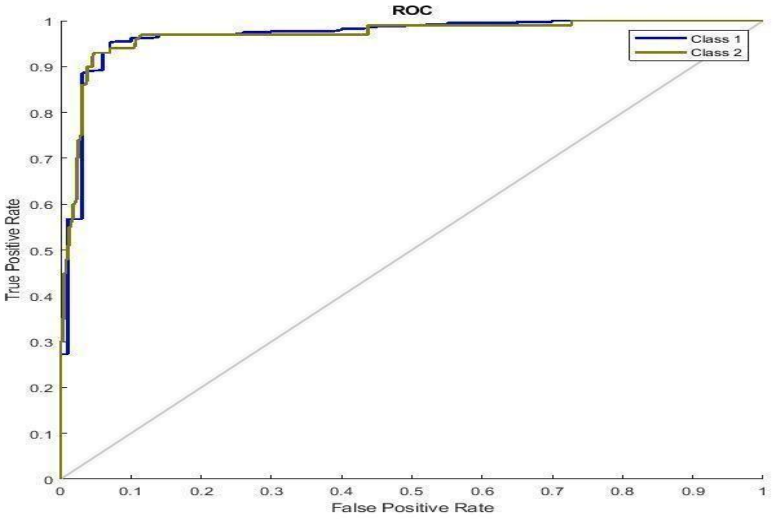

Figure 4.162: ROC Plot for Bat Algorithm 9 Population 35 Generation

Table 4.58 present the evaluation methods for BA-ANN using one, three, five, seven, and nine populations respectively for 5, 10, 15, 20, 25, 30, and 35 generations. The accuracy, precision, recall, F1 score, and AUC for various feature classes are shown as well. It shows that the best accuracy is achieved between using three populations and five-generation respectively. The best accuracy is achieved at 98.80%, with 76.10% precision, 99.20% sensitivity, and F1 score of 86.13% and 90.02% AUC respectively for 15 generations, and three populations for the BA-ANN algorithm. It is observed from Figure 4.163 that the accuracy is near 100%, indicating a strong performance of the BA-ANN classifier using one population. In the real classes, the classifier performs well, and vice versa for three features when compared to other classes of the feature's extraction.

Table 4.58: Summary of Epilepsy Detection Using Bat Algorithm with Various Generation

Evaluation Metrics	1 Population 15	3 Population 15 Generation	5 Population 15 Generation	7 Population 30 Generation	9 Population 5 Generation
Accuracy	**0.988**	0.982	0.984	0.978	0.98
Precision	**0.761**	0.9014	0.967	0.9705	0.9743
Sensitivity	**0.992**	0.99	0.986	0.982	0.982
F1 Measure	**0.8613**	0.9436	0.9764	0.9762	0.9781
AUC	**0.902**	0.964	0.981	0.979	0.98

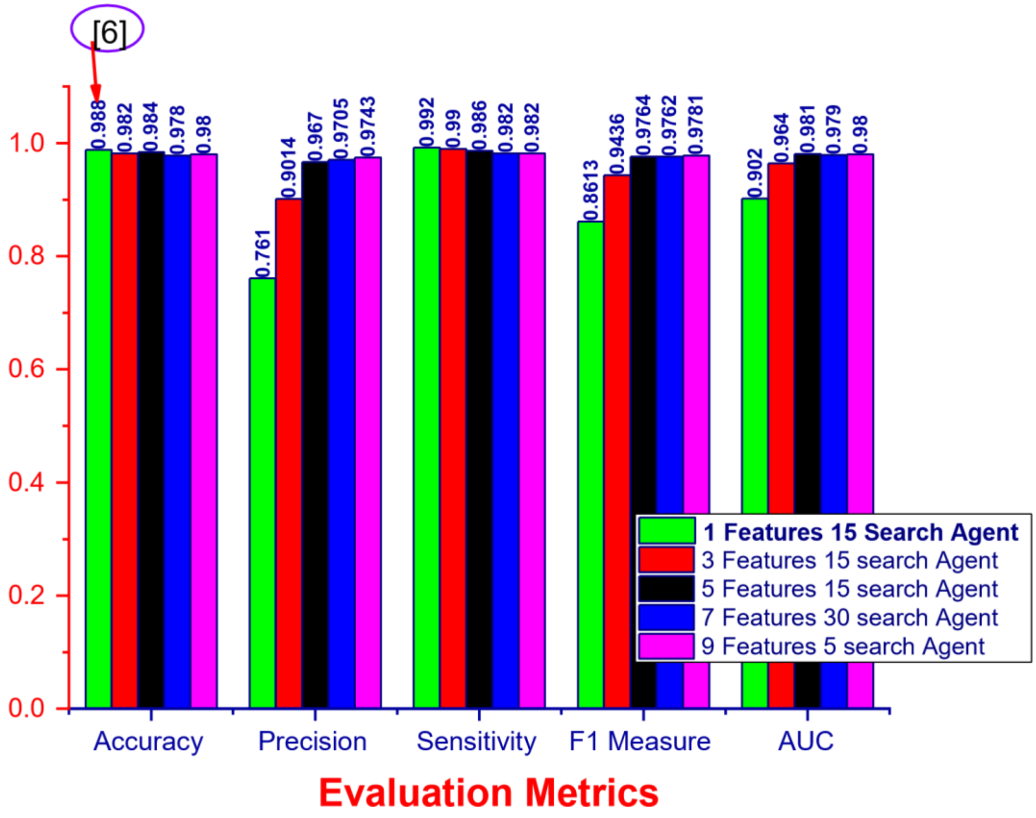

Figure 4.163: Bat Algorithm Performance with Various Generation

4.9 Epilepsy Classification Using Particle Swarm Optimization (PSO)

This section outlines the outcomes of the PSO algorithm's epilepsy detection and classification utilizing the artificial neural network and various features extracted from the populations of 1, 3,5, 7, and 9, and 5, 10, 15, 20, 25, 30, and 35, respectively.

4.9.1. Epilepsy classification using one feature and 5, 10, 15, 20, 25, 30, and 35 population

Tables 4.59 and 4.60 are the evaluation methods for PSO-ANN using one feature and 5, 10. 15, 25, 35, and 30 population respectively. The classifier predicted the true negative classes correctly and predicted 13,17,15, 89, and 24 of the false-positive classes incorrectly, and 5, 5, 8, 1, and 1 false-negative incorrectly for 5, 10, 15, 20, 25, 30, and 35 populations respectively. Table 4.60 presents the F1 score, recall, accuracy, precision, and AUC. It shows that the accuracy varies between 95.0% to 96.40% as the population increases from 5 to 35 for the PSO-ANN classifier. The best accuracy is achieved at 96.40%, with 86.45% precision, 99.20 sensitivity, F1 score of 92.39%, and 95.20% AUC respectively. It is observed from Figure 4.164 that the ROC curve for 5 populations performs better than 10,

15, 25, 30, and 35 respectively, indicating a better performance of the PSO-ANN classifier. The ROCs of various cases obtained using the PSO-ANN classifier are plotted as shown in Figures 4.164 to 4.170 for the representation of the obtained results for 5, 10, 15, 20, 25, 30, and 35 populations respectively.

Table 4.59: PSO Algorithm Metrics Using One Feature and 5, 10, 15, 20, 25, 30, and 35 Generation

Number of Features	Number of Population	TP	FP	TN	FN	Time (s)
[3]	5	395	13	87	5	33.754
[6]	10	395	17	83	5	60.639
[7]	15	392	15	85	8	129.721
[1]	20	386	14	86	14	186.581
[2]	25	399	89	11	1	153.233
[2]	30	399	24	76	1	174.68
[2]	35	399	89	11	1	330.410

Table 4.60: PSO Algorithm Performance Evaluation Using One Feature and 5, 10, 15, 20, 25, 30, and 35 Generation

Number of Features	Number of Population	Accuracy	Precision	Sensitivity	F1-Score	AUC
[3]	**5**	**96.4**	**0.8645**	**0.9920**	**0.9239**	**0.9520**
[6]	10	95.6	0.7610	0.8920	0.8613	0.9020
[7]	15	95.4	0.7824	0.9980	0.8772	0.9180
[1]	20	94.4	0.7539	0.9860	0.8544	0.8940

[2]	25	82.0	0.5570	0.9380	0.6989	0.6540
[2]	30	95.0	0.8353	0.9095	0.8403	0.9440
[2]	35	82.0	0.5570	0.9380	0.6989	0.6570

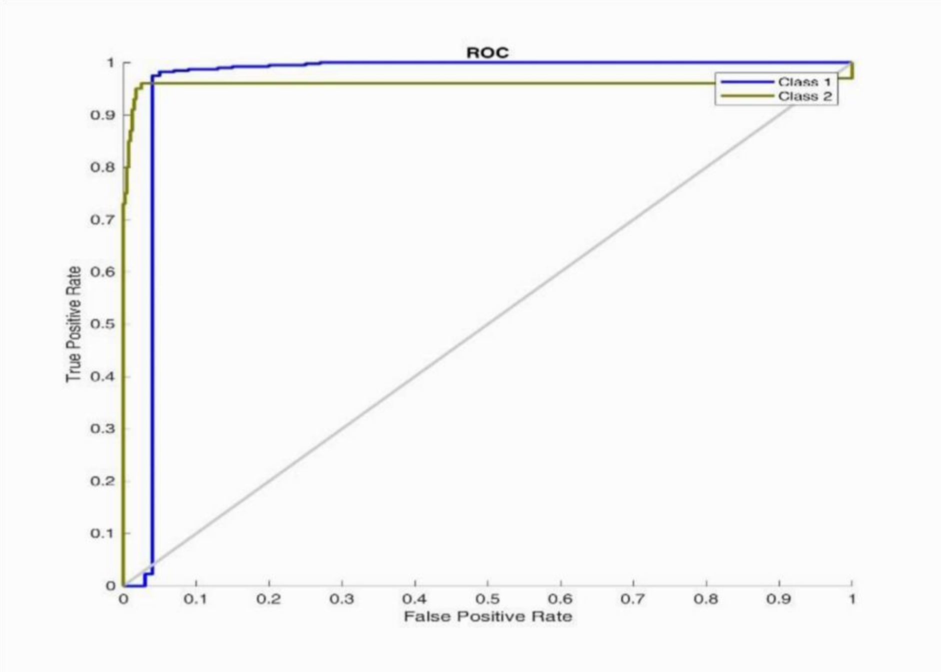

Figure 4.164: ROC Plot for PSO Algorithm 1 Feature 5 Generation

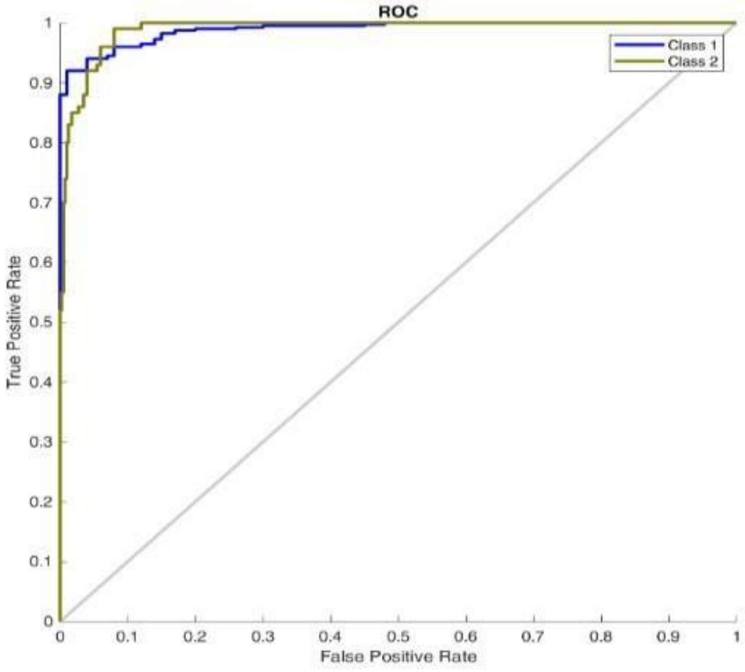

Figure 4.165: ROC Plot for PSO Algorithm 1 Feature 10 Generation

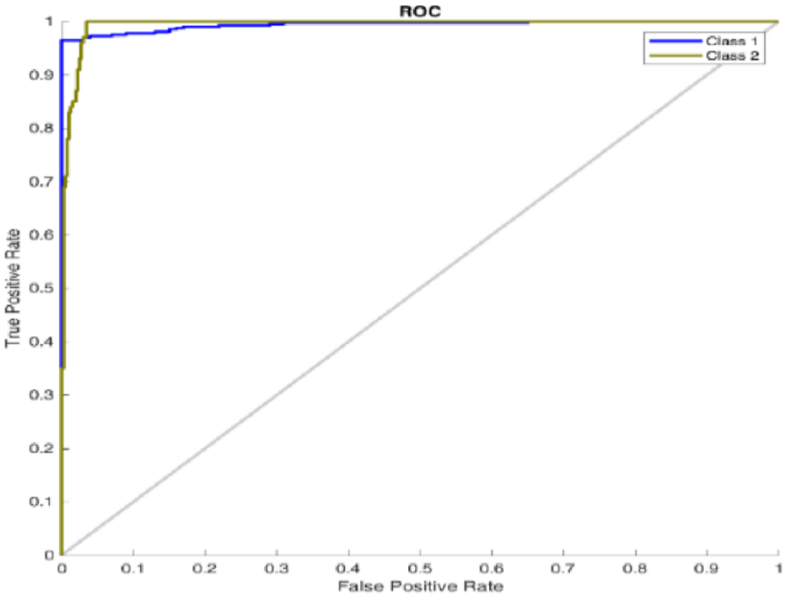

Figure 4.166: ROC Plot for PSO Algorithm 1 Feature 15 Generation

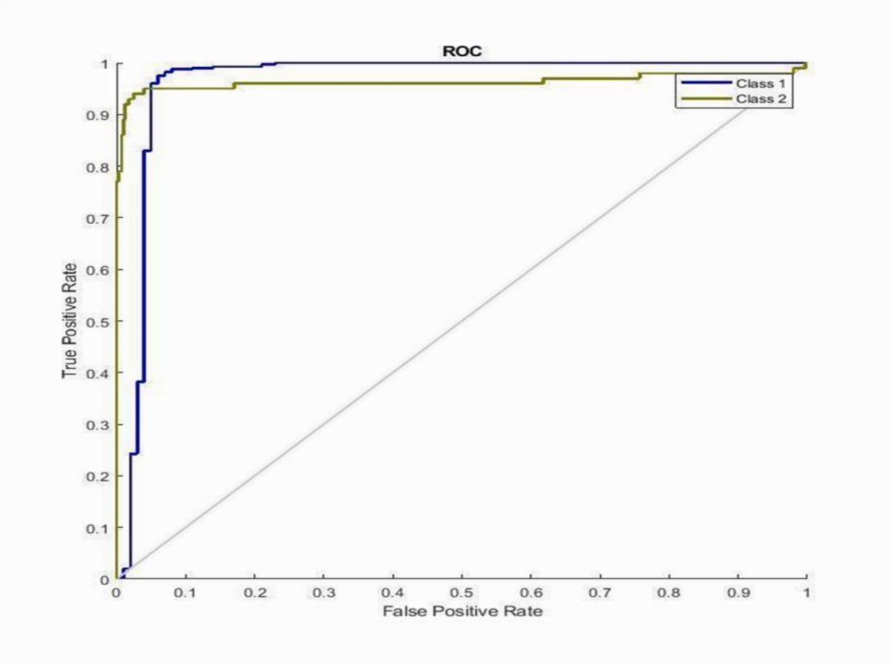

Figure 4.167: ROC Plot for PSO Algorithm 1 Feature 20 Generation

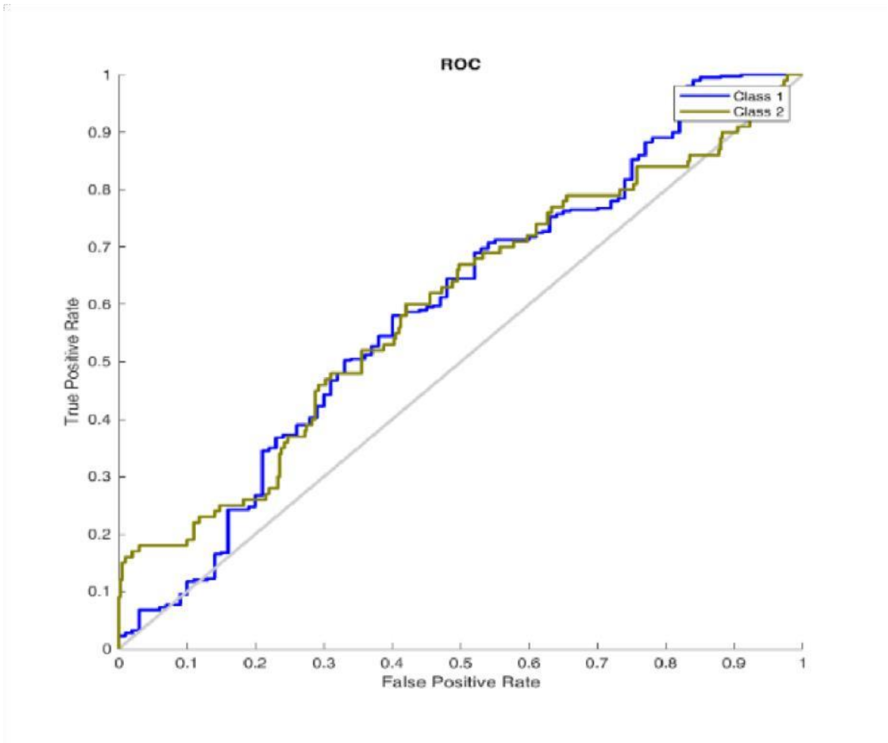

Figure 4.168: ROC Plot for PSO Algorithm 1 Feature 25 Generation

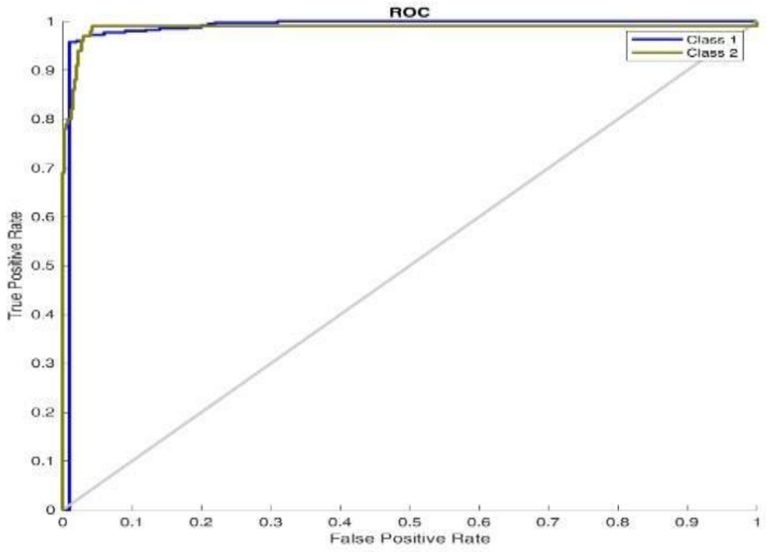

Figure 4.169: ROC Plot for PSO Algorithm 1 Feature 30 Generation

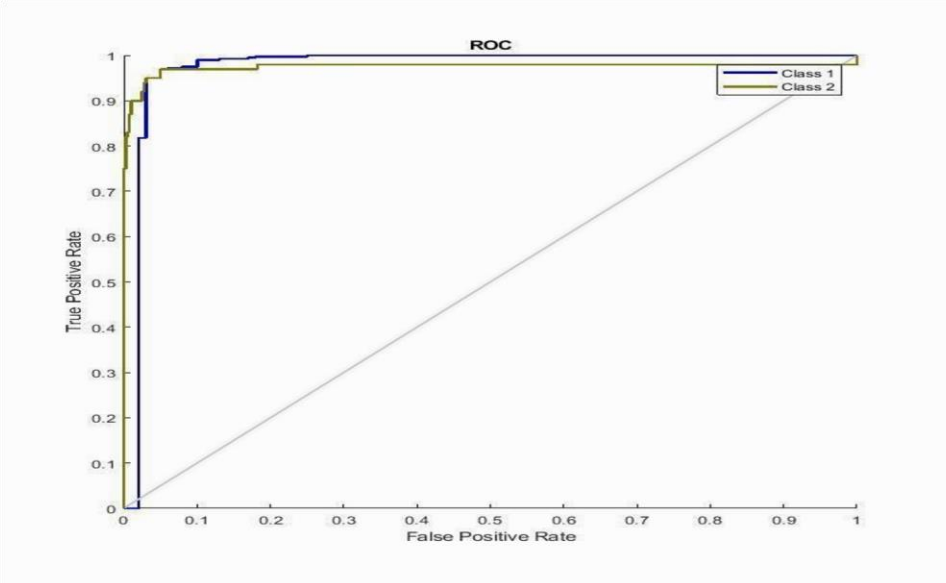

Figure 4.170: ROC Plot for PSO Algorithm 1 Feature 35 Generation

4.9.2 Epilepsy classification using three features and 5, 10, 15, 20, 25, 30, and 35 population

Table 4.62 presents the F1 score, recall, accuracy, precision, and AUC. It shows that the accuracy varies between 96.40% to 99.00% as the population increases from 5 to 35 for the PSO-ANN classifier. The best accuracy is achieved at 99.00%, with 91.70% precision, 99.40 sensitivity, and an F1 score of 95.39% and 97.20% AUC respectively. It is observed from Figure 4.175 that the ROC curve for 5 populations performs better than 10, 15, 25, 30, and 35 respectively, indicating a better performance of the PSO-ANN classifier. The ROCs of various cases obtained using the PSO-ANN classifier are plotted as shown in figures 4.171 to 4.177 for the representation of the obtained results for 5, 10, 15, 20, 25, 30, and 35 populations respectively.

Table 4.61: PSO Algorithm Metrics Using Three Features and 5, 10, 15, 20, 25, 30, and 35 Generation

Number of Features	Number of Population	TP	FP	TN	FN	Time (s)
[6 4 5]	5	399	4	96	1	48.002
[9 6 5]	10	398	6	94	2	81.215
[1 3 9]	15	399	11	89	1	131.477
[8 2 5]	20	400	25	75	0	303.368
[6 2 1]	25	399	10	90	1	215.311
[7 6 9]	30	396	14	86	4	243.702
[9 1 5]	35	397	17	83	3	512.624

Table 4.62: PSO Algorithm Performance Evaluation Using Three Features and 5,10, 15, 25, 30, 35 Generation

Number of Features	Number of Population	Accuracy	Precision	Sensitivity	F1-Score	AUC
[6 4 5]	**5**	**99.0**	**0.9170**	**0.9940**	**0.9539**	**0.9720**
[9 6 5]	10	98.4	0.8431	0.9960	0.9132	0.9460
[1 3 9]	15	97.6	0.8836	0.9860	0.9320	0.9550
[8 2 5]	20	95.0	0.8212	0.9880	0.8969	0.9310
[6 2 1]	25	97.8	0.8766	0.9940	0.9316	0.9580
[7 6 9]	30	96.4	0.8487	0.9940	0.9156	0.9470
[9 1 5]	35	96.0	0.8504	0.9900	0.9149	0.9450

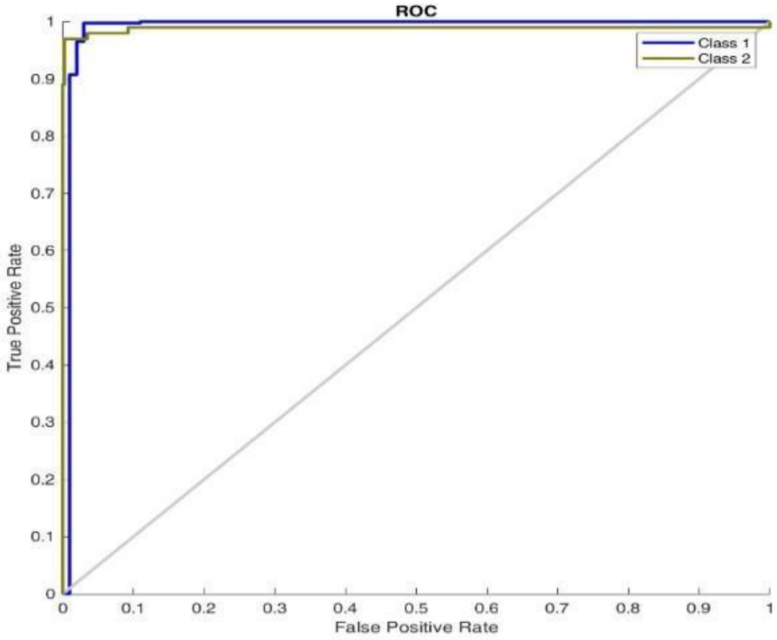

Figure 4.171: ROC Plot for PSO Algorithm 3 Feature 5 Generation

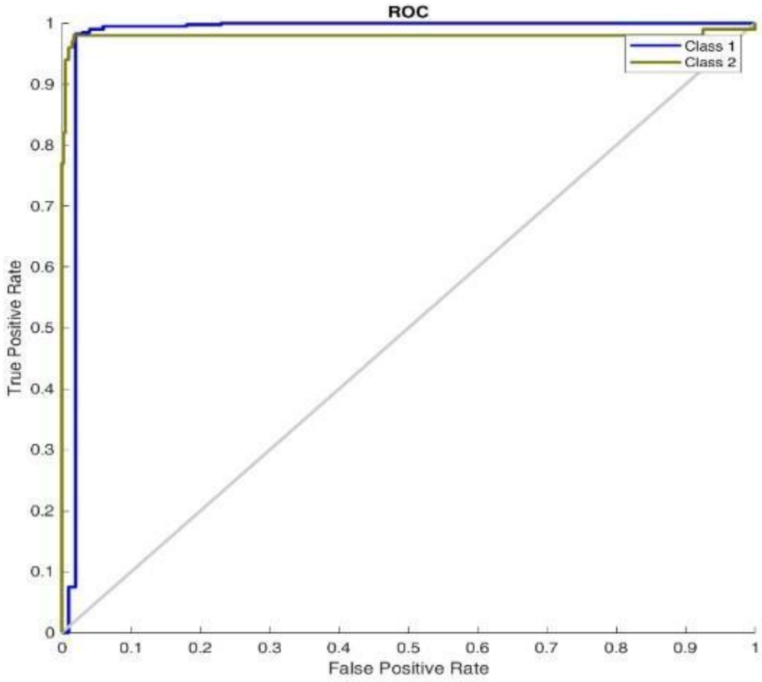

Figure 4.172: ROC Plot for PSO Algorithm 3 Feature 10 Generation

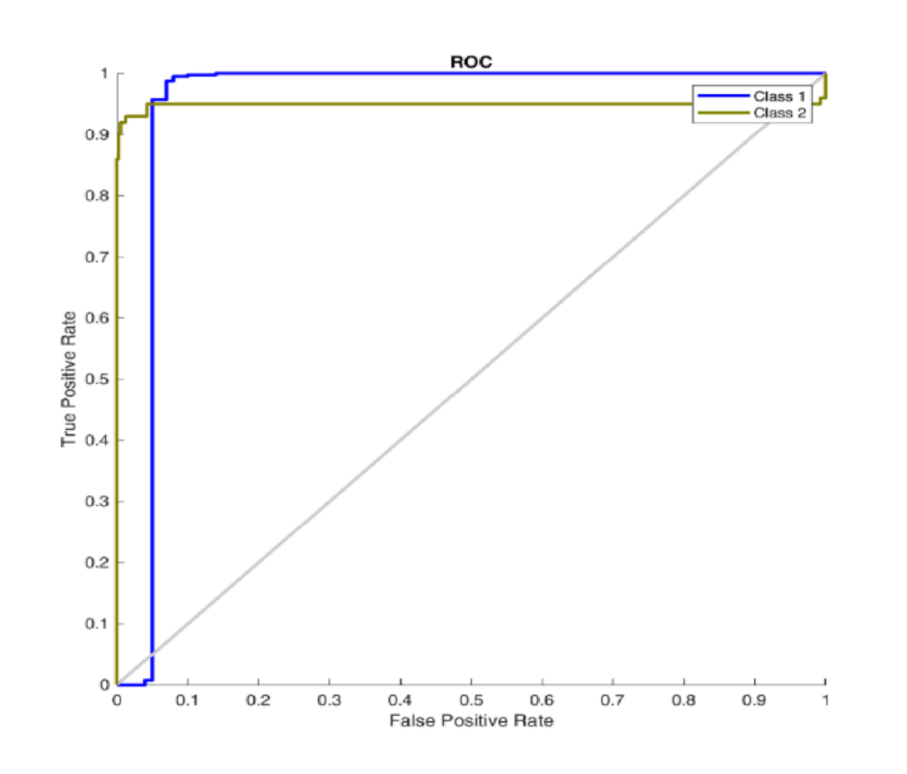

Figure 4.173: ROC Plot for PSO Algorithm 3 Feature 15 Generation

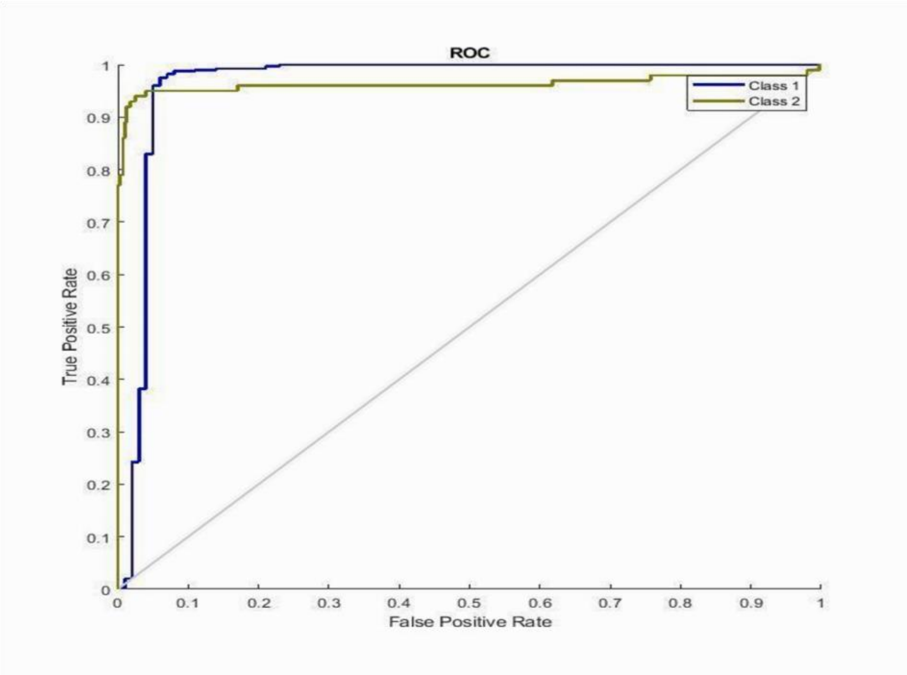

Figure 4.174: ROC Plot for PSO Algorithm 3 Feature 20 Generation

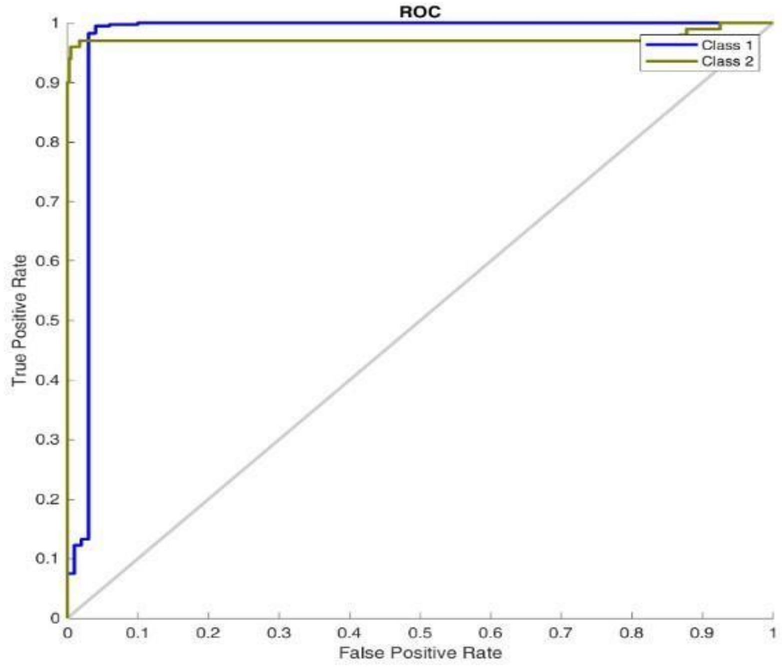

Figure 4.175: ROC Plot for PSO Algorithm 3 Feature 25 Generation

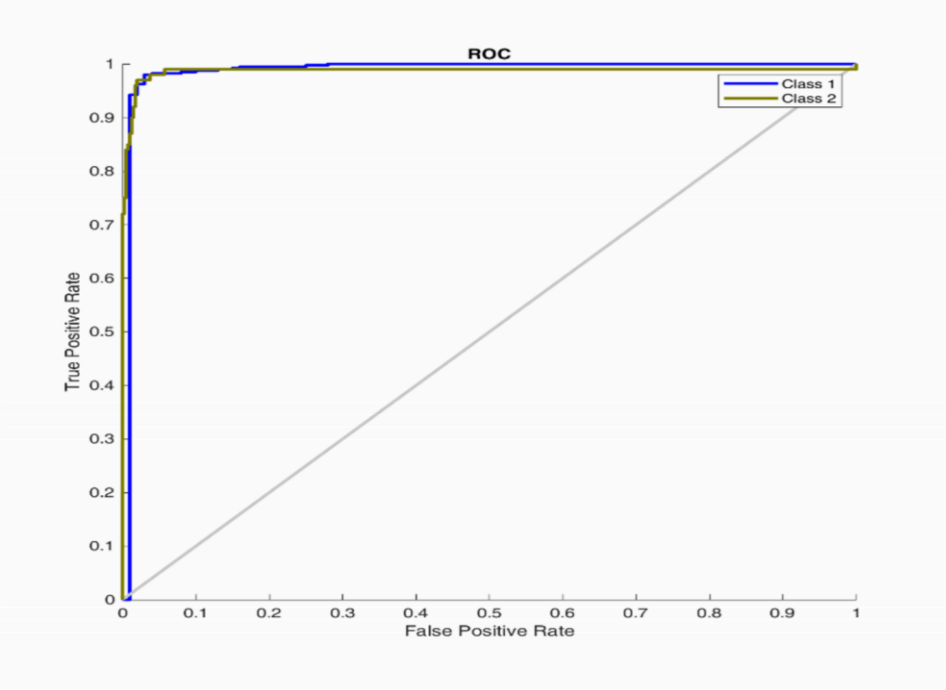

Figure 4.176: ROC Plot for PSO Algorithm 3 Feature 30 Generation

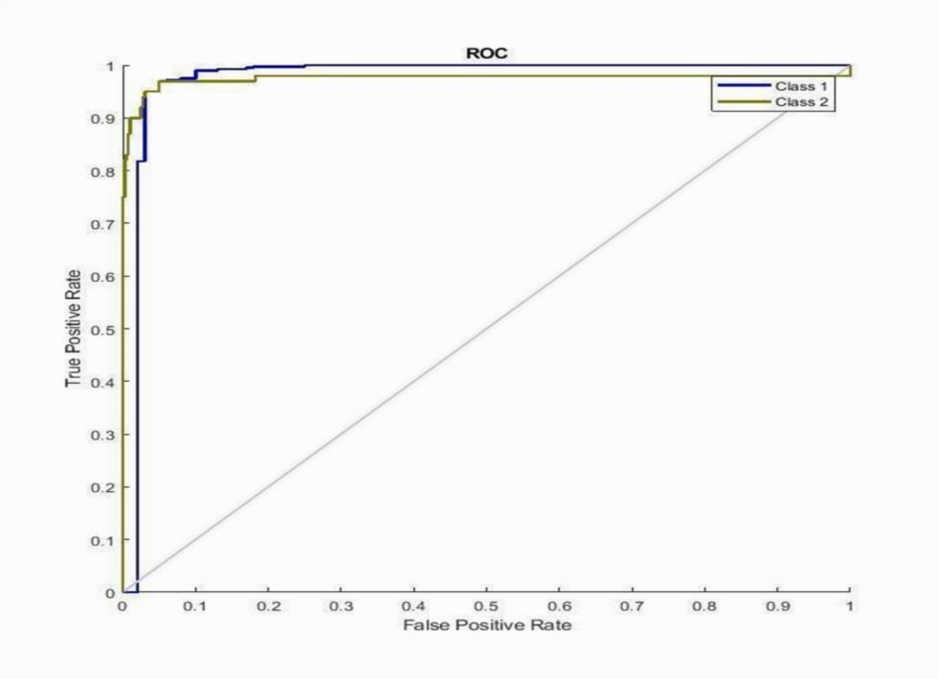

Figure 4.177: ROC Plot for PSO Algorithm 3 Feature 35 Generation

4.9.3 Epilepsy classification using five features and 5, 10, 15, 20, 25, 30, and 35 population

Table 4.64 presents the F1 score, recall, accuracy, precision, and AUC. It shows that the accuracy varies between 96.80% to 97.60% as the population increases from 5 to 35 for the PSO-ANN classifier. The best accuracy is achieved at 97.60%, with 84.63% precision,

99.40 sensitivity, F1 score of 91.43%, and 94.60% AUC respectively. It is observed from Figure 4.178 that the ROC curve for the 30 population performs better than 5, 10, 15, 25, and 30 respectively, indicating a better performance of the PSO-ANN classifier. The ROCs of various cases obtained using the PSO-ANN classifier are plotted as shown in Figures 4.178 to 4.184 for the representation of the obtained results for 5, 10, 15, 20, 25, 30, and 35 populations respectively.

Table 4.63: PSO Algorithm Metrics Using Five Features and 5, 10, 15, 20, 25, 30 and 35 Generation

Number of Features	Number of Population	TP	FP	TN	FN	Time (s)
[4 9 5 2 6]	5	400	16	84	0	79.670
[7 6 4 1 3]	10	400	14	86	0	202.598
[6 5 2 1 3]	15	400	19	81	0	303.025
[2 4 1 6 5]	20	399	14	86	1	312.509
[5 6 4 7 2]	25	400	24	76	0	414.427

[8 5 6 4 2]	30	400	12	88	0	596.900
[8 1 4 9 5]	35	400	8	92	0	579.920

Table 4.64: PSO Algorithm Performance Evaluation Using Five Features and 5, 10, 15, 20, 25, 30, and 35 Generation

Number of Features	Number of Population	Accuracy	Precision	Sensitivity	F1 Score	AUC
[4 9 5 2 6]	5	96.8	0.8625	0.9810	0.9175	0.9430
[7 6 4 1 3]	10	97.2	0.9702	0.9740	0.9721	0.9730
[6 5 2 1 3]	15	96.2	0.9476	0.9660	0.9567	0.9610
[2 4 1 6 5]	20	97.0	0.8415	0.9940	0.9114	0.9440
[5 6 4 7 2]	25	95.2	0.8830	0.9720	0.9253	0.9450
[8 5 6 4 2]	**30**	**97.6**	**0.8463**	**0.9940**	**0.9143**	**0.9460**
[8 1 4 9 5]	35	97.4	0.9599	0.9900	0.9747	0.9820

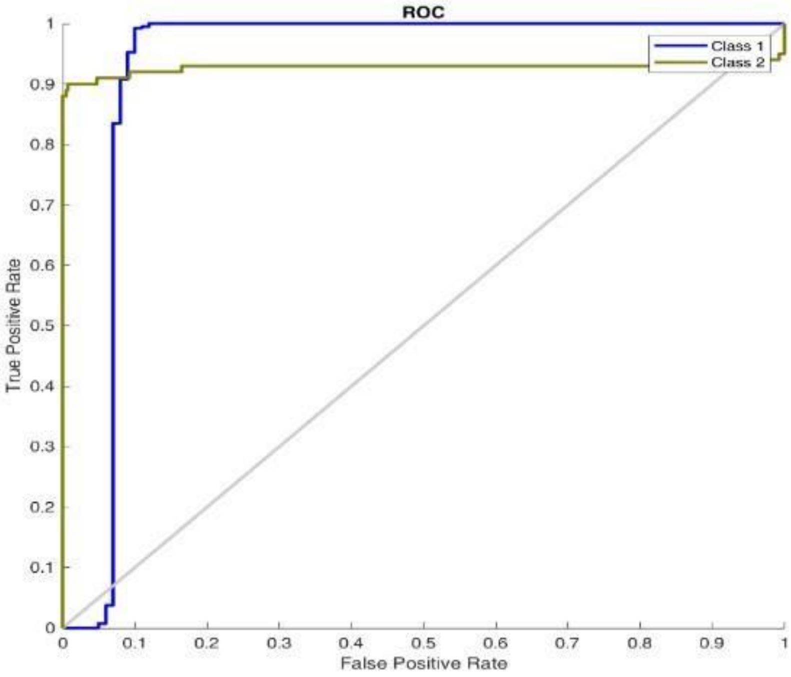

Figure 4.178: ROC Plot for PSO Algorithm 5 Feature 5 Generation

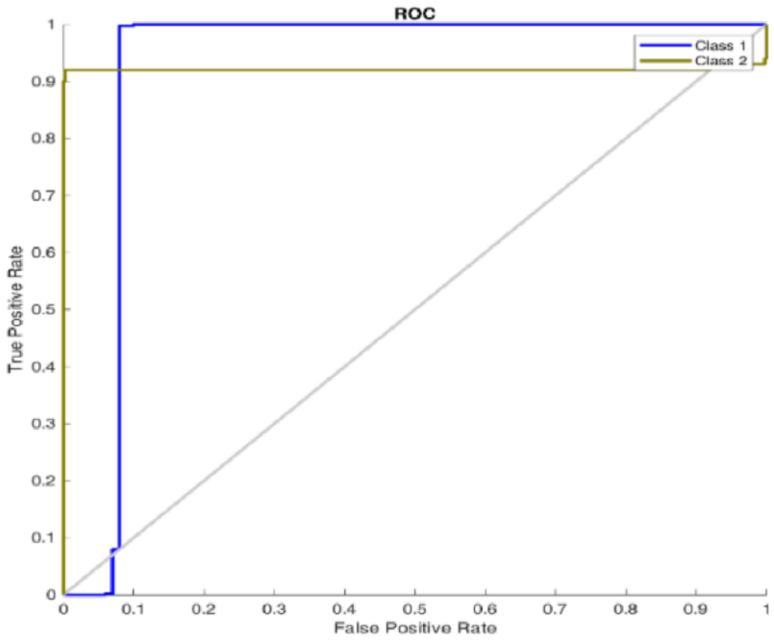

Figure 4.179: ROC Plot for PSO Algorithm 5 Feature 10 Generation

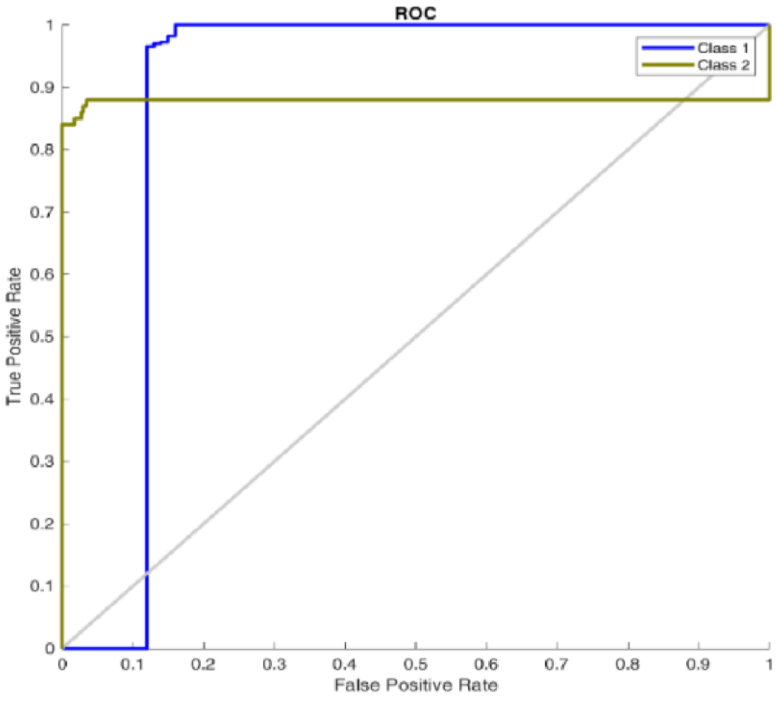

Figure 4.180: ROC Plot for PSO Algorithm 5 Feature 15 Generation

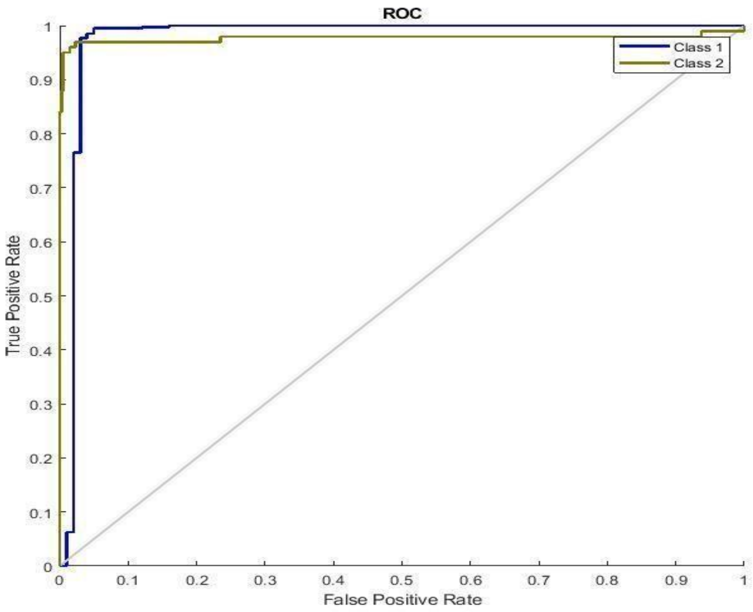

Figure 4.181: ROC Plot for PSO Algorithm 5 Feature 20 Generation

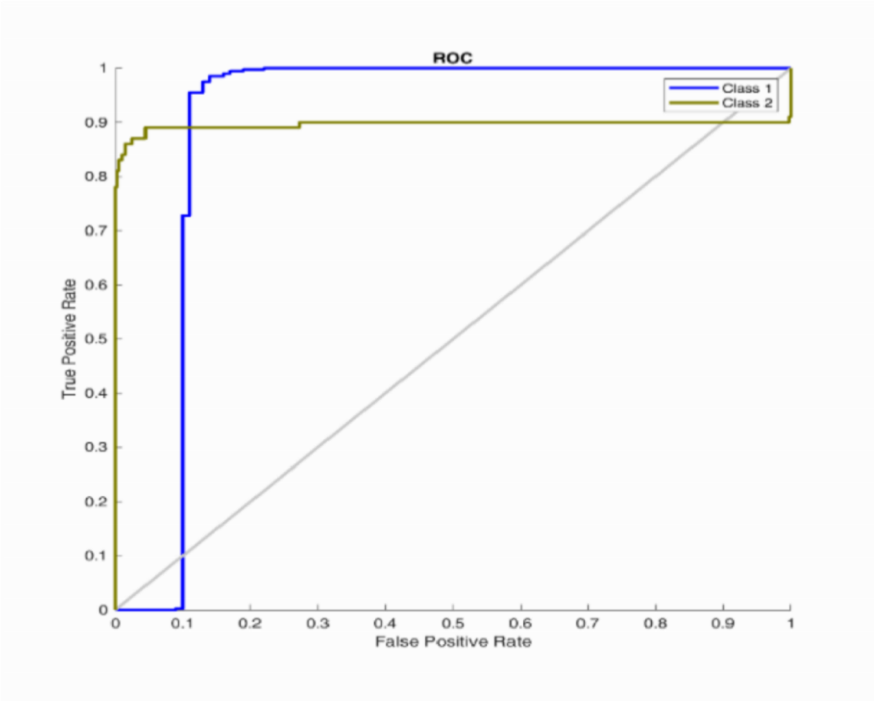

Figure 4.182: ROC Plot for PSO Algorithm 5 Feature 25 Generation

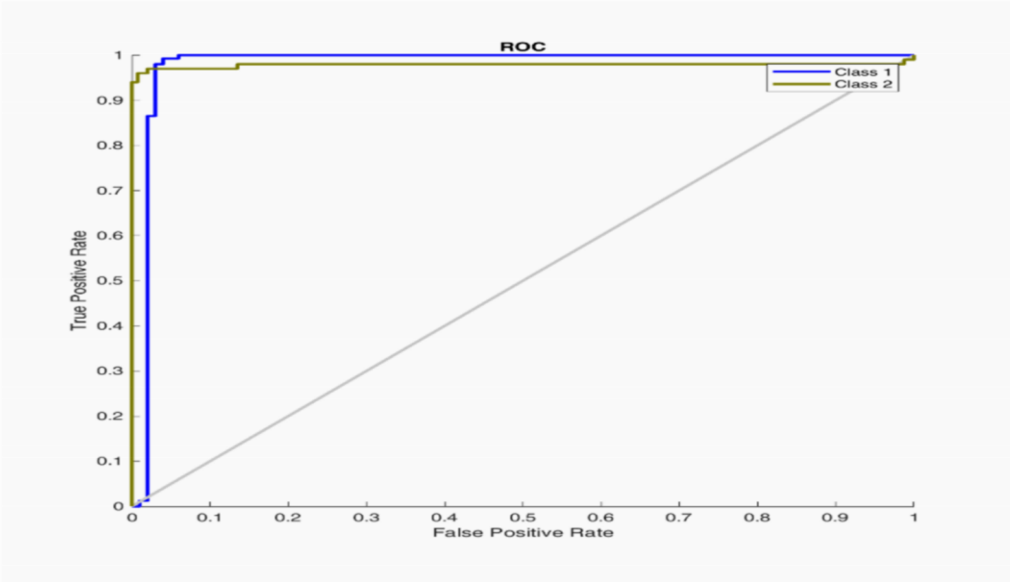

Figure 4.183: ROC Plot for PSO Algorithm 5 Feature 30 Generation

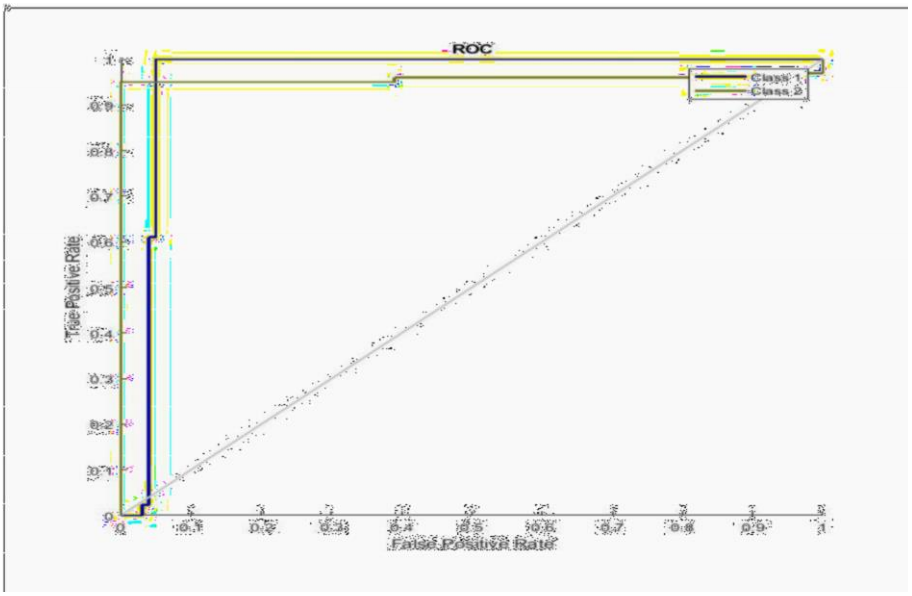

Figure 4.184: ROC Plot for PSO Algorithm 5 Feature 35 Generation

4.9.4 Epilepsy classification using seven features and 5, 10, 15, 20, 25, 30, and 35 population

Table 4.66 presents the F1 score, recall, accuracy, precision, and AUC. It shows that the accuracy varies between 95.50% to 98.60% as the population increases from 5 to 35 for the PSO-ANN classifier. The best accuracy is achieved at 98.60%, with 97.94% precision, 98.80 sensitivity, and an F1 score of 98.22% and 98.50% AUC respectively. It is observed from Figure 4.185 that the ROC curve for the 5 population performs better than 5, 10, 15, 20, 25, 30, and 35 respectively, indicating a better performance of the PSO-ANN classifier. The ROCs of various cases obtained using the PSO-ANN classifier are plotted as shown in

Figures 4.185 to 4.191 for the representation of the obtained results for 5, 10, 15, 20, 25, 30, and 35 populations respectively.

Table 4.65: PSO Algorithm Metrics Using Seven Features and 5, 10, 15, 20, 25, 30, and 35 Generation

Number of Features	Number of Population	TP	FP	TN	FN	Time
[7 3 6 9 4 1 8]	5	400	7	93	0	111.608
[1 3 6 7 9 5 8]	10	400	9	91	0	209.056
[3 9 6 7 5 2 1]	15	400	17	83	0	331.970
[5 6 7 2 3 8 9]	20	400	16	84	0	332.491
[7 6 8 3 4 2 5]	25	400	16	84	0	283.221
[2 1 5 4 9 7 8]	30	400	22	78	0	340.361
[6 9 7 2 5 3 4]	35	400	13	87	0	609.431

Table 4.66 PSO Algorithm Performance Evaluation Using Seven Features and 5, 10, 15, 20, 25, 30, and 35 Generation

Number of	Number of	Accuracy	Precision	Sensitivity	F1-	AUC
	Features	Population	Score			
[7 3 6 9 4 1 8]	**5**	**98.6**	**0.9794**	**0.9880**	**0.9822**	**0.9850**
[1 3 6 7 9 5 8]	10	98.2	0.9801	0.9840	0.9821	0.9830
[3 9 6 7 5 2 1]	15	96.6	0.9531	0.9680	0.9605	0.9640
[5 6 7 2 3 8 9]	20	98.8	0.9343	0.9740	0.9538	0.9630
[7 6 8 3 4 2 5]	25	96.8	0.9362	0.9760	0.9557	0.9650
[2 1 5 4 9 7 8]	30	95.5	0.9347	0.9600	0.9472	0.9530
[6 9 7 2 5 3 4]	35	97.4	0.9556	0.9780	0.9667	0.9720

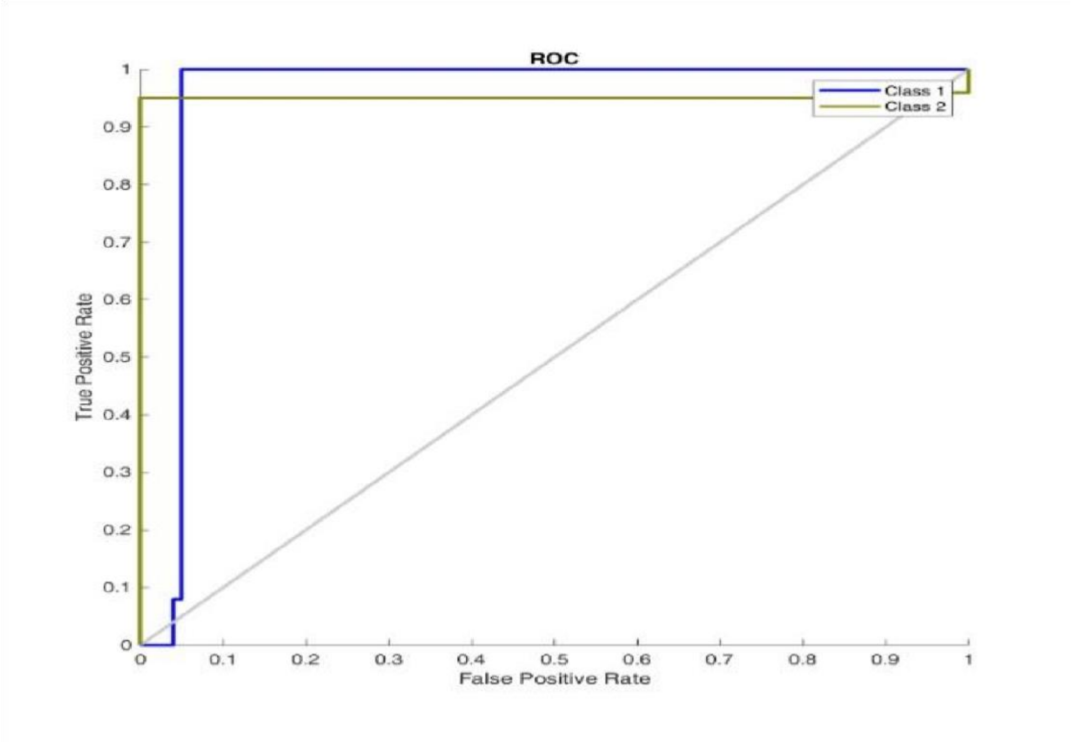

Figure 4.185: ROC Plot for PSO Algorithm7 Feature 5 Generation

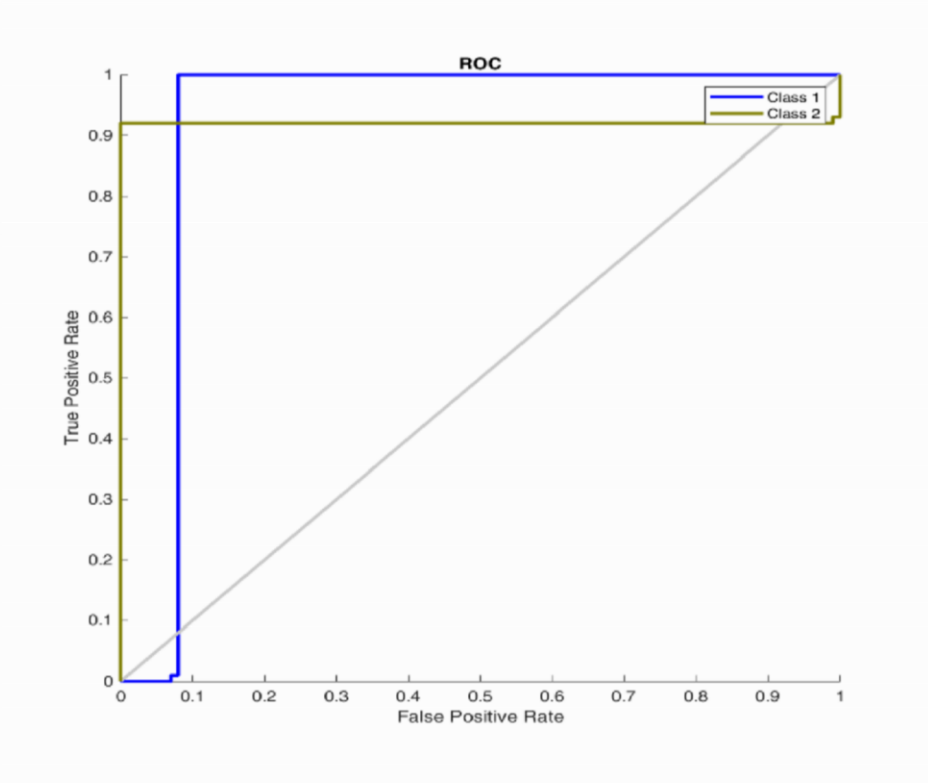

Figure 4.186: ROC Plot for PSO Algorithm 7 Feature 10 Generation

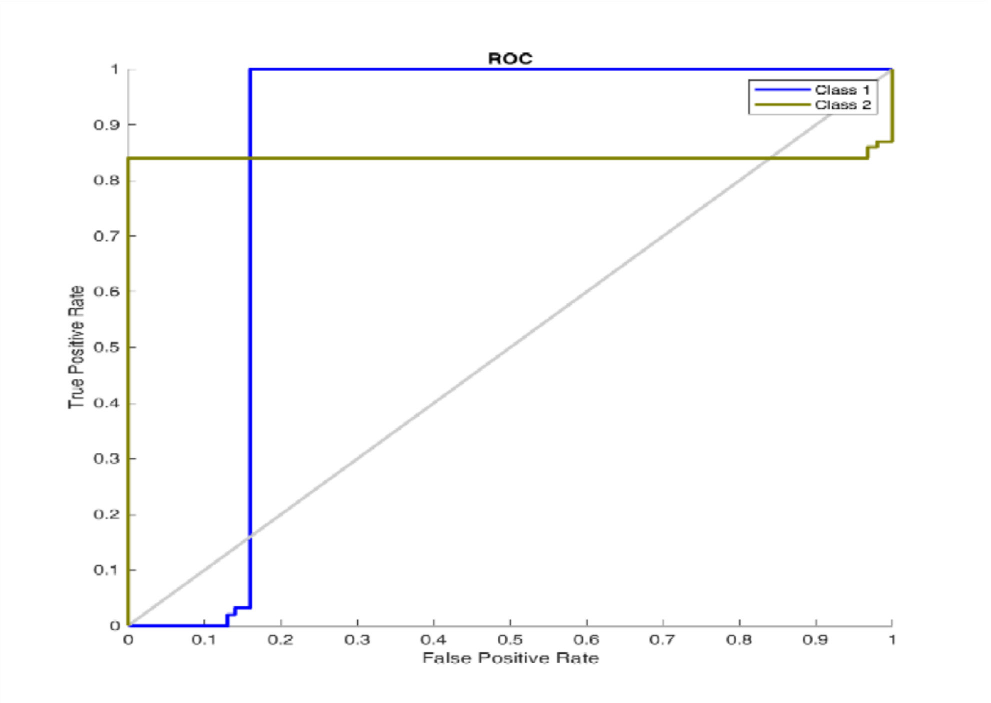

Figure 4.187: ROC Plot for PSO Algorithm 7 Feature 15 Generation

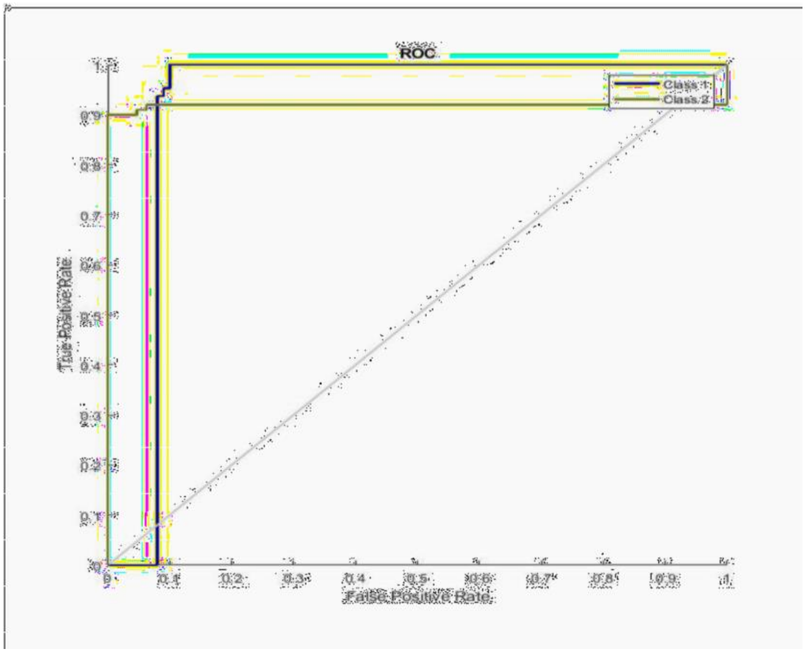

Figure 4.188: ROC Plot for PSO Algorithm 7 Feature 20 Generation

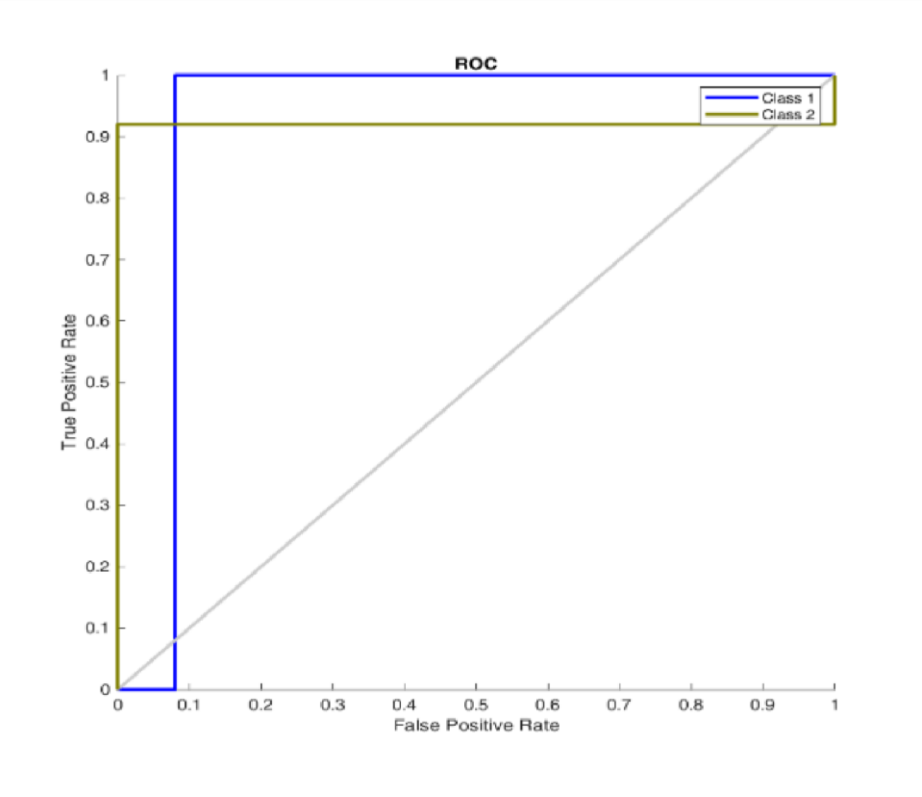

Figure 4.189: ROC Plot for PSO Algorithm 7 Feature 25 Generation

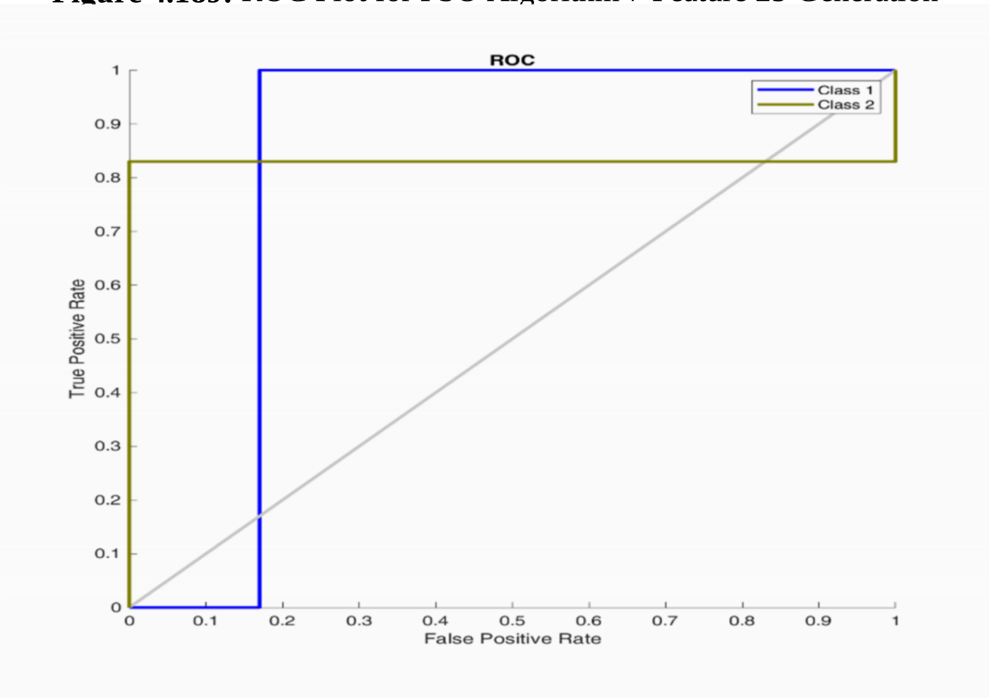

Figure 4.190: ROC Plot for PSO Algorithm 7 Feature 30 Generation

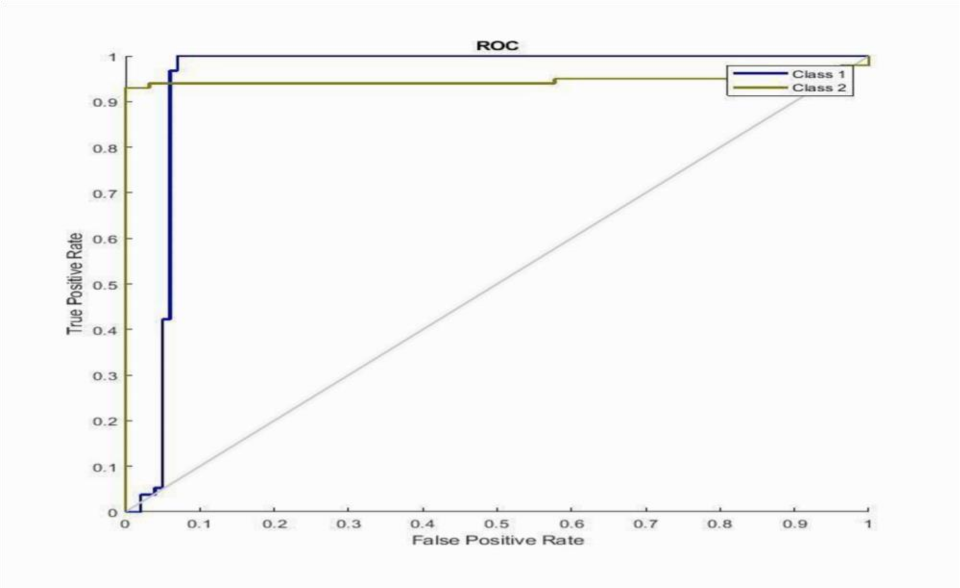

Figure 4.191: ROC Plot for PSO Algorithm 7 Feature 35 Generation

4.9.5 Epilepsy classification using nine features and 5, 10, 15, 20, 25, 30, and 35 population

Table 4.68 presents the F1 score, recall, accuracy, precision, and AUC. It shows that the accuracy varies between 96.60% to 98.60% as the population increases from 5 to 35 for the PSO-ANN classifier. The best accuracy is achieved at 98.60%, with 96.35% precision, 99.00 sensitivity, and an F1 score of 97.66% and 98.30% AUC respectively. It is observed from Figure 4.193 that the ROC curve for the 10 population performs better than 5, 10, 15, 20, 25, 30, and 35 respectively, indicating a better performance of the PSO-ANN classifier. The ROCs of various cases obtained using the PSO-ANN classifier are plotted as shown in Figures 4.192 to 4.198 for the representation of the obtained results for 5, 10, 15, 20, 25, 30, and 35 populations respectively.

Table 4.67: PSO Algorithm Metrics Using Nine Features and 5, 10, 15, 20, 25, 30, and 35 Generation

Number of Features	Number of Population	TP	FP	TN	FN	Time
[1 9 4 6 8 2 7 3 5]	5	400	17	83	0	52.250
[1 8 3 2 5 4 7 9 6]	10	400	7	93	0	107.911
[6 3 2 9 5 7 1 4 8]	15	400	11	89	0	198.124
[8 5 3 6 4 9 27 1]	20	400	15	85	0	331.961

[9 8 1 6 4 5 7 2 3]	25	400	14	86	0	273.683
[4 7 2 1 5 8 3 6 9]	30	400	13	87	0	397.021
[8 3 5 7 1 2 6 9 4]	35	400	10	90	0	550.932

Table 4.68: PSO Algorithm Performance Evaluation Using Nine Features and 5, 10, 15, 20, 25, 30, and 35 Generation

Number of	Number of	Accuracy	Precision	Sensitivity	F1-Score AUC	Features Population
[1 9 4 6 8 2 7 3 5]	5	96.6	0.9291	0.9720	0.9501	0.9600
[1 8 3 2 5 4 7 9 6]	**10**	**98.6**	**0.9635**	**0.9900**	**0.9766**	**0.9830**
[6 3 2 9 5 7 1 4 8]	15	97.8	0.9687	0.9840	0.9763	0.9800
[8 5 3 6 4 9 27 1]	20	97.0	0.9534	0.9720	0.9626	0.9670
[9 8 1 6 4 5 7 2 3]	25	97.2	0.9537	0.9760	0.9647	0.9700
[4 7 2 1 5 8 3 6 9]	30	97.4	0.9575	0.9800	0.9686	0.9740
[8 3 5 7 1 2 6 9 4]	35	98.0	0.9599	0.9900	0.9747	0.9820

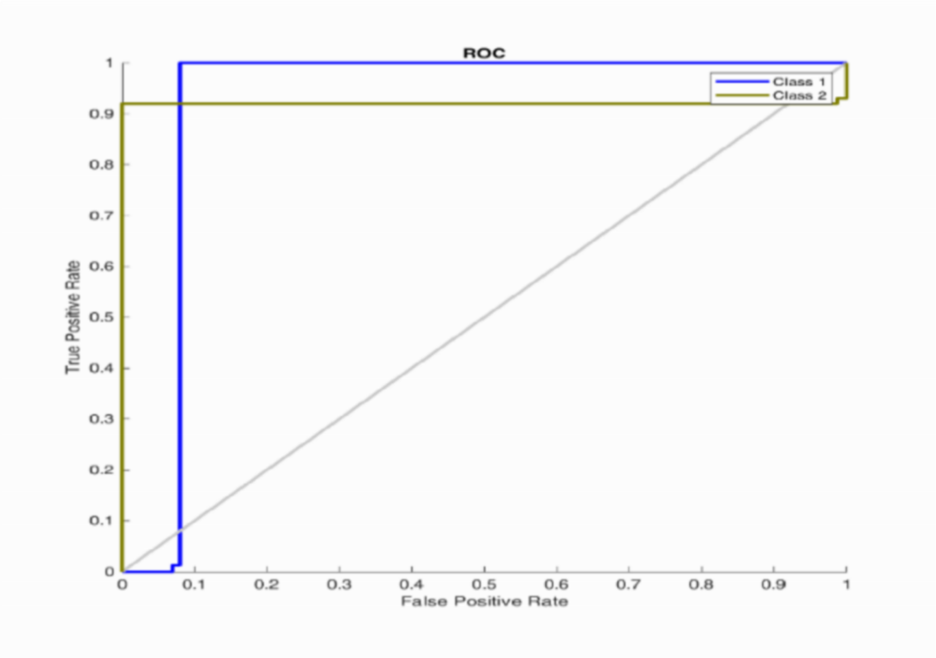

Figure 4.192: ROC Plot for PSO Algorithm 9 Feature 5 Generation

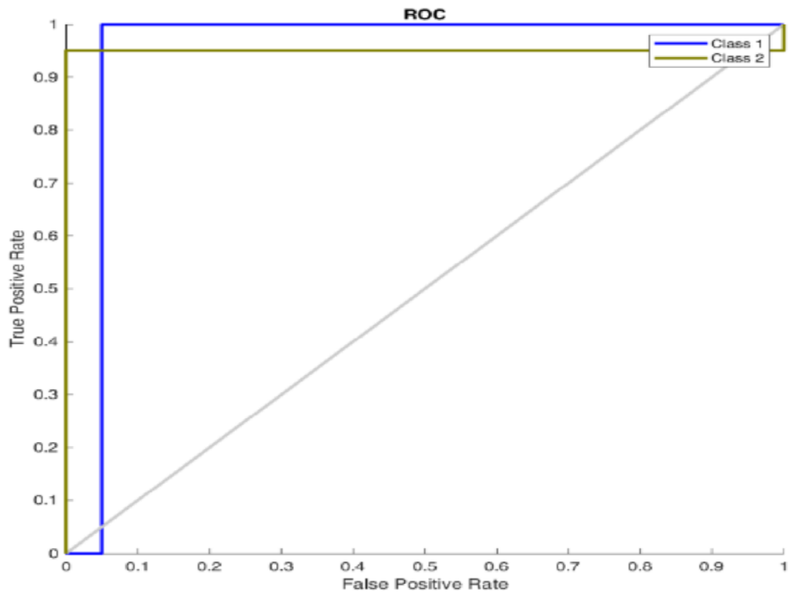

Figure 4.193: ROC Plot for PSO Algorithm 9 Feature 10 Generation

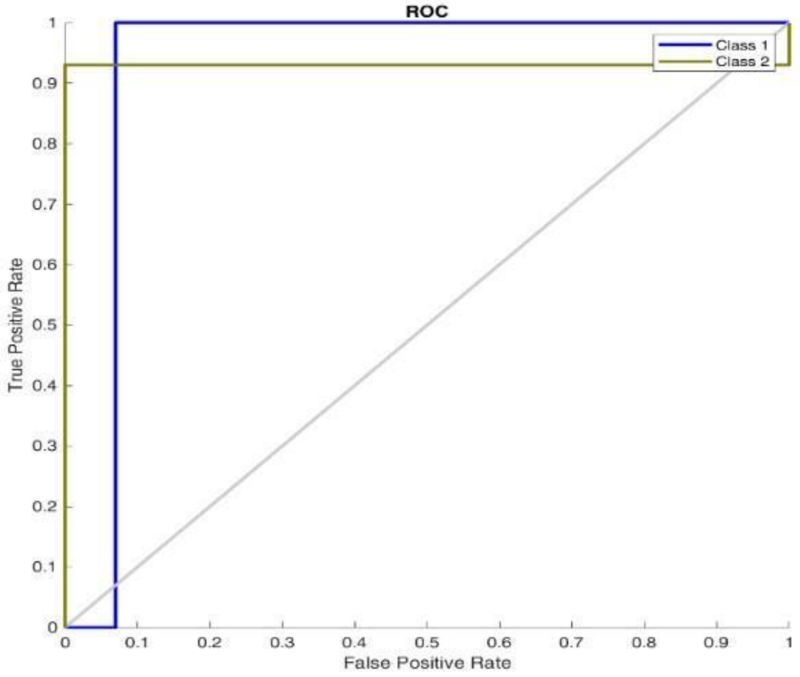

Figure 4.194: ROC Plot for PSO Algorithm 9 Feature 15 Generation

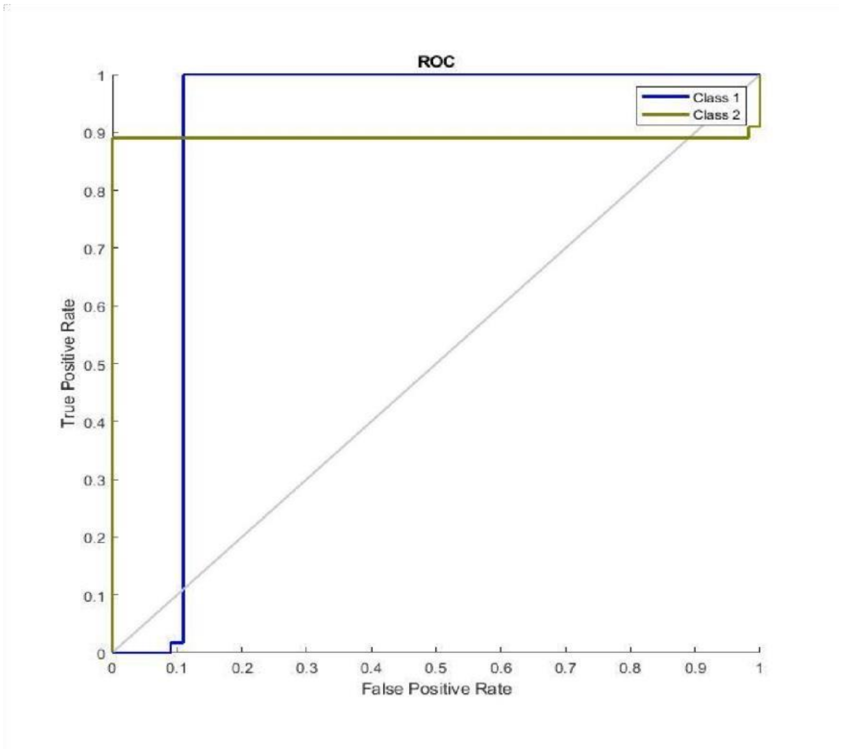

Figure 4.195: ROC Plot for PSO Algorithm 9 Feature 20 Generation

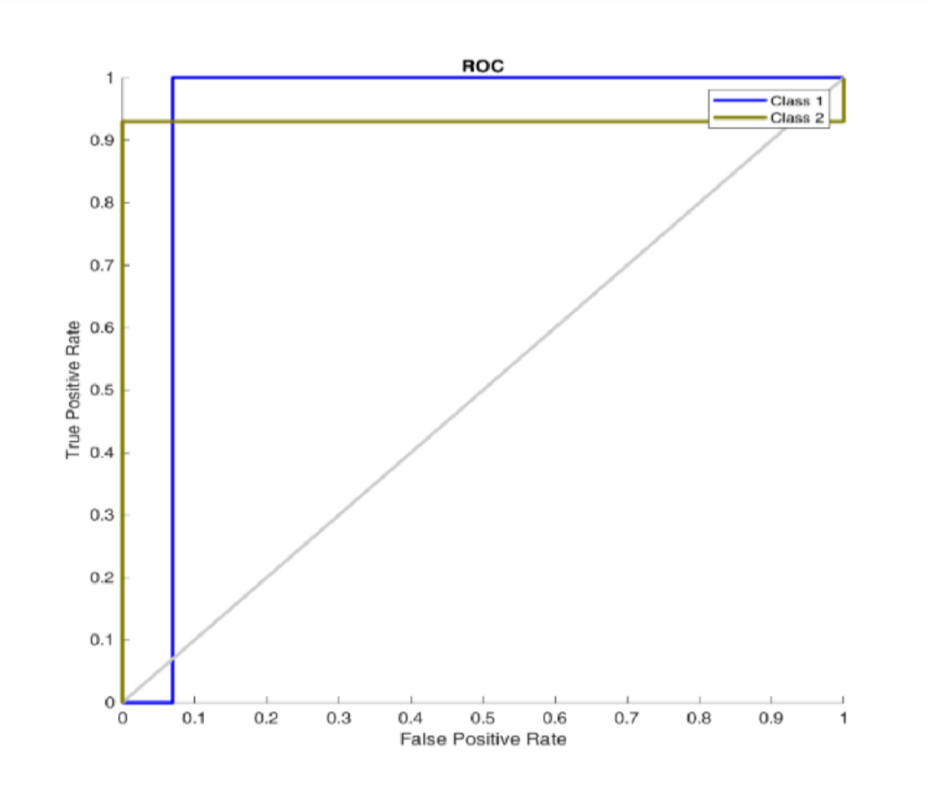

Figure 4.196: ROC Plot for PSO Algorithm 9 Feature 25 Generation

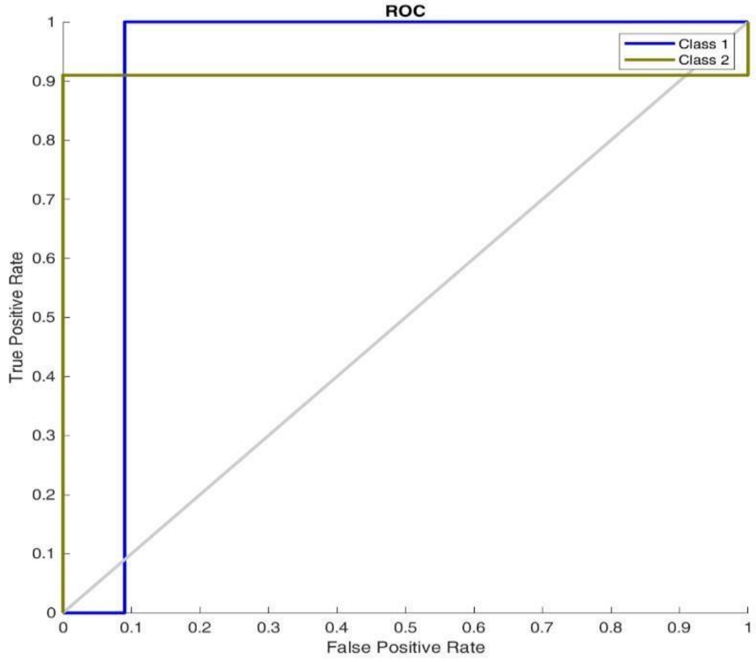

Figure 4.197: ROC Plot for PSO Algorithm 9 Feature 30 Generation

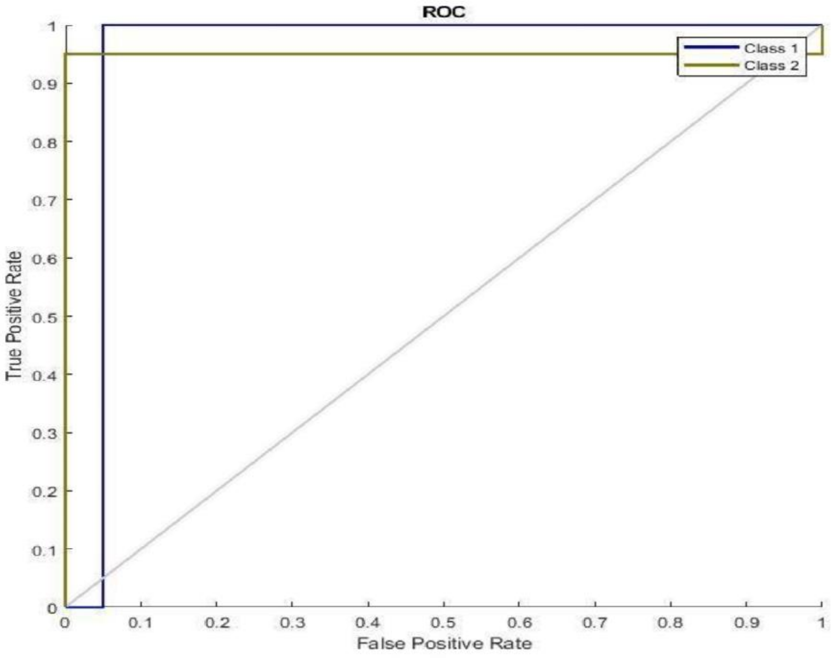

Figure 4.198: ROC Plot for PSO Algorithm 9 Feature 35 Generation Table 4.69 present the evaluation methods for PSO-ANN using one, three, five, seven, and nine features respectively for 5, 10. 15, 25, 30, and 35 population respectively. The AUC, recall,

accuracy, F1 score, and precision for various feature classes are shown as well. It shows that the best accuracy is achieved by using three features and five populations respectively. The best accuracy is achieved at 99.00%, with 91.70% precision, 99.40% sensitivity, and an F1 score of 95.39% and 97.20% AUC respectively for 3 features, and five populations for the PSO-ANN algorithm. It is observed from Figure 4.199 that the accuracy is near 100%, indicating a strong performance of the PSO-ANN classifier using five populations. In the real classes, the classifier performs well, and vice versa for three features when compared to other classes of the feature's extraction.

Table 4.69: Summary of Epilepsy Detection Using PSO with Various Search Agents

Evaluation Metrics	**1 Feature 5 Population**	**3 Features 5 Population**	**5 Features 30 Population**	**7 Features 5 Population**	**9 Features 10 Population**
Accuracy	0.964	**0.99**	0.976	0.986	0.986
Precision	0.8645	**0.917**	0.8463	0.9794	0.9635
Sensitivity	0.992	**0.994**	0.994	0.988	0.99
F1 Measure	0.9239	**0.9539**	0.9143	0.9822	0.9766
AUC	0.952	**0.972**	0.946	0.985	0.983

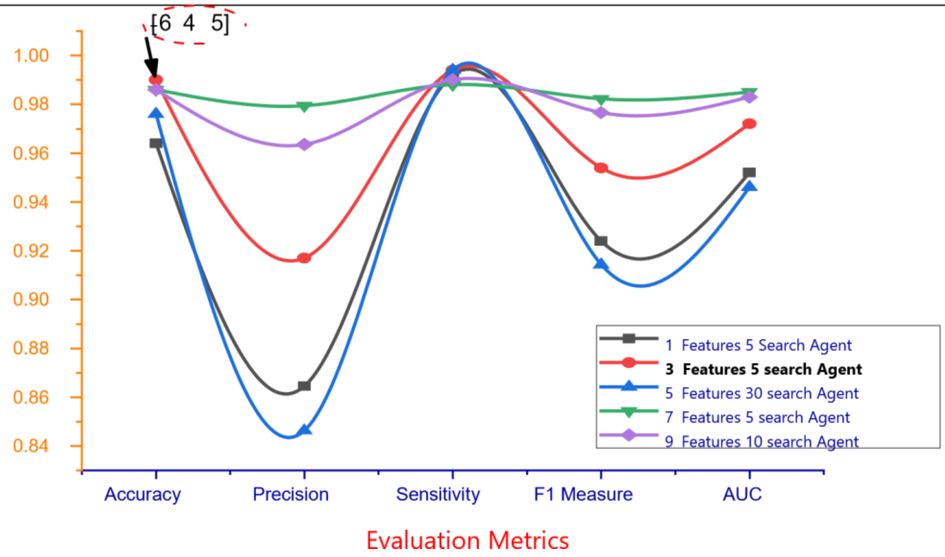

Figure 4.199: PSO Algorithm Performance Evaluation with Various Generation

4.10 Comparison of The Various Optimization Algorithm Results with IGOA

The findings and comparisons of the various optimization algorithms employed with the ANN classifier are presented in this section. Following the extraction of useful features from the EEG signals, five approaches were used to divide the data into "epileptic" and "non-epileptic" seizures, and performance parameters were developed to evaluate the model. The data were divided into two sets: training and testing, one for each classifier. The accuracy, sensitivity, specificity, F1 score, and AUC for each method were then calculated for the entire dataset. The findings of the performance measurement parameters for each method are shown in Table 4.70 and Figure 4.200 as the deliverable for objective three. It can be shown that the best result was obtained with GOA-ANN with 99.40% total accuracy, showing that the model accurately predicted all classes. GWO's overall accuracy of 98.40% was commendable. Similar results were obtained by PSO with 99.0% accuracy, SSOA with 98.49%, and BA with 98.80% accuracy. GWO and SSOA had the lowest overall accuracies, at 98.40% and 98.40%, respectively. The number of feature extraction and search agents in the GOA-ANN model that produced the best accuracy was seven and thirty, respectively.

Table 4.70: Summary of Epilepsy Detection Using the Various Optimization Algorithm

Evaluation	**IGOA**	**GOA -7**	**GWO- 9**	**SSOA -3**	**BA - 1**	**PSO -3**
Metrics	**5 Features**	**Features 30**	**Features**	**Features**	**Population**	**Features 5**
	30 Search	**Search 25 Search**	**30 Search 15**	**Population**	**Agent**	**Agent Agent**
	Agent	**Generation**				
Accuracy	**0.9960**	**0.994**	0.984	0.984	0.988	0.99
Precision	**0.9960**	**0.9881**	0.9689	0.984	0.761	0.917
Sensitivity	**0.9960**	**0.996**	0.988	0.984	0.992	0.994
F1 Measure	**0.9960**	**0.9921**	0.9784	0.984	0.8613	0.9539
AUC	**0.9960**	**0.994**	0.983	0.984	0.902	0.972

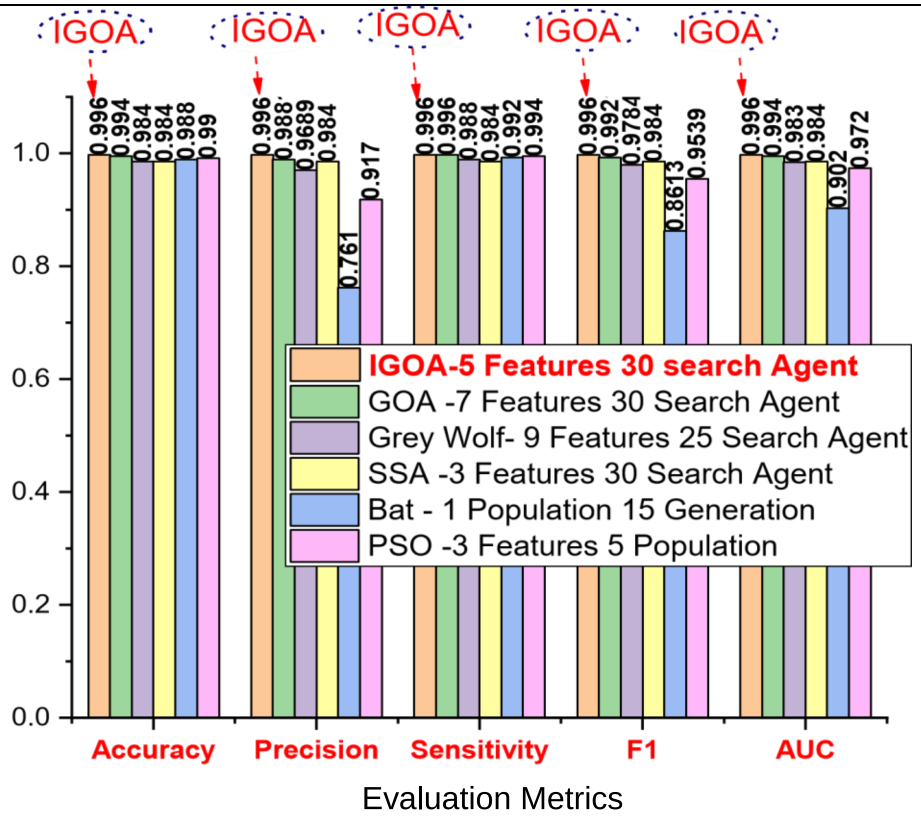

Figure 4.200: Performance Evaluation of the Various Algorithm Used

4.11 Comparison of the Research with Similar Studies

The accuracy levels of the five models were satisfactory, demonstrating that the proposed technique worked well with the EEG data. High overall accuracies of 99.60%, 99.40%, and 99.00% were recorded by the IGOA-ANN, GOA-ANN, and PSO-ANN models, respectively. The literature review's prior studies and the IGOA-ANN classifier's performance-based evaluation gave the classifier high results. The IGOA-ANN model, which obtained the best accuracy among the five models and the IGOA, surpassed the others in the experiment, achieving an overall accuracy of 99.60%. The GWO and SSOA models, on the other hand, had the algorithms' lowest overall accuracies, with 98.40% and 98.40% for the GWO and SSOA, respectively. The University of Bonn Dataset was used in other investigations, and when comparing the results, it was found that the IGOA-ANN approach gave results with higher sensitivity, specificity, and accuracy (99.60% each). The suggested model (IGOA-ANN), as demonstrated in Table 4.71, greatly outperformed more recent experiments with less complexity.

Figure 4.201 present the graphical user interface (GUI) for epilepsy classification. While Table 4.69 is the research objective five, Figure 4.198 is the deliverable for research objective four. Figure 4.201 make it easy for epilepsy classification for medical and nonmedical personnel with little or no knowledge of the algorithm which is the whole idea of the research.

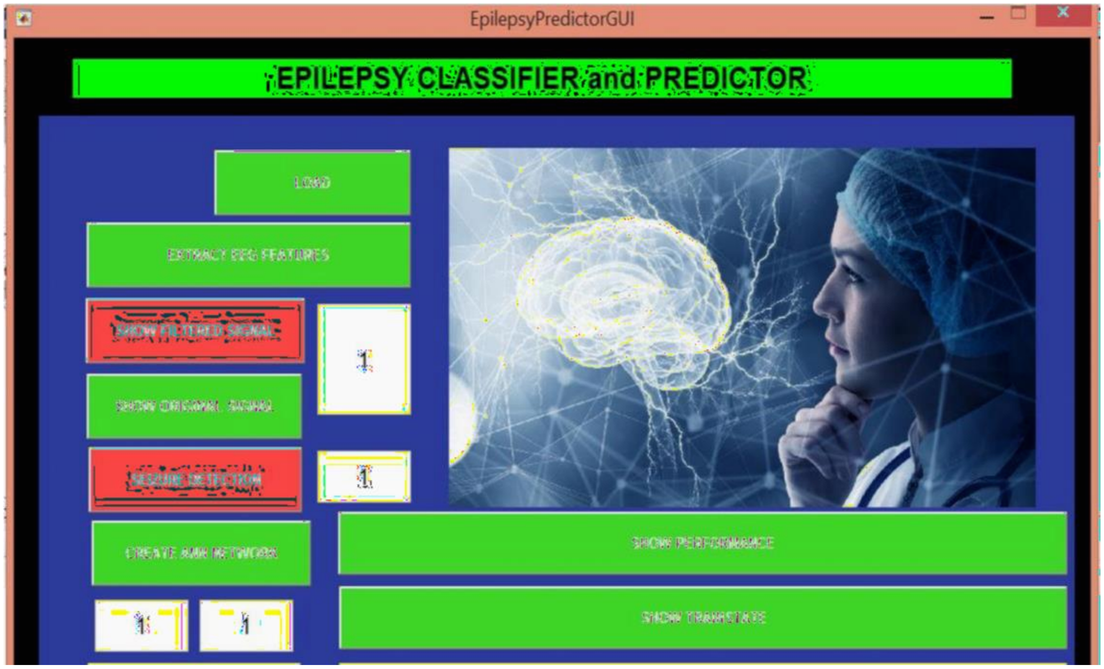

Figure 4.201: The System Graphical User Interface (GUI)

Table 4.71: Comparison of this Research with Similar Research in Literature

S/N	Author	Year	Classification Method	Features Used	Accuracy (%)
1.	(Zhao et al., 2020)	2020	1D Deep Neural Network	Automatic Feature Extraction	97.63
2.	(Glory et al., 2020)	2020	Deep Neural Network (DNN), SVM, KNN	Nonlinear and Entropy-based features	99.38
3.	(Ruchi Sharma, 2020)	2020	DNN	Entropy, linear and statistical features	98.12
4.	(Ruchi Sharma & Chopra)	2020	DNN, KNN, MSVM	Entropy, linear and statistical features	99.07
5.	(Mancha, Reddy, & Ch, 2021)	2021	Single Dimensional Pyramidal Ensemble Convolutional Neural Network (1D-PECNN)	Discriminative features	92.0
6.	(Shoeibi et al., 2021)	2021	SVM, KNN, and CNN-AE	Linear and Nonlinear features	73.3
7.	(Rashed AlMahfuz et al., 2021)	2021	CNN	Frequency features	99.21
8.	(Nkengfack, Tchiotsop, Atangana,	2021	LS-SVM and RBF Kernel	Beta and Gamma Rhythms	88.75

	Louis-Door, & Wolf, 2021)				
9.	(Chakraborty & Mitra, 2021)	2021	DT, KNN, SVM, and RF	Spectral Features	98.7
10.	(Khati & Ingle, 2020)	2021	Naïve Bayes, LR	Mean, standard deviation, and RMS features	99.0
11	**This Research**	**2021**	**IGOA-ANN**	**Maximum value, Mean value, RMS Variance, Standard Deviation, Entropy, Energy, Kurtosis, and Skewness**	**99.60**

CHAPTER FIVE

5.0 CONCLUSION AND RECOMMENDATIONS

5.1 Summary

It might be difficult to anticipate when someone may experience a seizure. Since most seizures occur unexpectedly, numerous researchers have looked for techniques to foresee seizures in advance. One can use classification algorithms to determine whether or not a person will experience a seizure. According to the research, epilepsy classification has insufficient power and is not appropriate for handling huge datasets. The number of characteristics utilized to generate the classifier, independent of the kind of classifier, has a significant impact on the classification performance as the quest for clinical epilepsy classification and prediction continues. Therefore, the quest for reliable biomarkers is essential, especially because EEG recordings are affected by a variety of physiological conditions, making it more challenging to distinguish between various brain states. As a result, utilizing the IGOA-ANN, this study proposes a hybrid model for epilepsy classification based on EEG signals that are both efficient and accurate. Thus, using the IGOA for feature selection and ANN to identify epilepsy is effective and efficient. This method makes use of the ANN data-driven methodology and the IGOA.

The influence of the search agent on the various attributes selected was investigated to select the search agent and feature extraction for the epilepsy classification approach. The search agent, as well as the best features, improve the accuracy. Other techniques such as standard GOA, PSO, SSOA, GWO, and BA were used to compare the results. The work was compared to other studies in the literature using the University of Bonn Epilepsy dataset. The performance of the developed system was assessed using its, F1 score measure, precision, accuracy, recall, and AUC.

5.2 Conclusion

The findings of the research study, experiments, findings, and pertinent debates can be summarized into the following conclusions:

When a set of ideal quicker features is chosen, epoch lengths are reduced, allowing for better epilepsy seizure classification. This aids in the identification of prominent traits and the elimination of those that may cause overfitting during the classification phase.

The investigation of the best features selection and search agent that was adopted for epilepsy classification adopted for IGOA-ANN indicates that the system was developed using five features and thirty search agents perform better when compared to using one, three, five, seven and nine features, and five, ten, fifteen, and twenty-five search agents respectively.

The results of the performance comparison with other optimization algorithms (GOA, SSOA, PSO, GWO, and BA) indicate that IGOA exceeds the other optimization algorithms for the majority of the performance metrics assessed in terms of accuracy, precision, AUC, F1 score measure, and recall. Furthermore, the accuracy test revealed that only IGOA obtained statistically significant findings with a 99.6 percent accuracy level. This means that the IGOA-ANN is the most effective in classifying epilepsy events into epilepsy and non-epilepsy events.

The experimental results of using IGOA-ANN to handle epilepsy classification problems show that IGOA-ANN is capable of solving all epileptic classification problems as the number of features, dimensions, and attributes grows. IGOA-ANN outperforms other studies with similar features and properties in the literature. As a result, IGOA-ANN is a good solution for a wide range of classification and optimization challenges, including traditional EEG and ECG detection and classification difficulties.

5.3 Recommendation

At the end of this research work, it is recommended that:

i. To increase the detection and classification of epilepsy seizures from EEG datasets, more feature extraction methods should be investigated. Additional univariate and bivariate features that incorporate spatial and temporal information from different brain regions could increase performance. Spectral entropy, mean phase coherence, and the Hjorth parameters, in particular, are potentially helpful properties that should be investigated.

ii. The IGOA-ANN classifier be tested on more epilepsy datasets to further determine its full classification performances.

iii. The IGOA-ANN classifier be tested on other engineering optimization problems. iv. The IGO-ANN classifier be tested on more real-world classification optimization problems such as Alzheimer's disease, migraine, stroke, and

cardiovascular disease classification.

5.4 Contribution to Knowledge

Using data from Bonn University in Germany, this research employs the IGOA-ANN approaches to classify seizure occurrences from non-seizure events. The ANN's vast collection of functions aids in learning the distinguishing aspects of seizure and non-seizure occurrences. The following are some of the contributions of this study:

i. An empirical comparative analysis of Optimization Algorithms to Feature Selection for Epilepsy Classification through a Systematic Literature Review (SLR) and performance comparison was carried out. ii. An improved Grasshopper Algorithm derived from elite opposition-based learning tested on unimodal and multimodal test functions was developed.

iii. Implementation of an improved Grasshopper algorithm with exponential switching parameters between local and random walks for updating the value of the Grasshopper Optimization Algorithm was achieved.

iv. The fourth contribution is the application of the Improved Grasshopper algorithm for the classification of epilepsy from Disruptive EEG Signals, with 99.60% accuracy, 99.60% sensitivity, 99.60%, precision, 99.60% AUC, and F1 measure of 99.60% achieved.

v. Development of a graphical user interface (GUI) for epilepsy classification.

REFERENCE

Aarabi, A., Wallois, F., & Grebe, R. (2006). Automated neonatal seizure detection: a multistage classification system through feature selection based on relevance and redundancy analysis. *Clinical Neurophysiology, 117*(2), 328-340.

Abbasi, M. U., Rashad, A., Basalamah, A., & Tariq, M. (2019). Detection of epilepsy seizures in neonatal EEG using LSTM architecture. *IEEE Access*, *7,* 179074179085.

Abderazek, H., Hamza, F., Yildiz, A. R., & Sait, S. M. (2021). Comparative investigation of the moth-flame algorithm and whale optimization algorithm for optimal spur gear design. *Materials Testing, 63*(3), 266-271.

Abedinpourshotorban, H., Shamsuddin, S. M., Beheshti, Z., & Jawawi, D. N. (2016). Electromagnetic field optimization: a physics-inspired metaheuristic optimization algorithm. *Swarm and Evolutionary Computation, 26*, 8-22.

Abualigah, L., Shehab, M., Alshinwan, M., Mirjalili, S., & Abd Elaziz, M. (2021). Ant lion optimizer: a comprehensive survey of its variants and applications. *Archives of Computational Methods in Engineering, 28*(3), 1397-1416.

Abusnaina, A. A., Ahmad, S., Jarrar, R., & Mafarja, M. (2018). *Training neural networks using salp swarm algorithm for pattern classification.* Paper presented at the Proceedings of the 2nd International Conference on Future Networks and Distributed Systems (ICFNDS 2018), ACM, New York, USA, 1-6.

Acharya, U. R., Chua, C. K., Lim, T.-C., Dorithy, & Suri, J. S. (2009). Automatic identification of epileptic EEG signals using nonlinear parameters. *Journal of Mechanics in Medicine and Biology, 9*(04), 539-553.

Acharya, U. R., Sree, S. V., Swapna, G., Martis, R. J., & Suri, J. S. (2013). Automated EEG analysis of epilepsy: a review. *Knowledge-Based Systems, 45*, 147-165.

Adam, V., Soldado-Magraner, J., Jitkritum, W., Strathmann, H., Lakshminarayanan, B., Ialongo, A. D., Dowling, S. (2012). Seizure Detection Challenge The Fitzgerald team solution, *Mechanics in Medicine and Biology,* 10(3) 9-15.

Aghdam, M. H., Ghasem-Aghaee, N., & Basiri, M. E. (2009). Text feature selection using ant colony optimization. *Expert Systems with Applications, 36*(3), 6843-6853.

Agrawal, P., Abutarboush, H. F., Ganesh, T., & Mohamed, A. W. (2021). Metaheuristic Algorithms on Feature Selection: A Survey of One Decade of Research (20092019). *IEEE Access, 9*, 26766-26791.

Ahammad, N., Fathima, T., & Joseph, P. (2014). Detection of epileptic seizure event and onset using EEG. *BioMed Research International*, 2014, 1-7.

Ahmad, M. A., Khan, N. A., & Majeed, W. (2014). *Computer-assisted analysis system of electroencephalogram for diagnosing epilepsy*. Paper presented at the 22nd IEEE International Conference on Pattern Recognition, Massachusetts Avenue, NW Washington, DC, United States, 3386-339.

Ahmed, S., Zhang, M., & Peng, L. (2013). *Enhanced feature selection for biomarker discovery in LC-MS data using GP*. Paper presented at the 2013 IEEE Congress on Evolutionary Computation, 584-591.

Akut, R. (2019). Wavelet-based deep learning approach for epilepsy detection. *Health Information Science and Systems, 7*(1), 1-9.

Alam, S. S., & Bhuiyan, M. I. H. (2013). Detection of seizure and epilepsy using higherorder statistics in the EMD domain. *IEEE Journal of Biomedical and Health Informatics, 17*(2), 312-318.

Ali, M. M., Khompatraporn, C., & Zabinsky, Z. B. (2005). A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems. *Journal of Global Optimization, 31*(4), 635-672.

Aliyu, I., Lim, Y. B., & Lim, C. G. (2019). *Epilepsy detection in EEG signal using recurrent neural network.* Paper presented at the Proceedings of the 2019 3rd International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence, Male, Maldives, 50-53.

Almustafa, K. M. (2020). Classification of epileptic seizure dataset using different machine learning algorithms. *Informatics in Medicine Unlocked, 21*, 100444.

Alotaiby, T., Abd El-Samie, F. E., Alshebeili, S. A., & Ahmad, I. (2015). A Review of channel selection algorithms for EEG signal processing. *EURASIP Journal on Advances in Signal Processing, 2015*(1), 1-21.

Amin, H. U., Malik, A. S., Ahmad, R. F., Badruddin, N., Kamel, N., Hussain, M., & Chooi, W.-T. (2015). Feature extraction and classification for EEG signals using wavelet transform and machine learning techniques. *Australasian Physical & Engineering Sciences in Medicine, 38*(1), 139-149.

Ansari, A. H., Cherian, P. J., Caicedo, A., Naulaers, G., De Vos, M., & Van Huffel, S. (2019). Neonatal seizure detection using deep convolutional neural networks. *International Journal of Neural Systems, 29*(04), 1850011.

Antoniou, A., & Lu, W.-S. (2007). General nonlinear optimization problems. *Practical optimization: algorithms and engineering applications*, 19, 501-531.

Assi, E. B., Gagliano, L., Rihana, S., Nguyen, D. K., & Sawan, M. (2018). Bispectrum features and multilayer perceptron classifier to enhance seizure prediction. *Scientific Reports, 8*(1), 1-8.

Assi, E. B., Nguyen, D. K., Rihana, S., & Sawan, M. (2017). Towards accurate prediction of epileptic seizures: A Review. *Biomedical Signal Processing and Control, 34*, 144-157.

Azlan, W. A. W., & Low, Y. F. (2014). *Feature extraction of electroencephalogram (EEG) signal-A review*. Paper presented at the 2014 IEEE Conference on Biomedical Engineering and Sciences (IECBES), Miri, Sarawak, Malaysia, 801-806.

Bandarabadi, M., Teixeira, C. A., Rasekhi, J., & Dourado, A. (2015). Epileptic seizure prediction using relative spectral power features. *Clinical Neurophysiology, 126*(2), 237-248.

Bangyal, W. H., Ahmad, J., Rauf, H. T., & Pervaiz, S. (2018). An Overview of Mutation Strategies in Bat Algorithm. *International Journal of Advanced Computer Science and Applications, 9*(8), 523-534.

Bertsekas, D. (1998). *Network optimization: continuous and discrete models*: Athena Scientific, 8.

Birjandtalab, J., Pouyan, M. B., Cogan, D., Nourani, M., & Harvey, J. (2017). Automated seizure detection using limited-channel EEG and non-linear dimension reduction. *Computers in Biology and Medicine, 82*, 49-58.

Bolón-Canedo,V., Rego-Fernández, D., Peteiro-Barral, D., Alonso-Betanzos, A., GuijarroBerdiñas, B., & Sánchez-Maroño, N. (2018). On the scalability of feature selection methods on high-dimensional data. *Knowledge and Information Systems, 56*(2), 395-442.

Boonyakitanont, P., Lek-Uthai, A., Chomtho, K., & Songsiri, J. (2020). A review of feature extraction and performance evaluation in epileptic seizure detection using EEG. *Biomedical Signal Processing and Control*, *57*, 101702.

Bosl, W. J., Tager-Flusberg, H., & Nelson, C. A. (2018). EEG analytics for early detection of autism spectrum disorder: a data-driven approach. *Scientific Reports, 8*(1), 1-20.

Boubchir, L., Daachi, B., & Pangracious, V. (2017). *A review of feature extraction for EEG epileptic seizure detection and classification.* Paper presented at the 2017 40th International Conference on Telecommunications and Signal Processing (TSP), Barcelona, Spain, 456-460.

Chakraborty, M., & Mitra, D. (2021). Epilepsy seizure detection using kurtosis based VMD's parameters selection and bandwidth features. *Biomedical Signal Processing and Control*, *64*, 102255.

Chakri, A., Khelif, R., Benouaret, M., & Yang, X.-S. (2017). New directional bat algorithm for continuous optimization problems. *Expert Systems with Applications, 69*, 159175.

Chang, R. S. K., Leung, C. Y. W., Ho, C. C. A., & Yung, A. (2017). Classifications of seizures and epilepsies, where are we?–A brief historical review and update. *Journal of the Formosan Medical Association*, *116*(10), 736-741.

Chen, D., Wan, S., Xiang, J., & Bao, F. S. (2017). A high-performance seizure detection algorithm based on Discrete Wavelet Transform (DWT) and EEG. *PloS one, 12*(3), e0173138.

Chickermane, H., & Gea, H. C. (1996). Structural optimization using a new local approximation method. *International Journal for Numerical Methods in Engineering, 39*(5), 829-846.

Covert, I. C., Krishnan, B., Najm, I., Zhan, J., Shore, M., Hixson, J., & Po, M. J. (2019). *Temporal graph convolutional networks for automatic seizure detection.* Paper presented at the Machine Learning for Healthcare Conference, University of Mitchigan, USA, 160-180.

Currey, D., Hsu, D., Ahmed, R., Venkataraman, A., & Craley, J. (2021). Cross-site Epileptic Seizure Detection Using Convolutional Neural Networks, In 2021 IEEE 55th Annual Conference on Information Sciences and Systems (CISS), Baltimore, MD, USA, 1-6 .

Dadgar-Kiani, E., Alkan, C., & Shameli, A. (2016). Applying machine learning for human seizure prediction. *Semanticscholar, 1-6*.

Daoud, H., & Bayoumi, M. A. (2019). Efficient epileptic seizure prediction based on deep learning. *IEEE Transactions on Biomedical Circuits and Systems, 13*(5), 804-813.

Direito, B., Teixeira, C. A., Sales, F., Castelo-Branco, M., & Dourado, A. (2017). A realistic seizure prediction study based on multiclass SVM. *International Journal of Neural Systems, 27*(03), 1750006.

Donos, C., Dümpelmann, M., & Schulze-Bonhage, A. (2015). Early seizure detection algorithm based on intracranial EEG and random forest classification. *International Journal of Neural Systems, 25*(05), 1550023.

Duda, R. O., & Hart, P. E. (1973). *Pattern classification and scene analysis*, Wiley, New York, 3.

El-Shorbagy, M., & Hassanien, A. E. (2018). Particle swarm optimization from theory to applications. *International Journal of Rough Sets and Data Analysis (IJRSDA), 5*(2), 1-24.

Emami, A., Kunii, N., Matsuo, T., Shinozaki, T., Kawai, K., & Takahashi, H. (2019). Seizure detection by convolutional neural network-based analysis of scalp electroencephalography plot images. *NeuroImage: Clinical, 22*, 101684.

Emary, E., Zawbaa, H. M., & Hassanien, A. E. (2016). Binary grey wolf optimization approaches for feature selection. *Neurocomputing, 172,* 371-381.

Esteller, R., Echauz, J., Tcheng, T., Litt, B., & Pless, B. (2001). *Line length: an efficient feature for seizure onset detection.* Paper presented at the 2001 Conference Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Sydney, Australia, 2, 1707-1710.

Faris, H., Aljarah, I., Al-Betar, M. A., & Mirjalili, S. (2018). Grey wolf optimizer: a review of recent variants and applications. *Neural Computing and Applications, 30*(2), 413435.

Faris, H., Mirjalili, S., Aljarah, I., Mafarja, M., & Heidari, A. A. (2020). Salp swarm algorithm: theory, literature review, and application in extreme learning machines. *Nature-inspired optimizers*, 185-199.

Fasil, O., & Rajesh, R. (2019). Time-domain exponential energy for epileptic EEG signal classification. *Neuroscience letters, 694*, 1-8.

Fergus, P., Hignett, D., Hussain, A., Al-Jumeily, D., & Abdel-Aziz, K. (2015). Automatic epileptic seizure detection using scalp EEG and advanced artificial intelligence techniques. *BioMed Research International,* 1-17.

Fernández-Delgado, M., Cernadas, E., Barro, S., & Amorim, D. (2014). Do we need hundreds of classifiers to solve real-world classification problems? *The Journal of Machine Learning Research, 15*(1), 3133-3181.

Fisher, R. S. (2017). The new classification of seizures by the International League Against Epilepsy 2017. *Current Neurology and Neuroscience Reports, 17*(6), 48.

Fister, I., Yang, X.-S., Fong, S., & Zhuang, Y. (2014). *Bat algorithm: Recent advances.* Paper presented at the IEEE 15th International symposium on computational intelligence and informatics (CINTI), Budapest, 163-167.

Gadhoumi, K., Gotman, J., & Lina, J. M. (2015). Scale invariance properties of intracerebral EEG improve seizure prediction in mesial temporal lobe epilepsy. *PloS one, 10*(4), e0121182.

Gadhoumi, K., Lina, J.-M., Mormann, F., & Gotman, J. (2016). Seizure prediction for therapeutic devices: A review. *Journal of neuroscience methods, 260*, 270-282.

Ghosh, A., Datta, A., & Ghosh, S. (2013). Self-adaptive differential evolution for feature selection in hyperspectral image data. *Applied Soft Computing, 13*(4), 1969-1977.

Giannakakis, G., Sakkalis, V., Pediaditis, M., & Tsiknakis, M. (2014). Methods for seizure detection and prediction: an overview. *Modern Electroencephalographic Assessment Techniques*, 131-157.

Gill, A. F., Fatima, S. A., Akram, M. U., Khawaja, S. G., & Awan, S. E. (2015). Analysis of EEG signals for detection of epileptic seizure using hybrid feature set. In *Theory and Applications of Applied Electromagnetics*, 49-57.

Glory, H. A., Vigneswaran, C., Jagtap, S. S., Shruthi, R., Hariharan, G., & Sriram, V. S. (2020). AHW-BGOA-DNN: a novel deep learning model for epileptic seizure detection. *Neural Computing and Applications*, 33, 1-29.

Glover, F. W., & Kochenberger, G. A. (2006). *Handbook of metaheuristics*: Springer Science & Business Media, 57.

Greene, B., Faul, S., Marnane, W., Lightbody, G., Korotchikova, I., & Boylan, G. (2008). A comparison of quantitative EEG features for neonatal seizure detection. *Clinical Neurophysiology, 119*(6), 1248-1261.

Guo, L., Rivero, D., Dorado, J., Rabunal, J. R., & Pazos, A. (2010). Automatic epileptic seizure detection in EEGs based on line length feature and artificial neural networks. *Journal of neuroscience methods, 191*(1), 101-109.

Hadoush, H., Alafeef, M., & Abdulhay, E. (2019). Brain complexity in children with mild and severe autism spectrum disorders: Analysis of multiscale entropy in EEG. *Brain Topography, 32*(5), 914-921.

Hamad, A., Houssein, E. H., Hassanien, A. E., & Fahmy, A. A. (2018). *Hybrid grasshopper optimization algorithm and support vector machines for automatic seizure detection in EEG signals.* Paper presented at the International conference on advanced machine learning technologies and applications, Cairo, Egypt, 81-91.

Hassanien, A. E., & Emary, E. (2018). *Swarm intelligence: principles, advances, and applications*: CRC Press.

Hatta, N., Zain, A. M., Sallehuddin, R., Shayfull, Z., & Yusoff, Y. (2019). Recent studies on optimisation method of Grey Wolf Optimiser (GWO): a review (2014–2017). *Artificial Intelligence Review, 52*(4), 2651-2683.

Hernández, D., Trujillo, L., Z-Flores, E., Villanueva, O., & Romo-Fewell, O. (2018). Detecting epilepsy in EEG signals using time, frequency and time-frequency domain features. In *Computer science and engineering—theory and applications,* 167-182.

Hively, L. M., & Protopopescu, V. A. (2003). Channel-consistent forewarning of epileptic events from scalp EEG. *IEEE Transactions on Biomedical Engineering, 50*(5), 584593.

Hoque, N., Bhattacharyya, D. K., & Kalita, J. K. (2014). MIFS-ND: A mutual informationbased feature selection method. *Expert Systems with Applications, 41*(14), 63716385.

Hossain, M. S., Amin, S. U., Alsulaiman, M., & Muhammad, G. (2019). Applying deep learning for epilepsy seizure detection and brain mapping visualization. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 15*(1s), 1-17.

Hussain, K., Salleh, M. N. M., Cheng, S., & Naseem, R. (2017). Common benchmark functions for metaheuristic evaluation: A review. *JOIV: International Journal on Informatics Visualization, 1*(4-2), 218-223.

Hussein, R., Ahmed, M. O., Ward, R., Wang, Z. J., Kuhlmann, L., & Guo, Y. (2019). Human intracranial EEG quantitative analysis and automatic feature learning for epileptic seizure prediction, *1904-03603*.

Hussein, R., Palangi, H., Ward, R., & Wang, Z. J. (2018). Epileptic seizure detection: A deep learning approach, *1803-09848*.

Ibrahim, H. T., Mazher, W. J., Ucan, O. N., & Bayat, O. (2017). Feature selection using salp swarm algorithm for real biomedical datasets, *International Journal of Computer Science and Network Security, 17*(12), 13-20.

Ibrahim, H. T., Mazher, W. J., Ucan, O. N., & Bayat, O. (2019). A grasshopper optimizer approach for feature selection and optimizing SVM parameters utilizing real biomedical data sets. *Neural Computing and Applications, 31*(10), 5965-5974.

Ihle, M., Feldwisch-Drentrup, H., Teixeira, C. A., Witon, A., Schelter, B., Timmer, J., & Schulze-Bonhage, A. (2012). EPILEPSIAE–A European epilepsy database. *Computer Methods and Programs in Biomedicine, 106*(3), 127-138.

Jamil, M., & Yang, X.-S. (2013). A literature survey of benchmark functions for global optimisation problems. *International Journal of Mathematical Modelling and Numerical Optimisation, 4*(2), 150-194.

Jović, A., Brkić, K., & Bogunović, N. (2015). *A review of feature selection methods with applications.* Paper presented at the 38th international convention on information and communication technology, electronics and microelectronics (MIPRO), Opatija, Croatia, 1200-1205.

Kashan, A. H. (2015). A new metaheuristic for optimization: optics inspired optimization (OIO). *Computers & Operations Research, 55*, 99-125.

Kaur, M., & Singh, G. (2017). Classification of seizure prone EEG signal using amplitude and frequency based parameters of intrinsic mode functions. *Journal of Medical and Biological Engineering, 37*(4), 540-553.

Kaveh, A., & Dadras, A. (2017). A novel meta-heuristic optimization algorithm: thermal exchange optimization. *Advances in Engineering Software, 110*, 69-84.

Kaveh, A., & Ghazaan, M. I. (2017). A new meta-heuristic algorithm: vibrating particles system. *Scientia Iranica. Transaction A, Civil Engineering, 24*(2), 551.

Khaire, U. M., & Dhanalakshmi, R. (2022). Stability of feature selection algorithm: A review. *Journal of King Saud University-Computer and Information Sciences,* 34,4, 1060-1073.

Khan, H., Marcuse, L., Fields, M., Swann, K., & Yener, B. (2017). Focal onset seizure prediction using convolutional networks. *IEEE Transactions on Biomedical Engineering, 65*(9), 2109-2118.

Khati, R. M., & Ingle, R. (2020). Feature extraction for epileptic seizure detection using machine learning. *Current Medicine Research and Practice, 10*(6), 266.

Khosla, A., Khandnor, P., & Chand, T. (2020). A comparative analysis of signal processing and classification methods for different applications based on EEG signals. *Biocybernetics and Biomedical Engineering, 40*(2), 649-690.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. S*cience, 220*(4598), 671-680.

Kitano, L. A. S., Sousa, M. A. A., Santos, S. D., Pires, R., Thome-Souza, S., & Campo, A. B. (2018). *Epileptic seizure prediction from EEG signals using unsupervised learning and a polling-based decision process.* Paper presented at the International Conference on Artificial Neural Networks, Rio de Janeiro,Brazil, 117-126.

Kochenderfer, M. J., & Wheeler, T. A. (2019). *Algorithms for optimization*: MIT Press.

Koolen, N., Jansen, K., Vervisch, J., Matic, V., De Vos, M., Naulaers, G., & Van Huffel, S. (2014). Line length as a robust method to detect high-activity events: automated burst detection in premature EEG recordings. *Clinical Neurophysiology, 125*(10), 1985-1994.

Kumar, J. S., & Bhuvaneswari, P. (2012). Analysis of Electroencephalography (EEG) signals and its categorization–a study. *Procedia Engineering, 38,* 2525-2536.

Kumar, N., Alam, K., & Siddiqi, A. H. (2017). Wavelet Transform for Classification of EEG Signal using SVM and ANN. *Biomedical and Pharmacology Journal, 10*(4), 2061-2069.

Kumar, Y., Dewal, M., & Anand, R. (2014). Epileptic seizures detection in EEG using DWT-based ApEn and artificial neural network. *Signal, Image and Video Processing, 8*(7), 1323-1334.

Lai, K.-T. (2021). Machine Learning Basics.

Larmuseau, M. (2016). *Epileptic seizure prediction using deep learning.* Master's thesis, Universiteit Gent, Belgique (2015–2016).

Lasefr, Z., Ayyalasomayajula, S. S. V., & Elleithy, K. (2017). *Epilepsy seizure detection using EEG signals.* Paper presented at the IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON), New York, USA, 385-400 .

Lawler, E. L. (2001). *Combinatorial optimization: networks and matroids*: Courier Corporation.

Lee, H., & Kim, S. (2016). Black-box classifier interpretation using decision tree and fuzzy logic-based classifier implementation. *International Journal of Fuzzy Logic and Intelligent Systems, 16*(1), 27-35.

Leszczyński, K. (2018). Seizure prediction using machine learning models.

Li, X., Song, D., Zhang, P., Zhang, Y., Hou, Y., & Hu, B. (2018). Exploring EEG features in cross-subject emotion recognition. *Frontiers in Neuroscience, 12*, 162.

Liu, H., & Yu, L. (2005). Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering, 17*(4), 491-502.

Liu, J., & Woodson, B. (2019). *Deep learning classification for epilepsy detection using single-channel electroencephalography (EEG).* Paper presented at the Proceedings of the 3rd International Conference on Deep Learning Technologies, Xiamen, China, 23-26.

Liu, Z., Wang, Y., Liu, X., Du, Y., Tang, Z., Wang, K., Dai, J. (2018). Radiomics analysis allows for precise prediction of epilepsy in patients with low-grade gliomas. *NeuroImage: Clinical, 19*, 271-278.

Löfhede, J., Thordstein, M., Löfgren, N., Flisberg, A., Rosa-Zurera, M., Kjellmer, I., & Lindecrantz, K. (2010). Automatic classification of background EEG activity in healthy and sick neonates. *Journal of Neural Engineering, 7*(1), 016007.

Logesparan, L., Casson, A. J., & Rodriguez-Villegas, E. (2012). Optimal features for online seizure detection. *Medical & Biological Engineering & Computing, 50*(7), 659-669.

Logesparan, L., Rodriguez-Villegas, E., & Casson, A. J. (2015). The impact of signal normalization on seizure detection using line length features. *Medical & Biological Engineering & Computing, 53*(10), 929-942.

Lu, D., & Triesch, J. (2019). Residual deep convolutional neural network for EEG signal classification in epilepsy, *1903-08100*.

Ma, H., Simon, D., Siarry, P., Yang, Z., & Fei, M. (2017). Biogeography-based optimization: a 10-year review. *IEEE Transactions on Emerging Topics in Computational Intelligence, 1*(5), 391-407.

Mancha, V. R., Reddy, S., & Ch, S. (2021). *Advanced Convolutional Neural Network Classification for Automatic Seizure Epilepsy Detection in EEG Signal.* Paper presented at the IOP Conference Series: Materials Science and Engineering, Samawa, Iraq, 1074(1), 012005.

Manzouri, F., Heller, S., Dümpelmann, M., Woias, P., & Schulze-Bonhage, A. (2018). A comparison of machine learning classifiers for energy-efficient implementation of seizure detection. *Frontiers in Systems Neuroscience, 12*, 43.

Mareli, M. (2018). *Development of Effective Cuckoo Search Algorithms for Optimisation Purposes*: University of Johannesburg, South Africa.

Meraihi, Y., Gabis, A. B., Mirjalili, S., & Ramdane-Cherif, A. (2021). Grasshopper Optimization Algorithm: Theory, Variants, and Applications. *IEEE Access*, *9*, 50001-50024.

Meraihi, Y., Ramdane-Cherif, A., Acheli, D., & Mahseur, M. (2020). Dragonfly algorithm: a comprehensive review and applications. *Neural Computing and Applications*, 122.

Mirjalili, S. (2015). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-Based Systems, 89*, 228-249.

Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software, 95*, 51-67.

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software, 69*, 46-61.

Moghim, N., & Corne, D. W. (2014). Predicting epileptic seizures in advance. *PloS one, 9*(6), e99334.

Mursalin, M., Zhang, Y., Chen, Y., & Chawla, N. V. (2017). Automated epileptic seizure detection using improved correlation-based feature selection with a random forest classifier. *Neurocomputing, 241*, 204-214.

Nayak, P. K., & Cholayya, N. U. (2006). *Independent component analysis of Electroencephalogram.* Paper presented at the IEEE Japan Papers of Technical Meeting on Medical and Biological Engineering, Japan, 6(95), 25-28.

Niknazar, M., Mousavi, S., Vahdat, B. V., & Sayyah, M. (2013). A new framework based on recurrence quantification analysis for epileptic seizure detection. *IEEE Journal of Biomedical and Health Informatics, 17*(3), 572-578.

Nkengfack, L. C. D., Tchiotsop, D., Atangana, R., Louis-Door, V., & Wolf, D. (2021). Classification of EEG signals for epileptic seizures detection and eye states identification using Jacobi polynomial transforms-based measures of complexity and least-square support vector machine. *Informatics in Medicine Unlocked, 23*, 100536.

Noureddine, S. (2015). An optimization approach for the satisfiability problem. *Applied Computing and Informatics, 11*(1), 47-59.

Oliva, J. T., & Rosa, J. L. G. (2019). Classification for EEG report generation and epilepsy detection. *Neurocomputing, 335*, 81-95.

Omerhodzic, I., Avdakovic, S., Nuhanovic, A., & Dizdarevic, K. (2013). Energy distribution of EEG signals: EEG signal wavelet-neural network classifier, *13077897*.

Ong, P., Zainuddin, Z., & Lai, K. H. (2018). A novel selection of optimal statistical features in the DWPT domain for discrimination of ictal and seizure-free electroencephalography signals. *Pattern Analysis and Applications, 21*(2), 515527.

Panayiotopoulos, C. P. (2010). Epileptic seizures and their classification. In *A clinical guide to epileptic syndromes and their treatment*, 21-63.

Park, Y., Luo, L., Parhi, K. K., & Netoff, T. (2011). Seizure prediction with spectral power of EEG using cost-sensitive support vector machines. *Epilepsia*, *52*(10), 17611770.

Polat, H., & Ozerdem, M. S. (2016). *Epileptic seizure detection from EEG signals by using wavelet and Hilbert transform.* Paper presented at the XVII International Conference on Perspective Technologies and Methods in MEMS Design (MEMSTECH), Lviv-Polyana, Ukraine, 66-69.

Quintero-Rincón, A., D'Giano, C., & Batatia, H. (2019). Seizure onset detection in EEG signals based on entropy from generalized gaussian pdf modelling and ensemble bagging classifier. In *Digital health approach for predictive, preventive, personalised and participatory medicine,* 1-10.

Rahman, M., & Karim, M. (2015). *Predicting epileptic seizure from Electroencephalography (EEG) using Hilbert Huang transformation and neural network, Doctoral dissertation, BRAC University.*

Rashed-Al-Mahfuz, M., Moni, M. A., Uddin, S., Alyami, S. A., Summers, M. A., & Eapen, V. (2021). A deep convolutional neural network method to detect seizures and characteristic frequencies using epileptic electroencephalogram (EEG) data. *IEEE Journal of Translational Engineering in Health and Medicine*, *9*, 1-12.

Rasheed, K., Qayyum, A., Qadir, J., Sivathamboo, S., Kwan, P., Kuhlmann, L., Razi, A. (2020). Machine learning for predicting epileptic seizures using EEG signals: A review. *IEEE Reviews in Biomedical Engineering, 14, 139-155.*

Rini, D. P., Shamsuddin, S. M., & Yuhaniz, S. S. (2011). Particle swarm optimization: technique, system and challenges. *International Journal of Computer Applications, 14*(1), 19-26.

Rosas-Romero, R., & Guevara, E. (2020). *Classification of functional Near Infra-Red Signals with Machine Learning for Prediction of Epilepsy*. Paper presented at the Proceedings of the 12th International Conference, South Africa, 70, 41-48.

Rothlauf, F. (2011). Optimization method. Design of Modern Heuristics, Natural Computing Series, Berlin, German.

Ruchi Sharma, K. C. (2020). Enhanced firefly optimizer with deep neural network for the detection of epileptic seizures using EEG signals. *International Journal of Engineering and Advanced Technology (IJEAT), 9*(5), 137-148.

San-Segundo, R., Gil-Martín, M., D'Haro-Enriquez, L. F., & Pardo, J. M. (2019). Classification of epileptic EEG recordings using signal transforms and convolutional neural networks. *Computers in Biology and Medicine, 109*, 148-158.

Sanei, S., & Chambers, J. A. (2013). *EEG signal processing*: John Wiley & Sons.

Saremi, S., Mirjalili, S., & Lewis, A. (2017). Grasshopper optimisation algorithm: theory and application. *Advances in Engineering Software, 105*, 30-47.

Savadkoohi, M., Oladunni, T., & Thompson, L. (2020). A machine learning approach to epileptic seizure prediction using Electroencephalogram (EEG) Signal. *Biocybernetics and Biomedical Engineering, 40*(3), 1328-1341.

Seyedali, M. (2015). The ant lion optimizer. *Advances in Engineering Software, 83*, 80-98.

Sharif, B., & Jafari, A. H. (2017). Prediction of epileptic seizures from EEG using analysis of ictal rules on Poincaré plane. *Computer Methods and Programs in Biomedicine, 145*, 11-22.

Sharma, M., Shah, S., & Achuth, P. (2019). A novel approach for epilepsy detection using time-frequency localized bi-orthogonal wavelet filter. *Journal of Mechanics in Medicine and Biology, 19*(01), 1940007.

Sharma, R., & Chopra, K. (2020). Epileptic Detection from the Eeg Signal using the Anti colony Optimization Technique with Deep Neural Network, *International Journal of Recent Technology and Engineering*, 9(1), 2726-2733.

Sharma, R., & Pachori, R. B. (2015). Classification of epileptic seizures in EEG signals based on phase space representation of intrinsic mode functions. *Expert Systems with Applications, 42*(3), 1106-1117.

Sharmila, A. (2018). Epilepsy detection from EEG signals: a review. *Journal of Medical Engineering & Technology*, *42*(5), 368-380.

Shehab, M., Khader, A. T., & Al-Betar, M. A. (2017). A survey on applications and variants of the cuckoo search algorithm. *Applied Soft Computing, 61,* 1041-1059.

Shoaib, M., Lee, K. H., Jha, N. K., & Verma, N. (2014). A 0.6–107 µW energy-scalable processor for directly analyzing compressively-sensed EEG. *IEEE Transactions on Circuits and Systems I: Regular Papers, 61*(4), 1105-1118.

Shoeb, A. H., & Guttag, J. V. (2010). *Application of machine learning to epileptic seizure detection.* Paper presented at the Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, 975-982.

Shoeibi, A., Ghassemi, N., Alizadehsani, R., Rouhani, M., Hosseini-Nejad, H., Khosravi, A., Nahavandi, S. (2021). A comprehensive comparison of handcrafted features and convolutional autoencoders for epileptic seizures detection in EEG signals. *Expert Systems with Applications, 163,* 113788.

Shoeibi, A., Ghassemi, N., Khodatars, M., Jafari, M., Hussain, S., Alizadehsani, R., Rouhani, M. (2020). Epileptic seizure detection using deep learning techniques: A Review, *2007-01276.*

Siddique, N., & Adeli, H. (2015). Nature-inspired computing: an overview and some future directions. *Cognitive Computation, 7*(6), 706-714.

Siddiqui, M. K., Islam, M. Z., & Kabir, M. A. (2019). A novel quick seizure detection and localization through brain data mining on the ECoG dataset. *Neural Computing and Applications, 31*(9), 5595-5608.

Singh, G., Kaur, M., & Singh, B. (2021). Detection of epileptic seizure EEG signal using multiscale entropies and complete ensemble empirical mode decomposition. *Wireless Personal Communications, 116*(1), 845-864.

Singh, G., Kaur, M., & Singh, D. (2015). *Detection of epileptic seizures using wavelet transformation and spike-based features.* Paper presented at the 2nd International Conference on Recent Advances in Engineering & Computational Sciences (RAECS), Belgium, 1-4.

Singh, G., Singh, B., & Kaur, M. (2019). Grasshopper optimization algorithm–based approach for the optimization of ensemble classifier and feature selection to classify epileptic EEG signals. *Medical & Biological Engineering & Computing, 57*(6), 1323-1339.

Singh, N., Singh, S., & Houssein, E. H. (2020). Hybridizing salp swarm algorithm with particle swarm optimization algorithm for recent optimization functions. *Evolutionary Intelligence*, 1-34.

Siuly, S., Li, Y., & Zhang, Y. (2016). EEG signal analysis and classification. *IEEE Trans Neural System Rehabilit Engineering, 11*, 141-144.

Sivasankari, N., Thanushkodi, K., & Naidu, H. K. (2010). An extensive review of significant researches on epileptic seizure detection and prediction using electroencephalographic signals. *Advances in Biomedical Research*, 330-353.

Snyder, E. (1990). The electroencephalogram (EEG). *Biomedical Instrumentation & Technology, 24*(4), 296-298.

Song, Y., & Liò, P. (2010). A new approach for epileptic seizure detection: sample entropybased feature extraction and extreme learning machine. *Journal of Biomedical Science and Engineering, 3*(06), 556.

Steczek, M., Jefimowski, W., & Szeląg, A. (2020). Application of Grasshopper Optimization Algorithm for Selective Harmonics Elimination in Low-Frequency Voltage Source Inverter. *Energies, 13*(23), 6426.

Stevenson, N., Tapani, K., Lauronen, L., & Vanhatalo, S. (2019). A dataset of neonatal EEG recordings with seizure annotations. *Scientific Data, 6*(1), 1-8.

Sui, L., Zhao, X., Zhao, Q., Tanaka, T., & Cao, J. (2019). *Localization of Epileptic Foci by Using Convolutional Neural Network Based on iEEG.* Paper presented at the IFIP International Conference on Artificial Intelligence Applications and Innovations, Hersonissos, Crete, Greece, 331-339.

Sujitha, V., Sivagami, P., & Vijaya, M. (2010). Support vector machine-based epilepsy prediction using textural features of MRI. *Procedia Computer Science*, *2*, 283-290.

Sumathi, S., & Kumar, L. A. (2018). *Computational intelligence paradigms for optimization problems using MATLAB®/SIMULINK®*: CRC Press.

Sun, M., Wang, F., Min, T., Zang, T., & Wang, Y. (2018). Prediction for high-risk clinical symptoms of epilepsy based on a deep learning algorithm. *IEEE Access, 6*, 7759677605.

Sun, Z., Bebis, G., & Miller, R. (2004). Object detection using feature subset selection. *Pattern Recognition, 37*(11), 2165-2176.

Swami, P., Gandhi, T. K., Panigrahi, B. K., Tripathi, M., & Anand, S. (2016). A novel robust diagnostic model to detect seizures in electroencephalography. *Expert Systems with Applications, 56*, 116-130.

Tang, J., Alelyani, S., & Liu, H. (2014). Feature selection for classification: A Review. *Data Classification: Algorithms and Applications*, 37, 1-33.

Teijeiro, A. E., Shokrekhodaei, M., & Nazeran, H. (2019). The conceptual design of a novel workstation for seizure prediction using machine learning with potential eHealth applications. *IEEE Journal of Translational Engineering in Health and Medicine, 7*, 1-10.

Tian, X., Deng, Z., Ying, W., Choi, K.-S., Wu, D., Qin, B., Wang, S. (2019). Deep multiview feature learning for EEG-based epileptic seizure detection. *IEEE Transactions on Neural Systems and Rehabilitation Engineering, 27*(10), 19621972.

Truong, N. D., Kuhlmann, L., Bonyadi, M. R., Querlioz, D., Zhou, L., & Kavehei, O. (2019). Epileptic seizure forecasting with generative adversarial networks. *IEEE Access, 7*, 143999-144009.

Truong, N. D., Nguyen, A. D., Kuhlmann, L., Bonyadi, M. R., Yang, J., & Kavehei, O. (2017). A generalised seizure prediction with convolutional neural networks for intracranial and scalp electroencephalogram data analysis, *1707-01976*.

Tsiouris, K. M., Pezoulas, V. C., Zervakis, M., Konitsiotis, S., Koutsouris, D. D., & Fotiadis, D. I. (2018). A long short-term memory deep learning network for the prediction of epileptic seizures using EEG signals. *Computers in Biology and Medicine, 99*, 24-37.

Türk, Ö., & Özerdem, M. S. (2019). Epilepsy detection by using a scalogram based convolutional neural network from EEG signals. *Brain Sciences, 9*(5), 115.

Tzallas, A. T., Tsipouras, M. G., & Fotiadis, D. I. (2009). Epileptic seizure detection in EEGs using time-frequency analysis. *IEEE Transactions on Information Technology in Biomedicine, 13*(5), 703-710.

Ullah, I., Hussain, M., & Aboalsamh, H. (2018). An automated system for epilepsy detection using EEG brain signals based on a deep learning approach. *Expert Systems with Applications, 107*, 61-71.

Usman, S. M., & Hassan, A. (2018). Efficient Prediction and Classification of Epileptic Seizures Using EEG Data Based on Univariate Linear Features. *Jounal Computer, 13*(6), 616-621.

Usman, S. M., Khalid, S., & Aslam, M. H. (2020). Epileptic seizures prediction using deep learning techniques. *IEEE Access, 8*, 39998-40007.

Usman, S. M., Usman, M., & Fong, S. (2017). Epileptic seizures prediction using machine learning methods. *Computational and Mathematical Methods in Medicine,* 1-7.

Uyttenhove, T., Maes, A., Van Steenkiste, T., Deschrijver, D., & Dhaene, T. (2020). *Interpretable Epilepsy Detection in Routine, Interictal EEG Data using Deep Learning. In Machine Learning for Health, 355-366.*

Walter, B., Wang, S., & Chun-An, C. (2020). Real-time Inference and Detection of Disruptive EEG Networks for Epileptic Seizures. *Scientific Reports (Nature Publisher Group), 10*(1).

Wang, G.-G., Gandomi, A. H., Alavi, A. H., & Gong, D. (2019). A comprehensive review of krill herd algorithm: variants, hybrids and applications. *Artificial Intelligence Review, 51*(1), 119-148.

Wang, L., Xue, W., Li, Y., Luo, M., Huang, J., Cui, W., & Huang, C. (2017). Automatic epileptic seizure detection in EEG signals using multi-domain feature extraction and nonlinear analysis. *Entropy, 19*(6), 222.

Wang, X., Gong, G., & Li, N. (2019). Automated recognition of epileptic EEG states using a combination of symlet wavelet processing, gradient boosting machine, and grid search optimizer. *Sensors, 19*(2), 219.

Wei, X., Zhou, L., Chen, Z., Zhang, L., & Zhou, Y. (2018). Automatic seizure detection using three-dimensional CNN based on multi-channel EEG. *BMC Medical Informatics and Decision Making, 18*(5), 71-80.

Wei, X., Zhou, L., Zhang, Z., Chen, Z., & Zhou, Y. (2019). Early prediction of epileptic seizures using a long-term recurrent convolutional network. *Journal of Neuroscience Methods, 327*, 108395.

Xu, X., Rong, H., Trovati, M., Liptrott, M., & Bessis, N. (2018). CS-PSO: chaotic particle swarm optimization algorithm for solving combinatorial optimization problems. *Soft Computing, 22*(3), 783-795.

Xu, Z., King, I., Lyu, M. R.-T., & Jin, R. (2010). Discriminative semi-supervised feature selection via manifold regularization. *IEEE Transactions on Neural Networks, 21*(7), 1033-1047.

Yang, X.-S. (2010). A new metaheuristic bat-inspired algorithm. *Nature Inspired Cooperative Strategies for optimization*, 65-74.

Yang, Y., Zhou, M., Niu, Y., Li, C., Cao, R., Wang, B., Xiang, J. (2018). Epileptic seizure prediction based on permutation entropy. *Frontiers in Computational Neuroscience, 12,* 55.

Yıldız, B. S. (2020). The spotted hyena optimization algorithm for weight-reduction of automobile brake components. *Materials Testing, 62*(4), 383-388.

Yıldız, B. S., Pholdee, N., Bureerat, S., Erdaş, M. U., Yıldız, A. R., & Sait, S. M. (2021). Comparision of the political optimization algorithm, the Archimedes optimization algorithm and the Levy flight algorithm for design optimization in industry. *Materials Testing, 63*(4), 356-359.

Yildiz, B. S., Pholdee, N., Bureerat, S., Yildiz, A. R., & Sait, S. M. (2021). Enhanced grasshopper optimization algorithm using elite opposition-based learning for solving real-world engineering problems. *Engineering with Computers*, 1-13.

Zainuddin, Z., Huong, L. K., & Pauline, O. (2012). On the use of wavelet neural networks in the task of epileptic seizure detection from electroencephalography signals. *Procedia Computer Science, 11*, 149-159.

Zhang, G., Lu, J., & Gao, Y. (2015). Multi-level decision making. *Models, Methods and Applications, 5(1), 20-28*.

Zhang, T., & Geem, Z. W. (2019). Review of harmony search with respect to algorithm structure. *Swarm and Evolutionary Computation, 48,* 31-43.

Zhang, Y., Guo, Y., Yang, P., Chen, W., & Lo, B. (2019). Epilepsy seizure prediction on EEG using common spatial pattern and convolutional neural network. *IEEE Journal of Biomedical and Health Informatics, 24*(2), 465-474.

Zhang, Y., Zhang, Y., Wang, J., & Zheng, X. (2015). Comparison of classification methods on EEG signals based on wavelet packet decomposition. *Neural Computing and Applications, 26*(5), 1217-1225.

Zhang, Z., & Parhi, K. K. (2015). *Seizure prediction using polynomial SVM classification.* Paper presented at the 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Milano, Italy, 5748-5751.

Zhao, W., Zhao, W., Wang, W., Jiang, X., Zhang, X., Peng, Y., Zhang, G. (2020). A novel deep neural network for robust detection of seizures using EEG signals. *Computational and Mathematical Methods in Medicine, 2020, 1-9.*

Zhou, W., Liu, Y., Yuan, Q., & Li, X. (2013). Epileptic seizure detection using lacunarity and Bayesian linear discriminant analysis in intracranial EEG. *IEEE Transactions on Biomedical Engineering, 60*(12), 3375-3381.

APPENDIX A

GREY WOLVES

```
%___________________________________________________________________%
%  Grey Wold Optimizer (GWO) source codes version 1.0                %
%                                                                    %
%  Developed in MATLAB R2011b(7.13)                                  %
%                                                                    %
%  Author and programmer: Seyedali Mirjalili                         %
%                                                                    %
%         e-Mail: ali.mirjalili@gmail.com                            %
%                 seyedali.mirjalili@griffithuni.edu.au              %
%                                                                    %
%       Homepage: http://www.alimirjalili.com                        %
%                                                                    %
%   Main paper: S. Mirjalili, S. M. Mirjalili, A. Lewis              %
%               Grey Wolf Optimizer, Advances in Engineering         %
%               Software , in press,                                 %
%               DOI: 10.1016/j.advengsoft.2013.12.007                %  %
%
%___________________________________________________________________%    %
Grey Wolf Optimizer
% function
[
Alpha_score,Alpha_pos,Convergence_curve]=GWO(SearchAgents_no,Max_iter,
lb,ub,dim,fobj) clc; clear; close all; data=LoadData();
  nf=5;  fobj = @(u)
FeatureSelectionCost(u,nf,data);  dim =
data.nx;
```

```
Max_iter     =     2;
SearchAgents_no  =  1;
lb=0; ub=1;
% lb=ones(dim,1)*lb;
% ub=ones(dim,1)*ub;
% If all the variables have equal lower bound you can just
% define lb and ub as two single number numbers
N=[1 dim];   % Size of Decision Variables Matrix
% N=10;
% initialize alpha, beta, and delta_pos
Alpha_pos=zeros(1,dim);
Alpha_score=inf; %change this to -inf for maximization problems
Beta_pos=zeros(1,dim);
Beta_score=inf; %change this to -inf for maximization problems
Delta_pos=zeros(1,dim);
Delta_score=inf; %change this to -inf for maximization problems %Initialize the
positions of search agents
Positions=initialization_Grey(SearchAgents_no,dim,ub,lb);
Convergence_curve=zeros(1,Max_iter);
   l=0;% Loop
counter
% Main loop while l<Max_iter     for
i=1:size(Positions,1)


       % Return back the search agents that go beyond the boundaries of the search
space
        Flag4ub=Positions(i,:)>ub;
        Flag4lb=Positions(i,:)<lb;


Positions(i,:)=( Positions(i,:).*(~(Flag4ub+Flag4lb)))+ub.*Flag4ub+lb.*F lag4lb;
%         elite opposition learning
for i=1:size(Positions,1)           k=rand();
          for j=1:size(dim, 2)
              Positions(i,:)=k.*(ub+lb)-Positions(i,j);            end
end


        % Calculate objective function for each search agent
fitness=fobj(Positions(i,:));


        % Update Alpha, Beta, and Delta         if
fitness<Alpha_score
            Alpha_score=fitness; % Update alpha
Alpha_pos=Positions(i,:);         end                  if fitness>Alpha_score
&& fitness<Beta_score              Beta_score=fitness; % Update beta
Beta_pos=Positions(i,:);         end                  if fitness>Alpha_score
&& fitness>Beta_score && fitness<Delta_score
            Delta_score=fitness; % Update delta
Delta_pos=Positions(i,:);         end     end                a=2-
l*((2)/Max_iter); % a decreases linearly fron 2
to 0
```

```
  % Update the Position of search agents including omegas   for
i=1:size(Positions,1)     for j=1:size(Positions,2)
r1=rand(); % r1 is a random number in [0,1]
r2=rand(); % r2 is a random number in [0,1]
      A1=2*a*r1-a; % Equation (3.3)
      C1=2*r2; % Equation (3.4)

      D_alpha=abs(C1*Alpha_pos(j)-Positions(i,j)); % Equation (3.5)-part 1
      X1=Alpha_pos(j)-A1*D_alpha; % Equation (3.6)-part 1
                  r1=rand();
r2=rand();

      A2=2*a*r1-a; % Equation (3.3)
      C2=2*r2; % Equation (3.4)

      D_beta=abs(C2*Beta_pos(j)-Positions(i,j)); % Equation (3.5)-part 2
      X2=Beta_pos(j)-A2*D_beta; % Equation (3.6)-part 2

r1=rand();       r2=rand();

      A3=2*a*r1-a; % Equation (3.3)
      C3=2*r2; % Equation (3.4)

      D_delta=abs(C3*Delta_pos(j)-Positions(i,j)); % Equation
(3.5)-part 3
      X3=Delta_pos(j)-A3*D_delta; % Equation (3.5)-part 3

      Positions(i,j)=(X1+X2+X3)/3;% Equation (3.7)

end   end   l=l+1;
  Convergence_curve(l)=Alpha_score; end
%% Results  figure;
plot(Convergence_curve,'LineWidth',2);
xlabel('Iteration'); ylabel('Best Cost');
```

BAT

```
%%%     clc;
clear; close
all;


%% Problem Definition
   data=LoadData();
nf=5; %
for j=1:nf


CostFunction=@(u) FeatureSelectionCost(u,nf,data);% Cost Function
     nVar=data.nx;       % Number of Decision
Variables
VarSize=[1 nVar];   % Size of Decision Variables Matrix
VarMin=0;         % Lower Bound of Variables
VarMax=1;         % Upper Bound of Variables
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% BAT Default parameters n=2;      % Population size,
typically 10 to 40
N_gen=2;  % Number of generations
A=0.5;      % Loudness  (constant or decreasing) r=0.5;
% Pulse rate (constant or decreasing) % This frequency
range determines the scalings
% You should change these values if necessary
Qmin=0;         % Frequency minimum
Qmax=1;         % Frequency maximum
% Iteration parameters
N_iter=0;       % Total number of function evaluations  %
Dimension of the search variables d=nVar;
% Number of dimensions
% Lower limit/bounds/ a vector  Lb=-2*ones(1,d);
% Upper limit/bounds/ a vector
Ub=2*ones(1,d);
% Initializing arrays Q=zeros(n,1);
%  Frequency   v=zeros(n,d);       %
Velocities    %    Initialize    the
population/solutions    for    i=1:n,
Sol(i,:)=Lb+(Ub-Lb).*rand(1,d);
Fitness(i)= CostFunction(Sol(i,:)); end
% Find the initial best solution
[fmin,I]=min(Fitness); best=Sol(I,:);


% Start the iterations -- Bat Algorithm  % for t=1:N_gen,  % Loop over
all bats/solutions         for i=1:n,            Q(i)=Qmin+(Qmin-
Qmax)*rand;           v(i,:)=v(i,:)+(Sol(i,:)best)*Q(i);
          S(i,:)=Sol(i,:)+v(i,:);
          % Apply simple bounds/limits
          Sol(i,:)=simplebounds(Sol(i,:),Lb,Ub);
          % Pulse rate           if
rand>r
          % The factor 0.001 limits the step sizes of random walks
```

```
              S(i,:)=best+0.001*randn(1,d);           end

     % Evaluate new solutions
           Fnew=CostFunction(S(i,:));
     % Update if the solution improves, or not too loud
if (Fnew<=Fitness(i)) & (rand<A) ,
Sol(i,:)=S(i,:);
Fitness(i)=Fnew;            end


          % Update the current best solution            if
Fnew<=fmin,                 best=(S(i,:));
fmin=Fnew;           end         end
        N_iter=N_iter+n; end
% end
% Output/display
disp(['Iterations: ',num2str(t)]); disp(['Best Cost=',num2str(best(t)),'
Optimization
Threshold=',num2str(fmin)]);


% Application of simple limits/bounds
%% Results   figure;
plot(best,'LineWidth',2);
xlabel('Iteration'); ylabel('Best
Cost');
```

GRASSHOPPER_IGOA

```
clc; clear;
close all; data=LoadData();
  nf=5;  fobj = @(u)
FeatureSelectionCost(u,nf,data);
dim = data.nx;
Max_iter     =     2;
SearchAgents_no  =  2;
lb=0; ub=1;
% lb=ones(dim,1)*lb;
% ub=ones(dim,1)*ub;
% If all the variables have equal lower bound you can just
% define lb and ub as two single number numbers
N=[1 dim];   % Size of Decision Variables Matrix
% N=10; disp('GOA is now estimating the global optimum for your problem....')
  flag=0; if
size(ub,1)==1     ub=ones(dim,1)*ub;
lb=ones(dim,1)*lb; end
if (rem(dim,2)~=0) % this algorithm should be run with a even number of
variables. This line is to handle odd number of variables     dim =
dim+1;     ub = [ub; 100];     lb = [lb; -100];     flag=1; end
%Initialize the population of grasshoppers
GrassHopperPositions=initialization(N,dim,ub,lb);
% GrassHopperFitness = zeros(1,N);
GrassHopperFitness    =    zeros(N);    %
fitness_history=zeros(N,Max_iter);
fitness_history=zeros(1,Max_iter);     %
position_history=zeros(N,Max_iter,dim);
position_history=zeros(1,dim);
Convergence_curve=zeros(1,Max_iter);
Trajectories=zeros(1,Max_iter);
cMax=1;
cMin=0.00004;
%Calculate the fitness of initial grasshoppers
   for i=1:size(GrassHopperPositions,1)
if flag == 1
        GrassHopperFitness(1,i)=fobj(GrassHopperPositions(i,1:end-1));     else
        GrassHopperFitness(1,i)=fobj(GrassHopperPositions(i,:));     end
    fitness_history(i,1)=GrassHopperFitness(1,i);
position_history(i,1,:)=GrassHopperPositions(i,1,:);
Trajectories(:,1)=GrassHopperPositions(1); end
[sorted_fitness,sorted_indexes]=sort(GrassHopperFitness);    %
Find the best grasshopper (target) in the first population   for
newindex=1:N


Sorted_grasshopper(newindex,:)=GrassHopperPositions(sorted_indexes(newi ndex),:);
end


TargetPosition=Sorted_grasshopper(1,:);
TargetFitness=sorted_fitness(1);


% Main loop
l=2; % Start from the second iteration since the first iteration was dedicated
to calculating the fitness of antlions while l<Max_iter+1
         c=cMax-l*((cMax-cMin)/Max_iter); % Eq. (2.8) in the
paper
          for      i=1:size(GrassHopperPositions,1)
temp= GrassHopperPositions';          for k=1:2:dim
S_i=zeros(2,1);           for j=1:N              if
i~=j
                    Dist=distance(temp(k:k+1), temp(k:k+1)); %
Calculate the distance between two grasshoppers
```

```
                                          r_ij_vec=(temp(k:k+1,j)-
temp(k:k+1,i))/(Dist+eps);
% xj-xi/dij in Eq. (2.7)                      xj_xi=2+rem(Dist,2); % |xjd
- xid| in Eq. (2.7)

s_ij=((ub(k:k+1) -
lb(k:k+1))*c/2)*S_func(xj_xi).*r_ij_vec; % The first part inside the
big bracket in Eq. (2.7)                      S_i=S_i+s_ij;                  end
end
            S_i_total(k:k+1, :) = S_i;
                     end


        X_new = c * S_i_total'+ (TargetPosition); % Eq. (2.7) in the paper
GrassHopperPositions_temp(i,:)=X_new';       end
    % GrassHopperPositions     GrassHopperPositions=GrassHopperPositions_temp;
for
i=1:size(GrassHopperPositions,1)
        % Relocate grasshoppers that go outside the search space
 Tp=GrassHopperPositions(i,:)>ub';Tm=GrassHopperPositions(i,:)<lb';Grass
 HopperPositions(i,:)=(GrassHopperPositions(i,:).*(~(Tp+Tm)))+ub'.*Tp+lb '.*Tm;
        % Calculating the objective values for all grasshoppers         if
flag == 1
            GrassHopperFitness(1,i)=fobj(GrassHopperPositions(i,1:end-
1));         else
            GrassHopperFitness(1,i)=fobj(GrassHopperPositions(i,:));         end
fitness_history(i,l)=GrassHopperFitness(1,i);
position_history(i,l,:)=GrassHopperPositions(i,1,:);
Trajectories(:,l)=GrassHopperPositions(1);


        % Update the target         if
GrassHopperFitness(1,i)<TargetFitness
TargetPosition=GrassHopperPositions(i,:);
TargetFitness=GrassHopperFitness(1,i);          end     end
    Convergence_curve(l)=TargetFitness;
    disp(['In iteration #', num2str(l), ' , target''s objective = ',
num2str(TargetFitness)])
         l  =  l
+ 1; end     if
(flag==1)
    TargetPosition = TargetPosition(1:dim-1); end
%% Results   figure;
plot(Convergence_curve,'LineWidth',2);
xlabel('Iteration'); ylabel('Best Cost');
```

SALP_SWARM

```
%_____________________________________________________________________ __________
%  Salp Swarm Algorithm (SSA) source codes version 1.0
%
%  Developed in MATLAB R2016a
%
%  Author and programmer: Seyedali Mirjalili %
%         e-Mail: ali.mirjalili@gmail.com
%                 seyedali.mirjalili@griffithuni.edu.au  %
%       Homepage: http://www.alimirjalili.com
% %   Main paper:
%   S. Mirjalili, A.H. Gandomi, S.Z. Mirjalili, S. Saremi, H. Faris,  S.M.
Mirjalili,
%   Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems
%   Advances in Engineering Software
%   DOI: http://dx.doi.org/10.1016/j.advengsoft.2017.07.002
%_____________________________________________________________________
______________
   %
function
[FoodFitness,FoodPosition,Convergence_curve]=SSA(N,Max_iter,lb,ub,dim,f obj)
clc; clear; close all; data=LoadData();
  nf=1;  fobj = @(u)
FeatureSelectionCost(u,nf,data);
dim = data.nx;
Max_iter     =     2;
SearchAgents_no  =  5;
lb=0; ub=2;
% lb=ones(dim,1)*lb;
% ub=ones(dim,1)*ub;
% If all the variables have equal lower bound you can just
% define lb and ub as two single number numbers
N=[1 dim];   % Size of Decision Variables Matrix
% N=10; if size(ub,1)==1
ub=ones(dim,1)*ub;
lb=ones(dim,1)*lb; end


Convergence_curve = zeros(1,Max_iter);
%Initialize the positions of salps
SalpPositions=initialization_sap(N,dim,ub,lb); FoodPosition=zeros(1,dim);
FoodFitness=inf;


%     for i=1:size(SalpPositions,1)
%           k=rand();
%           for j=1:size(dim, 2) %              ub=min(ub);
%               lb=min(lb);
%               SalpPositions(i,:)=k.*(ub+lb)-SalpPositions(i,j);
%           end
%         end
%calculate the fitness of initial salps
   for
i=1:size(SalpPositions,1)
    SalpFitness(1,i)=fobj(SalpPositions(i,:)); end
[sorted_salps_fitness,sorted_indexes]=sort(SalpFitness);
   for
newindex=1:N
    Sorted_salps(newindex,:)=SalpPositions(sorted_indexes(newindex),:); end
FoodPosition=Sorted_salps(1,:);
FoodFitness=sorted_salps_fitness(1);
% w_max = 1.5;
% w_min = 0.4;
% a = 0.02;
% b = 25;#



%Main loop
l=2; % start from the second iteration since the first iteration was
dedicated to calculating the fitness of salps while l<Max_iter+1
%      w = w_max-(w_max - w_min)*l/Max_iter;  %
p = rand/(1 + exp(a*l-b));
            c1 = 2*exp(-(4*l/Max_iter)^2); % Eq. (3.2) in the
paper
         for
i=1:size(SalpPositions,1)

        SalpPositions= SalpPositions';
if i<=N/2
for j=1:1:dim                 c2=rand();                 c3=rand();
%%%%%%%%%%%%% % Eq. (3.1) in the paper %%%%%%%%%%%%%%               if c3<0.5
SalpPositions(j,i)=FoodPosition(j)+c1*((ub(j)lb(j))*c2+lb(j));
else
                    SalpPositions(j,i)=FoodPosition(j)c1*((ub(j)lb(j))*c2+lb(j));
end
                %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
end
        elseif i>dim/2 && i<dim+1
point1=SalpPositions(:,i-1);             point2=SalpPositions(:,i);

            SalpPositions(:,i)=(point2+point1)/2; % % Eq. (3.4) in the paper
end
                  SalpPositions=
SalpPositions';     end
         for
```

```
i=1:size(SalpPositions,1)
Tp=SalpPositions(i,:)>ub';
        Tm=SalpPositions(i,:)<lb';

SalpPositions(i,:)=(SalpPositions(i,:).*(~(Tp+Tm)))+ub'.*Tp+lb'.*Tm;
        %    elite opposition learning
%         for i=1:size(SalpPositions,1)
%           k=rand();
%           for j=1:size(dim, 2)
%               SalpPositions(i,:)=k.*(ub+lb)-SalpPositions(i,j); %           end
%         end

        SalpFitness(1,i)=fobj(SalpPositions(i,:));
if
SalpFitness(1,i)<FoodFitness             FoodPosition=SalpPositions(i,:);
            FoodFitness=SalpFitness(1,i);

end     end
    Convergence_curve(l)=FoodFitness;     l
= l + 1; end %% Results   figure;
plot(Convergence_curve,'LineWidth',2);
xlabel('Iteration'); ylabel('Best
Cost');
```

PSO

```
% %
clc; clear;
close all;

%% Problem Definition

data=LoadData();
nf=5;  % for
j=1:nf

CostFunction=@(u) FeatureSelectionCost(u,nf,data);        % Cost
Function    nVar=data.nx;       % Number of Decision Variables
VarSize=[1 nVar];   % Size of Decision Variables Matrix
VarMin=0;         % Lower Bound of Variables VarMax=1;
% Upper Bound of Variables



%% PSO Parameters

MaxIt=2;      % Maximum Number of Iterations
   nPop=2;        % Population Size (Swarm
Size)
```

```
% w=1;            % Inertia Weight
% wdamp=0.99;     % Inertia Weight Damping Ratio
% c1=2;           % Personal Learning Coefficient
% c2=2;           % Global Learning Coefficient
% Constriction Coefficients phi1=2.05; phi2=2.05;
phi=phi1+phi2;   chi=2/(phi-2+sqrt(phi^2-4*phi));
w=chi;          % Inertia Weight wdamp=1;       %
Inertia
Weight Damping Ratio c1=chi*phi1;    % Personal
Learning Coefficient c2=chi*phi2;    % Global
Learning Coefficient

% Velocity Limits
VelMax=0.1*(VarMax-VarMin);
VelMin=-VelMax;

%% Initialization
   empty_particle.Position=[];
empty_particle.Cost=[]; empty_particle.Out=[];
empty_particle.Velocity=[];
empty_particle.Best.Position=[];
empty_particle.Best.Cost=[]; empty_particle.Best.Out=[];

particle=repmat(empty_particle,nPop,1);
BestSol.Cost=inf;
for i=1:nPop

    % Initialize Position
    particle(i).Position=unifrnd(VarMin,VarMax,VarSize);
    % Initialize Velocity     particle(i).Velocity=zeros(VarSize);

    % Evaluation     [particle(i).Cost,
particle(i).Out]=CostFunction(particle(i).Position);
    % Update Personal Best     particle(i).Best.Position=particle(i).Position;
particle(i).Best.Cost=particle(i).Cost;
particle(i).Best.Out=particle(i).Out;

    % Update Global Best      if
particle(i).Best.Cost<BestSol.Cost

        BestSol=particle(i).Best;
             end
     end
BestCost=zeros(MaxIt,1);

%% PSO Main Loop
```

```
   for it=1:MaxIt
for i=1:nPop

        % Update Velocity         particle(i).Velocity = w*particle(i).Velocity
...
+ c1*rand(VarSize).*(particle(i).Best.Positionparticle(i).P
osition) ...
            +c2*rand(VarSize).*(BestSol.Position-particle(i).Position);         %
Apply Velocity Limits         particle(i).Velocity =
max(particle(i).Velocity,VelMin);         particle(i).Velocity =
min(particle(i).Velocity,VelMax);
        % Update Position         particle(i).Position =
particle(i).Position + particle(i).Velocity;

        % Velocity Mirror Effect
        IsOutside=(particle(i).Position<VarMin |
particle(i).Position>VarMax);
particle(i).Velocity(IsOutside)=particle(i).Velocity(IsOutside)
;

        % Apply Position Limits         particle(i).Position =
max(particle(i).Position,VarMin);         particle(i).Position =
min(particle(i).Position,VarMax);

        % Evaluation
        [particle(i).Cost, particle(i).Out] =
CostFunction(particle(i).Position);

        % Update Personal Best          if
particle(i).Cost<particle(i).Best.Cost
particle(i).Best.Position=particle(i).Position;
particle(i).Best.Cost=particle(i).Cost;
particle(i).Best.Out=particle(i).Out;

            % Update Global Best             if
particle(i).Best.Cost<BestSol.Cost

                BestSol=particle(i).Best;
end                end                        end
         end
    BestCost(it)=BestSol.Cost;
          disp(['Iteration ' num2str(it) ': Best Cost = '
num2str(BestCost(it))]);
         w=w*wdamp;

% end
```

```
%% Results
   figure;
plot(BestCost,'LineWidth',2);
xlabel('Iteration'); ylabel('Best Cost');
```

APPENDIX B

List of Published and On-Going Papers From Thesis

1. Umar. B. U, Muazu M. B, Agajo. J, Kolo. J. G (2021), Epilepsy Seizure Classification Using Artificial Neural Network and Linear Discriminant Analysis Algorithm, Nigerian Journal of Engineering Science Research (NIJESR). Vol. 4, Issue 3, pp. 21-37.

2. Buhari U. Umar, Mohammed B. Muazu, Jonathan G. Kolo, James Agajo, (2019), Epilepsy Detection Using Artificial Neural Network and Grasshopper Optimization Algorithm (GOA), 15th International Conference on Electronics Computer and Computation (ICECCO 2019).