Heliyon 10 (2024) e30241

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

Leveraging ANFIS with Adam and PSO optimizers for Parkinson's disease

Akram Pasha [a], Syed Thouheed Ahmed [b], Ranjith Kumar Painam [c], Sandeep Kumar Mathivanan [d,*], Karthikeyan P [e], Saurav Mallik [f,g,**], Hong Qin [h,***]

[a] Department of Computer Science and Engineering, REVA University, Bengaluru, India
[b] Indian Institute of Technology Hyderabad, India
[c] Department of ECE, Kallam Haranadhareddy Institute of Technology, Guntur, Andhra Pradesh, India
[d] School of Computer Science and Engineering, Galgotias University, Greater Noida, 203201, India
[e] Department of Computer Applications, School of Computer Science Engineering and Information Systems, Vellore Institute of Technology, Vellore, 632014, Tamil Nadu, India
[f] Department of Environmental Health, Harvard T H Chan School of Public Health, Boston, MA, 02115, USA
[g] Department of Pharmacology & Toxicology, The University of Arizona, Tucson, AZ, USA
[h] Department of Computer Science and Engineering, University of Tennessee at Chattanooga, Chattanooga, TN, 37403, USA

A R T I C L E I N F O

Keywords:
Artificial intelligence
Adaptive Neuro-fuzzy inference system (ANFIS)
Particle swarm optimization (PSO)
Parkinson's disease (PD)
Neural networks
Machine learning
Feature extraction
Data mining
Predictive analytics
Feature selection
Data analysis
Classification
Clinical decision support
Healthcare information technology

A B S T R A C T

Parkinson's disease (PD) is an age-related neurodegenerative disorder characterized by motor deficits, including tremor, rigidity, bradykinesia, and postural instability. According to the World Health Organization, about 1 % of the global population has been diagnosed with PD, and this figure is expected to double by 2040. Early and accurate diagnosis of PD is critical to slowing down the progression of the disease and reducing long-term disability. Due to the complexity of the disease, it is difficult to accurately diagnose it using traditional clinical tests. Therefore, it has become necessary to develop intelligent diagnostic models that can accurately detect PD. This article introduces a novel hybrid approach for accurate prediction of PD using an ANFIS with two optimizers, namely Adam and PSO. ANFIS is a type of fuzzy logic system used for nonlinear function approximation and classification, while Adam optimizer has the ability to adaptively adjust the learning rate of each individual parameter in an ANFIS at each training step, which helps the model find a better solution more quickly. PSO is a metaheuristic approach inspired by the behavior of social animals such as birds. Combining these two methods has potential to provide improved accuracy and robustness in PD diagnosis compared to existing methods. The proposed method utilized the advantages of both optimization techniques and applied them on the developed ANFIS model to maximize its prediction accuracy. This system was developed by using an open access clinical and demographic data. The chosen parameters for the ANFIS were selected through a comparative experimental analysis to optimize the model considering the number of fuzzy membership functions, number of epochs of ANFIS, and number of particles of PSO. The performance of the two ANFIS models: ANFIS (Adam) and ANFIS (PSO) focusing at

* Corresponding author.
** Corresponding author. Department of Environmental Health, Harvard T H Chan School of Public Health, Boston, MA, 02115, USA.
*** Corresponding author.
E-mail addresses: akram.pasha@reva.edu.in (A. Pasha), syed.edu.in@gmail.com (S.T. Ahmed), ranjithkumar.painam@gmail.com (R.K. Painam), sandeepkumarm322@gmail.com (S.K. Mathivanan), mailbox7576@gmail.com (K. P), smallik@arizona.edu, sauravmtech2@gmail.com, smallik@hsph.harvard.edu (S. Mallik), hong-qin@utc.edu (H. Qin).

https://doi.org/10.1016/j.heliyon.2024.e30241
Received 11 August 2023; Received in revised form 13 April 2024; Accepted 22 April 2024
Available online 26 April 2024
2405-8440/

ANFIS parameters and various evaluation metrics are further analyzed in detail and presented, The experimental results showed that the proposed ANFIS (PSO) shows better results in terms of loss and precision, whereas, the ANFIS (Adam) showed the better results in terms of accuracy, f1-score and recall. Thus, this adaptive neural-fuzzy algorithm provides a promising strategy for the diagnosis of PD, and show that the proposed models show their suitability for many other practical applications.

1. Introduction

Parkinson's disease (PD) is a chronic and progressive neurological disorder affecting motor skills, such as mobility, balance and coordination in about 10 million people around the world. It is the second most common neurodegenerative disorder after Alzheimer's disease (AD) [1]. It affects an estimated seven to 10 million people worldwide, and its economic burden on society amounts to more than $25 billion each year in the United States alone. PD is a chronic, progressive neurodegenerative disorder with motor and non-motor symptoms, characterized by the progressive loss of dopamine-producing neurons in the brain - mainly the substantia nigra pars compacta (SNpc) [2]. As PD is an age-related disorder, its prevalence increases with age, usually affecting people over the age of 65. Diagnosis of PD is mainly based on clinical trials, with physicians rating patients based on the severity of their symptoms, such as tremor, rigidity, and gait problems. However, it is difficult to accurately diagnose PD based solely on clinical trials. Hence, accurate and early prediction of PD can help improve treatments, reduce morbidity and mortality, as well as save healthcare costs. Thus, accurate diagnosis and timely treatment are critical for alleviating some of the symptoms of PD [3]. To aid in the accurate diagnosis of PD, research has focused on leveraging of machine learning algorithms and different signal processing techniques, such as electroencephalography (EEG), electromyography (EMG) and sound recordings to accurately identify PD-related brain activity.

Over the years, several techniques have been used to diagnose PD, including anatomical imaging, electrophysiological studies, neuropsychological tests, genetic screenings, proteomic analysis, laboratory tests, and ultrasound [4–6]. However, these methods of diagnosis are expensive, time-consuming, and/or technically challenging. Furthermore, due to the complexity of PD, no single method can provide a complete view of the various signs and symptoms associated with the disease [7]. In the medical domain, Artificial Intelligence (AI) has been widely used for diagnosis and treatment purposes. In particular, Machine Learning (ML) techniques such as Neural Networks (NNs), Support Vector Machines (SVMs) and Fuzzy Logic (FL) are commonly employed for the diagnosis of PD [8–10]. While ML based systems have proven to deliver promising results for PD detection, certain limitations in accuracy and speed still persist. For example, NNs cannot effectively utilize the available prior knowledge through rule-based systems and require a long training period when dealing with large datasets. On the other hand, FL based systems are limited by their hand-crafted fuzzy rules, which may lead to inaccurate diagnosis.

Recently, Machine Learning (ML) approaches have been used to aid in the diagnosis of PD. ML has the potential to identify patterns and relationships in large datasets, providing a means of accurately diagnosing and monitoring the progress of PD [11]. However, ML algorithms require significant amounts of data, which is often unavailable or difficult to collect. To address this problem, researchers have proposed the use of hybrid ML models, combining multiple algorithms to take advantage of the strengths of each approach [12]. Recently, the combination of ANFIS and PSO have become increasingly popular for solving complex machine learning problems. ANFIS is a type of artificial neural network that combines the strengths of fuzzy logic and learning of neural networks. It is particularly suitable for modeling nonlinear systems, as it can take into account uncertain, incomplete, or imprecise input-output relationships. PSO is a computational process that mimics the behavior of social animals, such as birds and fish, in order to find the optimum solution to a problem. It has been used in many applications, including pattern recognition, optimization and feature selection.

To overcome these limitations, the current study propose the use of ANFIS, a combination of NN and FL, to accurately predict PD. However, the term 'predict' here, specifically pertains to the model's capability to diagnose or classify individuals as affected by PD based on learned patterns from input data. It focuses on the developed predictive model leveraging ANFIS integrated with Adam and PSO optimizers, aiming to accurately identify PD cases. The study emphasizes that 'predict' does not encompass forecasting disease progression, therapy responses, or developmental aspects of PD, but, the developed model aims to diagnose PD, discriminating between affected and unaffected individuals by analyzing a comprehensive set of features. The features that are employed in the study encompass both motor and non-motor aspects associated with PD, aiming for a holistic representation of the disease spectrum. Furthermore, while the major focus remains on PD diagnosis, the differentiation between PD and Parkinsonism, and distinguishing various forms of Parkinsonism, represents significant avenues for future research endeavors.

Subsequently, ANFIS is a supervised learning algorithm that combines NN and FL structures to capture the non-linearity of data and extract the relevant knowledge from it. Furthermore, to further enhance the model's performance, Adam optimizer, and PSO is introduced to optimize the ANFIS model. PSO is a stochastic optimization technique inspired by the behavior of a swarm of birds seeking food. Both the optimizers are used to optimize parameters in the ANFIS model to achieve the best fit for data.

Problem Statement: The objective is to design a model for accurate PD prediction. The model needs to identify the best predictive features from the given dataset, before training an ANFIS model with PSO as an optimization strategy. The Mathematical Formulation can be stated as follows:

Let $X = \{x_1, x_2, ..x_m\}$ be the feature vector; $Y = \{y_1, y_2, ..y_n\}$ be the target variable; $F = \{f_1, f_2, ..f_n\}$ be the set of selected features; $W = \{w_1, w_2, ..w_k\}$ be the weight vector of the ANFIS; and let $J(W)$ be the cost function to be optimized.

The optimization problem can be defined as: Minimize equation (1) subject to constraints in equations (2) and (3).

$$J(W) = min\left[\sum((Y_n - f(X, W)))\right] \tag{1}$$

$$F = ExtraTreeClassifier(X, Y) \tag{2}$$

$$W = PSO(J(W)) \tag{3}$$

In equation (1), $f(X, W)$ is the optimized ANFIS model with weights W obtained from PSO.

The aim of this study is to investigate the use of an ANFIS combined with the optimizers: Adam and PSO, for accurate PD prediction. Thereby, this research explores the potential of leveraging ANFIS with Adam and PSO for accurate PD prediction. These hybrid models will be evaluated using a well-known dataset containing clinical and demographic information related to PD patients. More specifically, this study aims to perform the performance analysis of ANFIS (Adam) and ANFIS (PSO) in predicting PD. The results of this study should provide insights into the effectiveness of the hybrid approach for PD diagnosis and help guide further development of ML-based PD diagnostic tools.

The main contributions of this paper are:

- Identifying the top ranked features of PD using ensemble learning.
- Developing the hybrid classification models by combining ANFIS with Adam and PSO optimizers for accurate PD diagnosis.
- Finding the optimal configuration of the ANFIS parameters that will yield the highest classification accuracy.
- Conducting the comprehensive comparative performance analysis of ANFIS (Adam) and ANFIS (PSO).
- The results demonstrate that the proposed models demonstrate their suitability in predicting PD.

The rest of the paper is organized as follows. Section 2 covers the related works in PD detection using various ML approaches. Section 3 introduces ANFIS based classification of PD and discusses the two models' components. Section 4 presents the detailed evaluation of the two models predicting PD based on series of evaluation metrics. Finally, the conclusion and future work are provided in Section 5.

2. Related work

This chapter presents the review of the literature on the use of ANFIS with PSO as a solution domain for solving any prediction problem clubbed with optimization through nature-inspired algorithms. It presents the current state of the art, summarize the results of recent studies, and explore the potential of ANFIS (PSO) for future PD diagnosis. The review is performed systematically to evaluate the current evidence on the use of ANFIS (PSO) for PD prediction. The relevant articles in scientific databases such as Google Scholar, PubMed, SciELO, EMBASE, and Web of Science were searched. The search terms used were "Adaptive Neuro-Fuzzy Inference Systems" OR "ANFIS" AND "Particle Swarm Optimization" OR "PSO" AND "Parkinson's Disease" OR "PD". Studies published in English between 2013 and 2023 were included in the review.

ANFIS is a supervised machine learning approach that combines the power of fuzzy logic and ANNs. PSO is a population-based optimization algorithm based on the concept of social behavior of birds, fish, bees and other animals, to identify optimal solutions for complex problems. PSO is used in the field of machine learning to optimize the choice of model parameters, which leads to improved accuracy in predictions. The combination of ANFIS tuned with PSO has been successfully used for various classification tasks in the past.

A combination of hybrid approaches using GA, PSO, and ANFIS was incorporated in the work of [13] to identify critical input variables that have a substantial impact on the amount of power a PV generation plant can produce. And, eventually to build the PV power forecasting model for the plant. Similar combination of hybrid approaches were employed in the works of [14,15], to perform landslide spatial modelling and its zonation, and to predict the stress intensity factor, respectively. However, in the works of [16,17], the authors incorporated many probabilistic and hybrid non-linear machine learning algorithms to classify PD. The authors in the work of [17] performed the GA and PSO based dimensionality reduction to find the most optimal features of the PD data set to further train many variety of ML algorithms to predict PD.

In addition to the use of GA with PSO to optimize ANFIS few of the works employed other optimization algorithms to optimize the parameters of the ANFIS. One such work is reported in Ref. [18], wherein, the authors perform the comparative study of the two optimization algorithms: PSO and Brain Storm Optimization; to determine the weight of neurons in the ANFIS technique, which is used to forecast when a bus would arrive at a bus stop. In their research, the authors report that the ANFIS with PSO algorithm produced superior results in terms of predicting bus arrival time than ANFIS BSO. The authors of [19] used the integration of ANFIS and PSO to forecast Iran's inflation rate. The authors trained the ANFIS by PSO to construct the model forecasting the inflation rate using time series data from the Central Bank of the Islamic Republic of Iran. The wavelet transform, ANFIS, and hybrid firefly and PSO method are combined in the forecasting model the authors created, known as WT-ANFIS-HFPSO (HFPSO). By quantifying carbon dioxide (CO2) dissolution in oil, the authors of [20] made an effort to address the issue of estimating the prospective and long-term behavior of CO2 in reservoir during secondary and tertiary oil recovery. In this study, a model built on ANFIS is created for precise CO2 diffusivity prediction in oils at high temperatures and pressures. To find the ideal ANFIS model parameters, PSO, a population-based stochastic search algorithm, was used. The authors of [21] suggested a hybrid evolutionary-adaptive methodology that successfully combined mutual information, wavelet transform, ANFIS, and PSO for short-term wind power forecasting. To forecast the ferrofluid's heat transfer coefficient in laminar and turbulent flows, the authors of [22] used a computational fluid dynamics simulation and an ANFIS optimized with PSO.

Further, there have been number of studies reported in the direction of solving optimization problems in the Power Distribution domain. For example, PSO and ANFIS were combined in the study of [23] to address the issue of reducing the imbalanced voltage sag with phase jumps by unified power-quality conditioner with little real power injection. PSO was primarily used in their study to reduce the real power infusion of UPQC as well as the restrictions. Using the PSO-based data for various voltage sag circumstances, ANFIS was then employed for minimum real power injection with UPQC. To accurately anticipate the electrical power generation depending on meteorological parameters, the authors of [24] used ANFIS with PSO to create a novel electrical power prediction model. In order to improve the performance of green energy, the authors of [25] used PSO and ANFIS to maximize the output power of photovoltaic systems while minimizing the energy payback time. In an effort to ensure the grid integration of renewable energy, the problem of solar power forecasting has been attempted to be solved in the work of [26] using ANFIS and PSO. The authors of [27] used machine learning to solve the challenge of predicting the voltage stability margin of power systems via the critical boundary index approach. On the basis of an ANFIS, prediction models were created, and its improved model was developed with PSO.

However, there have also been a few studies reported that employed ANFIS tuned with PSO in other variety of problem domains including the healthcare. For example, in agriculture domain, the authors of [28] used ANFIS and PSO to calculate the saffron yield based on the terrain characteristics in the Siminehrood catchment, which is located south of Urmia Lake, Iran. In the area of human resource management, the study conducted in Ref. [29], an algorithm based on the fusion of PSO with random weight and ANFIS is used to solve the problem of properly predicting the human resource structure. The authors of [30] used ANFIS optimized with PSO to identify Benzene, a carcinogen, using the hardware sensors to do so with only moderate operational efficiency. According to the authors of [31], a hybrid intelligent system that combined ensemble learning methods with ANFIS has the potential to help medical professionals in the practice of early diabetic retinopathy identification.

Most studies used ANFIS (PSO) as a tool for multi-class classification, while some used it for binary classification. Therefore, it is evident from the literature that there have been numerous studies that utilize ANFIS for developing the predictive models. In particular, ANFIS has the potential to detect PD from clinical data gathered from voice analysis, speech synthesis, gait analysis, and electroencephalography. The main advantage of ANFIS is that it can produce accurate predictions even when limited input data is available.

The evidence reviewed in the current article suggests that ANFIS (PSO) is a highly effective tool for accurately predicting PD. Furthermore, ANFIS (PSO) has the advantage of being easily interpretable, which makes it a suitable choice for medical diagnosis. The findings of this review provide valuable insights into the potential of ANFIS (PSO) for PD prediction. This paper leverages the capabilities of both ANFIS with Adam and PSO to predict PD through a process of optimizing the parameters of an ANFIS model. As this is a novel approach, the previous literature reviewed was mainly used for establishing the foundations for the proposed methodology.

Research Gap. The literature review shows that each study published is typically based on one or more of the following criteria: selection of the data sets, selection of the data set groups, and selection of the various parameter settings for the ANFIS and a variety of hybrid algorithms used for optimization, and finally, the evaluation techniques employed for the performance analysis. It is evident from the literature review that there is still significant scope for enhancing the prediction of PD through novel intelligent algorithms. As a result, the ANFIS Classification models tuned using Adam and PSO are developed and examined in the current study by using the PD data set. The materials and methods utilized to accomplish the study's main goal are discussed in the section that follows.

3. Materials and methods

3.1. Proposed system

ANFIS (Adam) and ANFIS (PSO) have been used as the predictive models to classify the PD dataset. ANFIS is a type of artificial neural network that is designed to combine the fuzzy logic and adaptive learning ability of backpropagation algorithms for better prediction accuracy. While Adam Optimizer helps to reduce the amount of time needed to find an optimal solution, the PSO is an

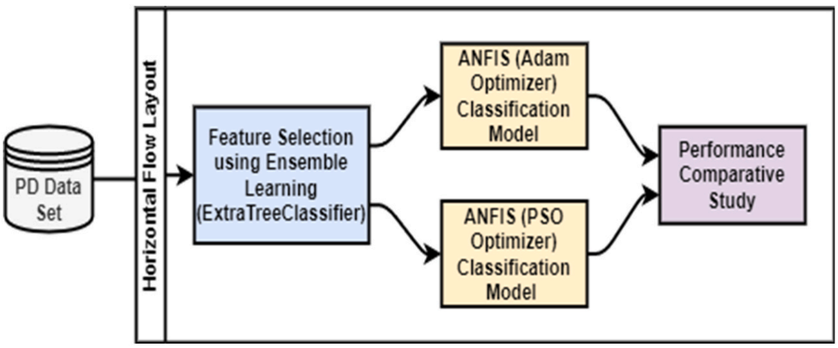


Fig. 1. Framework of the proposed system.

evolutionary optimization algorithm which can be used to tune the parameters of the ANFIS model. PSO was used to identify the optimal parameters of the ANFIS model and to improve the model's accuracy. The methodology employed in this research included analyzing empirical data and training the ANFIS model via Adam and PSO to determine the most appropriate parameters to represent the PD data. The proposed methodology adopted in this study is illustrated as in Fig. 1 in the most generic sense. The rest of the chapter gives the details of the proposed system.

3.1.1. PD data set description

Many research studies published in the domain of applications of ML algorithms have used the publicly available datasets. The PD data set used in this research study was collected from the UCI ML Repository [32]. Overall, there are 756 instances in the data set, each containing 755 floating-point features, and only one decision variable includes binary values. This data collection has been used in many research studies published with the aim of accurately identifying PD [17].

Table 1 briefly describes each of the attributes in this data collection [17]. The data was obtained from 252 patients, of which 188 (107 men and 81 women) had PD and 64 (23 men and 41 women) had no symptoms of the disease and were between the ages of 41 and 82. The entire data collection procedure was carried out in accordance with the qualified clinical expert's instructions. Three repeats of the sustained phonation of an "a" vowel were used to target each of the PD traits during the data gathering phase. The measurements were taken with a microphone tuned to 44.1 KHz. Then, from the patient's voice recordings, the various speech signal processing measurements, such as Time-Frequency, Mel Frequency Cepstral Coefficients (MFCC), Wavelet Transforms, Vocal Folds, and Tunable Q-factor Wavelet Transforms (TWQT), were taken. The total PD data set consists of 755 characteristics and 756 recorded observations, or rows and columns, respectively. One feature has values between 0 and 2, one feature has values between 0 and 2, and two features contain binary values out of a total of 755 features or columns. This data set's last feature's binary value denotes a choice or a class variable. An ML classifier can use the class variable to clinically extract the necessary data about a PD patient.

3.1.2. Data pre-processing

For training and testing purposes, the data was divided into 70:30 ratio respectively. The data pre-processing step involved feature scaling of the attributes present in the dataset. The main objective of the feature scaling step was to ensure that the trainable weights remain within the same range for all input attributes. This was achieved by bringing the values of each attribute between 0 and 1, which was done by using the Min-Max normalization technique. Normalizing is the process of rescaling a set of values so they fall within a specified range, usually 0 to 1 or -1 to +1. This is done by subtracting the minimum value of the data set from each value in the dataset, and then dividing each value by the difference between the maximum and minimum values of the dataset. Normalizing adjusts the data so all values are in the same range.

3.1.3. Feature selection using ensemble learning

Feature selection using ensemble learning is a powerful technique for reducing the number of features in a dataset while maintaining the accuracy of a predictive model. This technique can be applied to datasets with a large number of features, such as PD data. Ensemble learning combines predictions from multiple base learners (models) to create a more accurate and robust prediction than any single model can provide. In the case of feature selection, ensemble learning can be used to identify which of the many potential input features should be included in a predictive model. The Extra Tree Classifier class from the scikit-learn [33] python package is one way to perform feature selection using an ensemble approach. The Extra Trees Classifier uses an ensemble of decision trees to evaluate the importance of each feature in the dataset. Each tree is grown using a random set of the features, and then evaluated using some measure of performance; a feature that consistently performs well across multiple trees is considered "important". The Extra Tree Classifier then ranks the features according to importance and selects only the best performing ones. This method has several advantages over traditional feature selection techniques such as filter or wrapper methods. It is computationally efficient, and it can handle large numbers of features without becoming computationally intractable. Additionally, extra trees are easy to interpret, so it is possible to understand why certain features have been selected or excluded. Overall, feature selection using ensemble learning is a powerful tool for selecting the best features from PD data. It is efficient, robust, and interpretable, making it an ideal choice in this study having the PD dataset.

Table 1
Description of PD data set [17].

Attribute Name	Number of Columns, Type of Data
ID	1, Col_1 (Decimal)
Gender	1, Col_2 (Binary)
Baseline features	21, Col_3 to Col_23 (Real)
Intensity parameters	3, Col_24 to Col_26 (Real)
Formant frequencies	4, Col_27 to Col_30 (Real)
Bandwidth parameters	4, Col_31 to Col_34 (Real)
Vocal fold features	22, Col_35 to Col_56 (Real)
Mel frequency cepstral coefficients (MFCCs) features	84, Col_57 to Col_140 (Real)
Wavelet transform-based features	182, Col_141 to Col_322 (Real)
Tunable Q-factor wavelet transform (TQWT) features	432, Col_323 to Col_754 (Real)
Status (decision variable)	1, Col_755 (Binary)

3.2. ANFIS based classification of PD

This section provides a brief overview of ANFIS, which is a hybrid system that integrates Artificial Neural Network (ANN) and Fuzzy Inference System (FIS). It is a nonlinear prediction model, which is used in the current study to predict PD from patient data. ANFIS has two distinct stages where in the first stage, based on user-defined membership functions and input data, the fuzzy rules are generated. In the second stage, an iterative learning process is applied to tune up the parameters of the membership functions and optimizes the fuzzy rules. It is built to identify nonlinear relationships between input and output, and it is widely used for various practical applications such as prediction and control tasks.

The ANFIS model consists of five main layers:

- **Input Layer.** This layer collects input data, and has as many neurons as the number of inputs.
- **Fuzzification Layer.** This layer contains neurons that map the inputs from the input layer to their corresponding membership functions in the rule base.
- **Rule Base.** This layer contains a set of linguistic rules which describe the relationship between the inputs and outputs.
- **Inference Layer.** This layer evaluates the output of the rule base by combining the individual outputs of each rule.
- **Output Layer.** This layer performs the final mapping of the layer's inputs to the corresponding output variables.

The primary components of the ANFIS are the rulebase, the input membership functions (MFs), and output MFs. The rulebase contains information about the PD condition, including all relevant symptoms and diagnostic metrics. This rulebase is then used to determine the input MFs, which are defined as the fuzzy sets of data points corresponding to each symptom or diagnostic metric. The MFs act as the weights or importance associated with each data point in the rulebase. Finally, the output MFs are determined by the ANFIS, based on the inputs and the rulebase. These output MFs are then used to construct a prediction model for PD. This model is then used to make predictions about the likelihood of PD in a given patient.

The ANFIS model is a widely used machine learning model for classification and regression tasks. It is a combination of a traditional neural network and a fuzzy logic system, which can be used for tasks such as pattern recognition, data classification, and prediction. The best 5 input features are extracted using ensemble learning, and one output feature is used for training the ANFIS. First, the model takes the five features and converts them into fuzzy membership functions. These are sets of fuzzy numbers, which represent the degree to which each feature falls within a certain range. Next, the input membership functions are combined with a set of fuzzy rules to generate an output function that describes the likelihood of a given record being in either the PD or healthy class. Finally, an adaptive network is used to "tune" the model using the two optimizers: Adam and PSO, separately, so it accurately fits the data and makes accurate predictions. This includes adjusting the weights of the parameters and membership functions. Once trained, the ANFIS model is then used to classify unseen data points as either PD or healthy, depending on the fuzzy set membership functions and rules.

Premise functions are inputs of an ANFIS network that are used to define the fuzzy set membership function. A Gaussian membership function, which is used as a premise function in this research, is a bell-shaped curve that is used to represent the degree of membership on a scale from 0 to 1. Consequent functions are outputs of an ANFIS network and they are used to determine how the network will react when it receives certain inputs. An example of a consequent function is a linear rule, which determines how the output will respond when certain levels of input are received. Therefore, in the current study, experimentation is done to determine the optimum number of premise and consequent functions to be used for ANFIS. As the number of functions increases, the performance of the model can be monitored and compared, allowing for the selection of the best configuration. The results section of this article presents the results of experimentation conducted on determining the optimum number of premise and consequent functions.

In this current research, the Backpropagation is used for ANFIS (Adam) and Gradient Descent is used for ANFIS (PSO). Backpropagation is an optimization algorithm used to efficiently adjust the weights of a model according to the error gradient of a given cost

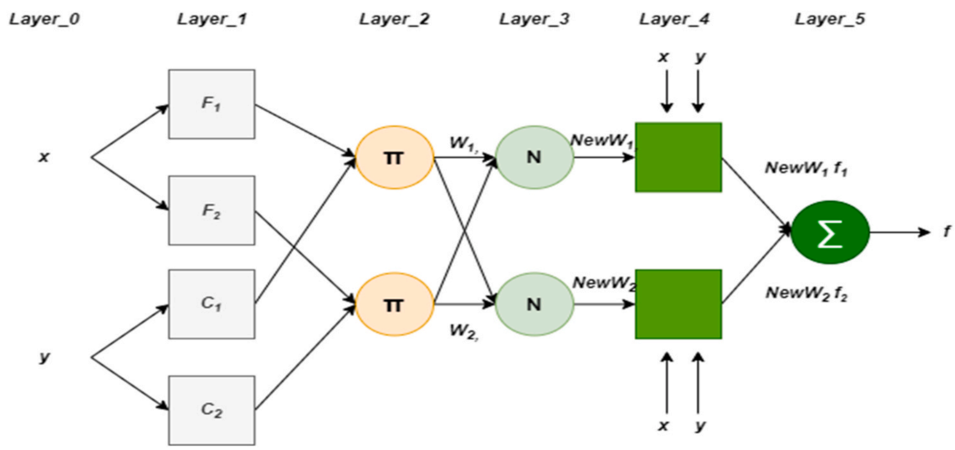


Fig. 2. ANFIS architecture.

function. When training ANFIS (Adam), backpropagation is used to update the weights of the model at each step in order to minimize the cost of the model and improve its accuracy. This approach is more efficient than simply using gradient descent, as the Adam Optimizer helps to reduce the amount of time needed to find an optimal solution. Whereas, Gradient descent is used to optimize a cost function by incrementally updating weights and parameter values in the model so that the cost of the model is minimized. ANFIS (PSO) is a data classification technique which uses the gradient descent algorithm to optimize various parameters in order to minimize the cost associated with misclassifications. This approach can be especially useful when dealing with large datasets and complex classifications.

3.2.1. Architecture

The generic architecture of an ANFIS system for the two inputs first order Takagi-Sugeno fuzzy model with two rules is as shown in Fig. 2. The ANFIS employed in the current study consists of an input layer, an output layer, three fuzzy layers, and two learning layers. The input layer is composed of membership functions for each input variable and the output layer is composed of the resulting output of the fuzzy system. The first two fuzzy layers are membership function layers, consisting of triangular membership functions to represent each input variable. These membership functions are used to define the range of input values and its corresponding decision values. The third fuzzy layer is a rule layer, which provides the "firing strength" of the rules. This layer is responsible for comparing the values of the input variables and deciding which rule to fire. The learning layers are composed of a network of neurons connected to the input, output, and fuzzy layers. The learning layers use training algorithms such as backpropagation to adjust the weighting of the neurons so that the error between the desired output and the fuzzy output is minimized.

After being trained, the ANFIS system can be used to make predictions using new input data. The performance of an ANFIS system is measured using many evaluation metrics such as loss, accuracy, f1-score, precision and recall.

3.2.2. Training

In this research article, the two ANFIS models are implemented to classify PD data. To train the two models, the two different optimization techniques: Adam Optimizer and PSO are employed separately. For Adam Optimizer, Tensor Flow library is used, and the model is optimized on a PD dataset by adjusting various parameters such as learning rate, batch size and number of iterations. For PSO, PySwarm [35] library is used, and the model is tuned by optimizing the particle movement parameters such as inertia factor, acceleration coefficients and neighbourhood size. Both models were then tested using the test set of the PD data set to evaluate their performance. The ANFIS (Adam) achieved higher classification accuracy than the ANFIS (PSO). This indicates that careful parameter selection is the key to achieving accurate results. However, further tuning of the parameters with different optimization techniques, such as ensemble and multi-objective optimization, can further improve the performance of these models.

3.2.3. Parameter estimation

The objective of both Adam Optimizer and PSO when they are used to tune an ANFIS model is to find the optimal configuration of the parameters that will yield the highest classification accuracy. This is done by calculating the gradients of the loss function with respect to each parameter of the model, and then updating the parameters accordingly. Adam Optimizer utilizes a momentum term which helps it avoid local minima and allows it to find better values for the parameters. Whereas, PSO uses a swarm intelligence technique to optimize the parameters of the model by adjusting the particle movement parameters such as inertia factor, acceleration coefficients, and neighbourhood size. Both Adam Optimizer and PSO are effective optimization algorithms for tuning an ANFIS model and can help it achieve higher classification accuracy. However, Adam Optimizer is considered to be computationally efficient and has a faster training time than PSO, while PSO is preferred when we need better generalizability to new data than Adam Optimizer. Adam optimizer is a gradient-based optimization algorithm used to efficiently train ANN. It works by calculating the gradients of the loss function with respect to each parameter of the model, and then updating the parameters accordingly. Both algorithms are effective at tuning an ANFIS model for classifying the PD data set and can help the model achieve higher classification accuracy.

3.2.4. PSO

PSO is an evolutionary optimization technique developed by Kennedy and Eberhart in 1995 [34]. It is based on the behavior of swarms and has been used to solve many difficult optimization problems. It has several advantages over other evolutionary algorithms such as genetic algorithms, simulated annealing and tabu search. Some of these advantages are that it requires fewer parameters, is computationally efficient, and easy to implement. The PSO algorithm updates the position (vector of the estimated parameters) of each particle in the swarm according to its own best position, the current global best position and the random velocity vector. The global best position of the swarm is updated for every iteration. PSO takes inspiration from the natural process of swarming behavior of animals in their search for food. In PSO, each particle is a point in the search space and modifies its position according to the combined effects of its own and others' experiences as it searches for an optimum solution.

PSO Algorithm. The basic step-by-step algorithm for PSO is described as follows:

Step 1. Initialize the particle positions randomly.

Step 2. For each particle calculate its fitness value.

Step 3. For each particle update the velocity and position based on equations (4) and (5).

$$v_i^{t+1} = v_i^t + C_1 \,.\, r_1 \,.\, \left(pbest_i - x_i^t\right) + C_2 \,.\, r_2 \,.\, \left(gbest_i - x_i^t\right) \tag{4}$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{5}$$

in equations (4) and (5).

- v_i^t is the current velocity of particle i at time t
- w is the inertia weight
- C_1 and C_2 are the cognitive and social parameters respectively
- r_1 and r_2 are random numbers between 0 and 1
- $pbest_i$ is the personal best position for particle i.
- $gbest_i$ is the global best position for particle i.
- x_i^t is the current position of particle i at time t.

Step 4. If the maximum number of iterations has been reached, then terminate the process, else go back to Step 2.

PSO Parameters. In PSO, each particle represents a candidate solution to the problem and is composed of position and velocity vectors. The position vector represents the values of the decision variables and the velocity vector determines the direction and magnitude of the movement of each particle. The objective of PSO is to minimize or maximize an objective function.

The following parameters are used by PSO in order to achieve this goal:

Swarm size. This parameter is used to define the number of particles present in the swarm. A larger swarm size leads to better exploration of the search space.

Inertia weight. This parameter is used to define how quickly the particle velocity should be changed. If the inertia weight is too large, then particles will maintain their current velocity and will not explore the search space efficiently. On the other hand, if the inertia weight is too small, particles may be stuck in local minima and not converge to the global optimum.

Velocity limits. These parameters are used to constrain the maximum and minimum velocities at which the particles can move.

Cognitive acceleration (C_1). This parameter governs the influence of a particle's own experience on its movement.

Social acceleration (C_2). This parameter governs the influence of a particle's neighbor's experience on its movement.

Neighbourhood size. This parameter is used to define the number of neighbors a particle interacts with in the swarm.

3.3. Mathematical formulation of the proposed system

The mathematical formulation for leveraging ANFIS (PSO) for accurate PD prediction is formulated as an optimization problem. The objective function is comprised of two terms: the first term is an ANFIS performance metric which measures the quality of the model, and the second term is a cost function which measures the accuracy of the prediction. The optimization problem is then solved using PSO, which searches for the optimal parameters such that the objective function is minimized.

The ANFIS model is represented by equation (6), where, m is the number of fuzzy sets and N is the number of memberships, A_j and H_j are the output function and membership function parameters respectively which need to be optimized. The optimization of the model parameters (A_j and H_j) is done using PSO.

$$Y = \sum_{j=1}^{N} m\left(A_j * H_j\right) \tag{6}$$

The cost function used for training the model is given by equation (7), where, N is the total number of training samples and Y_i and T_i are the predicted and actual class labels respectively.

$$J = \sum_{i=1}^{N} \|(Y_i - T_i)\|^2 \tag{7}$$

Extra Tree Classifier extracts the best 5 features from the dataset which are then used to train the ANFIS model. This method is particularly useful in identifying complex nonlinear relationships among a large number of features such as the PD data set. Subsequently, the parameters of the ANFIS model (A_j and H_j) are optimized using PSO, which is a population-based search algorithm inspired by the social behaviors of animals. The velocity and position of the particle (p) are updated according to the simple rules as defined in equations (4) and (5). After training, the model is tested on the test dataset to evaluate the performance of the model using metrics such as loss, accuracy, f1 score, precision and recall,.

The overall objective of the problem is to optimize the parameters of an ANFIS model in order to produce a model with the maximum performance while minimizing the cost of the model. To do this, the ANFIS performance metric is used to measure the quality of the model, while the cost function is used to measure the accuracy of the prediction. The ANFIS performance metric is formulated as the *MSE*, shown in equation (8), between the model output and the actual output of the PD patient data set.

$$MSE = \frac{\sum_{i=1}^{n*p} \|(y_i - \widehat{y})\|^2}{n * p} \tag{8}$$

In equation (9), n = number of data points, p = number of features. The cost function is then used to evaluate the accuracy of the prediction by penalizing the *MSE* with a parameter W, which represents the importance of accuracy over model performance using equation (9).

$$Cost = W * MSE \tag{9}$$

The optimization problem is then formulated as a multi-objective optimization problem, where the objective function is defined as the combination of the ANFIS performance metric and the cost function as shown in equation (10).

$$Objective\ Function = MSE + Cost \tag{10}$$

This problem is then solved by PSO, which searches the parameter space to find the parameter values that minimize the objective function. The parameters are optimized using a swarm of particles, where each particle is assigned a position, velocity and fitness value and is updated iteratively through the PSO algorithm using equations (4) and (5). The particles of the swarm used in PSO correspond to solutions in the search space and act as an optimization heuristic. The particles move through the search space and update their positions according to a few simple rules. At each iteration, the particle moves towards the position in the search space for which it holds the highest fitness value.

3.4. Evaluation metrics

The loss, accuracy, f1-score, precision and recall are the main evaluation metrics used to assess the performance of the proposed models. Accuracy shown in equation (11) measures how many times a model correctly predicts the class of a piece of data. It is measured by the number of correct predictions divided by the total number of predictions made. Precision shown in equation (12) measures the proportion of correctly identified positive results in a dataset, while recall shown in equation (13) measures the proportion of actual positive cases that were identified correctly. A perfect classifier would have both high precision and high recall. F1 score shown in equation (14) is the harmonic mean of precision and recall. It combines both accuracy and completeness in one measure and is generally considered a better measure than any of the individual metrics. Additionally, the f1-score is the best metric to evaluate the performance of a classifier used for medical data sets. It makes it easier to get an understanding of how well the classifier is performing overall. F1-score has been found to be better suited for imbalanced datasets such as medical data sets due to its ability to assign different weights to different classes. In equations 11–14, TP stands for True Positive, TN stands for True Negative, FP stands for False Positive and FN stands for False Negative.

$$Accuracy = \frac{(TP + TN)}{(TP + FP + FN + TN)} \tag{11}$$

$$Precision = TP \,/\, (TP + FP) \tag{12}$$

$$Recall = \frac{TP}{(TP + FN)} \tag{13}$$

$$F1 - Score = 2 * \frac{(Precision * Recall)}{(Precision + Recall)} \tag{14}$$

The results obtained show that the proposed approach is able to predict the PD based on the above evaluation metrics. The subsequent chapter discusses the details of the experimental results of the study.

4. Results and discussion

In the first part of this section, the results of the feature selection performed using an Extra Tree Classifier is discussed. In the second part, the performance of ANFIS (Adam) is presented, followed by the presentation of the results of ANFIS (PSO). In the last part, the comparative performance analysis of ANFIS (Adam) and ANFIS (PSO) is presented.

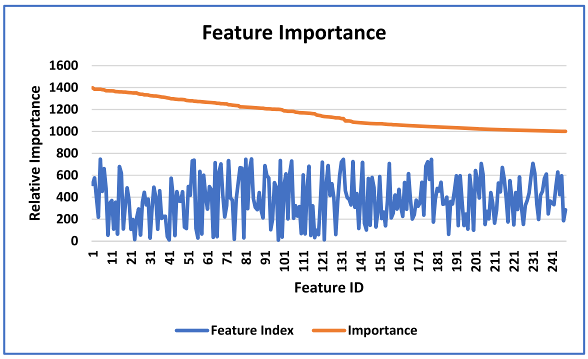


Fig. 3. Feature importance (all features (total = (755–1(Patient id)) = 754).

4.1. Feature selection using ensemble learning

Fig. 3 presents the Feature Importance generated using an Extra Tree Classifier after fitting a model to the PD data having all 754 features. The figure illustrates the relative importance of each feature when predicting the target, represented by a percentage. In this case, the features are ranked in order of importance, with the most influential feature at the top. The Feature Importance obtained offers insight into the predictive power of each feature and helps identify the most important predictors for more accurate results. Furthermore, it can be used to evaluate which features have the highest relevance for model performance and, if necessary, to eliminate those that are not significant. Overall, the Feature Importance is an essential part of any machine learning workflow and can help to improve model accuracy and performance.

Table 2 lists the best 10 features reflecting the results of an Extra Tree Classifier applied to a dataset. These features can be used to build a more reliable model for predicting outcomes. Furthermore, the extra tree classification result can serve as a useful tool for feature selection when creating predictive models.

Fig. 4 shows the feature importance of the five best features that were identified using the Extra Tree Classifier to analyze a PD dataset. These are the five most influential features in predicting the target outcome. The top three features, according to their importance, are 3, 0 and 2. Their importance is represented by the width of the bars in the chart. Feature 3 has been identified as the most important feature, followed by 0 and 2 respectively. The remaining two features, 1 and 4, have also been identified as important but with lesser importance.

Table 3 lists the top 5 features of a dataset that were obtained after Extra Tree Classifier was trained with the 10 best features in Table 1. These features are then used in a training sessions of ANFIS. The features have been ranked in order of importance, with the first listed feature being the most important. These features can be thought of as being the main predictors of the outcome of the model being trained. Understanding which variables are the most important can help optimize the selection of input variables and inform the design of the model. This can lead to more accurate results and better decision making.

Overall, the feature importance chart generated from the Extra Tree Classifier analysis serves to highlight the key features which can be utilized for predicting the target outcome and optimizing the ANFIS model accordingly.

4.2. ANFIS (Adam) based classification of PD

In this section, the performance of ANFIS (Adam) is evaluated. The ANFIS model was trained on a dataset of 529 instances of PD patients, with the aim to accurately predict PD from clinical tests. The evaluation was conducted based on: the number of epochs: 200, 400, 600, 800 and 1000; and, number of fuzzy rules per feature: 2, 3 and 4. The ANFIS model was designed with 5 input features shown in Table 3, representing the symptoms of PD. The output node of the model was configured to provide a binary classification output, with a value of 1 indicating PD and 0 indicating no PD. The fuzzy logic parameters of the ANFIS were optimized using the Adam optimizer and the training data was split into two separate sets; one for training and validation, and a second set as a testing dataset.

Table 4 shows the overall results of the ANFIS (Adam) along with its graphical representation shown in Fig. 5. The performance of ANFIS (Adam) is represented in terms of the evaluation metrics: loss, accuracy, f1-score, precision, and recall.

Fig. 6 shows the best evaluation scores achieved along with the number of epochs it required. However, the number of rules per feature consumed to achieve all the best scores is equal to 4 except for the Loss achieved for testing set, wherein, it required 3 rules per feature.

As the f1-score achieved by the ANFIS (Adam) is higher than the accuracy achieved, it usually indicates that the model is able to accurately identify the minority class more often than not, even though it may not be as accurate with the majority class. This suggests that the model has some degree of bias towards the minority class and could benefit from further tuning or optimization. Additionally, the f1-score, precision, and recall scores of the models failed to improve linearly with the increasing number of epochs or the number of rules per feature. While the accuracy, precision, recall and f1-score varied slightly, the overall performance of the model remained consistent across all epochs, indicating the model was unlikely to suffer from overfitting. However, there was a slight decrease in accuracy, precision, recall and f1-score with increasing epochs beyond 600 and number of rules per feature equals 3, indicating that the model may have been over trained after this point. In conclusion, the results showed that the ANFIS (Adam) model was able to achieve high levels of accuracy, precision, recall and f1-score when used to predict PD with Adam optimizer. This demonstrates the potential of

Table 2
Best 10 features out of all features.

Sl. No.	Feature No.	Feature Name	Cost
1	517	tqwt_medianValue_dec_16	0.001397
2	576	tqwt_stdValue_dec_3	0.001385
3	397	tqwt_entropy_log_dec_3	0.001385
4	218	app_det_TKEO_mean_7_coef	0.001385
5	747	tqwt_kurtosisValue_dec_30	0.001385
6	453	tqwt_TKEO_mean_dec_22	0.001381
7	661	tqwt_maxValue_dec_14	0.001379
8	479	tqwt_TKEO_std_dec_12	0.001371
9	53	IMF_SNR_TKEO	0.001371
10	348	tqwt_energy_dec_25	0.001370

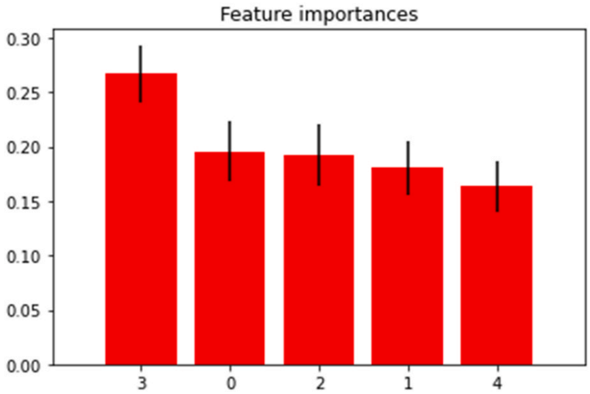


Fig. 4. Feature importance best five features.

Table 3
Feature Importance of Top 5 Features to be trained on ANFIS.

Sl. No.	Feature No.	Feature Name	Cost
1	218	app_det_TKEO_mean_7_coef	0.266957
2	517	tqwt_medianValue_dec_16	0.195777
3	397	tqwt_entropy_log_dec_3	0.192736
4	576	tqwt_stdValue_dec_3	0.180960
5	747	tqwt_kurtosisValue_dec_30	0.163570

Table 4
Evaluation scores of ANFIS (Adam).

Epochs	Evaluation Scores	No. of Rules = 10		No. of Rules = 15		No. of Rules = 20	
		Train	Test	Train	Test	Train	Test
200	loss	40.89	62.57	38.87	56.06	32.12	59.67
	accuracy	82.04	73.57	85.44	38.87	88.28	77.09
	f1-score	88.86	83.61	90.71	83.57	92.51	85.56
	precision	82.75	77.66	86.64	78.95	88.45	80.63
	recall	95.95	90.53	95.19	88.76	**96.96**	91.12
400	loss	36.07	63.49	37.36	56.87	28.95	66.21
	accuracy	86.96	74.45	85.82	75.33	89.60	74.01
	f1-score	91.62	83.62	90.91	84.27	93.17	83.19
	precision	88.08	80.00	87.21	80.21	91.46	80.22
	recall	95.44	87.57	94.94	88.76	94.94	86.39
600	loss	35.63	63.71	36.94	57.85	28.62	68.52
	accuracy	86.77	74.01	85.82	74.89	89.79	74.45
	f1-score	91.48	83.38	90.89	83.94	93.23	83.43
	precision	88.06	79.57	87.38	80.11	92.31	80.66
	recall	95.19	87.57	94.68	88.17	94.18	86.39
800	loss	34.93	66.50	36.34	61.42	28.04	71.08
	accuracy	86.77	74.01	85.63	74.45	**89.98**	75.77
	f1-score	91.46	83.38	90.75	83.62	**93.35**	84.33
	precision	88.24	79.57	87.35	80.00	92.54	81.32
	recall	94.94	87.57	94.43	87.57	94.18	87.57
1000	loss	34.83	68.04	34.10	63.58	**27.88**	71.06
	accuracy	86.77	74.01	86.77	74.89	**89.98**	76.21
	f1-score	91.46	83.38	91.44	83.85	93.33	84.66
	precision	88.24	79.57	88.42	80.43	**92.75**	81.42
	recall	94.94	87.57	94.68	87.57	93.92	88.17

ANFIS (Adam) as an accurate and reliable method for predicting PD, and highlights the potential benefits of leveraging this model in order to further improve its performance.

4.3. ANFIS (PSO) based classification of PD

ANFIS (PSO) is a hybrid machine learning algorithm using both ANFIS and PSO, to accurately classify the PD data. The data set

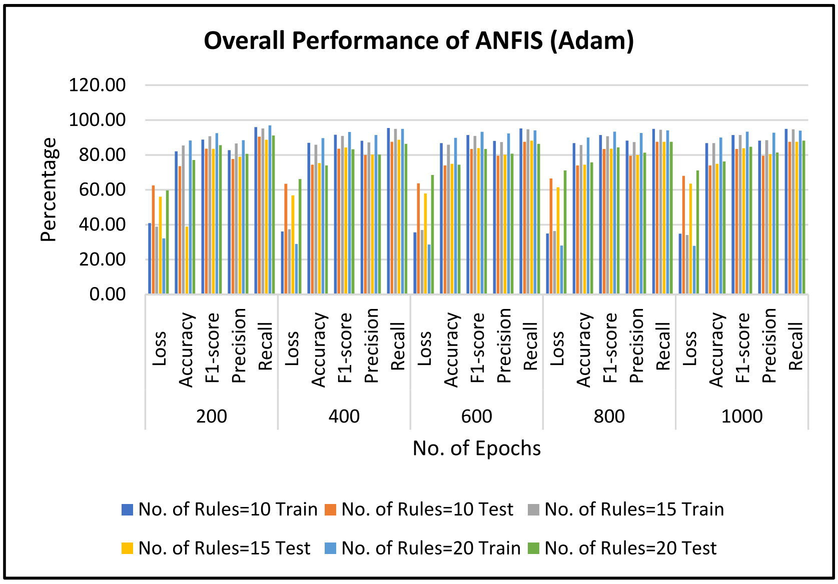


Fig. 5. Performance of ANFIS (Adam).

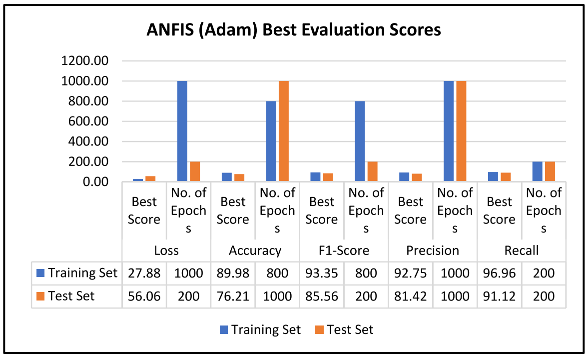


Fig. 6. Best evaluation scores of ANFIS (Adam).

consists of numerical values of the patient's physiological symptoms such as tremors, stiffness, gait, depression, etc. Using ANFIS (PSO), the input dataset is taken into account for a series of membership functions and partitioned into a fuzzy rule base. After the fuzzy rule base is established, PSO is applied to optimize the parameters of each rule in the fuzzy rule base, which in turn leads to the formation of an ANFIS. The output of the ANFIS model is then used to classify the patient's condition (e.g., PD or no PD).

The following are the ANFIS (PSO) model's hyperparameters.

Number of Inputs. This is the number of input features used to build the model.

Number of Membership Functions. This determines how many fuzzy membership functions are used for each input variable.

Learning Rate. This is the rate of change that ANFIS uses to adjust its weights and biases during the learning process.

Number of Fuzzy Rules. This is the number of rules that ANFIS uses to build its fuzzy inference system.

Error Tolerance. This is the maximum error value allowed before the model stops training.

Maximum Number of Epochs (Iterations). This is the maximum number of iterations the model is allowed to go through during the training phase.

Inertia Weight. This is the weight of the inertia that is incorporated into the PSO algorithm in order to keep the particles away from local optima.

Social Learning Weight. This is the weight of the social learning that is incorporated into the PSO algorithm in order to give particles a "nudge" towards attractive solutions.

The following are the factors and their reasons that are considered to asses ANFIS (PSO) targeted to classify PD data.

Fuzzy structure. The Fuzzy structure of ANFIS is an important factor to assess when using it on PD data. It is important to determine the accuracy of the model by assessing the underlying fuzzy layers, rules, and parameters. This is done by calculating the accuracy of the model with various combinations of these factors. For example, different membership functions or change the rules of the fuzzy system are used to assess the accuracy of the model.

Training algorithm. Another important factor to consider when using ANFIS on PD data is the training algorithm chosen. Different algorithms may produce different results and it is important to assess which algorithm produces the most accurate results for your data. Therefore, if backpropagation is used for ANFIS (Adam), the gradient descent is used for ANFIS (PSO).

Network architecture. The network architecture of ANFIS is also an important factor to consider when tuning the model for CSV data. The architecture is chosen in such a way that it accurately fits the data. For example, the number of input nodes, layers, and rules are adjusted through experimentation in order to accurately classify the data.

The rest of the section presents the results obtained from the performance evaluation of ANFIS (PSO). The ANFIS (PSO) model was trained on a dataset of 529 instances of PD patients, with the aim to accurately predict PD from clinical tests. The evaluation was conducted based on: the number of epochs: 200, 400, 600, 800 and 1000; number of fuzzy rules per feature: 2, 3 and 4; and number of particles: 40 and 50. The ANFIS (PSO) model was designed with five input features shown in Table 3, representing the symptoms of PD. The output node of the model was configured to provide a binary classification output, with a value of 1 indicating PD and 0 indicating no PD. This work was carried out for the accurate prediction of PD, and consequently the results of experimentation were recorded based on the parameters and evaluation scores shown in Table 5 to further evaluate the performance of the model.

Tables 6 and 7 present the performance of ANFIS (PSO). These tables show the results of running ANFIS (PSO) on PD data set for epochs: 200, 400, 600, 800, and 1000; number of rules per feature: 2, 3 and 4; and, with the number of PSO particles: 40 and 50. The results indicated that the model achieved improved predictive accuracy consistently as the number of epochs, the number of PSO particles, and number of rules per feature are increased. Additionally, the analysis found that the model reached its peak accuracy performance of 84.5 % at 1000 epochs, 50 PSO particles and, 4 rules per feature, for the training data set. These results suggest that ANFIS (PSO) is an effective method of achieving high accuracy predictions from large data sets such as PD dataset.

Tables 6 and 7 also provide a comparison of the results obtained from a correlation analysis, best learners, and accuracy when running ANFIS (PSO) on a PD data set. These tables provide a detailed statistics of the performance of ANFIS (PSO) for varied number of epochs and particles of PSO.

In all the runs of the ANFIS (PSO), the number of premise functions, number of consequent functions and the number of total variables were calculated based on the number of rules per feature. Tables 8 and 9 show the best scores obtained for the ANFIS parameters and Evaluation Scores, along with the best configuration of ANFIS (PSO) that achieved the aforementioned best results. In Table 8, the best value of 0.7315 for Minimum is achieved when numbers of rules per features were 4 at 1000 epochs when number of PSO particles were set to 50 with 20 premise functions, 1024 consequent functions and 12348 total variables. The best correlation of 0.26 for training set is achieved when only 2 rules per features were used at 600 epochs. The best correlation of 0.15 for testing set is achieved when 4 rules per features were used at 1000 epochs when only 40 PSO particles were used. Similarly, the 47 of best learners and 50 of closed learners were achieved for varied number of epochs as shown in the table.

In Table 9, the least loss for the training set was achieved when 4 rules per feature at 1000 epochs and 50 particles were employed. Whereas, the least loss for the testing set was achieved when 2 or 3 rules per feature at different epochs and 50 particles were employed. The best accuracy of 84.5 % for training set is achieved with the maximum number of rules per feature, epochs and particles of PSO. Whereas, the best accuracy of 74.89 % for testing set is achieved when 2 or 3 rules per feature at different epochs and 50 particles were employed. The ANFIS (PSO) outperformed ANFIS (Adam) in achieving the best precision of 96.47 % and 91.02 % for training and test set, respectively, which were achieved with maximum 800 epochs employing 50 particles. However, only 2 rules per feature were used to achieve the best precision for training set. The best recall of 85.39 % and 80.11 % for training and test set, respectively, were achieved with maximum 1000 epochs employing 50 particles and 4 rules per feature. Finally, the best f1-score of 90.26 % and 84.21 % for training and test set, respectively, were achieved with maximum 1000 epochs employing 50 particles and 2 or 3 rules per feature. Higher precision achieved by ANFIS (PSO) indicates a model's ability to accurately identify positive cases and avoid false positives. Therefore, ANFIS (PSO) outperformed ANFIS (Adam) in terms of precision, though ANFIS (Adam) also produced the highest precision. This also indicate that the ANFIS (PSO) is more reliable than ANFIS (Adam) while classifying the PD data set.

Table 10 shows the sample plots of membership functions for one of the top 5 features, "app_det_TKEO_mean_7_coef", out of the total of 50 membership functions, that were used to train the ANFIS (PSO). The membership functions were generated for each of the feature for each of the number of epochs of ANFIS: 200, 400, 600, 800 and 1000; and number of particles of a PSO: 40 and 50. These membership functions represent the fuzzy rule base that was created using the ANFIS and help to accurately predict how input values influence a given output. The plots also show how the membership functions change over time as the ANFIS cycles through each combination of epochs and PSO particles. The data demonstrates the effectiveness of the ANFIS in creating an effective predictive model with relatively low computational costs.

Table 10 also show the graphs which appear as a bell-shaped curve that extends from the minimum to maximum values of the input variable, with a peak value in the middle. The shape of the graph does not appear too linear or flat, indicating that the PSO has

Table 5
ANFIS (PSO) parameters and evaluation scores.

Parameters	Evaluation Scores		
No. of Epochs	J Minimum	accuracy Training Data	precision Test Data
No. of Particles in PSO	Correlation Training Data	accuracy Test Data	recall Training Data
No. of Premise Functions	Correlation Test Data	loss Training Data	recall Test Data
No. of Consequent Functions	No. of Best Learners	loss Test Data	f1-score Training Data
No. of Variables	No. of Close Learners	precision Training Data	f1-score Test Data

Table 6
Performance of ANFIS (PSO).

ANFIS Layout	A	B	C	D	E	F	G	H	I	J
[2,2,2,2,2]	200	40	10	32	414	0.9296	0.37	0.25	14	40
	400					0.9098	0.4	0.23	13	40
	600					0.897	0.37	0.24	38	40
	800					0.8873	0.38	0.24	15	40
	1000					0.8794	0.38	0.24	16	40
	200	50				0.9364	0.38	0.22	29	46
	400					0.9164	0.41	0.25	34	**50**
	600					0.9056	**0.26**	0.26	5	**50**
	800					0.8987	0.42	0.26	43	**50**
	1000					0.8925	0.4	0.26	14	**50**
[3,3,3,3,3]	200	40	15	243	2961	0.8951	0.42	0.19	26	1
	400					0.8511	0.45	0.19	0	12
	600					0.8283	0.45	0.19	15	28
	800					0.8114	0.46	0.21	29	36
	1000					0.7982	0.47	0.21	25	36
	200	50				0.905	0.37	0.28	23	1
	400					0.8617	0.45	0.23	29	12
	600					0.8349	0.47	0.21	**47**	36
	800					0.8153	0.47	0.22	28	49
	1000					0.7989	0.48	0.19	36	46
[4,4,4,4,4]	200	40	20	1024	12348	0.8909	0.44	0.25	7	1
	400					0.8448	0.46	0.19	20	1
	600					0.8197	0.49	0.19	11	1
	800					0.8028	0.48	0.19	12	4
	1000					0.7905	0.52	**0.15**	2	2
	200	50				0.8927	0.46	0.24	3	1
	400					0.8334	0.47	0.27	19	1
	600					0.7853	0.53	0.29	37	1
	800					0.7564	0.54	0.28	26	3
	1000					**0.7315**	0.55	0.29	6	1
Average						0.8533	0.44	0.23	20.73	23.96

Column Headers:A. No. of Epochs, **B**. No. of Particles in PSO, **C**. No. of Premise Functions, **D**. No. of Consequent Functions, **E**. No. of Variables, **F**. J Minimum, **G**. Correlation Training Data, **H**. Correlation Test Data, **I**. No. of Best Learners, **J**. No. of Close Learners.

optimized on all the parameters of ANFIS accurately. The vertical lines of a graph drawn for a membership function in ANFIS shows that it has crisp boundaries and a smooth overall slope that is consistent with the expected behavior from the analysis of the data. It indicated that there are no such areas of divergence or accuracy issues that deviate significantly from the expected value. However, the Gaussian function is used as a membership function in this ANFIS, with parameters m (mean) and s (standard deviation), which is then used to fine tune the parameters m and s to generate the best curve that corresponds to the input data. After the training process is completed, the parameters are updated according to the curve generated and this will provide a more accurate representation of the data. Further, it is important to make sure that the shape and range of values of the graph accurately reflect the terms used to define the structure of the fuzzy rule. In order to assess the graph of a membership function in ANFIS (PSO) developed for classifying PD data, we analyzed the accuracy and relevance of the graph. The first step in assessing the results of an ANFIS (PSO) is to evaluate the accuracy of the model. This was done by comparing the predicted labels from the model to the true labels in the dataset, and then calculating the accuracy (or other performance metrics) of those predictions. The second step in assessing the results of an ANFIS (PSO) is to evaluate the interpretability of the membership functions. This was done by looking at the overall shape of the functions and evaluating how well each function is able to capture the underlying patterns in the data. This was done by examining things like the width of the shape, the location of the apexes, and the steepness of the sides.

Table 11 shows the closed form solution obtained for ANFIS (PSO), which is an optimal set of parameters (weights and biases) that was further used to make predictions from the PD data. The solution obtained shows that the process of tuning ANFIS (PSO) has been completed, and the resulting set of parameters can be used without further tuning or optimization.

Table 12 presents the average evaluation scores of the ANFIS (PSO) applied to the PD prediction task along with its graphical representation in Fig. 7. The table compares the performance of the ANFIS (PSO) model at different epochs: 200, 400, 600, 800 and 1000; and for different number of rules per feature: 10, 15 and 20; averaged on the number of particles used in the PSO. The results reveal that the model consistently achieved good accuracy with increasing epochs, with the highest average accuracy of 89.98 % obtained at epoch 1000 when 4 rules per feature were used. Further, the model achieved a highest average precision score of 92.75 % and highest average recall score of 96.96 at epochs 1000 and 200 respectively. This indicates that the model is capable of correctly classifying individuals with PD with a high degree of reliability. Overall, this study demonstrates that combining ANFIS and PSO can produce accurate classification models for predicting PD.

Table 7
Performance of ANFIS (PSO) (contd.).

ANFIS Layout	A	B	C	D	E	F	G	H	I	J	K	L
[2,2,2,2,2]	200	40	20.42	26	79.58	74.01	95.47	89.22	80.81	78.42	87.53	83.47
	400		19.66	26.43	80.34	73.57	95.47	89.22	81.51	78.01	87.94	83.24
	600		20.42	25.99	79.58	74.01	95.21	89.82	80.94	78.12	87.5	83.57
	800		20.23	25.99	79.77	74.01	94.96	89.82	81.25	78.13	87.57	83.57
	1000		20.23	25.99	79.77	74.01	94.47	89.82	80.98	78.13	87.63	83.57
	200	50	20.23	26.87	79.77	73.13	94.96	88.62	81.25	77.89	87.57	82.91
	400		19.66	25.55	80.34	74.45	94.96	90.42	81.78	78.24	87.88	83.89
	600		19.85	**25.11**	80.15	**74.89**	95.71	**91.02**	81.2	78.35	87.86	**84.21**
	800		19.28	**25.11**	80.72	**74.89**	95.72	**91.02**	81.72	78.35	88.17	**84.21**
	1000		19.85	**25.55**	80.15	74.44	93.95	89.82	82.16	78.53	87.66	83.8
[3,3,3,3,3]	200	40	19.09	27.31	80.91	72.69	96.22	89.82	81.62	76.92	88.32	82.87
	400		18.53	27.75	81.47	72.25	94.96	88.62	82.86	77.08	88.5	82.45
	600		18.52	28.19	81.47	71.81	94.96	88.62	82.86	76.69	88.5	82.22
	800		18.34	27.31	81.66	72.69	94.71	88.02	83.19	77.78	88.57	82.58
	1000		17.96	27.75	82.04	72.25	95.21	86.82	83.26	77.96	88.84	82.15
	200	50	20.6	**25.11**	79.4	**74.89**	95.21	89.82	80.77	78.95	87.4	84.03
	400		18.53	26.87	81.47	73.13	95.21	88.02	82.71	78.19	88.52	82.82
	600		17.96	27.31	82.04	72.69	95.21	88.02	83.26	77.78	88.84	82.58
	800		17.77	27.31	82.23	72.69	95.21	87.43	83.44	78.07	88.94	82.49
	1000		17.58	28.63	82.42	71.37	94.71	85.63	83.93	77.72	88.99	81.48
[4,4,4,4,4]	200	40	18.71	26.87	81.29	73.13	94.46	86.83	82.96	78.8	88.34	82.62
	400		18.15	28.19	81.85	71.81	95.72	86.83	82.79	77.54	88.79	81.92
	600		17.39	28.63	82.61	71.37	95.47	85.63	83.66	77.72	89.17	81.48
	800		17.39	28.63	82.61	71.37	95.97	85.63	83.37	77.72	89.23	81.48
	1000		16.45	30.4	83.55	69.6	95.47	83.83	84.6	76.92	89.7	80.23
	200	50	18.15	26.87	81.85	73.12	96.22	87.43	82.51	78.49	88.84	82.72
	400		17.77	26.43	82.23	73.57	95.21	86.23	83.44	79.56	88.94	82.76
	600		16.07	25.99	83.93	74.01	95.97	86.23	84.67	80	89.96	83
	800		15.69	26.43	84.31	73.57	**96.47**	85.63	84.73	79.89	90.22	82.66
	1000		**15.5**	25.55	**84.5**	74.45	95.72	86.83	**85.39**	**80.11**	**90.26**	83.33
Average			18.53	26.87	81.47	73.12	95.30	88.02	82.65	78.20	88.54	82.81

Column Headers:A. No. of Epochs, **B**. No. of Particles in PSO, **C.** Loss Training Data, **D.** LOss Test Data, **E.** accuracy Training Data, **F.** accuracy Test Data, **G.** precision Training Data, **H.** precision Test Data, **I.** recall Training Data, **J.** recall Test Data, **K.** f1-score Training Data, **L.** f1-score Test Data.

Table 8
ANFIS (PSO): Analysis of best scores (J minimum, correlation and learners).

ANFIS Configuration	J Minimum = 0.7315	Correlation Training Data = 0.26	Correlation Test Data = 0.15	No. of Best Learners = 47	No. of Close Learners = 50
No. of Rules per Feature	4	2	4	3	2
No. of Epochs	1000	600	1000	600	200, 400, 600, 800, 1000
No. of Particles in PSO	50	50	40	50	50
No. of Premise Functions	20	10	20	15	10
No. of Consequent Functions	1024	32	1024	243	32
No. of Variables	12348	414	12348	2961	414

Table 9
ANFIS (PSO): Analysis of best evaluation scores.

ANFIS Configuration	A = 15.5	B = 25.11	C = 84.5	D = 74.89	E = 96.47	F = 91.02	G = 85.39	H = 80.11	I = 90.26	J = 84.21
No. of Rules per Feature	4	2, (3)	4	2, (3)	4	2	4	4	4	2
No. of Epochs	1000	600, 800, (200)	1000	600, 800, (200)	800	600, (800)	1000	1000	1000	600, (800)
No. of Particles in PSO	50	50	50	50	50	50	50	50	50	50
No. of Premise Functions	20	10	20	10, (15)	20	10	20	20	20	10
No. of Consequent Functions	1024	32	1024	32, (243)	1024	32	1024	1024	1024	32
No. of Variables	12348	414	12348	414, (2961)	12348	414	12348	12348	12348	414

Column Headers:A. Loss Training Data, B. Loss Test Data, C. accuracy Training Data, D. accuracy Test Data, E. precision Training Data, F. precision Test Data, G. recall Training Data, H. recall Test Data, I. f1-score Training Data, J. f1-score Test Data.

Table 10
Sample plots of membership functions (best Feature-1 at epochs = 200) of ANFIS (PSO).

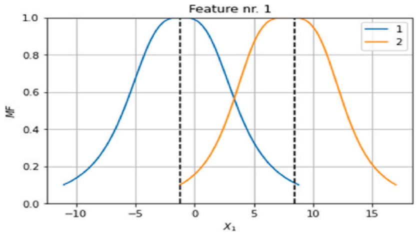
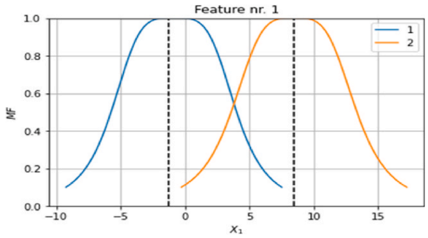
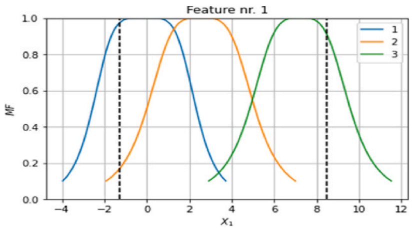
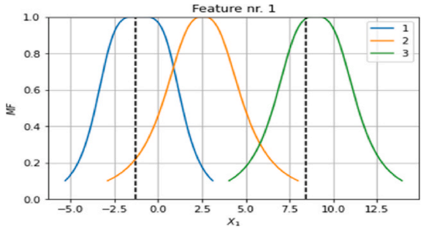
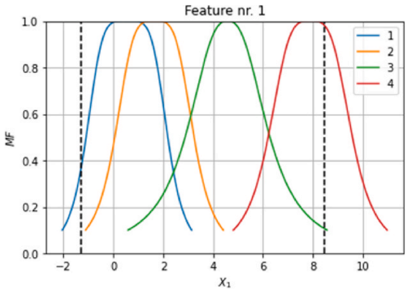
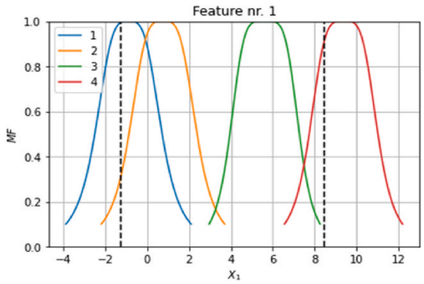
<table>
<tr><th>No. of Rules per Feature</th><th>No. of Particles used in PSO = 40</th><th>No. of Particles used in PSO = 50</th></tr>
<tr><td>2</td><td>
</td><td>
</td></tr>
<tr><td>3</td><td>
</td><td>
</td></tr>
<tr><td>4</td><td>
</td><td>
</td></tr>
</table>

Table 11
Closed-form solution of ANFIS (PSO).

Theta	[[0.75047259] [-0.13464271] [-0.08298691] [0.02156321] [0.00727497] [-0.03747104]]
Correlation Training Data	0.3814
Correlation Test Data	0.2407

4.4. *Performance analysis of ANFIS (Adam) versus ANFIS (PSO)*

Figs. 8–12 show the performance analysis of ANFIS (Adam) versus ANFIS (PSO), averaged over number of epochs and number of rules per feature. The results obtained for ANFIS (PSO) are initially averaged for number of particles in PSO, to get the average scores based on the number of epochs and number of rules per feature. The experimentation was performed for each epoch, the number of rules per feature, and number of particles of PSO. The evaluation scores of both ANFIS (Adam) and ANFIS (PSO) are then averaged and tabulated in the figures. The results indicate that ANFIS (Adam) has better accuracy compared to the ANFIS (PSO) approach. This can be attributed to the optimization capabilities of Adam optimizer, which helps to improve the accuracy of the model. As more epochs are used, the accuracy of both ANFIS (Adam) and ANFIS (PSO) increases, although ANFIS (Adam) generally shows a better performance than ANFIS (PSO). Consequently, this table indicates that both the models can provide improved results over the standard ANFIS approach in terms of evaluation scores achieved.

Loss is an evaluation metric that measures how well a model is able to fit the data it is being trained on. This is important when classifying medical datasets, as it indicates how well the model is able to predict outcomes and detect patterns in the data. High loss indicates that the model is not able to accurately predict outcomes and can lead to incorrect classifications, so focusing on low loss values is important when classifying medical data sets. Fig. 8 shows that the ANFIS (PSO) is better than the ANFIS (Adam) in terms the evaluation metric loss, irrespective of the number of epochs and number of rules per features used in implementing the ANFIS. Due to the inherent capabilities of the PSO optimizer, the ANFIS (PSO) incurs the loss better than the loss (from incurred by ANFIS (Adam). The ANFIS (Adam) incurred the loss from 22.46 to 23.02 and from 22.26 to 22.92, when the evaluation scores were averaged for number of epochs and number of rules per feature, respectively.

Table 12
Evaluation scores of ANFIS (PSO) (averaged on No. of particles (40 and 50)).

Epochs	Evaluation Scores	No. of Rules = 10		No. of Rules = 15		No. of Rules = 20	
		Train	Test	Train	Test	Train	Test
200	loss	40.89	62.57	38.87	56.06	32.12	59.67
	accuracy	82.04	73.57	85.44	38.87	88.28	77.09
	f1-score	88.86	83.61	90.71	83.57	92.51	85.56
	precision	82.75	77.66	86.64	78.95	88.45	80.63
	recall	95.95	90.53	95.19	88.76	**96.96**	91.12
400	loss	36.07	63.49	37.36	56.87	28.95	66.21
	accuracy	86.96	74.45	85.82	75.33	89.60	74.01
	f1-score	91.62	83.62	90.91	84.27	93.17	83.19
	precision	88.08	80.00	87.21	80.21	91.46	80.22
	recall	95.44	87.57	94.94	88.76	94.94	86.39
600	loss	35.63	63.71	36.94	57.85	28.62	68.52
	accuracy	86.77	74.01	85.82	74.89	89.79	74.45
	f1-score	91.48	83.38	90.89	83.94	93.23	83.43
	precision	88.06	79.57	87.38	80.11	92.31	80.66
	recall	95.19	87.57	94.68	88.17	94.18	86.39
800	loss	34.93	66.50	36.34	61.42	28.04	71.08
	accuracy	86.77	74.01	85.63	74.45	**89.98**	75.77
	f1-score	91.46	83.38	90.75	83.62	**93.35**	84.33
	precision	88.24	79.57	87.35	80.00	92.54	81.32
	recall	94.94	87.57	94.43	87.57	94.18	87.57
1000	loss	34.83	68.04	34.10	63.58	**27.88**	71.06
	accuracy	86.77	74.01	86.77	74.89	**89.98**	76.21
	f1-score	91.46	83.38	91.44	83.85	93.33	84.66
	precision	88.24	79.57	88.42	80.43	**92.75**	81.42
	recall	94.94	87.57	94.68	87.57	93.92	88.17

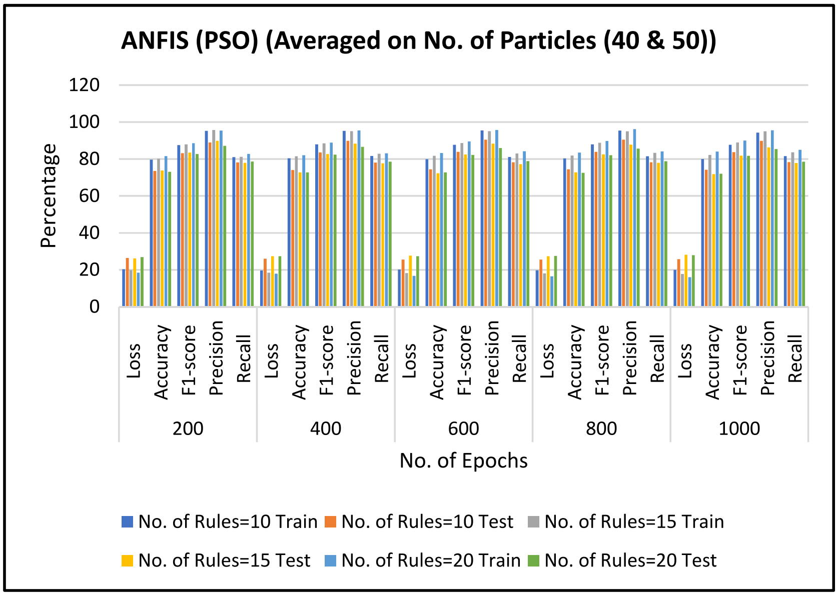


Fig. 7. ANFIS (PSO) (averaged on No. of particles (40 and 50)).

Accuracy is an evaluation metric that measures how well a model performs in predicting elements in a dataset. It is especially important when classifying PD because of the nature of this type of data, as there are often severe consequences for incorrect predictions. High accuracy indicates that a model is able to correctly identify cases and avoid false positives or false negatives. Fig. 9 shows that the performance of ANFIS (Adam) was better than ANFIS (PSO) in terms the evaluation metric accuracy. However, the average accuracies increased linearly as the number of epochs and number of rules were increased in the both the models. The best average accuracy of 82.52 % for ANFIS (Adam) was achieved when the number of rules per feature were 20, and the best average accuracy of 77.76 % for ANFIS (PSO) was also achieved when the number of rules per feature were 20. This shows that the number of rules used per feature has higher impact on the performance of the ANFIS models irrespective of the number of epochs used.

F1-score is an evaluation metric that combines precision and recall into a single measure. In other words, it provides an overall

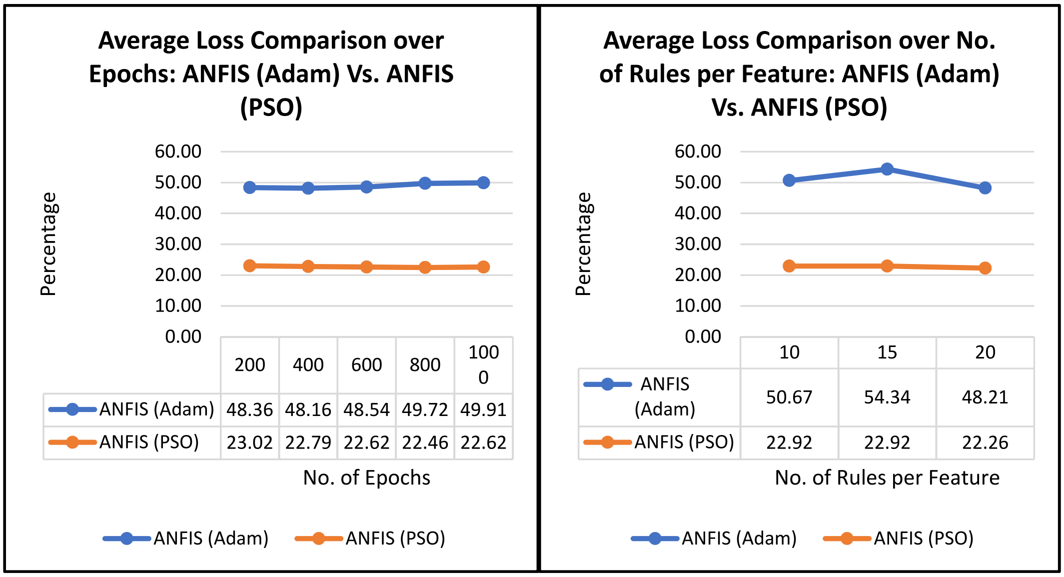


Fig. 8. Comparison of loss (ANFIS (Adam) Vs. ANFIS (PSO)).

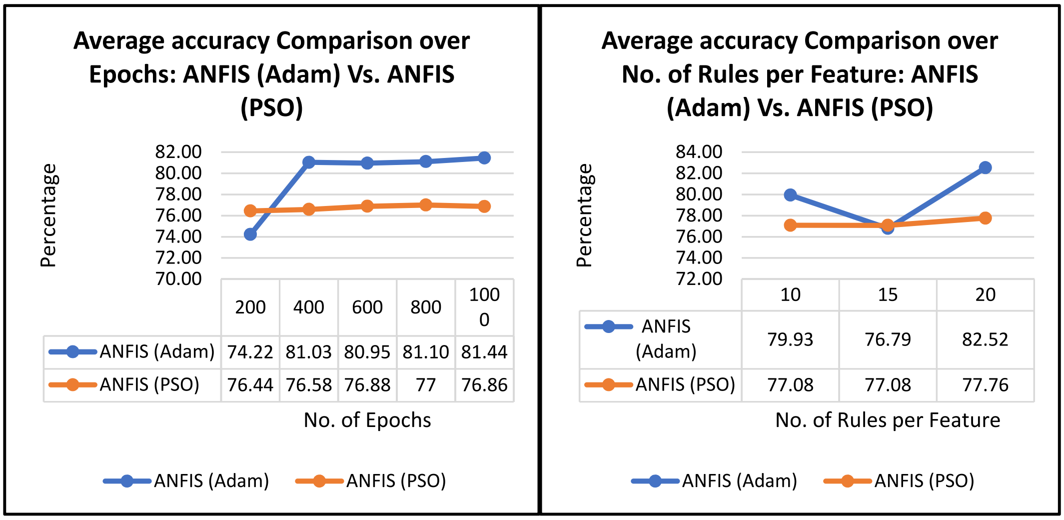


Fig. 9. Comparison of accuracy (ANFIS (Adam) Vs. ANFIS (PSO)).

score of how well the model performed in predicting elements in the dataset. This is especially important when classifying PD as it takes into account both precision and recall, allowing for a more comprehensive analysis of the model's performance. High F1 score of ANFIS (Adam) in Fig. 10 indicates that the model is able to accurately identify positive cases while avoiding false positives, making it a useful metric for evaluating the performance. Fig. 10 shows that the performance of ANFIS (Adam) was better than ANFIS (PSO) in terms the evaluation metric f1-score. However, the average f1-score increased linearly (from 87.47 % to 88.02 %) with a smaller value as the number of epochs and number of rules were increased in ANFIS (Adam). Whereas, the f1-score increased linearly (from 85.55 % to 85.81 %) with a smaller value until the number of epochs were increased up to 800, and when it reached 1000 epochs it reduced to 85.64 %, indicating that the PSO was stuck at the local optima after 800 epochs. However, both the models showed the linear increase of f1-score: from 87.23 % to 88.68 % and 85.69 %–85.78 %, for ANFIS (Adam) and ANFIS (PSO), respectively.

Precision is an evaluation metric that measures how accurate a classification model is in correctly predicting elements in a dataset. It is especially important when classifying medical datasets such as PD because some diseases or treatments may have a high rate of false-positives, meaning that the model could end up incorrectly labeling patients. Higher precision achieved by ANFIS (PSO) of 91.02 % at 1000 epochs, and 90.89 % at 20 rules per feature, as shown in Fig. 11 indicates a model's ability to accurately identify positive cases and avoid false positives. Therefore, ANFIS (PSO) outperformed ANFIS (Adam) in terms of precision, though ANFIS (Adam) also produced the highest precision of 85.14 % and 86.18 % at 1000 epochs and 20 rules per feature, respectively. This also indicate that the

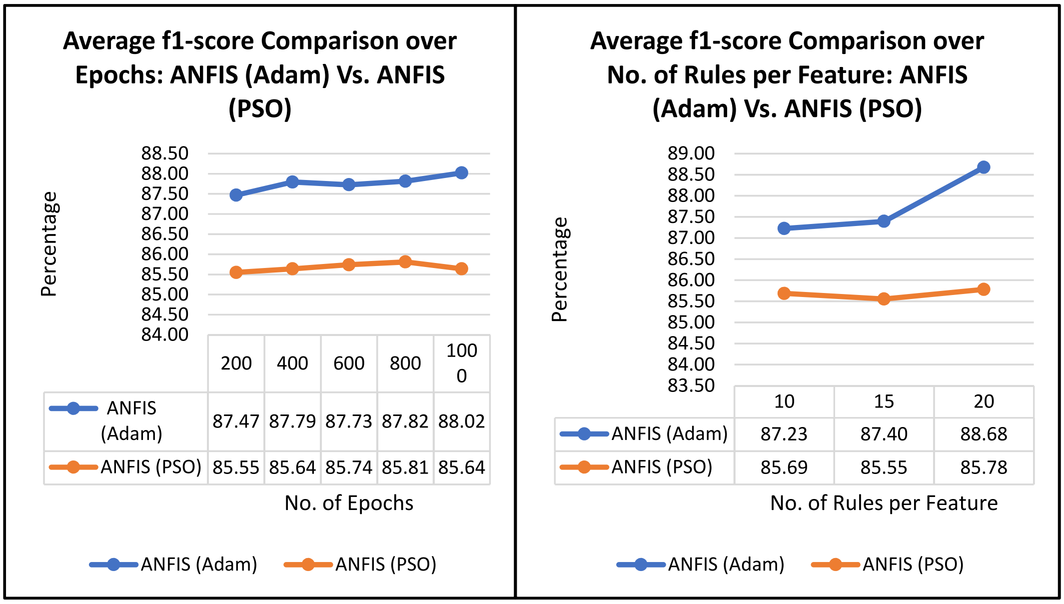


Fig. 10. Comparison of f1-score (ANFIS (Adam) Vs. ANFIS (PSO)).

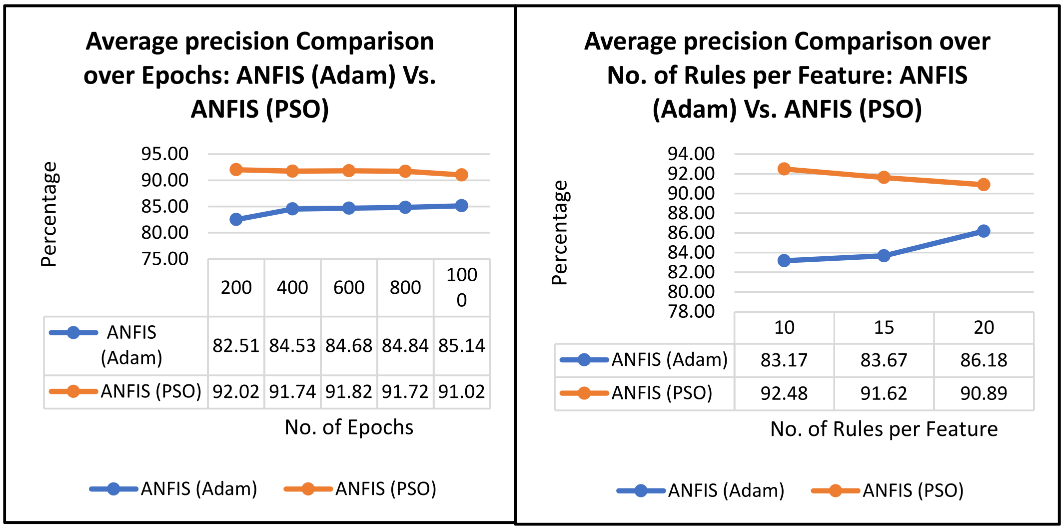


Fig. 11. Comparison of precision (ANFIS (Adam) Vs. ANFIS (PSO)).

ANFIS (PSO) is more reliable than ANFIS (Adam) while classifying the PD data set.

Recall is an evaluation metric that measures the model's ability to identify the positive cases from a dataset. This is especially important when dealing with medical datasets such as PD because some diseases or treatments may have a low prevalence in the population and therefore can be difficult for models to accurately identify. Higher recall indicates that a model has a greater ability to correctly identify positive cases, which is important when it comes to PD. Focusing on high recall is beneficial when classifying PD as it ensures that the model is able to accurately detect positive cases that would otherwise be missed. Fig. 12 shows that the performance of ANFIS (Adam) was better than ANFIS (PSO) in terms the evaluation metric recall. However, the average recall was fluctuating and decreased (from 93.09 % to 91.14 %) between the number of epochs, with a smaller value as the number of epochs and number of rules were increased in ANFIS (Adam). Whereas, the recall increased linearly (from 79.95 % to 80.85 %) with a smaller value as the number of epochs were increased from 200 to 800. However, the behavior of the ANFIS (Adam) was same (fluctuating from 91.73 % to 91.38 %) even when the number of rules per feature were increased. And, the ANFS (PSO) showed linear increase in recall (from 79.79 % to 81.24 %) as the number of rules per feature were increased.

Finally it is evident from the findings in this research that if the tasks of the classification are more focused on accuracy and avoiding false positives, then ANFIS (Adam) is the better choice since it has better results in terms of accuracy, f1-score, and recall.

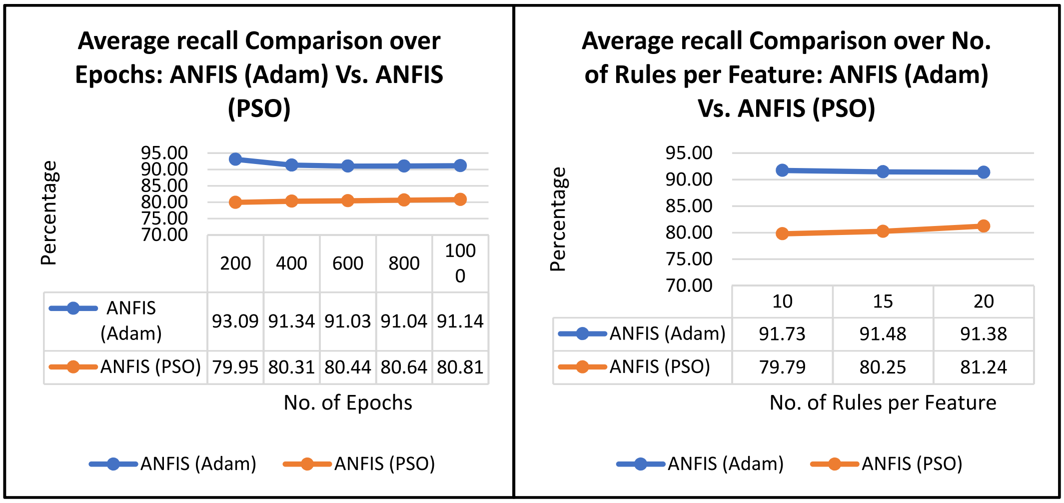


Fig. 12. Comparison of recall (ANFIS (Adam) Vs. ANFIS (PSO)).

However, if tasks require more precision and fewer false negatives, then ANFIS (PSO) is the better choice since it has better results in terms of loss and precision. Overall, the results presented in this study demonstrate the utility of the two ANFIS models tuned with the two different optimizers for the accurate prediction of PD. These models, which were designed to replicate the skills of clinical experts and minimize the human bias in clinical decision support systems, holds a potential for its further development as a diagnostic tool for PD.

Although the proposed ANFIS models yields promising results, certain limitations should be acknowledged. First, the training and testing performance evaluated in this study were based on a single PD dataset, which means that the generalizability of the model is unknown. Therefore, further studies should be conducted to assess the model's performance on datasets from different patient populations and with more data points. Second, the features used in this study may not capture all relevant factors affecting the condition. For example, psychological factors, such as anxiety and depression, are known to be associated with PD, but were not included in the current dataset. Consequently, the performance of the model could be further improved by incorporating additional features into the dataset.

5. Conclusion and future enhancement

In the current study, the problem of classifying the PD data is addressed. Ensemble learning based feature selection approach is employed to select the top 5 subset of features from the PD data set, with the objective of developing the two efficient ANFIS models to classify PD. The study employed hybrid ANFIS models for classifying, and coupled with two optimizers: Adam and PSO, for tuning the two important ANFIS parameters: number of epochs and number of rules per feature. The PD data set is split into training and testing set in the ratio of 0.7:0.3, to train and test the two ANFIS models for number of epochs (200, 400, 600, 800 and 1000) and number of rules per feature (2, 3 and 4). The performance of these ANFIS models are evaluated based on the classification evaluation metrics: loss, accuracy, f1-score, precision and recall. The results showed that ANFIS (PSO) had better results in terms of loss and precision, while ANFIS (Adam) had better results in terms of accuracy, f1-score, and recall. Our experiments showed that the ANFIS tuned with Adam Optimizer was able to achieve higher classification accuracy for PD data than when tuned with PSO. This suggests that parameter selection plays a critical role in how well a model can classify data. Our findings can be used to better understand the progression of PD and develop more accurate diagnostic and therapeutic measures.

However, the dataset employed in this study did not categorize patients based on different PD stages, like the early motor or prodromal phases. While the incorporated methodology aimed at efficient PD classification, future studies could explore integrating stage-specific features to enhance the model's diagnostic capabilities across various disease stages, including the early motor or prodromal phases. Additionally, the primary focus on utilizing ANFIS models for PD classification highlights their potential contribution to understanding PD progression. Leveraging these models to analyze learned patterns and feature importance could reveal crucial factors or markers associated with PD progression, aiding in delineating disease advancement patterns or identifying predictive markers indicative of disease progression. Furthermore, this study did not explicitly address the differentiation between PD and other forms of Parkinsonism. Future research directions could involve refining ANFIS models or employing methodologies aimed specifically at distinguishing PD from various Parkinsonian subtypes. This endeavor could significantly enhance the models' accuracy in distinguishing accurately between these conditions. These potential extensions offer avenues to explore alternative optimization techniques, diverse ANN architectures, ensemble and multi-objective optimization, and the potential use of evolutionary techniques like genetic algorithms. Investigating the generalizability of these models across different datasets, such as EEG and MRI data, presents an opportunity to further enhance the accuracy of PD prediction and potentially advance the diagnosis and treatment of PD.

Data availability statement

Not applicable.

Funding

HQ thanks the USA NSF award 1663105, 1761839, and 2200138, a catalyst award from the USA National Academy of Medicine, AI Tennessee Initiative, and the support at the University of Tennessee at Chattanooga.

CRediT authorship contribution statement

Akram Pasha: Writing – original draft, Conceptualization. **Syed Thouheed Ahmed:** Writing – original draft, Conceptualization. **Ranjith Kumar Painam:** Visualization, Methodology. **Sandeep Kumar Mathivanan:** Writing – review & editing, Visualization, Validation, Supervision, Methodology, Investigation. **Karthikeyan P:** Supervision, Project administration, Data curation. **Saurav Mallik:** Writing – review & editing, Visualization, Supervision, Methodology. **Hong Qin:** Writing – review & editing, Supervision, Project administration, Funding acquisition, Formal analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] R.L. Nussbaum, C.E. Ellis, Alzheimer's disease and Parkinson's disease, N. Engl. J. Med. 348 (14) (2003) 1356–1364.
[2] J.M. Beitz, Parkinson's disease: a review, Frontiers in Bioscience-Scholar 6 (1) (2014) 65–74.
[3] B.R. Bloem, M.S. Okun, C. Klein, Parkinson's disease, Lancet 397 (10291) (2021) 2284–2303.
[4] S. Sharma, C.S. Moon, A. Khogali, A. Haidous, A. Chabenne, C. Ojo, M. Ebadi, Biomarkers in Parkinson's disease (recent update), Neurochem. Int. 63 (3) (2013) 201–229.
[5] J. Thome, A.C. Ehlis, A.J. Fallgatter, K. Krauel, K.W. Lange, P. Riederer, M. Gerlach, Biomarkers for attention-deficit/hyperactivity disorder (ADHD). A consensus report of the WFSBP task force on biological markers and the World Federation of ADHD, World J. Biol. Psychiatr. 13 (5) (2012) 379–400.
[6] A. Jeromin, R. Bowser, Biomarkers in neurodegenerative diseases, Neurodegener. Dis.: Pathology, Mechanisms, and Potential Therapeutic Targets (2017) 491–528.
[7] L.V. Kalia, A.E. Lang, Parkinson's disease, Lancet 386 (9996) (2015) 896–912.
[8] Y. Kumar, A. Koul, R. Singla, M.F. Ijaz, Artificial intelligence in disease diagnosis: a systematic literature review, synthesizing framework and future research agenda, J. Ambient Intell. Hum. Comput. (2022) 1–28.
[9] J. Shen, C.J. Zhang, B. Jiang, J. Chen, J. Song, Z. Liu, W.K. Ming, Artificial intelligence versus clinicians in disease diagnosis: systematic review, JMIR medical informatics 7 (3) (2019) e10010.
[10] C. Ao, S. Jin, H. Ding, Q. Zou, L. Yu, Application and development of artificial intelligence and intelligent disease diagnosis, Curr. Pharmaceut. Des. 26 (26) (2020) 3069–3075.
[11] K.J. Kubota, J.A. Chen, M.A. Little, Machine learning for large-scale wearable sensor data in Parkinson's disease: concepts, promises, pitfalls, and futures, Movement disorders 31 (9) (2016) 1314–1326.
[12] M. Fatima, M. Pasha, Survey of machine learning algorithms for disease diagnostic, J. Intell. Learn Syst. Appl. 9 (1) (2017) 1.
[13] Y.K. Semero, J. Zhang, D. Zheng, PV power forecasting using an integrated GA-PSO-ANFIS approach and Gaussian process regression based feature selection strategy, CSEE Journal of Power and Energy Systems 4 (2) (June 2018) 210–218, https://doi.org/10.17775/CSEEJPES.2016.01920.
[14] Wei Chen, et al., Performance evaluation of GIS-based new ensemble data mining techniques of adaptive neuro-fuzzy inference system (ANFIS) with genetic algorithm (GA), differential evolution (DE), and particle swarm optimization (PSO) for landslide spatial modelling, Elsevier BV, Catena 157 (Oct. 2017) 310–324, https://doi.org/10.1016/j.catena.2017.05.034. Crossref.
[15] Seghier Ben, Mohamed El Amine, et al., Novel hybridized adaptive neuro-fuzzy inference system models based particle swarm optimization and genetic algorithms for accurate prediction of stress intensity factor, Wiley, Fatig. Fract. Eng. Mater. Struct. 43 (11) (Aug. 2020) 2653–2667, https://doi.org/10.1111/ffe.13325. Crossref.
[16] A. Pasha, P. Latha, Well-calibrated probabilistic machine learning classifiers for multivariate healthcare Data, Int. J. Adv. Res. Comput. Sci. 12 (2) (2021).
[17] A. Pasha, P.H. Latha, Bio-inspired dimensionality reduction for Parkinson's disease (PD) classification, Health Inf. Sci. Syst. 8 (2020) 1–22.
[18] I.B. Mores, M. Fauzan, Y.Y. Nazaruddin, P. Ishaya Siregar, Using Particle Swarm and Brain Storm Optimization for Predicting Bus Arrival Time," 2019 6th International Conference on Electric Vehicular Technology (ICEVT), Bali, Indonesia, 2019, pp. 58–62, https://doi.org/10.1109/ICEVT48285.2019.8993978.
[19] Fateme Nazari Robati, Saeed Iranmanesh, Inflation rate modeling: adaptive neuro-fuzzy inference system approach and particle swarm optimization algorithm (ANFIS-PSO), Elsevier BV, MethodsX 7 (2020) 101062, https://doi.org/10.1016/j.mex.2020.101062. Crossref.
[20] Ahmadreza Ejraei Bakyani, et al., Prediction of CO2–oil molecular diffusion using adaptive neuro-fuzzy inference system and particle swarm optimization technique, Elsevier BV, Fuel 181 (Oct. 2016) 178–187, https://doi.org/10.1016/j.fuel.2016.04.097. Crossref.
[21] G.J. Osório, et al., Short-term wind power forecasting using adaptive neuro-fuzzy inference system combined with evolutionary particle swarm optimization, wavelet transform and mutual information, Elsevier BV, Renew. Energy 75 (Mar. 2015) 301–307, https://doi.org/10.1016/j.renene.2014.09.058. Crossref.
[22] Mohammad Malekan, Khosravi Ali, Investigation of convective heat transfer of ferrofluid using CFD simulation and adaptive neuro-fuzzy inference system optimized with particle swarm optimization algorithm, Elsevier BV, Powder Technol. 333 (June 2018) 364–376, https://doi.org/10.1016/j.powtec.2018.04.044. Crossref.
[23] G.S. Kumar, B.K. Kumar, M.K. Mishra, Mitigation of voltage sags with phase jumps by UPQC with PSO-based ANFIS, IEEE Trans. Power Deliv. 26 (4) (Oct. 2011) 2761–2773, https://doi.org/10.1109/TPWRD.2011.2165301.
[24] Samuel Atuahene, et al., Short-term electric power forecasting using dual-stage hierarchical wavelet- particle swarm optimization- adaptive neuro-fuzzy inference system PSO-ANFIS approach based on climate change, MDPI AG, Energies 11 (10) (Oct. 2018) 2822, https://doi.org/10.3390/en11102822. Crossref.
[25] A.A. Sarhan, M.A. Kamel, A.T. Hafez, S. Givigi, Comparative Investigation of MPPT Controller for Grid Connected Photovoltaic System," 2020 IEEE International Systems Conference (SysCon), Montreal, QC, Canada, 2020, pp. 1–7, https://doi.org/10.1109/SysCon47679.2020.9460985.
[26] Nor Azliana Abdullah, et al., Forecasting solar power using hybrid firefly and particle swarm optimization (HFPSO) for optimizing the parameters in a wavelet transform-adaptive Neuro fuzzy inference system (WT-ANFIS), MDPI AG, Appl. Sci. 9 (16) (Aug. 2019) 3214, https://doi.org/10.3390/app9163214. Crossref.

[27] Oludamilare Bode Adewuyi, et al., Power system voltage stability margin estimation using adaptive neuro-fuzzy inference system enhanced with particle swarm optimization, MDPI AG, Sustainability 14 (22) (Nov. 2022) 15448, https://doi.org/10.3390/su142215448. Crossref.
[28] Hosnie Nazari, et al., Saffron yield estimation by adaptive neural-fuzzy inference system and particle swarm optimization (ANFIS-SCM-PSO) hybrid model, Archives of Agronomy and Soil Science, Informa UK Limited (Nov. 2021) 1–15, https://doi.org/10.1080/03650340.2021.2004588. Crossref.
[29] Xiaoyong Liu, Zhili Zhou, A novel prediction model based on particle swarm optimization and adaptive neuro-fuzzy inference system, 33, no. 5, IOS Press, in: L. Juan, G. Guirao, Wei Gao (Eds.), Journal of Intelligent & Fuzzy Systems, Oct. 2017, pp. 3137–3143, https://doi.org/10.3233/jifs-169365. Crossref.
[30] Husanbir S. Pannu, et al., Improved particle swarm optimization based adaptive neuro-fuzzy inference system for Benzene detection, Wiley, CLEAN – Soil, Air, Water 46 (5) (Mar. 2018) 1700162, https://doi.org/10.1002/clen.201700162. Crossref.
[31] M. Nilashi, S. Samad, E. Yadegaridehkordi, A. Alizadeh, E. Akbari, O. Ibrahim, Early detection of diabetic retinopathy using ensemble learning approach, Journal of Soft Computing and Decision Support Systems 6 (2) (2019) 12–17.
[32] M. Little, P. Mcsharry, S. Roberts, D. Costello, I. Moroz, Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection, 1-1, Nature Precedings (2007).
[33] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, E. Duchesnay, Scikit-learn: machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830.
[34] R. Eberhart, J. Kennedy, Particle swarm optimization, Neural Network. 4 (1995, November) 1942–1948.
[35] L.J. Miranda, PySwarms: a research toolkit for particle swarm optimization in Python, J. Open Source Softw. 3 (21) (2018) 433.