

Heba Al Nsour, Mohammed Alweshah\*, Abdelaziz I. Hammouri, Hussein Al Ofeishat and Seyedali Mirjalili

# A Hybrid Grey Wolf Optimiser Algorithm for Solving Time Series Classification Problems

<https://doi.org/10.1515/jisys-2018-0129>

Received March 7, 2018.

**Abstract:** One of the major objectives of any classification technique is to categorise the incoming input values based on their various attributes. Many techniques have been described in the literature, one of them being the probabilistic neural network (PNN). There were many comparisons made between the various published techniques depending on their precision. In this study, the researchers investigated the search capability of the grey wolf optimiser (GWO) algorithm for determining the optimised values of the PNN weights. To the best of our knowledge, we report for the first time on a GWO algorithm along with the PNN for solving the classification of time series problem. PNN was used for obtaining the primary solution, and thereby the PNN weights were adjusted using the GWO for solving the time series data and further decreasing the error rate. In this study, the main goal was to investigate the application of the GWO algorithm along with the PNN classifier for improving the classification precision and enhancing the balance between exploitation and exploration in the GWO search algorithm. The hybrid GWO-PNN algorithm was used in this study, and the results obtained were compared with the published literature. The experimental results for six benchmark time series datasets showed that this hybrid GWO-PNN outperformed the PNN algorithm for the studied datasets. It has been seen that hybrid classification techniques are more precise and reliable for solving classification problems. A comparison with other algorithms in the published literature showed that the hybrid GWO-PNN could decrease the error rate and could also generate a better result for five of the datasets studied.

**Keywords:** Classification problems, probabilistic neural networks, grey wolf optimiser, time series classification.

## 1 Introduction

One of the most important data mining tasks involves classification. In any classification problem, the input dataset is called the training set, which is used for building the model of a class label. This model is used for generating the output for cases when the preferred result is unknown. Neural network (NN) is a popular method for solving classification problems. Different NN models have been developed and published, like the probabilistic NN (PNN), radial basis function network, feed-forward network, multilayer perceptron, and modular networks [10]. The NN models differ from one other based on their architecture, behaviour, and learning approaches. Because of these differences, some of the models are more reliable than others and were used for solving many different problems like the time series classification problem. This is a form of a

---

\*Corresponding author: **Mohammed Alweshah**, Department of Computer Science, Prince Abdullah Bin Ghazi Faculty of Information Technology, Al-Balqa Applied University, Salt, Jordan, e-mail: weshah@bau.edu.jo

**Heba Al Nsour and Abdelaziz I. Hammouri:** Department of Computer Science, Prince Abdullah Bin Ghazi Faculty of Information Technology, Al-Balqa Applied University, Salt, Jordan

**Hussein Al Ofeishat:** Department of Computer Engineering, Al-Balqa Applied University, Salt, Jordan

**Seyedali Mirjalili:** School of Information and Communication Technology, Griffith University, Nathan Campus, Brisbane QLD 4111, Australia

supervised learning process wherein the input is mapped to the final output using historical data. The main objective of the techniques is determining some noteworthy patterns that are present in these data [3, 7].

Recently, metaheuristic algorithms have been hybridised with different kinds of classifiers, and this has resulted in better performance when compared to standard classification approaches [2, 57]. Single-based and population-based metaheuristics have been used for training NNs. Single-solution-based approaches, such as tabu search [54], simulated annealing [13], and the population-based approach, have been found to be more effective when combined with NNs [2]. The combination of an NN and an evolutionary algorithm results in superior intelligent systems, as compared to relying only on either an NN or an evolutionary algorithm [1]. For instance, particle swarm optimisation, on its own and hybridised with a local search operator, has been employed to train an NN [44, 62]. Other swarm intelligence approaches such as ant colony optimisation have also been used to train NNs [12, 51]. Furthermore, Chen et al. [15] have proposed a novel hybrid algorithm based on the artificial fish swarm algorithm. Moreover, genetic algorithms [41] and differential evolution [50] have been effectively employed to enhance bacterial chemo-taxis optimisation [59], while an electromagnetism-like mechanism-based algorithm [56] and the harmony search algorithm have been used to solve various classification problems [31–33, 36], as well as the firefly algorithm (FA) [6, 8], biogeography-based optimisation [7, 9, 26], and other approaches [3–6].

Various works have investigated many classifier algorithms, and it was seen that a single classifier was unable to satisfy all the requirements of a dataset. However, every technique has a distinct applicable area and is favourable to a particular domain based on its characteristics. After understanding the positive and negative benefits of the classification approaches, it becomes important to investigate the probability of assimilating two or more techniques for addressing the classification problems, wherein the advantages of one method would overcome the drawbacks of the second method. Based on this perspective, the effectiveness of the metaheuristic algorithms must be studied for its application in hybridisation [16, 37, 47, 49, 55, 60, 61].

In this study, the researchers studied and applied the grey wolf optimisation (GWO) algorithm [23, 29] for improving the PNN performance for solving the classification problem. Some preliminary solutions were randomly generated, which used the PNN, and the improvement was suggested using the GWO, which optimised the weights of the PNN. To explain further, the potential of using the search ability of the GWO in order to increase the performance of the PNN was analysed. Further, it was analysed how this can be achieved by exploiting and exploring the search space more effectively and by regulating the random steps. Finally, it was assessed how the GWO can avoid premature convergence and immobility of the population, so that the PNN classification technique can find the optimal solution. This was carried out by monitoring the step of randomness and studying the search space for determining the optimal PNN weights.

The paper is organised as follows. In Section 2, the background information and the published literature about GWO are presented. In Section 3, the proposed hybrid method is described. In Section 4, the experimental results are discussed, and its computational complexity is discussed in Section 5. In Section 6, the final conclusions of the paper are presented.

## 2 GWO Algorithm: Background Information and Published Literature

The GWO algorithm was first developed by Mirjalili and Lewis [40], who described a swarm-based metaheuristic algorithm. The GWO algorithm mimics the hunting technique and the social leadership displayed by the grey wolves in nature. The researchers used mathematical modelling for explaining the major stages of the hunting process by the wolves, and used it for solving the optimisation problems. While explaining the mathematical model for the social hierarchy of grey wolves, the GWO algorithm population was divided into four groups, i.e. alpha ( $\alpha$ ), beta ( $\beta$ ), delta ( $\delta$ ), and omega ( $\omega$ ). The three parameter fittest wolves were considered to be  $\alpha$ ,  $\beta$ , and  $\delta$ , and they guided the other remaining wolves ( $\omega$ ) towards the favourable areas within a search space. While optimising, the wolves surrounded their prey, and the mathematical equations

describing this type of behaviour are as follows:

$$\vec{D} = |\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)|, \quad (1)$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D}, \quad (2)$$

where  $\vec{A}$  and  $\vec{C}$  represent the coefficient vectors,  $\vec{X}_p$  describes the position vector of the prey, and  $\vec{X}$  describes the position vector of the grey wolf.  $\vec{A}$  and  $\vec{C}$  vectors are computed as follows:

$$\vec{A} = 2\vec{a} \cdot r_1 - \vec{a}, \quad (3)$$

$$\vec{C} = 2 \cdot \vec{r}_2, \quad (4)$$

where the  $\vec{a}$  constituents are linearly decreased to 0 from 2 using some iterations and  $r_1, r_2$  are the random vectors used in [0, 1].

Using the above equations, the grey wolf present in an  $(X, Y)$  position could update its own position based on the position of the prey  $(X^*, Y^*)$ . Different spots surrounding the best agent could be reached with respect to the current position after adjusting the  $\vec{A}$  and  $\vec{C}$  vector values. For instance,  $(X^*, Y^*)$  is achieved by establishing that  $\vec{A} = (1, 0)$  and  $\vec{C} = (1, 1)$ . It must be noted that the random vectors  $r_1$  and  $r_2$  enable the grey wolves to get access to any position present between these two points. Hence, with the help of the above equations, the wolf could update its present position within the space that surrounds the prey randomly [21].

The GWO algorithm assumes that  $\alpha, \beta$ , and  $\delta$  are the likely positions of the prey. During the optimisation process, the initial three best solutions are assumed to be  $\alpha, \beta$ , and  $\delta$ , respectively. Thereafter, the other wolves, which were considered to be  $\omega$ , could reposition themselves based on the positions of the  $\alpha, \beta$ , and  $\delta$  wolves. The mathematical model that represents the readjustment of the  $\omega$  wolves' position is described below [39]:

$$\vec{D}\alpha = |\vec{C}1 \cdot \vec{X}\alpha - \vec{X}|, \quad (5)$$

$$\vec{D}\beta = |\vec{C}2 \cdot \vec{X}\beta - \vec{X}|, \quad (6)$$

$$\vec{D}\delta = |\vec{C}3 \cdot \vec{X}\delta - \vec{X}|, \quad (7)$$

where  $\vec{X}\alpha$  represents the  $\alpha$  position,  $\vec{X}\beta$  represents the  $\beta$  position, and  $\vec{X}\delta$  describes the  $\delta$  position.  $\vec{C}1, \vec{C}2, \vec{C}3$  are the random vectors, while  $\vec{X}$  indicates the position used in this solution.

As shown in Eqs. (5), (6), and (7), the step size of the  $\omega$  wolf towards the  $\alpha, \beta$ , and  $\delta$  wolves can be defined, respectively. Once this distance is defined, the final positions of the wolves, based on the current solution, are estimated as follows:

$$\vec{X}1 = \vec{X}\alpha - \vec{A}1 \cdot (\vec{D}\alpha), \quad (8)$$

$$\vec{X}2 = \vec{X}\beta - \vec{A}2 \cdot (\vec{D}\beta), \quad (9)$$

$$\vec{X}3 = \vec{X}\delta - \vec{A}3 \cdot (\vec{D}\delta), \quad (10)$$

$$\vec{X}(t+1) = \frac{\vec{X}1 + \vec{X}2 + \vec{X}3}{3}, \quad (11)$$

where  $\vec{X}\alpha, \vec{X}\beta$ , and  $\vec{X}\delta$  show the positions of the  $\alpha, \beta$ , and  $\delta$  wolves, respectively.  $\vec{A}1, \vec{A}2, \vec{A}3$  are the random vectors, while  $t$  indicates the number of iterations used in the study [27].

As shown in Eqs. (8), (9), (10), and (11), the final position of the  $\omega$  wolves is determined. It can be seen that the two vectors  $\vec{A}$  and  $\vec{C}$  are adaptive and random, which helps in exploring and exploiting in the GWO algorithm [48].

As seen in Figure 1, the occurring exploration was  $<1$  or  $>-1$ . Furthermore, even vector  $C$  promoted exploration, when it was  $>1$ . On the contrary, exploitation was more emphasised when  $|A| < 1$  and  $C < 1$  (Figure 1). It must be noted that  $A$  decreases linearly during the process of optimisation for emphasising

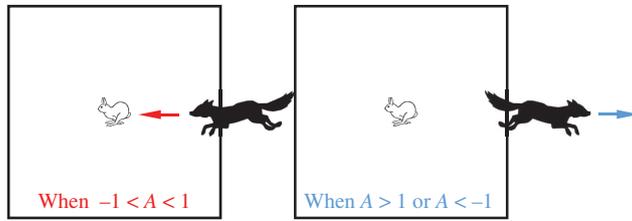


Figure 1: Exploitation and Exploration in GWO.

exploitation, with an increase in the iteration counter. However, vector  $C$  is randomly generated throughout the optimisation process for emphasising the exploration/exploitation taking place at any step, which is a very useful mechanism that helps in resolving the local optima entrapment, which is  $<1$  or  $>-1$ .

Mirjalili and Lewis [40] proposed the GWO as the population-based approach for solving the optimisation problems. In one study, the researchers [42] proposed a better GWO version, whereas Song et al. [52] suggested a GWO-based technique for solving the economic dispatch problems. Studies were carried out using GWO. In one study, Chaman-Motlagh [14] suggested using an optimisation approach using the superdefect photonic crystal filter based on the GWO. Furthermore, El-Fergany and Hasanien [18] also proposed the single- and multi-objective optimal power flow (OPF) technique, which uses GWO, whereas El-Gaafary et al. [19] applied the multi-input and multi-output mechanism. In their study, Emary et al. [20] applied the feature subset selection process based on the GWO intelligent search, and noted that the GWO application improved the performance. Also, Gupta and Saxena [24] proposed a robust generation control strategy in their study. Many optimising strategies were developed by Gupta et al. [25]. The GWO toolkit must be investigated thoroughly for improving the outcome in real life, and addressing and optimising engineering problems. Moreover, Jayapriya and Arock [28] aligned many molecular sequences with the help of the parallel GWO process. In their study, Kamboj et al. [30] explained that a robust balance between exploration-exploitation helped in evading high local optima. Their results suggested that this GWO could be used for addressing many problems related to the economic load dispatch. El-Gaafary et al. [19] suggested using the GWO technique for improving the voltage profile and decreasing the system losses. In one study, Gupta and Saxena [24] suggested the use of the GWO technique for projecting the parameters of the proportional integral controller for an automatic generation control. The GWO technique was also used by Kamboj et al. [30] for solving the capacitated vehicle routing issue problem, whereas Mahdad and Srairi [38] applied it for a blackout risk deterrence in smart grids that was centred on the flexible optimal strategy. Mirjalili [39] used the GWO technique for training multilayer perceptrons, and it was used by Mustafa et al. [43] for training least-squares support vector machines in order to forecast prices.

Pan et al. [45] suggested that the communication strategy must be used in parallel to the GWO. Dzung et al. [17] applied this method for a selective harmonic elimination of the cascaded multilevel inverters. In their study, Emary et al. [21] suggested determining the end-optimised regions of a complicated search space using the GWO technique. In one study [18], the researchers applied the GWO and the differential evolution algorithms for solving the OPF problems. Komaki and Kayvanfar [34] suggested using the GWO algorithm while investigating the two-stage assembly in a problem related to the job shop schedule. Here, the release times for the different jobs along with their sequences were optimised, so that a minimal time was wasted after the last processed job was completed. Recently, Jayakumar et al. [27] suggested using the GWO algorithm for solving the collective heat and the power dispatch issues noted in a cogeneration mechanism.

The GWO algorithm mimics the leadership hierarchy and hunting mechanism of grey wolves in nature. Four types of grey wolves are simulated:  $\alpha$ ,  $\beta$ ,  $\delta$ , and  $\omega$ , which are employed for simulating the leadership hierarchy. This social hierarchy is similar to the water cycle algorithm (WCA) hierarchy with  $N_{sr} = 3$ , where  $\alpha$  could be seen as the sea,  $\beta$  and  $\delta$  as the rivers, and  $\omega$  as the streams. Although the hierarchy is similar, the way in which the GWO algorithm updates the positions of the individuals is different. GWO positions update depending on the hunting phases: searching for prey, encircling prey, and attacking prey. Those hunting phases are the way in which the GWO deals with exploration and exploitation. As mentioned before, the WCA

uses the evaporation process, which is very different to the hunting phases [22]. In this section, we calculate the computational complexity of the GWO as follows:  $Complexity = Loop\ O(n) * fitness\ O(f) = O(nf)$ .

### 3 Proposed Method: Hybridised GWO

In 1990, Specht proposed that PNNs must be used for classifying the patterns based on learning from examples [53]. Most researchers working on PNNs base the algorithm on ‘The Bayes Strategy for Pattern Classification’. Different rules determine the pattern statistics from the training samples to obtain knowledge about the underlying function. The strength of a PNN lies in the function that is used inside the neuron. A PNN consists of one input layer and two hidden layers. The first hidden layer (pattern layer) contains the pattern units. The second hidden layer consists of one summation unit and one output layer, as shown Figure 2. The PNN approach differs from that of a back-propagation NN. The biggest advantage of the PNN is that the probabilistic approach works with a one-step-only learning method. The learning used by back-propagation NNs can be described as one of trial and error. This is not the case for the PNN; it learns not by trial and error but by experience.

A PNN has a very simple structure and very stable procedures. It performs very well with only a few training samples. Moreover, the quality increases as the number of training samples increase, as shown below.

The PNN model used in this study (Figure 1) contains four neuronal layers, i.e. input layer, pattern layer, summation layer, and output layer. The first layer, i.e. input layer, contains several neurons wherein every neuron represents a different attribute within the test or the training dataset (from  $x_1$  to  $x_n$ ). The number of the inputs is seen to be similar to the number of the attributes present in the dataset. Thereafter, the values generated from this input dataset are multiplied by appropriate weights  $w(ij)$ , determined using the PNN

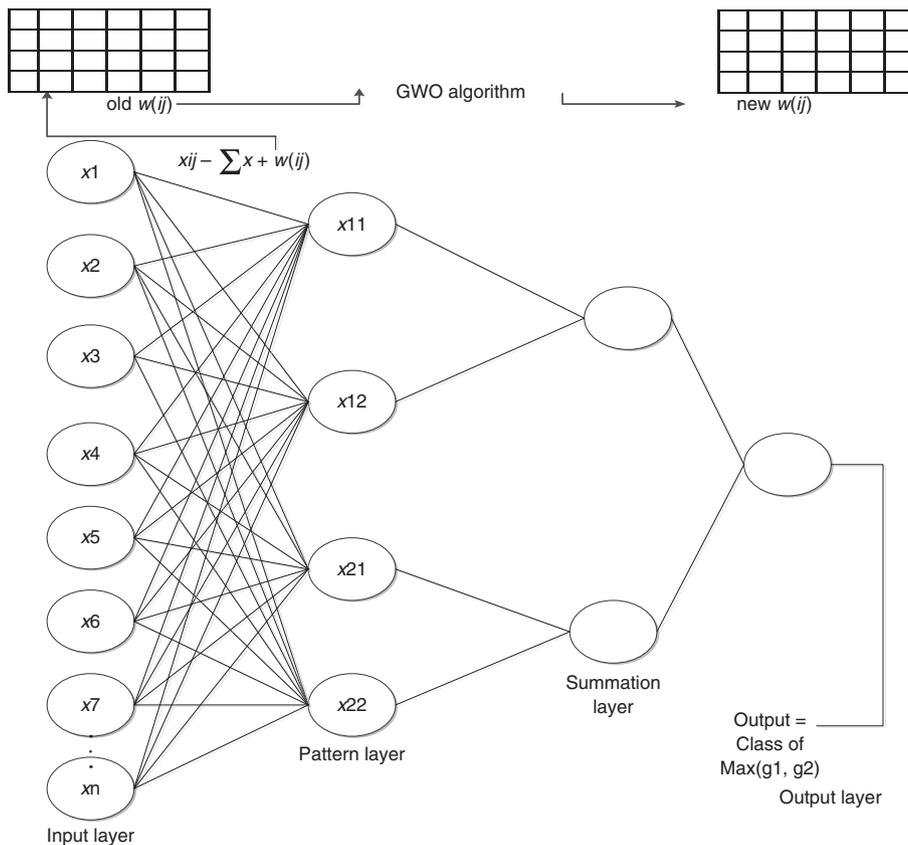
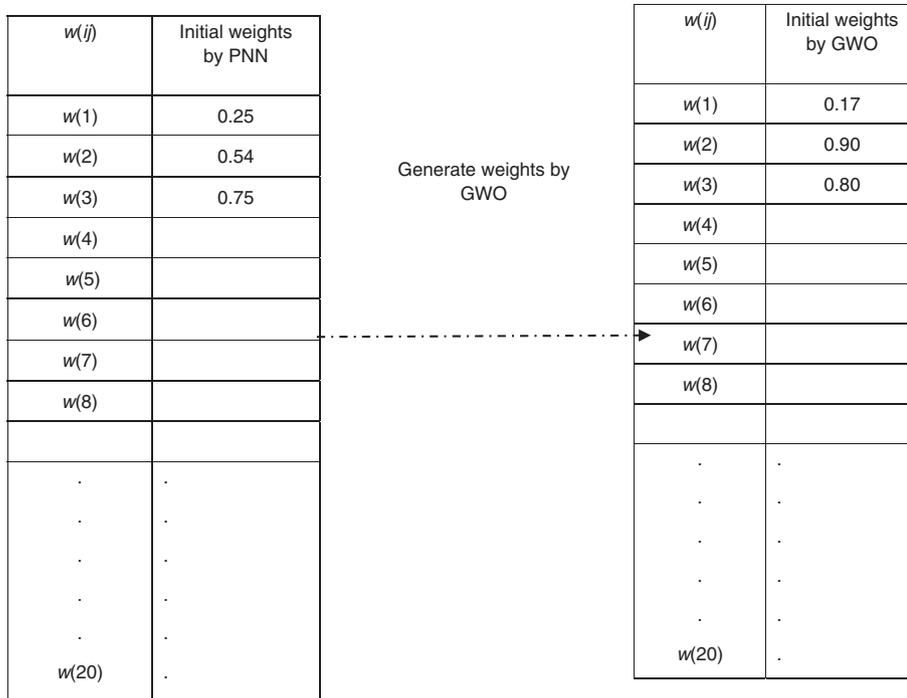


Figure 2: Mechanism of GWO Algorithm with PNN.



**Figure 3:** Representation of the Initial Weights.

algorithm as described in Figure 2, and were then transmitted to the next layer, i.e. pattern. These are then converted using a transfer function to the summation and the output layers, as shown earlier [53]. The last layer is the output layer, which typically consists of one class, as only one output is requested. While carrying out the training process, the main objective is determining the most precise weights that were assigned to a connector line. Herein, the output gets repeatedly computed and the resultant output is then compared to a preferred output, which is generated using the test or the training datasets.

As described in Figure 3, the process begins from the initial weights that are generated randomly using the PNN classification model. The input data values are then multiplied using appropriate weights  $w(j)$ , which have been determined with the help of the PNN algorithm.

Here, the researchers have primarily focused on exploration and exploitation [58], as a balance between these components is very important for any metaheuristic algorithm to be successful [11, 58]. The GWO algorithm was selected for obtaining optimised parameter settings for the PNN training and for achieving a good accuracy.

## 4 Experimental Results

This study makes several contributions to the field by solving classification problems with fast convergence speed and good accuracy. This was done as follows: initially, by using the hybrid technique of PNN and GWO, better results were achieved as it used the GWO algorithm for optimising the PNN weights for obtaining good classification accuracy. The main objective of the GWO is controlling a random step within the algorithm for balancing between exploration and exploitation and determining a near-optimised solution quickly.

Here, the proposed novel algorithm was tested using six benchmark from University of California (UCR) time series datasets (Table 1) using the MATLAB R2010a program (The MathWorks, Natick, MA, USA). Furthermore, simulations were also carried out using the Intel<sup>®</sup> Xeon<sup>®</sup> system with a E5-1630 v3@3.70 GHz CPU that could carry out 20 independent test runs for each data set studied.

The outcome (solution quality) of these experiments obtained with the help of the hybrid algorithm was compared to other methods described earlier, which have dealt with similar problems. The six benchmark

**Table 1:** Characteristics of Datasets.

No.	Name	No. of classes	Size of training set	Size of testing set	Time series length
1	Gun-Point	2	50	150	150
2	Wafer	2	1000	6174	152
3	Lightning 2	2	60	61	637
4	ECG	2	100	100	96
5	Yoga	2	300	3000	426
6	Coffee	2	28	28	286

UCR classification datasets used in this study ranged from a size of 96–637 examples and contained different numbers of attributes [46].

The classification quality can be measured using the accuracy as estimated using Eq. (12). Accuracy can be referred to as true positive ( $TP$ ), true negative ( $TN$ ), false positive ( $FP$ ), and false negative ( $FN$ ) [37], which is further described in Table 2.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \quad (12)$$

where  $TP$  refers to a true positive rate, indicating a fraction of positive cases that are likely to be positive;  $FP$  refers to a false positive rate, indicating a fraction of negative cases that are likely to be positive;  $TN$  refers to a true negative rate, indicating a fraction of negative cases that are likely to be negative; and  $FN$  refers to a false negative rate, indicating a fraction of positive cases that are likely to be negative.

Along with the accuracy, in this study, three additional performance measures were considered, i.e. error rate, sensitivity, and specificity. Equations describing the error rate [Eq. (13)], sensitivity [Eq. (14)], and specificity [Eq. (15)] are as follows:

$$\text{Error Rate} = 1 - \frac{TP + TN}{TP + TN + FP + FN}. \quad (13)$$

$$\text{Sensitivity} = \frac{TP}{TP + FN}. \quad (14)$$

$$\text{Specificity} = \frac{TN}{TN + FP}. \quad (15)$$

Table 3 shows the various parameters and settings for the GWO-PNN algorithm that have been determined after preliminary experimentation.

For determining the effectiveness of the proposed algorithm, this was compared to an already published FA-artificial NN (FA-ANN) [6] algorithm with respect to its accuracy, error rate, sensitivity, and specificity.

**Table 2:** Confusion Matrix for a Classifier.

		Predicted class	
		Yes	No
Actual class	Yes	True positive ( $TP$ )	False negative ( $FN$ )
	No	False positive ( $FP$ )	True negative ( $TN$ )

**Table 3:** Parameter Settings.

Parameter	Value
Population size (no. of grey wolves)	50
Number of iterations	100
$\alpha$ (alpha), $\beta$ (beta), $\delta$ (delta)	Random value from 0 to 1

As shown in Table 6, the hybrid GWO-PNN hybrid algorithm displayed better results in comparison to the FA-ANN. Furthermore, the results showed that the error rate was parallel, and the GWO showed low error rates when compared to other techniques.

Sensitivity refers to the proportion of the TPs that can be appropriately determined using the time series dataset. In the case of the ECG dataset, GWO showed 98% sensitivity, which was better than the value obtained by other algorithms for all the datasets. Specificity refers to the fraction of the TNs that could be determined appropriately. GWO showed 100% specificity for two datasets studied (i.e. Gun-Point and Lightning 2).

Based on the results obtained in Table 4, it can be concluded that GWO showed better performance as compared to FA, due to its ability to preserve the best solution obtained, which further helped the algorithm retain the best position. Furthermore, the hybrid GWO-PNN algorithm was better than the other technique of FA-ANN, with respect to error rate, classification accuracy, sensitivity, and specificity.

The hybrid GWO-PNN algorithm was also compared to some sophisticated algorithms like the FA-ANN using the same dataset for their error rate, and the results are described in Table 5. The best results have been shown in bold letters. The results show that the hybrid GWO-PNN approach (GWO-PNN) performed better than the other algorithms studied for five of the six datasets studied (i.e. Gun-Point, ECG, Wafer, Lightning 2, and Yoga). Furthermore, GWO-PNN classified the Wafer dataset and showed an error rate of 0.001%.

The best performance with a minimal error rate displayed by the GWO-PNN algorithm was due to the fact that the algorithm contained adaptive parameters for effectively balancing exploration and exploitation.

In this study, the GWO-PNN performance was further investigated to determine if there was a statistical difference between GWO-PNN and FA by conducting a t-test with a significant interval of 95% ( $\alpha = 0.05$ ) for the classification accuracy, sensitivity, and specificity. Table 6 shows the p-values obtained.

**Table 4:** Classification Accuracy, Sensitivity, Specificity, and Error Rate (%) of GWO-PNN.

Dataset	TP	FP	TN	FN	PNN accuracy	GWO-PNN accuracy	Sensitivity	Specificity	Error rate
Gun-Point	76	0	71	3	88.67	98.00	0.96	1	0.026
Wafer	5494	5	664	1	99.55	99.00	1.00	0.99	0.001
Lightning 2	33	0	22	6	75.41	90.00	0.85	1	0.10
ECG200	63	1	35	1	88.00	98.00	0.98	0.97	0.02
Yoga	1224	169	1432	175	83.17	88.00	0.87	0.89	0.11
Coffee	11	2	10	5	75.00	75.00	0.68	0.83	0.25

**Table 5:** A Comparison of the Results by GWO-PNN and Other State-of-the-Art Methodologies for Error Rate.

No.	Dataset name	1 – NN Euclidean distance	1 – NN best warping window DTW	1 – NN DTW no warping window	FA-ANN	GWO-PNN
1	Gun-Point	0.087	0.087	0.093	0.080	<b>0.02</b>
2	Wafer	0.005	0.005	0.020	0.004	<b>0.01</b>
3	Lightning 2	0.246	0.131	0.131	0.197	<b>0.10</b>
4	ECG	0.120	0.120	0.230	0.080	<b>0.02</b>
5	Yoga	0.170	0.155	0.164	0.161	<b>0.11</b>
6	Coffee	0.250	0.179	0.179	0.107	<b>0.25</b>

**Table 6:** p-Values of t-Test for GWO vs. FA for Accuracy, Sensitivity, and Specificity.

Dataset	Accuracy	Sensitivity	Specificity
Gun-Point	0.000	0.000	0.000
Wafer	0.000	0.000	0.000
Lightning 2	0.000	0.000	0.000
ECG200	0.000	0.000	0.000
Yoga	0.000	0.000	0.000
Coffee	0.000	0.000	0.000

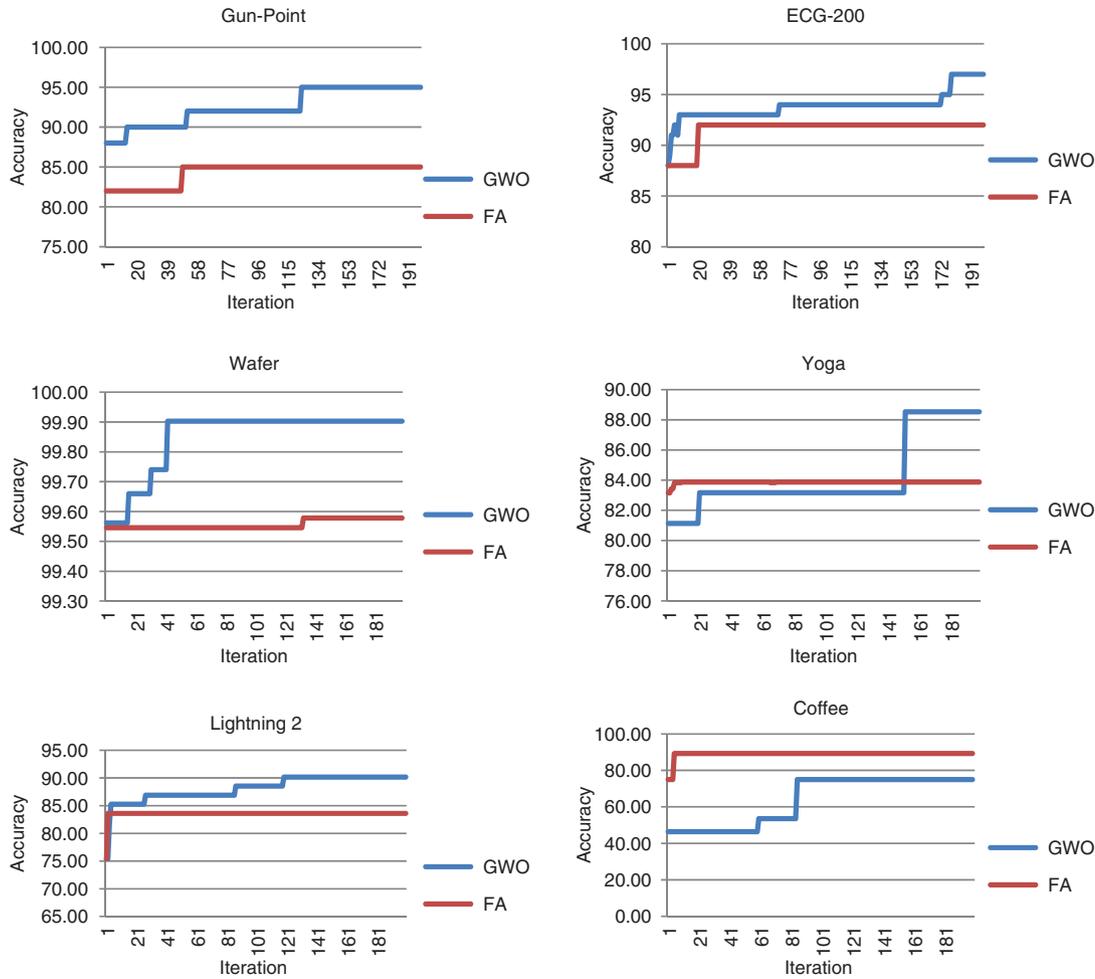


Figure 4: Convergence Characteristics of GWO and FA.

In Table 6, the p-values were  $\leq 0.0$ , which showed that there was a significant difference between the two performances. Hence, it could be concluded that the hybrid GWO-PNN approach could compete with the other published approaches. In particular, the GWO-PNN algorithm showed much better performance as compared to the other techniques, which were equivalent to the best results published in the literature earlier. This is mainly because the hybrid GWO algorithm has improved exploration and exploitation capabilities.

Along with comparing the classification accuracies, the researchers also studied the convergence attributes displayed by the two algorithms. This was done by simulations, and its results are described in Figure 4.

## 5 Conclusion

The main objective of this study was to develop a hybrid algorithm based on GWO and PNN for solving the problems related to the classification of the time series. This was carried out by using the GWO for generating a much better error rate after optimising the weights for the PNN. When tested against six benchmark UCR time series datasets, the hybrid algorithm displayed better performance than the PNN algorithm. Furthermore, the hybrid algorithm also showed better classification accuracy as compared to some of the state-of-the-art methodologies, by displaying better results for five of the six datasets. As a future study, the authors plan to make a hybrid with other search algorithms that possess high exploration, so that a balance

between exploitation and exploration during optimisation can be achieved and population diversity can be maintained.

## Bibliography

- [1] E. Alba and J. Chicano, Training neural networks with GA hybrid algorithms, in: *Genetic and Evolutionary Computation – GECCO 2004*, 2004.
- [2] E. Alba and R. Martí, *Metaheuristic Procedures for Training Neural Networks*, Springer, 2006.
- [3] A. M. Alshareef, A. A. Bakar, A. R. Hamdan, S. M. S. Abdullah and M. Alweshah, A case-based reasoning approach for pattern detection in Malaysia rainfall data, *Int. J. Big Data Intell.* **2** (2015), 285–302.
- [4] A. Alshareef, A. Alkilany, M. Alweshah and A. A. Bakar, Toward a student information system for Sebha University, Libya, in: *2015 Fifth International Conference on Innovative Computing Technology (INTECH)*, pp. 34–39, 2015.
- [5] A. Alshareef, S. Ahmida, A. A. Bakar, A. R. Hamdan and M. Alweshah, Mining survey data on university students to determine trends in the selection of majors, in: *Science and Information Conference (SAI), 2015*, pp. 586–590, UK, 2015.
- [6] M. Alweshah, Firefly algorithm with artificial neural network for time series problems, *Res. J. Appl. Sci. Eng. Technol.* **7** (2014), 3978–3982.
- [7] M. Alweshah, Construction biogeography-based optimization algorithm for solving classification problems, *Neural Comput. Appl.* **29** (2018), 1–10.
- [8] M. Alweshah and S. Abdullah, Hybridizing firefly algorithms with a probabilistic neural network for solving classification problems, *Appl. Soft Comput.* **35** (2015), 513–524.
- [9] M. Alweshah, A. I. Hammouri and S. Tedmori, Biogeography-based optimisation for data classification problems, *Int. J. Data Mining Modell. Manage.* **9** (2017), 142–162.
- [10] M. Alweshah, H. Rashaideh, A. I. Hammouri, H. Tayyeb and M. Ababneh, Solving time series classification problems using support vector machine and neural network, *Int. J. Data Anal. Tech. Strat.* **9** (2017), 237–247.
- [11] C. Blum and A. Roli, Metaheuristics in combinatorial optimization: overview and conceptual comparison, *ACM Comput. Surv.* **35** (2003), 268–308.
- [12] C. Blum and K. Socha, Training feed-forward neural networks with ant colony optimization: an application to pattern classification, in: *Fifth International Conference on Hybrid Intelligent Systems (HIS'05)*, pp. 233–238, Brazil, 2005.
- [13] S. Chalup and F. Maire, A study on hill climbing algorithms for neural network training, in: *Congress on Evolutionary Computation*, University of York, UK, pp. 2014–2021, 1999.
- [14] A. Chaman-Motlagh, Superdefect photonic crystal filter optimization using grey wolf optimizer, *IEEE Photon. Technol. Lett.* **27** (2015), 2355–2358.
- [15] X. Chen, J. Wang, D. Sun and J. Liang, A novel hybrid evolutionary algorithm based on PSO and AFSA for feedforward neural network training, in: *IEEE 4th International Conference on Wireless Communications, Networking and Mobile Computing*, Dalian, China, 2008.
- [16] Z. Chen, S. Zhou and J. Luo, A robust ant colony optimization for continuous functions, *Expert Syst. Appl.* **81** (2017), 309–320.
- [17] P. Q. Dzung, N. T. Tien, N. D. Tuyen and H.-H. Lee, Selective harmonic elimination for cascaded multilevel inverters using Grey Wolf Optimizer algorithm, in: *2015 9th International Conference on Power Electronics and ECCE Asia (ICPE-ECCE Asia)*, pp. 2776–2781, 2015.
- [18] A. A. El-Fergany and H. M. Hasanien, Single and multi-objective optimal power flow using grey wolf optimizer and differential evolution algorithms, *Electr. Power Comp. Syst.* **43** (2015), 1548–1559.
- [19] A. A. El-Gaafary, Y. S. Mohamed, A. M. Hemeida and A.-A. A. Mohamed, Grey wolf optimization for multi input multi output system, *Generations* **10** (2015), 11.
- [20] E. Emary, H. M. Zawbaa, C. Grosan and A. E. Hassenian, Feature subset selection approach by gray-wolf optimization, in: *Afro-European Conference for Industrial Advancement*, pp. 1–13, Cham, 2015.
- [21] E. Emary, H. M. Zawbaa and A. E. Hassenian, Binary grey wolf optimization approaches for feature selection, *Neurocomputing* **172** (2016), 371–381.
- [22] H. Eskandar, A. Sadollah, A. Bahreininejad and M. Hamdi, Water cycle algorithm – a novel metaheuristic optimization method for solving constrained engineering optimization problems, *Comput. Struct.* **110** (2012), 151–166.
- [23] H. Faris, I. Aljarah, M. A. Al-Betar and S. Mirjalili, Grey wolf optimizer: a review of recent variants and applications, *Neural Comput. Appl.* **30** (2018), 413–435.
- [24] E. Gupta and A. Saxena, Robust generation control strategy based on grey wolf optimizer, *J. Electr. Syst.* **11** (2015), 174–188.
- [25] P. Gupta, K. P. S. Rana, V. Kumar, P. Mishra, J. Kumar and S. S. Nair, Development of a Grey Wolf Optimizer Toolkit in LabVIEW™, in: *2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE)*, pp. 107–113, Noida, India, 2015.

- [26] A. I. Hammouri, M. Alweshah, I. A. Alkadasi and M. Asmaran, Biogeography based optimization with guided bed selection mechanism for patient admission scheduling problems, *Int. J. Soft Comput.* **12** (2017), 103–111.
- [27] N. Jayakumar, S. Subramanian, S. Ganesan and E. B. Elanchezhian, Grey wolf optimization for combined heat and power dispatch with cogeneration systems, *Int. J. Electr. Power Energy Syst.* **74** (2016), 252–264.
- [28] J. Jayapriya and M. Arock, A parallel GWO technique for aligning multiple molecular sequences, in: *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 210–215, Kochi, India, 2015.
- [29] H. Joshi and S. Arora, Enhanced grey wolf optimization algorithm for global optimization, *Fundam. Inform.* **153** (2017), 235–264.
- [30] V. K. Kamboj, S. Bath and J. Dhillon, Solution of non-convex economic load dispatch problem using Grey Wolf Optimizer, *Neural Comput. Appl.* **27** (2016), 1301–1316.
- [31] A. Kattan and R. Abdullah, A parallel & distributed implementation of the harmony search based supervised training of artificial neural networks, in: *International Conference on Intelligent Systems, Modelling and Simulation (ISMS)*, United Kingdom, 2011.
- [32] A. Kattan and R. Abdullah, An enhanced parallel & distributed implementation of the harmony search based supervised training of artificial neural networks, in: *Computational Intelligence, Communication Systems and Networks (CICSYN)*, Bali, Indonesia, 2011.
- [33] A. Kattan, R. Abdullah and R. A. Salam, Harmony search based supervised training of artificial neural networks, in: *ISMS'10 Proceedings of the 2010 International Conference on Intelligent Systems, Modelling and Simulation*, Washington, DC, USA, 2010.
- [34] G. M. Komaki and V. Kayvanfar, Grey Wolf Optimizer algorithm for the two-stage assembly flow shop scheduling problem with release time, *J. Comput. Sci.* **8** (2015), 109–120.
- [35] L. Korayem, M. Khorsid and S. Kassem, Using grey wolf algorithm to solve the capacitated vehicle routing problem, in: *IOP Conference Series: Materials Science and Engineering*, p. 012014, Bali, Indonesia, 2015.
- [36] S. Kulluk, L. Ozbakir and A. Baykasoglu, Self-adaptive global best harmony search algorithm for training neural networks, *Proc. Comput. Sci.* **3** (2011), 282–286.
- [37] Q. Luo, S. Zhang, Z. Li and Y. Zhou, A novel complex-valued encoding grey wolf optimization algorithm, *Algorithms* **9** (2015), 4.
- [38] B. Mahdad and K. Srairi, Blackout risk prevention in a smart grid based flexible optimal strategy using Grey Wolf-pattern search algorithms, *Energy Convers. Manage.* **98** (2015), 411–429.
- [39] S. Mirjalili, How effective is the Grey Wolf optimizer in training multi-layer perceptrons, *Appl. Intell.* **43** (2015), 150–161.
- [40] S. Mirjalili and A. Lewis, Grey wolf optimizer, *Adv. Eng. Softw.* **69** (2014), 46–61.
- [41] D. J. Montana and L. Davis, Training feedforward neural networks using genetic algorithms, in: *International Joint Conference on Artificial Intelligence, USA*, 1989.
- [42] N. Muangkote, K. Sunat and S. Chiewchanwattana, An improved grey wolf optimizer for training q-Gaussian Radial Basis Functional-link nets, in: *2014 International Computer Science and Engineering Conference (ICSEC)*, pp. 209–214, Khon Kaen, Thailand, 2014.
- [43] Z. Mustafa, M. H. Sulaiman and M. N. M. Kahar, Training LSSVM with GWO for price forecasting, in: *2015 International Conference on Informatics, Electronics & Vision (ICIEV)*, pp. 1–6, Fukuoka, Japan, 2015.
- [44] M. G. H. Omran, Using opposition-based learning with particle swarm optimization and barebones differential evolution, in: *Particle Swarm Optimization*, vol. 23, InTech Education and Publishing, pp. 343–384, 2009.
- [45] T.-S. Pan, T.-K. Dao and S.-C. Chu, A communication strategy for paralleling grey wolf optimizer, in: *International Conference on Genetic and Evolutionary Computing GEC 2015: Genetic and Evolutionary Computing*, pp. 253–262, Springer International Publishing, Switzerland, 2016.
- [46] H. Pham and E. Triantaphyllou, A meta-heuristic approach for improving the accuracy in some classification algorithms, *Comput. Oper. Res.* **38** (2011), 174–189.
- [47] R.-E. Precup, M.-C. Sabau and E. M. Petriu, Nature-inspired optimal tuning of input membership functions of Takagi-Sugeno-Kang fuzzy models for anti-lock braking systems, *Appl. Soft Comput.* **27** (2015), 575–589.
- [48] R.-E. Precup, R.-C. David and E. M. Petriu, Grey wolf optimizer algorithm-based tuning of fuzzy control systems with reduced parametric sensitivity, *IEEE Trans. Indust. Electron.* **64** (2017), 527–534.
- [49] M. Shams, E. Rashedi, S. M. Dashti and A. Hakimi, Ideal gas optimization algorithm, *Int. J. Artif. Intell.* **15** (2017), 116–130.
- [50] A. Slowik and M. Bialko, Training of artificial neural networks using differential evolution algorithm, in: *Conference on Human System Interactions*, Amsterdam, 2008.
- [51] K. Socha and C. Blum, An ant colony optimization algorithm for continuous optimization: application to feed-forward neural network training, *Neural Comput. Appl.* **16** (2007), 235–247.
- [52] H. M. Song, M. H. Sulaiman and M. R. Mohamed, An application of Grey Wolf Optimizer for solving combined economic emission dispatch problems, *Int. Rev. Modell. Simul. (IREMOS)* **7** (2014), 838–844.
- [53] D. F. Specht, Probabilistic neural networks, *Neural Netw.* **3** (1990), 109–118.
- [54] N. K. Treadgold and T. D. Gedeon, Simulated annealing and weight decay in adaptive learning: the SARPROP algorithm, *Neural Netw.* **9** (1998), 662–668.
- [55] J. Vaščák, Adaptation of fuzzy cognitive maps by migration algorithms, *Kybernetes* **41** (2012), 429–443.

- [56] X. J. Wang, L. Gao and C. Y. Zhang, Electromagnetism-like mechanism based algorithm for neural network training, in: *Advanced Intelligent Computing Theories and Applications with Aspects of Artificial Intelligence*, pp. 40–45, 2008.
- [57] D. Whitley, T. Starkweather and C. Bogart, Genetic algorithms and neural networks: optimizing connections and connectivity, *Parallel Comput.* **14** (1990), 347–361.
- [58] X. S. Yang, *Nature-Inspired Metaheuristic Algorithms*, Luniver Press, UK, 2008.
- [59] Y. D. Zhang and L. Wu, Weights optimization of neural network via improved BCO approach, *Progr. Electromagn. Res.* **83** (2008), 185–198.
- [60] S. Zhang, Y. Zhou, Z. Li and W. Pan, Grey wolf optimizer for unmanned combat aerial vehicle path planning, *Adv. Eng. Softw.* **99** (2016), 121–136.
- [61] S. Zhang, Q. Luo and Y. Zhou, Hybrid grey wolf optimizer using elite opposition-based learning strategy and simplex method, *Int. J. Comput. Intell. Appl.* **16** (2017), 1750012.
- [62] F. Zhao, Z. Ren, D. Yu and Y. Yan, Application of an improved particle swarm optimization algorithm for neural network training, in: *Conference on Neural Networks and Brain (ICNN&B'05)*, Beijing, China, 2005.