

# Adaptive Multilayered Particle Swarm Optimized Neural Network (AMPSONN) for Pipeline Corrosion Prediction

Kien Ee Lee

Department of Computer and  
Information Sciences  
Universiti Teknologi PETRONAS  
Tronoh, Perak

Izzatdin bin Abdul Aziz

Department of Computer and  
Information Sciences  
Universiti Teknologi PETRONAS  
Tronoh, Perak

Jafreezal bin Jaafar

Department of Computer and  
Information Sciences  
Universiti Teknologi PETRONAS  
Tronoh, Perak

**Abstract**—Artificial Neural Network (ANN) design has long been a complex problem because its performance depends heavily on the network topology and algorithm to train the set of synaptic weights. Particle Swarm Optimization (PSO) has been the favored optimization algorithm to complement ANN, but a thorough literature study has shown that there are gaps with current approaches which integrate PSO with ANN, including the optimization of network topology and the unreliable weight training process. These gaps have caused inferior effect on critical Artificial Intelligence (AI) applications and systems, particularly when predicting plant machinery and piping failure due to corrosion. The problem of corrosion prediction in the oil and gas domain remains unanswered due to the lack of a flexible prediction method which targets specific damage mechanisms that caused corrosion. This paper proposes a hybrid prediction method known as the Adaptive Multilayered Particle Swarm Optimized Neural Network (AMPSONN), which integrates several layers of PSO to optimize different parameters of the ANN. The multilayered PSO enables the method to optimize the network topology and train the set of synaptic weights at the same time using a hierarchical optimization approach. Through detailed discussion and literature study, the damage mechanism focused in this research is the CO<sub>2</sub> corrosion and the dataset for this research is obtained from the NORSOK empirical model. The proposed AMPSONN method is tested against BP, MPPO and PSOBP methods on an industrial corrosion dataset for different test conditions. The results showed that AMPSONN performs best on all three problems, exhibiting high classification accuracies and time efficiency.

**Keywords**—Corrosion; damage mechanism; prediction method; artificial neural network; particle swarm optimization

## I. INTRODUCTION

According to the American Petroleum Institute (API), the oil and gas industry is one of the largest and most capital-intensive industries in the world. There are currently more than tens of millions of kilometers of oil and gas pipelines being installed and used daily across the globe [1], [2]. Most of the pipelines in use are made of steel as they deliver the safest means to transport large quantities of oil and gas related products. Despite the use of insulation on these steel pipelines, they are still prone to deterioration when exposed to various damage mechanisms over time [2]-[4]. Damage mechanisms

affecting a certain pipeline depends on the environment in which the pipeline is installed and also the material being transported.

Throughout the years, the API has documented a list of over 60 of these damage mechanisms under API RP 571, which includes CO<sub>2</sub> corrosion, sulfidation corrosion, amine acid corrosion and many more. Damage mechanisms eventually lead to corrosion and the pipeline is subjected to leakages and ruptures, resulting in major financial losses and poses substantial health, environment and safety hazards to the surrounding ecosystem [5], [6].

Therefore, operators have for a long time practiced regular inspection on pipelines to ensure that they operate smoothly and to minimize the risk of accidents [7], [8]. These inspections make use of historical corrosion data and prediction methods are used to predict and monitor the state of pipelines to determine preventive actions to be taken ahead of a potential incident. Despite the effort, pipelines are still failing and pipeline incidents are still occurring throughout the globe, bringing about deadly consequences [9].

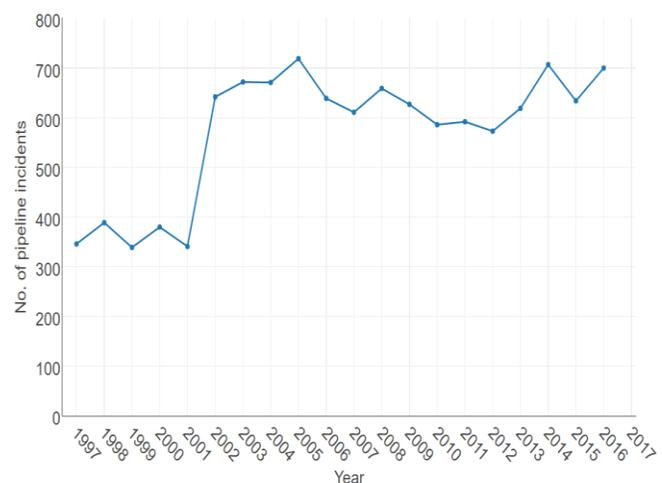


Fig. 1. Pipeline incident count from years 1997-2016 [10].

Fig. 1 shows the annual total of oil and gas pipeline incidents that happened between the years 1997 to 2016, as

collected by [10]. Although there is an intermittent pattern of increase and decline, it can be seen that the overall trend of pipeline incidents is increasing. During an expert interview with Staff Corrosion Engineer from a Malaysian oil and gas company [11], he stated that pipeline corrosion affects Malaysia severely and accounts for over 35% of pipeline failures and incidents. It was revealed that the main reason behind the failure of existing prediction methods used in the oil and gas domain to address the problem of corrosion is due to the different forms of damage mechanisms. Each damage mechanism affects corrosion in their own unique way, which makes it difficult to model their relationship.

Pipeline incidents always bring about casualties and it is therefore of paramount importance to have a solid and good way to monitor and predict the state of corrosion. Thus, this project hopes to develop the AMPSONN method which could adapt to design the most optimum ANN to predict the corrosion severity caused by different damage mechanisms.

## II. LITERATURE REVIEW

### A. Artificial Neural Networks

Artificial Neural Networks (ANN) are powerful mathematical models and universal approximators that have been used to solve various real-world problems. Among all types of ANNs, multilayered perceptron (MLP) is considered as the best, and consists of three layers; the input layer, hidden layer and output layer, with every layer comprising of several neurons. The neurons are connected to each other by a set of synaptic weights, which consists of values representing the strength of the connection. During the training process, the ANN continuously adjusts the values of these weights until a certain termination condition is achieved, usually measured by the error value of the network or the number of maximum iterations.

Although various learning algorithms have been suggested for the training of ANNs, the most popular technique used to train the ANN is still the backpropagation (BP) algorithm [12]-[14]. The BP algorithm has a good acceptance by the community because of its robustness and versatility while providing the most efficient learning procedure for MLP networks [15], [16]. In addition, it is a gradient-descent algorithm which adjusts the weights of ANN by using gradients of their error. Hence, the adjustments done on every weight depends on how much they affect the final output, and this offers a more refined local searching capability while looking for the global minimum [17]. The BP algorithm iterates through the same dataset over and over again until the network converges. Generally, as the number of trained epochs increases, the accuracy of the ANN to predict the output increases, at the cost of a longer training time.

### B. Existing Works on ANN

Table 1 shows the comparison between six recent works on the ANN model for prediction in various domains. From all

the six papers [12]-[14], [18]-[20], there is an agreement that a major problem that most ANN researchers are facing is in initializing the ANN topology. According to Saima et al. [20], the influence of the network topology on the final output is tremendous despite not having a direct interaction with the external environment. The influence the network topology on the output is consequently reflected in the form of other problems such as the slow convergence [12], [13], [19] and getting trapped in a local minimum [18], [19]. Currently, there have been no formally established methods to determine the optimal topology of an ANN for any given problem, especially in the number of neurons in the hidden layer. If the number of neurons is inadequate, it may result in underfitting, meaning that the neural network is unable to learn all the information contained in the dataset. Conversely, an overabundance of neurons in the hidden layer leads to overfitting, a situation in which the neural network captures the noise of the training data, negatively impacting its ability to predict new data [20]. Some literature [21]-[24] offer rules of thumb methods or guidelines for selecting the number of hidden neurons by using any value between the number of input and output neurons, but a good topology cannot be decided solely based on the number of inputs and outputs.

TABLE I. TABLE OF COMPARISON BETWEEN EXISTING WORKS ON ANN

Authors	Domain	Problems Identified	Suggested Future Work
Supriyatman et al. [13]	Oil and gas	Manual selection of topology. Slow convergence.	Identify related parameters to be used to improve accuracy.
Ren et al. [14]	Oil and gas	Manual selection of topology.	-
Sinha & Pandey [12]	Oil and gas	Manual selection of topology. Slow convergence.	Improve convergence rate.
Zhang et al. [18]	General	Number of nodes in the hidden layer affects accuracy and time efficiency. BP algorithm can easily get trapped in a local minimum.	Improve convergence rate by implementing global search algorithms.
Mohammadi & Mirabedini [19]	General	BP algorithm can get trapped in a local minimum. Slow convergence.	Improve convergence rate by implementing exploration algorithms.
Saima et al. [20]	General	There is a major problem in establishing the ANN topology.	-

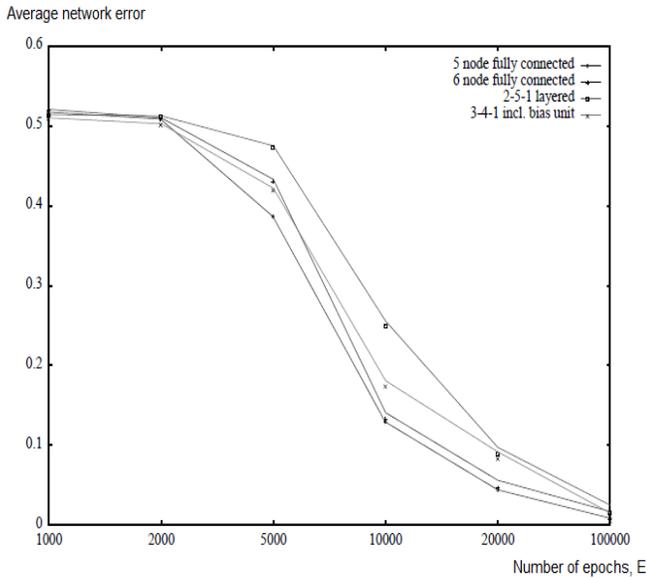


Fig. 2. The learning curve of ANN models of varying topologies [25].

Thus, in all three researches within the oil and gas domain [12]-[14], the selection of ANN topology has been done manually, through trial and error of all possible topologies, and selecting the best performing ANN. Another highlighted problem is the slow convergence of the ANN model [12], [13]. Zhang et al. [18] referred to this problem as an inherent problem of the BP algorithm, as it exhibits a very strong local but poor global searching capability. Thus, the BP algorithm has a high tendency to get trapped in a local minimum while searching for the global minimum [17]. Even in cases when global minimum can be achieved, the entire training process takes a long time to complete and the convergence rate is low. A research [25] has shown that different network topologies affect how long an ANN takes to learn.

Fig. 2 shows the learning curve of four different ANN models with varying topologies. Despite being given the same number of training epochs, each model has different average errors and they learn at different rates. It is therefore possible to implement a form of optimization algorithm that can perform selection of an optimal network topology which converges faster, instead of a manual selection [25]. In [12], [13], [18], [19], authors have proposed several suggestions for future work, such as to have a better research that identifies the right input parameters for specific damage mechanisms when training the neural network and to improve the convergence rate of the model.

### C. Particle Swarm Optimization

Particle Swarm Optimization (PSO) is chosen as the global optimization algorithm to be employed in this research. Although various optimization algorithms were developed in the recent years, population-based evolutionary algorithms remain the most popular, due to their reliability in approximating non-linear problems [26], [27]. PSO has particularly been the favored algorithm as interaction between the swarm particles has shown to be highly effective in finding the global optimum in high-dimensional search spaces [28],

[29]. PSO has also been found to be faster and exhibit a higher computational efficiency as compared to other optimization algorithms [30]-[32]. However, researchers [29], [33], [34] argue that the performance of optimization algorithms depend on several factors, such as the data set involved, type of problem to be optimized, and the selection of parameters. Therefore, there is no way to generalize the performance of different optimization algorithms. Hence, the choice of PSO in this research is heavily weighted with respect to ANN optimization. We have selected PSO to be employed because research done on integrating PSO to optimize various aspects of the ANN is more active compared to other optimization algorithms such as Genetic Algorithm (GA) and Ant Colony Optimization (ACO) [35]. This enables a more thorough literature study to be done when studying existing hybrid methods that have been developed. In addition, PSO is increasingly being used alongside ANN and machine learning applications because it is relatively easier to implement and provides more robustness [28], [36].

The PSO algorithm [37], [38] is a bioinspired optimization method for continuous nonlinear functions as originally proposed by Eberhart et al. [39]. It is based on the behavior of bird flocks in search of food in a search space, and every individual move with respect to their personal best experience and the social best experience of the entire flock. The population can be thought of as a collection of particles  $i$  where each represents a position,  $x_i$  in a multidimensional search space as denoted by the dimensionality,  $D$ . At every iteration, the particles are evaluated based on a fitness function, to identify the global best position  $p_g$ , and also their personal best position  $p_i$  so far. Based on these components, all the particles would update their position within the search space using a velocity function,  $v_i$  at every iteration until an optimum position or solution is found. The velocity function is given as

$$v_i(t + 1) = \omega v_i(t) + c_1 r_1 [p_i(t) - x_i(t)] + c_2 r_2 [p_g(t) - x_i(t)], \quad (1)$$

where  $\omega$  is inertia weight which governs the rate of search space exploration;  $c_1$  and  $c_2$  are acceleration coefficients;  $r_1$  and  $r_2$  are uniformly distributed random numbers between [0, 1]. The velocity,  $v_i$  is controlled between the range of  $[v_{min}, v_{max}]$ . By updating their velocity in this way, particles are able move to a more optimized position,  $x_i(t + 1)$  by the formula

$$x_i(t + 1) = x_i(t) + v_i(t + 1) \quad (2)$$

In the case of using PSO for ANN weight training, every particle represents a set of synaptic weights, which are encoded into a vector as shown in Fig. 3.

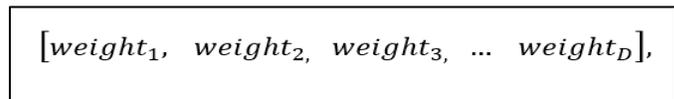


Fig. 3. A vector containing the set of synaptic weights.

$D$  is the dimensionality, or the total number of synaptic weights in the ANN, and can be calculated with the equation

below, with  $I$ ,  $H$  and  $O$  representing the number of input neurons, hidden neurons and output neurons respectively.

$$D = (I \times H) + (H \times O) \quad (3)$$

#### D. Existing Work on PSO with ANN

The first paper by Armaghani et al. uses PSO to completely replace BP as the training algorithm for ANN in the field of geoscience. The findings of their research showed that PSO can evolve the set of optimal weights to achieve an accuracy of 93% on the test dataset. However, the authors did not incorporate any forms of topology optimization in their research and the selection of topology is done via trial and error.

In the second paper, Dang & Hoshino introduced an improved version of PSO, which shall be referred to as Modified Particle Swarm Optimization (MPSO). In their research, MPSO incorporates a method proposed by Shi & Eberhart [41] which reduces the inertia weight,  $\omega$  linearly over the course of iterations, to further improve the searching capability of PSO. By promoting exploration in early iterations and convergence in later iterations, it has been experimentally proven to increase the swarm performance when trying to look for the optimum solution [41]. The equation below shows how the inertia weight can be reduced linearly, given that  $w_1$  and  $w_2$  are the largest and smallest values for the inertia weight respectively,  $MAXITER$  is the maximum number of iterations, and  $iter$  is the current iteration number.

$$\omega = (w_1 - w_2) \times \frac{MAXITER - iter}{MAXITER} + w_2 \quad (4)$$

Besides, Dang & Hoshino proposed for a new component to be added into the velocity update equation in (1), which is called a seed factor,  $Pseed$ . Seed factors of all particles are randomly generated when the algorithm is initialized, and would help to pull the particles to the initial positions of the seeds. The main purpose of adding the seed factor as a third component in the velocity update equation is to help reduce the possibility of the PSO algorithm from getting trapped in a local minimum [40]. The new velocity update equation used in the MPSO algorithm is shown below:

$$v_i(t+1) = \omega v_i(t) + c_1 r_1 [p_i(t) - x_i(t)] + c_2 r_2 [p_g(t) - x_i(t)] + c_3 r_3 [Pseed_i(t) - x_i(t)] \quad (5)$$

From the results of their paper, Dang & Hoshino have proven that their proposed MPSO method outperforms the basic PSO method in terms of accuracy and time efficiency.

The third paper by Koochi & Groza [35], studies the integration between PSO and BP in terms of the weight training process under a method known as Particle Swarm Optimization with Backpropagation (PSOBP). Instead of using classical BP or classical PSO to fully train an ANN, Koochi & Groza divided the ANN training phase into two, the initial and final phases. The initial training phase is done solely using PSO and terminates at an accuracy of 90%, which is a common threshold which most researchers consider a method to have high accuracy [17], [18], [35]. The final training phase is then done using the BP algorithm to perform

a local search around the promising particle to further improve its accuracy. In their study, the ANN was able to exhibit a final accuracy of 98%. The authors have proved that the performance of ANN can be improved by adopting the strong global search capability of PSO to speed up the earlier stage of training and the strong local search capability of BP to refine the final training stage.

TABLE II. COMPARISON BETWEEN EXISTING WORKS ON PSO WITH ANN

Authors	Novelty	Findings	Gaps
Armaghani et al. [17]	Uses PSO to perform weight training on ANN instead of BP.	ANN trained using PSO has a higher convergence rate than ANN trained using BP.	Topology selection is still unoptimized. Training done using PSO may get trapped in a local minimum.
Dang & Hoshino [40]	Introduces Modified Particle Swarm Optimization (MPSO), which includes a seed component into the velocity update equation.	MPSO displayed a better performance than classical PSO.	Topology selection is still unoptimized. Training done using PSO may get trapped in a local minimum.
Koochi & Groza [35]	Introduces Particle Swarm Optimization with Backpropagation (PSOBP). PSO is used in initial phase of training, which then is then switched to BP.	PSO helps to speed up the earlier stages of training, and improves the overall convergence rate.	Topology selection is still unoptimized. Training done using PSO may get trapped in a local minimum.

After studying all three papers, several gaps in the literature have been identified, as shown in Table 2. Firstly, all the authors, Armaghani et al. [17], Dang & Hoshino [40] and Koochi & Groza [35] included the implementation of PSO for weight training purposes and did not implement any form of ANN topology optimization. In their studies, the best ANN topology is still decided by using trial and error, with multiple ANN models of varying topologies being developed and trained to find the one with the best performance. Besides, the range of number of hidden neurons used to develop the multiple ANNs during the trial and error is limited to (3,  $MNN$ ), where  $MNN$  is the maximum number of hidden neurons that the ANN can have. The equation to calculate  $MNN$  is given by (6) [38], [42].

$$MNN = (I + O) \times 2 \quad (6)$$

However, no studies have shown that the optimum number of hidden neurons resides within the range (3,  $MNN$ ) and ANN initialized within this range might not yield the best possible result. Therefore, our proposed method would implement a

layer of topology optimization using PSO within a larger search space than what is used in these three studies.

Secondly, in all three studies, there exists a training component which is solely done using PSO. In the first and second paper, the authors have used PSO to completely replace the original BP algorithm as the training method, while in the third paper, PSO is used solely to initially train the ANN until 90% accuracy. Although PSO has reported success in various problems, there are researches which have shown negative results for PSO used in training. Several studies [18], [19] have shown that while PSO demonstrates strong global searching capabilities, its local searching capability around that global optimum is poor. It is proven that ANN training done solely using PSO achieve convergence very slowly during the later stages of training and would exhibit a poor overall time efficiency. The PSO algorithm also has a disadvantage of easily getting into a local minimum when applied in a high-dimensional search space due to its poor local searching capability around the most optimum particles [18], [20]. This shows that training done solely using PSO in high-dimensional search spaces may not be fully reliable. Thus, our proposed method will involve modifications done within the PSO itself to improve its local searching capability.

#### E. Significance of Study

According to a report by the Institute of Energy Technology, most oil and gas corporations and operators base their current predictions on mechanistic and empirical models [43]. These are developed using data from laboratory testing as well as field data for calibration, and have an advantage which lies in the thorough calibration of the model from laboratory experiments and are verified over a range of data to be safely used with a high confidence within this range of data.

However, mechanistic and empirical models require a long time to be developed and must be backed with years of field experience. A good example of such model is the NORSOK-506 empirical model, which is currently used as a standard for Norwegian oil and gas corporations to predict corrosion under CO<sub>2</sub> corrosion damage mechanism, which has been developed over the past 40 years [43]. Due to the long and thorough process needed to calibrate mechanistic and empirical models, researchers have started to shift their focus towards ANN, which learns faster and does not require as much understanding of the phenomenon. However, as studied in Section II (B), even the most current ANN models developed for corrosion in the oil and gas domain [12]-[14] still depend on trial and error for the establishment of an optimum topology.

Therefore, current prediction methods in the oil and gas domain are still unable to address the problem of having multiple damage mechanisms due the limitations of methods used and researchers chose to target only the more critical damage mechanisms in their studies [5]. The fact that over 60 damage mechanisms exist to cause corrosion [44] should not be overlooked and corrosion prediction for a particular pipeline must be done while taking into account the specific

damage mechanism which is affecting the pipeline. This is because different damage mechanisms affect corrosion in different ways and the number of operational parameters to inspect individual mechanisms varies. Hence, corrosion prediction for a damage mechanism must be done using a model that has been trained for it [11].

This research would be significant for the oil and gas industry by developing a flexible prediction method that can adapt to target different corrosion damage mechanisms to produce an accurate prediction for whichever damage mechanism that requires prediction. A more detailed analysis of this method will be discussed in Section III.

### III. METHODOLOGY

#### A. Proposed Methodology

As studied in Section II, there are gaps in the literature where a novel algorithm and method can be established. To overcome these gaps, we have identified several improvements that can be made by integrating PSO together with ANN in our AMPSONN method.

Firstly, the problem of manual topology selection can be solved by implementing a layer of optimization, where the global search capability of the PSO can be used to design multiple ANN candidates in a large search space. From Section II, we have studied that the main problem in selecting a suitable ANN topology lies in the number of hidden neurons,  $H$  while the number of input neurons,  $I$  and output neurons,  $O$  can be obtained from the dataset used for the training. Therefore, every particle in the PSO is an ANN which has a different topology depending on the value of  $H$  which it is holding, before being trained to identify its performance as compared to all other particles based on certain fitness functions. Through PSO iterations, the particles would have searched through the search space to find a more optimal value of  $H$  while maintaining a memory of the best value of  $H$  so far. When the PSO terminates, the most optimum value of  $H$  would be used to design the topology of the final ANN to serve as the prediction method.

Secondly, to overcome the unreliability of BP and PSO when used separately as ANN training algorithms, we have formulated a novel training approach to capitalize the strong global searching capability of PSO with the strong local searching capability of BP. We propose to improve the searching efficiency of the classical PSO algorithm by providing it with additional gradient-descent information with the use of BP at every single iteration. As mentioned by [18], [19], the PSO is a very promising searching algorithm which has a weakness when it comes to performing local search to attain the global minimum. Although several improvements have been proposed to improve the PSO algorithm over the years, which includes the MPSO [40] and PSOBP [35] algorithms which we have covered in Section II, the capability of the PSO to perform on its own remains an issue and the tendency to become trapped in some undesirable local minimum increases when size of the dimension to be optimized increases. This would become a problem in this research as the size of the dimension increases as the value of

$H$  is increased, due to the increase in number of synaptic weights in the ANN.

Therefore, our proposed training algorithm utilizes BP within each PSO iteration as the BP algorithm offers a more refined local search, by adjusting every single weight value in terms of how much they contribute to the calculated error. We believe that by integrating BP within the PSO itself, our proposed training algorithm would have a more efficient and superior searching capability as compared to PSOBP which only utilizes BP after PSO has completed iterating.

until the objective function of the masterswarm is satisfied. Once the layered PSO has completed its iteration, the obtained masterswarm and subswarm global bests would be the optimized ANN topology and initial weight vector respectively. With that, an optimized ANN will be designed for the research. This ANN will then be trained using BP, a method guaranteed to stop at the global minimum point to fine-tune the weight vectors to achieve an intended level of classification accuracy.

IV. EXPERIMENTATION AND RESULTS

A. Experimental Setup

Table 3 shows the selection of parameters used for the experiments. The parameters for BP, MPSO, and PSOBP algorithms are based on configurations done by their respective authors. On the other hand, the parameters for our AMPSONN algorithm are adopted partially from other authors and through sensitivity analysis. For the masterswarm, the number of maximum iterations is set to 10 with a population size of 5, which would be enough for all the particles to search through most of the search space within the range (3,50). For the subswarm, the number of maximum iterations is set to 100 as inspired by [35], while the population size is set to 10 after performing an optimization run using different values.

TABLE III. EXPERIMENT PARAMETERS FOR THE TESTED METHODS

Method	BP	MPSO	PSOBP	AMPSONN
Maximum BP Iteration	5,000	-	2,000	2,000
Maximum PSO Iteration	-	200	100	10 (Masterswarm), 100 (Subswarm)
Population Size	-	200	200	5 (Masterswarm), 10 (Subswarm)
Initialized Weight Range	(0,1)	(-4,4)	(-1,1)	(-1,1)
BP Learning Rate	0.3	-	0.3	0.3 (PSO run), 0.15 (BP run)

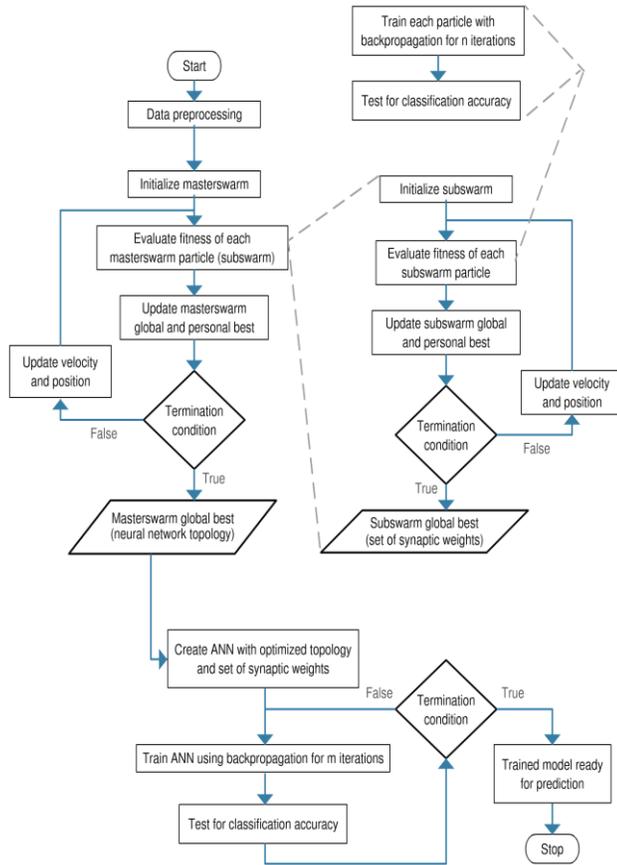


Fig. 4. The flowchart of AMPSONN.

Fig. 4 shows the flowchart of the Adaptive Multilayered Particle Swarm Optimized Neural Network (AMPSONN). It starts from data preprocessing, where the corrosion data is normalized and divided into a training set and a validation set. Next, the layered PSO initializes a masterswarm, in which every particle of the masterswarm is a swarm by itself, called a subswarm. The goal of the subswarms is to provide the best initial weight vector, for a given topology that is given by the masterswarm particle, through a combination of PSO and BP algorithm. The purpose of combining PSO and BP in the training is to provide initial weight vector that will not get stuck in a local minimum, and is a promising solution to the global minimum. In each iteration of the masterswarm, all subswarms would complete an entire optimization run on the objective function and return their fitness value back to the masterswarm. The masterswarm would then continue to iterate

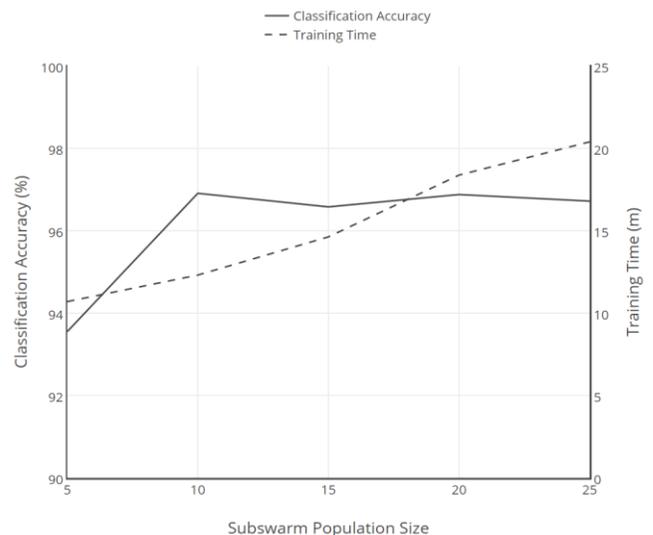


Fig. 5. Sensitivity analysis on the optimal subswarm size.

Fig. 5 above shows sensitivity analysis done to determine the optimal subswarm population size for our AMPSONN method. The increase in subswarm population size shows a linear increase in time taken to train the method. The performance of the method, as denoted by the classification accuracy, shows improvement as the subswarm population size is initially increased from 5 to 10, after which further increments to the population size show no significant improvements in performance. Therefore, the optimum value for the subswarm population size is set as 10.

**B. NORSOK Corrosion Test**

The first test is done by testing the performance of the AMPSONN, PSOBP, MPSO and BP methods on NORSOK corrosion dataset. Since the AMPSONN method optimizes its own ANN topology during runtime, which differs from PSOBP, MPSO and BP which require manual initialization during compile time, several considerations will be taken into account for this test. To provide an unbiased result for the BP, MPSO and PSOBP methods, all three methods are initialized with all possible values of *H* within the range (3,14). The range of (3,14) for the value *H* is the range of number of hidden neurons which the researchers would consider initializing their methods with, as determined by [38], [42] in (6). The best performances of the three methods out of all of their possible topologies are compared to the best performance of AMPSONN across 10 repetitions.

From Table 4 and Fig. 6, we can see that the AMPSONN method outperformed the other three methods in terms of classification accuracy and also training time. AMPSONN can be said to have reached the global minimum by achieving the highest, 97.1% accuracy during the training, with the second highest being the PSOBP method at 93.85%, followed by BP at 91.23% and lastly, MPSO at 86.55%. The close values between the accuracies of testing and validation sets of each method shows that none have suffered from overfitting. The BP, MPSO and PSOBP methods were able to achieve high accuracies, but are still far below that of the AMPSONN method. This signifies that they may have become trapped in local minimums, which supports findings by [18], [19] who stated that training done using PSO may not always be reliable. This also explains why the three methods took more than 40 minutes to train, while AMPSONN completed its training in under 12.77 minutes. The long time taken by the other methods indicates that their trainings were completed after their maximum number of iterations were reached, while AMPSONN terminated the training earlier after having met the PSO termination conditions. By using PSO to design an optimized topology for the base ANN model, the method was able to select a value of *H*, which is neither too large or small. The value of *H* which is selected is optimum, and provides the right number of synaptic connections to support the mapping of relationships between variables in the NORSOK generated dataset. With the right number of synaptic connections, the optimized ANN was able to avoid being underfitted or overfitted, and produces the best performance over the complexity of the network. Therefore, optimization of topology has enabled the AMPSONN to improve its accuracy

and time efficiency in comparison to the other unoptimized method.

TABLE IV. RESULTS OF THE NORSOK CORROSION TEST

Method	Training Time (m)	Classification Accuracy on Validation Set (%)	Classification Accuracy on Testing Set (%)
AMPSONN	12.77	97.1	96.7
PSOBP	48.2	93.85	93.37
MPSO	45.2	86.55	87.67
BP	50.05	91.23	90.89

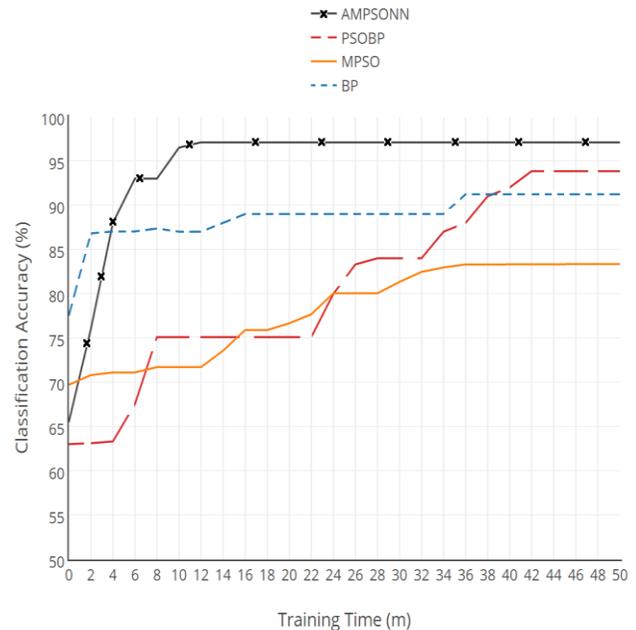


Fig. 6. The training curve of all methods showing accuracy improvement over time.

TABLE V. CALCULATED SPEEDUP PROVIDED BY AMPSONN AGAINST OTHER METHODS

Method	Speedup
PSOBP	$\frac{48.2}{12.77} = 3.77 \text{ times}$
MPSO	$\frac{45.2}{12.77} = 3.54 \text{ times}$
BP	$\frac{50.05}{12.77} = 3.92 \text{ times}$

Table 5 shows the speedup in terms of training time provided by the AMPSONN method against the PSOBP, MPSO and BP methods. From Table 5, it can be concluded that there is a very significant speedup provided by the AMPSONN method, calculated at above 3.5 times against the other three methods.

**C. Optimized Topology Test**

To further test the improvement in performance of the AMPSONN method, an optimized topology test was carried

out. The optimized number of hidden neurons,  $H$  selected by the AMPSONN method will be used to initialize the BP, MPSO and PSOBP method for another round of evaluation on the NORSOK corrosion dataset. This serves as a test to evaluate if the other three methods could perform better than the AMPSONN method if they were provided with an optimized topology. This test would also be significant to show the speedup capability provided by the novel integration of BP into every iteration of subswarm in our proposed AMPSONN method.

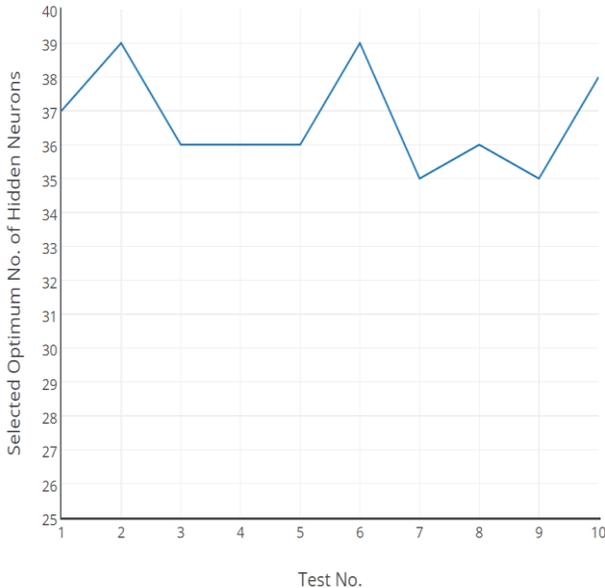


Fig. 7. The number of hidden neurons selected by AMPSONN method across 10 tests.

Fig. 7 shows the optimum number of hidden neurons,  $H$  that was selected by the AMPSONN method across 10 repeated tests. It can be seen that despite the AMPSONN method being given a search space of (3,50), all the selected values of  $H$  are very close to each other and are optimized within the range of (35,39). This indicates that our proposed AMPSONN method is consistent across repetitions, and the classification accuracy achieved is not purely a case of lucky initialization into a position near the global minimum. The selected range of  $H$  within (35,39) also showed that the optimum value of  $H$ , is outside the initialization range that most researches [18], [21]-[24], [38] have considered in their research.

From Fig. 7, the value of 36 has been selected to be the optimum  $H$  for 40% of the time, and this value was used to initialize the BP, MPSO and PSOBP methods. The best performances of the three methods across 10 repetitions are recorded and compared against that of the AMPSONN method. The results for this test are tabulated in Table 6.

TABLE VI. RESULTS OF THE OPTIMIZED TOPOLOGY TEST

Method	Training Time (m)	Classification Accuracy on Validation Set (%)	Classification Accuracy on Testing Set (%)
AMPSOBP	12.77	97.1	96.7
PSOBP	19.62	95.31	95
MPSO	50.07	81.33	81.7
BP	58.03	93.57	93.73

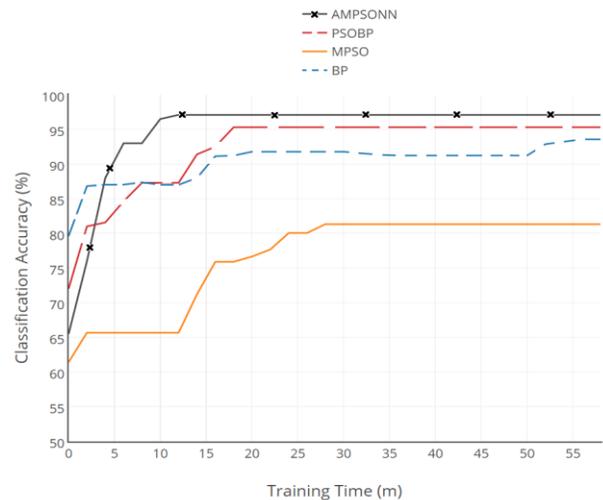


Fig. 8. The training curve of all methods during the optimized test.

Table 6 and Fig. 8 show the results of the BP, MPSO, PSOBP and AMPSONN methods with optimized topologies. It can still be seen that AMPSONN still performed better than the other methods, scoring an accuracy of 97.1%, followed closely by PSOBP at 95.31%, BP at 93.57% and MPSO at 81.33%. By optimizing the topology instead of a manual selection, we can see that the performance of BP and PSOBP have improved significantly, and have managed to obtain accuracies higher than their best accuracies obtained during the previous test. This result is fully supported by Saima et al. [20], who stated that the topology of an ANN greatly affects the performance of the method, regardless of the weight training algorithm used. In terms of training time however, the optimized BP method ended up taking more time to converge in comparison to its unoptimized state, while the optimized PSOBP method was able to save time needed to converge than previously. This is because the BP method is a standalone method while the PSOBP method uses PSO to optimize the earlier stages of training. Therefore, as the number of hidden neurons,  $H$  is increased from within a range of (3,14) to 36, it

incurs more computation effort for the classical BP algorithm in every epoch of training while not affecting PSOBP as much, because the latter only uses BP during the later stage of training. On the other hand, the performance of the MPSO method has not improved even after an optimized topology has been provided. The method is still trapped in a local minimum, and is supported by findings from [20], stating that purely training an ANN using PSO in a high-dimensional search space has a high tendency of getting trapped in a local minimum.

Overall, the proposed AMPSONN method exhibits the best performance over the other methods after optimization, with the lowest training time of 12.77 minutes, followed by PSOBP at 19.62 minutes. This result has proven another important point. With BP, MPSO and PSOBP methods optimized with an optimum topology, this test has taken the AMPSONN masterswarm component out of the picture and set all methods on an equal footing. However, their performance lag behind the AMPSONN method, proving that the weight training (subswarm) component of AMPSONN is more optimized than the other three methods. Although both AMPSONN and PSOBP integrate the PSO algorithm together with BP algorithm, only AMPSONN incorporates BP training into the earlier training stage as well as the later stage of training. By training the subswarm particles with several epochs of BP in every iteration of the subswarm, the particles were to learn faster while being able to capitalize on the ability of PSO to avoid local minimum. The gradient descent information provided by BP has allowed the particles to adjust each synaptic weight with respect to their contribution to the network error, before the next iteration of PSO. This relationship has allowed the method to perform a better local search in between global searches.

With that, the AMPSONN method has shown to be able to save on computational time needed in order to converge to the global minimum. Therefore, the overall performance of the method is attributed to both the topology-optimizing masterswarm as well as the weight-optimizing subswarm.

TABLE VII. CALCULATED SPEEDUP PROVIDED BY AMPSONN AGAINST OTHER METHODS IN THE OPTIMIZED TEST

Method	Speedup
PSOBP	$\frac{19.62}{12.77} = 1.54 \text{ times}$
MPSO	$\frac{50.07}{12.77} = 3.92 \text{ times}$
BP	$\frac{58.03}{12.77} = 4.54 \text{ times}$

Table 7 shows the speedup in terms of training time provided by the AMPSONN method against PSOBP, MPSO and BP methods for the optimized topology test. From the table, the speedup provided by AMPSONN against MPSO and BP is still significant, with 3.92 times against MPSO and 4.54 times against BP. AMPSONN shows a very positive speedup of 1.54 times against PSOBP, even after PSOBP is developed with an optimized topology.

## V. CONCLUSION

This research presented a novel approach to predicting the severity of pipeline corrosion by targeting damage mechanisms individually. After conducting a thorough literature study, a framework which utilizes the combination of PSO and ANN algorithms were used to establish a flexible hybrid method which adaptively designs and trains an ANN until it is fully able to model a given damage mechanism. The hybrid model proposed in this research was compared with other hybrid prediction models based on their classification accuracy and time efficiency. The comparisons are aimed to demonstrate that combining PSO and BP algorithms lead to improved performance over other prediction models in the case of predicting corrosion data of CO<sub>2</sub> damage mechanism. Results obtained have proven that the proposed AMPSONN method has in fact, demonstrated a better performance as compared to BP, MPSO and PSOBP methods. The proposed use of a masterswarm has been justified when an optimized topology allowed the AMPSONN method to achieve accuracy rates higher than the other methods, as well as improving the performance of the other methods when optimized. The proposed use of a subswarm which integrates BP to add gradient-based local information into every iteration of PSO for ANN weight training has shown to outperform that of a pure PSO. The results have also revealed that the AMPSONN method shows a better performance even when the topologies of all methods are optimized, and when the size of training set is reduced. The adaptive multilayered PSO approach has proven itself to be a promising solution to target various individual damage mechanisms.

For future directions for this research, we suggest to incorporate additional ANN parameters into the masterswarm for optimization, such as the transfer function. Although most researchers argue that logistic is sufficient for most cases, some researchers have started to research into optimizing transfer functions [38] for different types of problems. It would be interesting to observe the effect on the performance of the whole method and the improvement on its adaptiveness at the same time. It is also possible to use more datasets in the testing of the AMPSONN method to prove its effectiveness in other application areas. As the field of data analytics and prediction is constantly growing, there is a need to fine-tune new algorithms to adapt to more real-world applications, such as image processing [45] and prediction in the medical field [46].

## REFERENCES

- [1] Demma, P. Cawley, M. Lowe, A. G. Roosenbrand and B. Pavlakovic, "The reflection of guided waves from notches in pipes: a guide for interpreting corrosion measurements," *NDT & E International*, vol. 37, no. 3, pp. 167-180, 2004.
- [2] K. Reber, M. Beller, H. Willems and O. A. Barbian, "A new generation of ultrasonic in-line inspection tools for detecting, sizing and locating metal loss and cracks in transmission pipelines," *Ultrasonics Symposium*, vol. 1, pp. 665-671, 2002.
- [3] M. Lowe, D. N. Alleyne and P. Cawley, "Defect detection in pipes using guided waves," *Ultrasonics*, vol. 36, pp. 147-154, 1998.
- [4] J. A. Beavers and N. G. Thompson, "External corrosion of oil and natural gas pipelines," *Corrosion: Environments and Industries*, vol. 13, 2006.

- [5] M. Singh and T. Markeset, "A methodology for risk-based inspection planning of oil and gas pipes based on fuzzy logic framework," *Engineering Failure Analysis*, vol. 16, no. 7, pp. 2098-2113, 2009.
- [6] M. Hirao and H. Ogi, "An SH-wave EMAT technique for gas pipeline inspection," *NDT & E International*, vol. 32, no. 3, pp. 127-132, 1999.
- [7] F. I. Khan and M. Haddara, "Risk-based maintenance; a quantitative approach for maintenance, inspection scheduling and planning," *Journal of Loss Prevention in the Process Industries*, vol. 16, no. 6, pp. 561-573, 2003.
- [8] J. L. Rose, "Ultrasonic guided waves in structural health monitoring," *Key Engineering Materials*, Vols. 270-273, pp. 14-21, 2004.
- [9] P. K. Dey, "Risk-based maintenance model for offshore oil and gas pipelines," *Journal of Quality in Maintenance Engineering*, vol. 7, no. 1, pp. 25-43, 2001.
- [10] U.S. Department of Transportation, "Pipeline and Hazardous Materials Safety Administration," 2016. [Online]. Available: <http://www.phmsa.dot.gov/pipeline/library/data-stats/pipelineincidenttrends>. [Accessed 29 March 2016].
- [11] A. K. Naim, Interviewee, *Corrosion Engineer*. [Interview]. 25 February 2016.
- [12] S. K. Sinha and M. D. Pandey, "Probabilistic neural network for reliability assessment of oil and gas pipelines," *Computer-Aided Civil and Infrastructure Engineering*, vol. 17, pp. 320-329, 2012.
- [13] D. Supriyatman, Sumarni, K. A. Sidarto and R. Suratman, "Artificial neural networks for corrosion rate prediction in gas pipelines," *SPE International*, 2012.
- [14] C. Ren, W. Qiao and X. Tian, "Natural gas pipeline corrosion rate prediction model based on bp neural network," in *Fuzzy Engineering and Operations Research*, Berlin, Springer, 2012, pp. 449-455.
- [15] Rezaei, M., Monjezi, M., Moghaddam, S. G. and Farzaneh, F., "Burden prediction in blasting operation using rock geometrical properties," *Arab Journal of Geoscience*, vol. 5, pp. 1031-1037, 2012.
- [16] Tawadrous, A. S. and Katsabanis, P. D., "Prediction of surface crown pillar stability using artificial neural networks," *Numerical and Analytical Methods in Geomechanics*, vol. 31, no. 7, pp. 917-931, 2007.
- [17] Armaghani, D. J., Tonnizam, E., Hajihassani, M. and Marto, A., "Blasting-induced flyrock and ground vibration prediction through expert artificial neural network based on particle swarm optimization," *Arabian Journal of Geosciences*, vol. 7, no. 12, pp. 5383-5396, 2013.
- [18] Zhang, J., Zhang, J., Lok, T. and Lyu, M., "A hybrid particle swarm optimization-back-propagation algorithm for feedforward neural network training," *Applied Mathematics and Computation*, no. 185, pp. 1026-1037, 2007.
- [19] Mohammadi, N. and Mirabedini, S. J., "Comparison of particle swarm optimization and backpropagation algorithms for training feedforward neural network," *Journal of Mathematics and Computer Science*, no. 12, pp. 113-123, 2014.
- [20] Saima, H., Khosravi, A. and Jaafar, J., "Neural network ensemble: evaluation of aggregation algorithms in electricity demand forecasting," *International Joint Conference on Neural Networks*, pp. 1-6, 2013.
- [21] Dawson, M. R., "How many hidden units?," Wiley-BI, 2005, pp. 137-144.
- [22] Xu, S. and Che, L., "A novel approach for determining the optimal number of hidden layer neurons for FNN and its application in data mining," *International Conference on Information Technology and Applications*, vol. 5, pp. 683-686, 2008.
- [23] Stathakis, D., "How many hidden layer and nodes?," *International Journal of Remote Sensing*, vol. 30, no. 8, pp. 2133-2147, 2009.
- [24] Blum, A., *Neural networks in c++*, Jo, 1992.
- [25] P. Koehn, "Combining genetic algorithms and neural networks: the encoding problem," 1994. [Online]. Available: <http://homepages.inf.ed.ac.uk/pkoehn/publications/gann94.pdf>. [Accessed 23 March 2016].
- [26] Bergh, F. V. D., "An analysis of particle swarm optimization," University of Pretoria, 2006. Poli, R., "An analysis of publications on particle swarm applications," University of Essex, Essex, UK, 2007.
- [27] Saini, S., Zakaria, N., Rambli, D. R. A. and Sulaiman, S., "Markerless human motion tracking using hierarchical multiswarm cooperative particle swarm optimization," *PloS One*, vol. 10, no. 5, p. e0127833, 2015.
- [28] Khan, K. and Sahai, A., "A comparison of BA, GA, PSO, BP and LM for training feedforward neural networks in e-learning context," *Intelligent Systems and Applications*, vol. 7, pp. 23-29, 2012.
- [29] Mohcene, B., Moussa, S. and Kahla, S., "Maximum power point tracking for photovoltaic systems: A comparison between GA and PSO approaches," *International Journal of Hydrogen Energy*, vol. 10, no. 6, pp. 393-398, 2015.
- [30] V. Kachitvichyanukul, "Comparison of three evolutionary algorithms," *Industrial Engineering & Management Systems*, vol. 11, no. 3, pp. 215-223, 2012.
- [31] R. Hassan, B. Cohanim and O. Weck, "A comparison of particle swarm optimization and the genetic algorithm," *Proceedings of the 1st AIAA multidisciplinary design optimization specialist conference*, pp. 18-21, 2005.
- [32] D. J. Montana and L. Davis, "Training feedforward neural networks using genetic algorithms," *IJCAI*, vol. 89, pp. 762-767, 1989.
- [33] R. Dybowski, P. Weller, R. Chang and V. Gant, "Prediction of outcome in critically ill patients using artificial neural network synthesized by genetic algorithm," *The Lancet*, vol. 347, pp. 1146-1150, 1996.
- [34] Koochi, I. and Groza, V., "Optimizing particle swarm optimization algorithm," *Canadian Conference on Electrical and Computer Engineering*, vol. 27, 2014.
- [35] Ding, S., Li, H., Su, C., Yu, J. and Jin, F., "Evolutionary artificial neural networks: A review," *Artificial Intelligence Review*, vol. 39, no. 3, pp. 251-260, 2013.
- [36] Meissner, M., Schmuker, M. and Schneider, G., "Optimized particle swarm optimization and its application to neural network training," *BMC Bioinformatics*, vol. 7, no. 125, 2006.
- [37] V. R. A. Garro, B. A., "Designing artificial neural networks using particle swarm optimization algorithm,"
- [38] *Computational Intelligence and Neuroscience*, no. 2015, p. 61, 2015.
- [39] Eberhart, R. C., Shi, Y. and Kennedy, J., *Swarm Intelligence, The Morgan Kaufmann Series in Evolutionary Computation*, Boston: Morgan Kaufmann, 2001.
- [40] Dang, T. L. and Hoshino, Y., "An FPGA based classification system by using a neural network and an improved particle swarm optimization algorithm," *Soft Computing and Intelligent Systems*, pp. 97-102, 2016.
- [41] Shi, Y. and Eberhart, R. C., "Empirical study of particle swarm optimization," in *Congress on Evolutionary Computation*, Washington, DC, 1999.
- [42] B. A. Garro, H. Sossa and R. A. Vazquez, "Design of artificial neural networks using a modified particle swarm optimization," in *International Joint Conference on Neural Networks*, Georgia, 2009.
- [43] Nyborg, R., "Guidelines for prediction of CO2 corrosion in oil and gas production systems," Institute for Energy Technology, Kjeller, 2009.
- [44] Inspectioneering, "Damage mechanisms," 2 July 2015. [Online]. Available: <https://inspectioneering.com/tag/failure+mechanisms>. [Accessed 5 April 2016].
- [45] Shi, W., Caballero, J., Huszar, F., Totz, J., Aitken, A. P., Bishop, R. and Rueckert, D., "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," *The IEEE Conference of Computer Vision and Pattern Recognition*, pp. 1874-1883, 2016.
- [46] Li, Q., Cai, W., Wang, X., Zhou, Y., Feng, D. D. and Chen, M., "Medical image classification with convolutional neural network," *International Conference on Control Automation Robotics and Vision*, vol. 13, pp. 844-848, 2014.