

Creep modelling of polypropylenes using artificial neural networks trained with Bee algorithms

Muharrem Düğenci¹, Alpay Aydemir³, İsmail Esen², Mehmet Emin Aydın⁴

¹Karabuk University, Engineering Faculty, Department of Industrial Engineering, Karabuk, TURKEY, mdugenci@karabuk.edu.tr

²Karabuk University, Engineering Faculty, Department of Mechanical Engineering, Karabuk, TURKEY, iesen@karabuk.edu.tr

³Wavin Technology & Innovation B.V. NETHERLAND, alpay.aydemir@wavin.com

⁴University of the West of England, Computer Science and Creative Technologies, UK mehmet.aydin@uwe.ac.uk

Abstract

Polymeric materials, being capable of high mouldability, usability of long lifetime up to 50 years and availability at low cost properties compared to metallic materials, are in demand but finite element-based design engineers have limited means in terms of the limited material data and mathematical models. In particular, in the analysis of products with complex geometry, the stresses and strains of various amounts formed in the product should be known and evaluated in terms of a precise design of the product to fulfil life expectancy. Due to time and cost constraints, experimental data cannot be available for all cases required in analysis, therefore, finite element method-based simulations are commonly used by design engineers. This is also computationally expensive and requires a simpler and more precise way to complete the design more realistically. In this study, the whole creep behaviour of polypropylene for all stresses were obtained with 10% accuracy errors by artificial neural networks trained using existing experimental test results of the materials for a particular working range. The artificial neural network model was trained with traditional as well as heuristic based methods. It is demonstrated that heuristically trained ANN models have provided much accurate and precise results, which are in line with 10% accuracy of experimental data.

Keywords: creep, polypropylene, artificial neural networks, bees algorithms, heuristically trained neural networks.

1. Introduction

The creep behaviour of materials for making analysis of post and corrective design activities is of importance in terms of appropriate product design, since a product design must guarantee the expected lifetime of the product will be put in service. Furthermore, the creep behaviour of a plastic material varies in different stress levels. In this respect, a realistic design requires lots of real data to realistically analyse creep behaviour of a material, where a variety of stress levels require to be

considered and each different stress level in the material necessities producing corresponding data. A part of this data can be produced experimentally, while conducting experimental work for all stress levels is impractical since it costs so much of labour time and incurs design cost. In this respect, new, speedy and reliable methods are always of interest of the researchers to mitigate these circumstances.

Most of the methods available in the literature are material specific, i.e. they are developed for one material, and most of the times, they rely on very complex mathematical formulations with several parameters, which create serious difficulties for many industrial cases. One way to overcome such industrial difficulties, the Finite Element (FE) method is commonly used nowadays as it allows assigning constitutive models to any geometrical form, which can be very complicated. On the other hand, the most commonly used FE codes for structural analysis are limited with their implemented material models. What is, in fact, needed by the practitioner engineers is a tool which is sufficiently simple in use with in the design phase, so that it could easily be interfaced with the most commonly used FE codes for structural analysis; moreover, that tool should be sufficiently general in order to be applied to a vast class of cases, in terms of materials, load types and loading histories, thus providing a reliable and quick prediction within the accuracy of the state of the art of the FE analysis [16].

Artificial neural networks (ANN) are computational models having good records of success and reputation in accurate and precise prediction as well as classification. Once well-configured and trained, an ANN model can be simply integrated into engineering solutions which are commonly used implementing FE-like powerful simulation models. This is because of the fact that an ANN model is able to learn the complex relationships (mostly non-linear) between input and output data collected from a particular problem domain. This makes ANN models increasingly and more commonly used in the mathematical constitutive modelling, but, these are not adequately attempted for creep modelling of materials. Furukawa [9] and Sen [20] are two studies reporting earlier attempts to model the behaviours of materials with ANNs. Also, the study reported in [6] described a procedure particularly for modelling the primary creep behaviour of polypropylene (PP) materials, where the proposed procedure involves using ANSYS [1] to verify an assumed mathematical creep model and calculated creep behaviour material properties. Bano [3] defines a comprehensive ANN approach to describe and predict the behaviour, damage, and life of metals used in turbine components. Bal et al [2] and El-Shafie et al [8] are two relatively recent studies conducted for creep behaviour of construction materials while Sarkar et al [19] have introduced an ANN model for predicting creep behaviour of a particular material in nuclear plant systems. All of these approaches developed their proposals on use of traditional feed-forward ANN models trained with conventional methods, which are proven to provide limited accuracy and success.

A more comprehensive approach has been introduced, in this paper, to overcome the reported limitations of well-known conventional techniques. The design of a realistic water infiltration system has been considered with respect to creep behaviour of its particular material, which is, in fact, a polypropylene material engineered for long-term use of infiltration systems. The creep analyses have been made with ANSYS FE software alongside experimental data. Then, a swarm intelligence-based training algorithm is proposed for feed-forward ANN models to achieve higher accuracy and precision in predictions through more comprehensive training using hybrid Bees algorithm, which is recently developed and proven of success for numerical optimization problems [7].

In the rest of this paper, the problem is introduced in the following section, while the ANN model proposed, and the training algorithms including hybrid Bee algorithm (Hybrid-BA) is presented in

Section 3. The details of experimental study and relevant discussions are provided in Section 4, which is followed by the conclusions in Section 5.

2. The Problem

The problem under consideration for this study is the design and production of rain and sewer water infiltration units, which are made of a particular material so called Polypropylene (PP). The purpose of use these products is to help bring rain and sewer water under control and retain in temporary storage for later use. The lifetime of the units are designed to be up to 50 years. These requirements impose that the products should be produced of plastic materials rather than metals so as to achieve long lasting timeline of use. Water infiltration units are produced in the form of cubic cages in various shape and geometries, where the plastic rods made of PP are assembled; therefore, the mechanical behaviour of an infiltration unit is estimated over the behaviour of the rods, which form up the entire cage. The units are assembled into multiple layers and deployed/placed underground so that the product can function for collecting and retaining the rain and/or sewer water within the underground temporary storages. The units can be subject to various constant and/or sudden/shock waives of load and, hence, pressure/stress/strain, which can last in different time periods. This compels a rigorous engineering process for identifying long term creep behaviour of the units. Figure 1 presents a typical infiltration unit, which has been subject to a particular stress, where part of the grid is brought under focus to reveal the level of stress occurs once the unit remains under pressure.

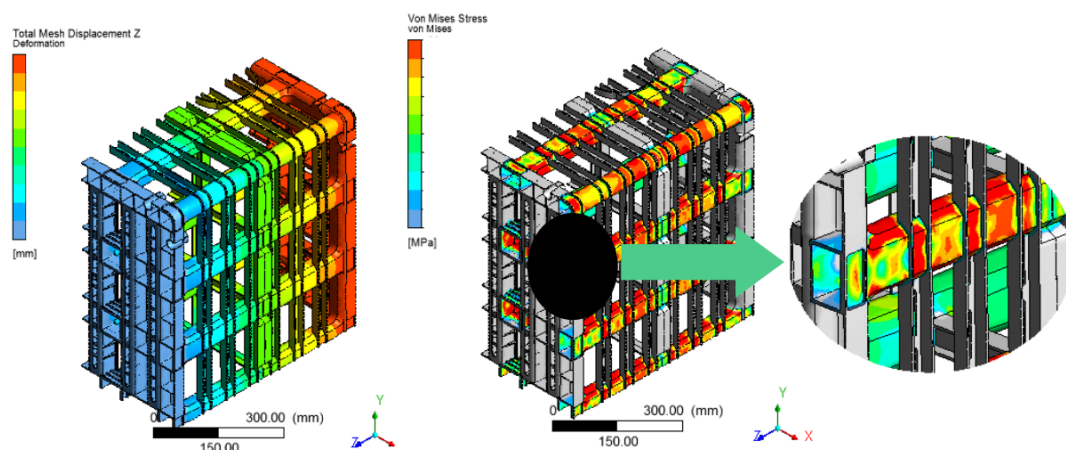


Figure 1: Deformation and stress contours in a particularly engineered infiltration unit.

Figure 1 also provides the stress distribution of the particularly engineered products with specific geometries, where the stress distribution in the structure varies and can be at countless values due to the loading and dimensions of the geometry. In this particular case, the product was loaded in between two rigid plates. The non-uniformity of the stress contour is noticeable although the deformation contour is uniform. Thus, the incurred stress values of particular cases require rigorous attention on experimental data, where it is almost impossible to obtain all the data via experimental tests due to cost and time constraints. Thus, the creep behaviour of this products made of PP enforces to study using FE method-based simulation; here ANSYS software is used for this purpose in this study.

Polypropylene (PP) is selected for this study because of its versatility with respect to physical and especially mechanical properties [10]. PP material is available in three forms: homopolymer,

copolymer, and random copolymer grades. The polypropylene grade used for this study is the random copolymer one, which is detailed in Table 1 with respect to various properties. Injection moulding is one of the most widely employed mass production methods for manufacturing PP products. The material properties of injection moulded products of semi-crystalline polymers strongly depend on the final morphology, which itself depends on the complete (processing) history of the material [25]. This means the actual mechanical properties of plastic products can be quite different than the datasheet figures, which are mostly defined by the material suppliers. Because of this fact, in this study, the creep test samples are taken from the injection-moulded product, not from a typical injection moulded plaque. Consequently, the experimental data used in this study is very specific to the selected design and production.

Table 1: Technical properties of the Polypropylene Random Copolymer

Specifications	Value	Standard
Vicat Softening Point (1 Kg)	132°C	(ISO 1183)
Melting Point	146°C	(ISO 527/1)
Melt Flow Rate (Mn 230/2.16 Kg)	0.3	(ISO 527/1)
Linear Extension Coefficient	0.15 mm/m(°C)	(ISO 527/1)
Density (23 °C)	0.91 g/cm ³	(ISO 527/1)
Yield Strength (at 23 °C) (50 mm/mm)	25 N/mm ²	(DIN 53505)
Elongation at Yield (23 °C) (50 mm/mm)	12%	(ISO 1 79/1 eA)
Breaking Strength (at 23 °C) (50 mm/mm)	34 N/mm ²	(ASTM D 746)
Elongation at Break Point (23 °C) (50 mm/mm)	> 500 %	(ISO 306)
Shore D Hardness (3 sec. Value)	65	(ISO 306)
Charpy Impact Resistance (Notched) (at + 23 °C)	52 KJ/m ²	(ISO 1133)
Brittleness Temperature	-13 °C	(ASTM D 696)

Polymer engineers have proposed some theoretical models of viscoelastic behaviours of the polymer materials. Spring and damper elements can be combined in a variety of arrangements to produce a simulated viscoelastic response. Early models due to Maxwell and Kelvin combine a linear spring in series or in parallel with a Newtonian damper. Other basic arrangements include the three-parameter solid and the four-parameter fluid. These models are very useful in understanding the physical relation between stress and strain that occurs in polymers and other viscoelastic materials. Moreover there are the others that combine these models as can be found in [4]. However, using any of the theoretical models cannot satisfy the whole needs of a design engineer to guarantee realistic viscoelastic behaviour of any polymer material. Further details can be found in [4].

As it is mentioned above, there is not one viscoelastic model available yet to solve any kind of engineering problem dealing with creep. Therefore, most of the commercially available general purpose FE programs come with some implemented (ready-to-use) creep models and an added feature of a user defined creep equation. The designers have still the daunting task of either matching the materials performance with one of these empirical formulas or creating their own equation and subroutine.

In this study, the infiltration unit design is tested (or simulated) according to some local and global (international) norms where few short-term and long-term tests are defined. Since the design is approved, prior to the mould order, based on the simulation results with creep, accuracy in the calculated values is crucial. The accuracy needed for the product release required very intensive material testing and FE modelling.

In order to define the strain vs time behaviour of the PP material, a variety of experimental tests were conducted. A number of accordingly dimensioned rods as test samples have been prepared and examined under certain levels of pressure in which the samples were hanged and each loaded with stress levels in between 2 and 26 MPa. Figure 2 shows the experimental data recorded during the creep tests. The experiments were performed at room temperature (23 °C) since the rigor of varying temperatures is not within the intent of this study. As can be seen from Figure 2, the creep behaviour of the material changes according to the stress. From designer point of view, there should be too many data of the creep behaviour to evaluate the design expectation of the structure in order to satisfy customer specifications such as life expectancy and reliability of the product.

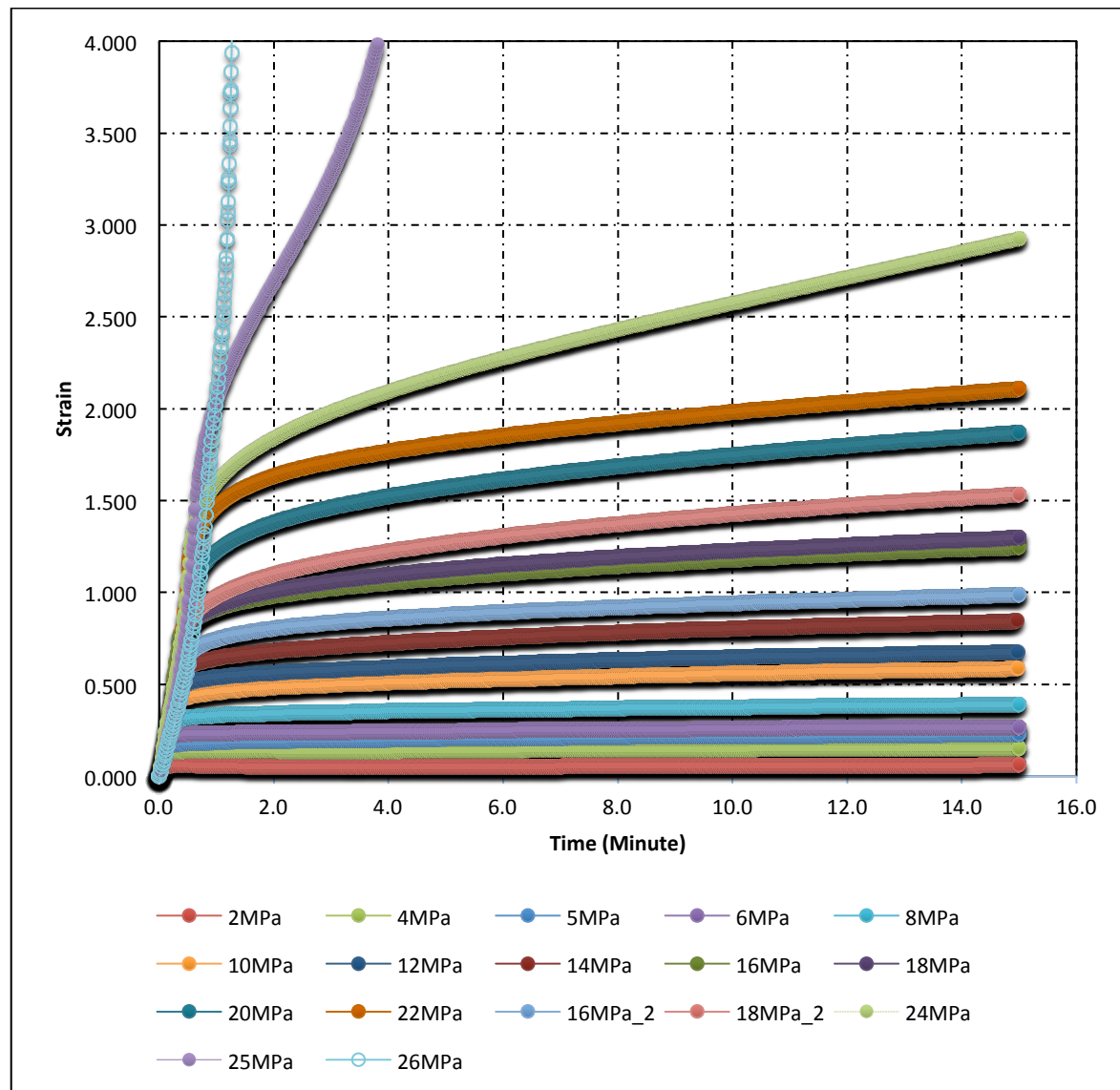


Figure 2: Experimental creep test data of the material for various stresses.

3. Artificial Neural Network and training with Heuristic algorithms

Artificial neural networks (ANNs) are computational modelling tools devised to develop data models for various purposes. They are well-known approaches inspired of human information processing system, where the models are created imitating the working principles of the human brain. An ANN

model is composed of a number of processing units, so-called artificial neurons, organised in particular patterns in which different types of models can be varied in their distinctive patterns. A feed-forward ANN model consists of a layered set of neurons interconnected, where the processing information is done forwardly and backwardly. The model admits data inputs via input layer nodes/neurons and passes them to hidden layer nodes and finally the information is passed to output nodes afterwards. The neurons are connected to every neuron in the next layer by communication links that are associated with connection weights. Signals, which are the form of information units, are passed through neurons over the connection weights. Each neuron receives multiple inputs from the neurons in the previous layer proportional to their connection weights and generates a single output [15,22,23]. A typical feed-forward ANN model is presented in Figure 3, where the model consists of 2-input nodes, 3-hidden nodes and 1-output node associated with 2-bias nodes; all nodes of two adjacent layers are fully connected.

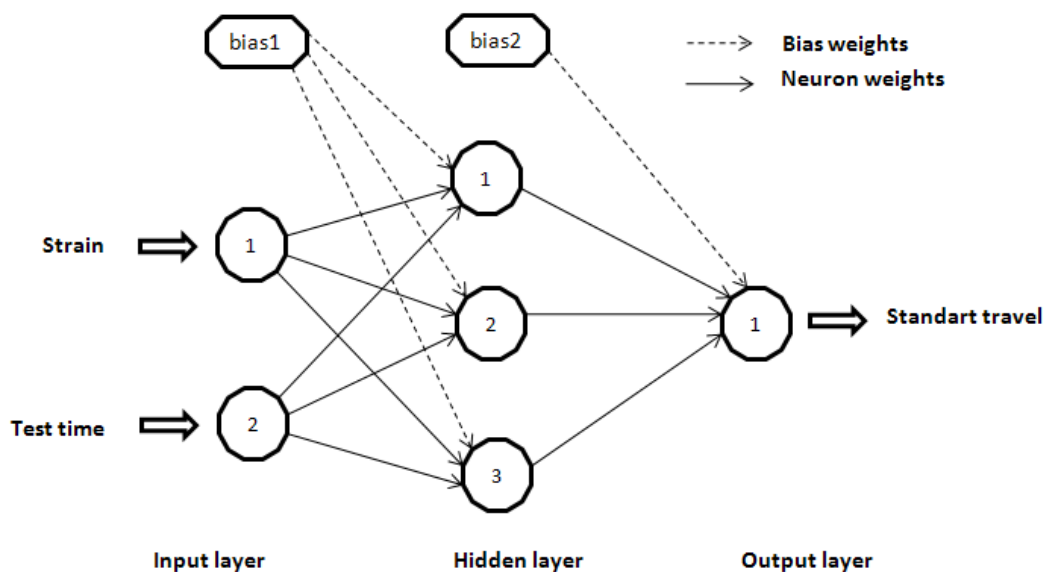


Figure 3; A typical feed-forward Artificial neural network model

The ANN models are configured and trained to adapt the data/problem domain, where the training process can be viewed as minimization of the error that substantiates between the desired (expected) output and the actual output of the model. Training process, in each case, is performed by introducing randomly selected samples of the input data and desired outputs to the configured model of ANN. Once samples are introduced to the ANN model, the sum of squares of the differences between expected and obtained output values is determined. This is applied to the whole network to obtain the total error of the model. Minimisation of the error produced is the process of training the model in which the connection weights linking the neurons are revised/optimised to produce minimum error between the output of ANN and the expected values. Back-propagation (BP) is one of the most well known training algorithms used for this purpose [11]. There are various training approaches used with different strengths and weaknesses. Among them are the metaheuristic-based approaches such as genetic algorithms [5], particle swarm optimisation [17] or bee algorithms [18]. A trained ANN model is tested to verify its readiness for use and to validate that the model produces meaningful results. The model is made ready for use when the predicted results remain within the affordable tolerance limits [15,22,23].

3.1 Training ANN Models

The back propagation (BP) algorithm is the most popular and extensively used algorithm, which has been developed based on gradients. The procedure consists of two phases: the forward and backward phases. The model takes inputs and puts forward and calculates the output of the model during the forward phase, while the error substantiates between the output and desired results are propagated back to the connection weights for better approximation. The details of this algorithm can be found in Haykin [11].

Metaheuristic-based training algorithms are developed based on search algorithms in which the set of connection weights is adopted to be the problem state and the best state is search through a population-based heuristic algorithm such as genetic algorithm (GA), particle swarm optimisation (PSO) and bees algorithm (BA). GA and PSO have been quite widely used algorithms for these purposes with certain level of success records [5].

In this study, we have used GA and Bees Algorithm as training procedures alongside back propagation. GA is one of very well known population based search frameworks developed and extensively used for solving optimisation problems. As mentioned before, training ANN models is also an optimisation process in which the error yielded of ANN models is minimised. GA is a nature inspired metaheuristic approach offers a search mechanism using genetic operators, so-called crossover, mutation and selection, where crossover and mutation operators are used to generate new solution from existing ones while selection helps promote solutions based on their fitness strengths. An implemented GA algorithm is applied to the problems and run generation-by-generation. Finally, the algorithm offers a set of useful solutions, which are usually very close to the optimum [21]. An implementation of GA is run to find the optimum connection weights of inter-neurons, which help yield/predict outputs with minimum deviations.

Bees Algorithm (BA) is one of mainstream swarm intelligence algorithms devised inspiring of collective behaviour of honeybees. There are a number of BA implementations [17, 24] devised for solving various optimisation problems. Another mainstream honeybees-inspired algorithms is Artificial Bee Colony (ABC) algorithm [12, 13], which is successfully implemented and used. A hybrid BA algorithm has recently been developed and used for function optimisation problems [7], henceforth, the same algorithm is decided for use in this study due to the similarity of the problems undertaken and the level of accomplishment.

3.2 Hybrid Bees Algorithm

This hybrid algorithm has been developed based on the framework of original Bees algorithm devised by Pham et al [17] and extended with the operators revised and developed borrowing ideas from ABC algorithm [12] and its variants [13]. The main idea of original Bees algorithm is initially to create a swarm of bees, and configure a search mechanism imitating honeybees' collective behaviours in which the bees keep investigating for food sources, once found inform all other fellow bees to join collective investigation. The original BA algorithm works as follows:

An initial swarm comprised of N individual solutions, where each is considered as an independent source of nectar discovered by individual bees, is generated using the rule inequation (1).

$$\vec{x}_i = \vec{x}_{min} + \vec{\rho} * (\vec{x}_{max} - \vec{x}_{min}) \quad \text{for } \forall i \in N \quad (1)$$

where \vec{x}_i is the solution vector of i^{th} bee generated between \vec{x}_{max} and \vec{x}_{min} with the random vector of $\vec{\rho}$. Then, the solutions are evaluated and ranked based on the richness of the nectars, which is measured by the usefulness of the each individual solution, $F(\vec{x}_i)$. In another word, the quality of solution of each individual is defined as its closeness to the optimum. While moving to the next

generation/ swarm, N_m number of top ranked bees, so called fit-bees, are selected as the set of bees holding promising nectar sources, and among them, N_e of the are identified as elite bees. These promising bees are granted with extra support with sending more bees to the neighbourhood of each. The support team of bees is generated with equation (2) as follows:

$$\vec{x}_i = \vec{x}_i + \vec{\rho} * \delta \quad \text{for } \forall i \in N \quad \text{and } \delta \in \mathbb{R} \quad (2)$$

Where δ is a random number generated within the range of (-1,1) and ρ is another predetermined fixed value to be the step-size of change in any input of a solution/ a bee. The elite bees are granted with more support, say with ε bees, while the other non-elite fit-bees are supported with μ number of bees. As a result, $N_e(1+\varepsilon)$ and $N_m(1+\mu)$ bees are deployed within the new swarm at this stage. For the remaining places in the swarm, $N-(N_e(1+\varepsilon) + N_m(1+\mu))$ number of randomly generated bees using equation (1). This process is repeated until the stopping criterion is satisfied.

The hybrid BA algorithm takes the same framework as explained above, but implies generating supporting bees and sending them to the neighbourhood of each of those identified as elite and fit ranked bees with more operators as given below with equations (3) – (5).

$$\vec{x}_i = \vec{x}_i + \vec{\phi}_i(\vec{x}_i - \vec{x}_k) \quad k \in N \quad \text{and for } \forall i \in N \quad (3)$$

$$\vec{x}_i = \vec{x}_i + \vec{\rho} * \delta * \vec{x}_i \quad \text{for } \forall i \in N \quad \text{and } \delta \in [0, 1] \quad (4)$$

$$\vec{x}_i = \vec{x}_i + \vec{\phi}_i(\vec{x}_i - \vec{x}_k) \quad k \in \{Q_1 \text{ of } N\} \quad \text{and for } \forall i \in N \quad (5)$$

Equation (3) is borrowed from ABC algorithm [13], in which a supporting bee is generated based on the differences between the i^{th} solution/bee and a randomly selected k^{th} solution normalised with the randomly generated number denoted with Φ_i . Equation (4) is a revised version of equation (2), which implies to generate the new candidate solution with more finely granulated level so that the approximation step can be further atomised to fit in any precision level. Finally, equation (5) is the revised version of equation (3), where the k^{th} solution is decided to be selected among the first quartile of the ranked solutions/bees. This hybrid algorithm systematically harmonises/reuses the equations (1) – (5) for generating new solutions/bees as well as neighbours for the existing elite and fit bees, where equation (1) is used for generating the initial swarm and independently exploring for better nectar sources while equations (2) – (5) are used to send supporting bees around each elite bee.

Equation (2), (3), (4) and (5) are the neighbourhood rules used, respectively, by the ordinary BA algorithm, the revised BA algorithm, ABC and revised ABC algorithms [13] to explore around a local nectar source, which means a local region of the search space in optimisation context. The hybrid algorithm randomly selects one of these rules to generate a neighbouring solution of a particular elite solution, each time, to complete up ε supporting bees for each elite so that $N_e \times \varepsilon$ bees can be placed in the new generation. The fit-ranked search bees use only equation (2) and (3) for generating their neighbouring solutions to complete μ number of supporting bees so as to place $N_m \times \mu$ solutions in the next swarm while the independent bees explore with equation (1) for further generations of randomly searched nectars. The rest of algorithmic mechanics of ordinary BA algorithm applies to the hybrid BA until a certain satisfactory level is achieved.

2.3 ANN-BA configuration

Heuristic-based training for ANN is not a very new concept, but is under consideration for further improvements with more powerful algorithms. Bees algorithm is, as mentioned above, one relatively new swarm intelligence algorithms attracts further research efforts as it proves success in various areas of applications, especially in numerical optimisation problems. Dugenci [7] has recently demonstrated that the algorithm is capable of solving high-dimensioned complex numerical

functions. On this basis, BA-based training is devised for optimising the set of connection weights of feed-forward ANN models, so that they can predict and classify more precisely and successfully [14, 18, 21]. In order to achieve this, first an ANN model should be configured and then the set of weights can be retrieved. A typical feed-forward ANN model with 2 inputs, few hidden nodes and 1 output is displayed in Figure 4, where the connection weights labelled with honeybees and the nodes are signified with honeycombs. The total number of connections is the size of a solution state, which is subjected to optimisation process.

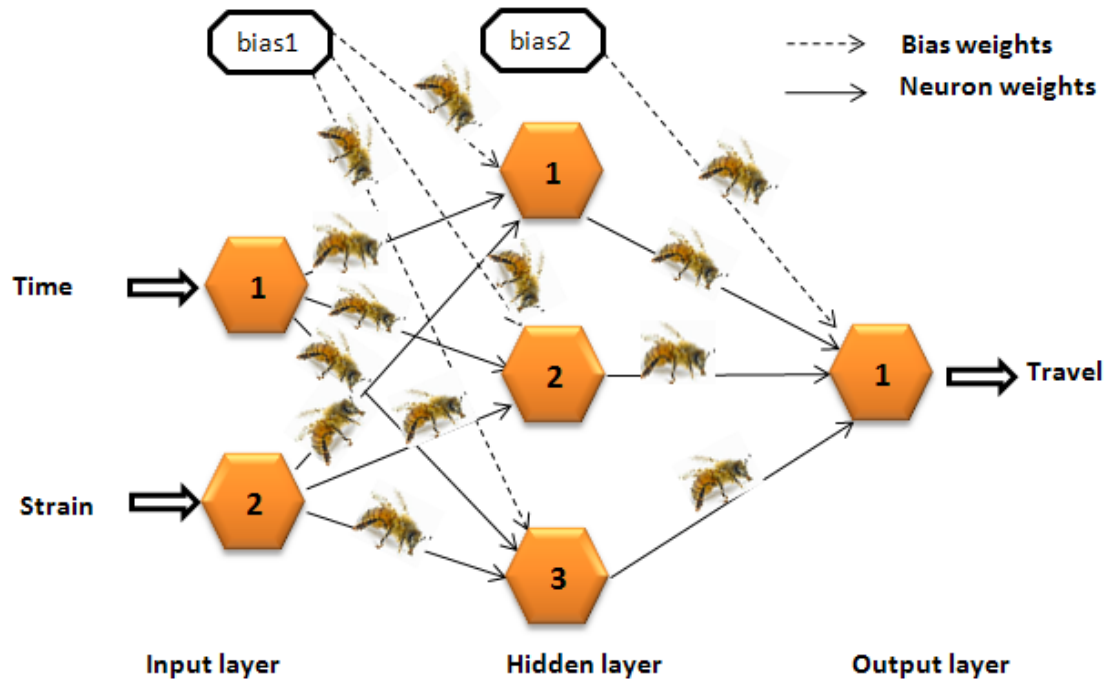


Figure 4; The Feed-forward ANN model configured and trained with Hybrid BA

Given the circumstances, a feed-forward ANN model with I number of input nodes, H number of hidden nodes and O number of output nodes. The number of connections constituted between input and hidden layer is $(I + 1)H$, while the number of connections required between hidden layer and output layer is $(H + 1)O$, where in each level 1 bias node is also considered as part of feed-forward ANN to facilitate learning more smoothly. The set of weights between I and H is $w_i^{I-H} = \{w_{i,j}^{I-H} | j = 1, \dots, (I + 1)H\}$ while the weight set for connections between H and O is $w_i^{H-O} = \{w_{i,j}^{H-O} | j = 1, \dots, (H + 1)O\}$. The ultimate set of weights is $w_i = \{w_i^{I-H} \cup w_i^{H-O}\}$ with the size of $|w_i| = (I + 1)H + (H + 1)O$. For example, given the model in Figure 7, there are 2 input neurons, 3 hidden layer neurons and 1 output neuron with bias nodes in each layer of hidden and output level; hence the total number of connection weights is $(2+1)*3 + (3+1)*1 = 12$. Therefore, a typical bee will represent the whole ANN with a vector of 12 weights including 9 w_i^{I-H} and 4 w_i^{H-O} values. The total number of weights will change accordingly if any of I , or H or O changes.

4. Experimental Results and discussions

In this experimental study, two types of data models are developed; FE method-based simulation model using ANSYS software and artificial neural network based models. In order to realistically touch the ground of the problem and have designed models that can be used in real world, realistic experimental data is used for development, verification and validation purposes.

4.1 Data collection

Experimental data required for this study has been collected from a realistic experimental test set up, where standard travel (in millimetres) data was collected from the test set of two rods with length of 65.00 mm examined/stressed under 8, 10, 12, 14, 16, 18, 20 and 22 MPa pressure over 15 minutes time frame recording the level of standard travel, accordingly. Each rod was examined applying pressure of each of abovementioned levels for 15 minutes and 900 data points were recorded within different time ranges in seconds, where each data point corresponds to each second.

The ANN model developed with this data, has been trained and tested with 400 out of 900 data points which are obtained from evenly distributed 50 data points per abovementioned pressure level. The data has been normalized within the range of [0, 1] over the test period of 0 – 900 seconds, up to 2 mm standard travel values. This normalized data has been represented as in Table 2. Here, 50 data points per pressure level were collected from the complete data set with an even resolution so that the representation of data remains very high and consistent.

Table 2; Normalised standard travel values over test time period given the circumstances of 65 mm of length and 10 MPa of pressure.

L=65.00 mm 10 MPa	
Test time	Standard travel
min	mm
0.001	0.000
0.011	0.008
.....
15.000	0.497

4.2 Feed-forward ANN models

Feed-forward ANN models are the most straightforwardly used models in various areas of data modelling in which the collection of data are used to develop a model so that details of unknown aspects/states of the particular problem under consideration to be predicted. Obviously, these models may not be adequate for highly complex problem, but they are found very useful for the majority of real world problems. Withholding this knowledge in mind, a number of feed-forward ANN models are developed and trained with different learning algorithms. Table 3 and 4 include performance data for 5 ANN models trained with 3 different learning algorithms. First columns of both tables present these 5 ANN models, where all have 2 input nodes, various hidden nodes and 1 output node. This is due to the fact that Creep design requires estimation of one value, which is standard travel results under stressing the rods, while 2 parameters are needed to map the resulted travel with, which are the level of strain and the duration on which strain is applied to the rod. Figure 7 indicates these inputs and output clearly. On the other hand, the number of hidden nodes varies so as to find out with which level the model remains under-trained and when it turns to be over-trained.

All 3 learning algorithms (Back Propagation (BP), Genetic Algorithm (GA) and Hybrid Bees algorithm (Hybrid-BA)) have used the same data sets, which described below, and the performance results are presented in Table 3. The training, for all cases, is stopped once no improvement is persistently observed. BP runs for 5000 iterations using the classical gradient-based propagation algorithm with learning rate of 0.8 and momentum value of 0.2, while GA runs 3000 generations as a generational GA using ranked-based selection, 2 point crossover operator with definite use and mutation

operator with rate of 0.005. It is crucial to indicate that GA does not make any improvement after 1000 generations. On the other hand, Hybrid-BA uses swarm size of 100 solutions, exactly the same as population size of GA, appointing 10 top-ranked bees as elite bees, next 20 top-ranked as fit bees. It is devised to support each elite bee with 5 more supporting bees and each fit bees with no support while 20 randomly searching bees are also appointed. Hybrid BA is run over 2500 generations using neighbourhood factor, δ , of 1. The performance measure adopted is Mean Absolute Percentage Error (MAPE), which is calculated with $MAPE = 1 - error(\%)$. Table 3 and 4 present the mean, minimum, maximum and standard deviation values in MAPE per ANN model trained with each of the learning algorithm. The statistics of each model is calculated over 30 runs.

Table 3; the performance of ANN models trained with three algorithms

ANN-Model		Training (%)			Test (%)		
		Mean	Max	Mean	Max	Mean	Max
BP	2-3-1	87.64	95.09	0.92	75.38	92.67	3.35
	2-4-1	93.38	95.27	1.61	88.13	92.96	4.36
	2-5-1	93.37	95.32	1.94	87.25	92.90	5.43
	2-8-1	95.21	95.58	0.12	91.90	93.21	0.87
	2-12-1	95.09	95.48	0.04	90.79	91.73	0.23
GA	2-3-1	93.33	95.63	1.67	91.81	94.12	2.36
	2-4-1	90.47	95.45	1.36	89.97	93.74	1.72
	2-5-1	94.07	95.49	0.35	93.47	93.74	0.33
	2-8-1	91.46	95.51	0.52	91.08	93.80	0.60
	2-12-1	93.05	95.67	0.26	92.39	94.28	0.24
Hybrid-BA	2-3-1	95.54	96.16	0.09	93.90	95.63	0.22
	2-4-1	95.33	96.23	0.12	93.55	96.27	0.18
	2-5-1	95.48	96.82	0.01	93.73	96.56	0.01
	2-8-1	95.50	96.40	0.01	93.78	96.14	0.02
	2-12-1	95.67	97.25	0.01	94.27	96.84	0.01

The training and test data sets for ANN models indicated in both tables (Table 3 and 4) are designed from 400 experimental data points, where randomly selected 300 out of 400 selected data points are used for training and the remaining 100 of them are used for testing purposes. It is apparent from Table 3 that the highest score by BP with both training and test data is achieved with the model of 2-8-1 in the level of 95.2% and 91.9% from training and test sets respectively. On the other hand, the model of 2-12-1 achieves better performance with both GA and Hybrid-BA. This may be due to that BP can stick in local optima and may not let to improve the performances. It is also observed that the models trained with Hybrid-BA outperform the others, which are trained with both BP and GA with respect to both mean and max measures, while the variations of all results seem slightly but not significantly different as the standard deviations do not suggest very high differences.

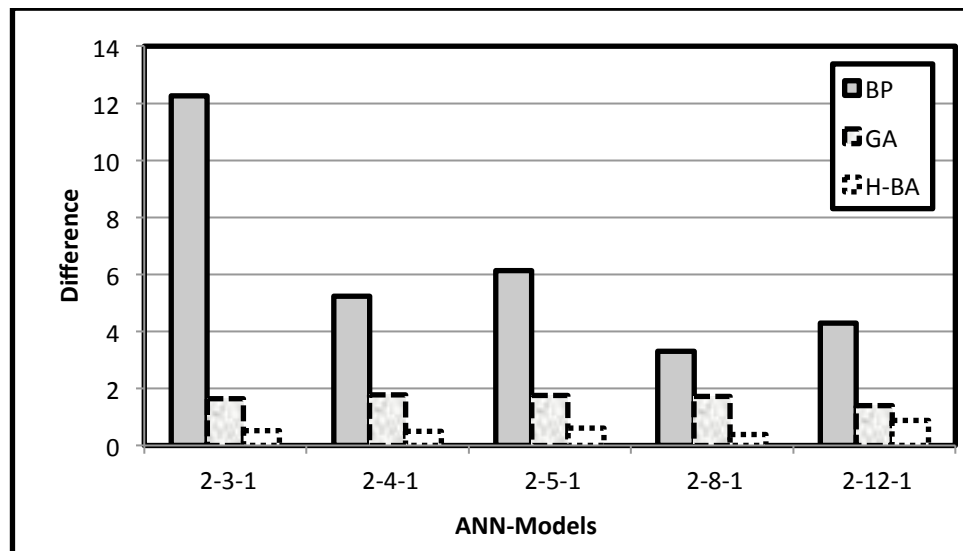
The results in Table 4 represent the performance of all existing models trained with a different data set, which deliberately exclude data for the case of 16 MPa pressure. Once trained, the tests have been conducted as usual. The performances for both of training and test cases look slightly different with all three algorithms, while the results by Hybrid-BA look outperforming the other 2 algorithms. It is observed that the variations of the results also remain alike.

Table 4; the performance of ANN models trained with three algorithms, which did not include case 16MPa in the training

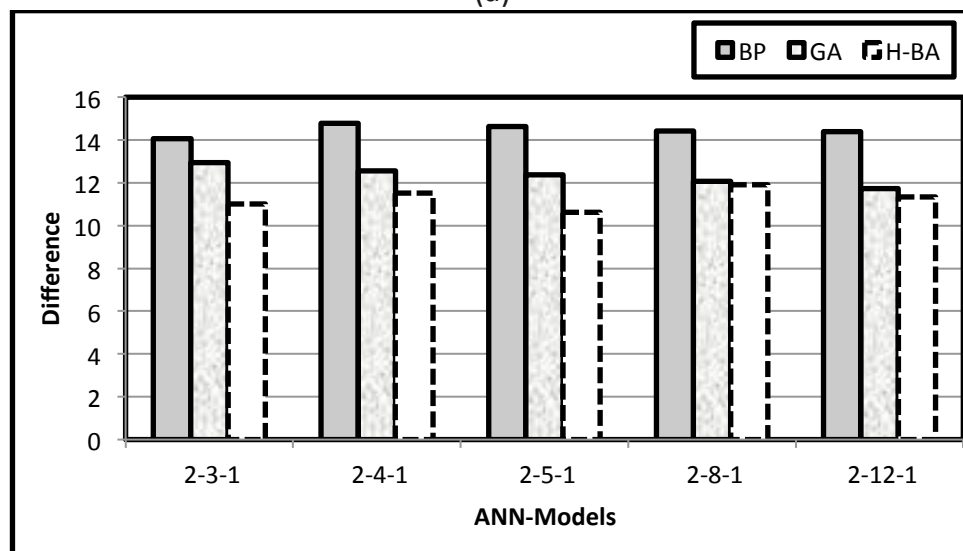
ANN-Model		Training (%)			Test (%)		
		Mean	Max	St Dev	Mean	Max	St Dev
BP	2-3-1	94.48	96.47	1.87	80.43	81.86	1.26
	2-4-1	96.23	96.48	1.20	81.44	82.05	1.80
	2-5-1	96.40	96.46	0.03	81.79	82.29	0.24
	2-8-1	96.35	96.72	0.00	81.93	83.93	0.13
	2-12-1	96.23	96.41	0.05	81.83	82.72	0.23
GA	2-3-1	91.28	96.63	0.73	80.27	83.97	0.12
	2-4-1	92.23	96.72	0.48	80.70	84.67	0.41
	2-5-1	92.19	96.79	1.83	81.56	84.59	0.11
	2-8-1	94.89	96.98	0.10	82.00	85.48	0.22
	2-12-1	93.27	97.01	0.44	80.93	85.42	0.17
Hybrid-BA	2-3-1	96.60	96.89	0.03	83.66	85.01	1.29
	2-4-1	96.67	97.15	0.06	84.11	89.30	0.69
	2-5-1	96.75	97.58	0.04	84.38	86.62	0.41
	2-8-1	96.94	97.89	0.03	84.87	88.20	0.16
	2-12-1	97.00	97.54	0.01	85.28	88.96	0.54

An attempt is made to investigate how robust and precisely the ANN-models can estimate unseen samples; a training set excluding data from the case 16MPa is developed and all ANN models are trained with all 3 algorithms and then all seen and unseen cases have been tested accordingly. The performance results are indicated in Table 4. It is clearly observed that the gap between test performances and the training performances increases by roughly 10%. Figure 5 (a) plots the achievement of all 3 algorithms for the models, where apparently the lowest differences clearly and significantly achieved by Hybrid-BA, where the gap significantly rises by BP as expected, while the other 2 seems stable. On the other hand, Figure 5 (b) reflects a scattered pattern for all while GA looks more stable than other 2, but the ultimate lower differences archived with Hybrid-BA. The rough difference level around 10% is observable within the experimental data, which is considered as the tolerable range. Table 5 includes the differences between 2 rods while subjected to the same strain level, while first column presents the strain levels, the second column shows the mean difference calculated over many experimental results, and the third column indicates the last difference while subjected to a particular strain just before the end of 15 min test duration. The bottom row of Table 5 presents the mean of all differences, which are both just above 10%. This clarifies that even the experimental data have a rough tolerance of 10%-12%, hence, the prediction

results of ANN models for unseen samples remain within this range, which is the safe region, and no harms causes to the robustness of the prediction.



(a)



(b)

Figure 5; Differences substantiated between performances of ANN models with training data and test data. (a) differences while all cases are included in the training set, (b) differences while 16 MPa case is excluded of training data.

Table 5; The differences measured between two experimental data w.r.t. the standard travel level resulted subject to various stress levels.

Strain (MPa)	Mean Difference	Last Difference
2	18.99%	22.16%
4	12.43%	11.98%
5	6.51%	5.00%
6	5.15%	6.09%
8	3.44%	3.04%

10	16.75%	17.57%
12	6.48%	7.73%
14	13.47%	14.89%
16	7.54%	7.95%
18	12.18%	13.38%
20	14.35%	17.21%
22	6.27%	2.43%
Mean	10.30%	10.79%

As a result, the trained and tested Hybrid-BA model is made ready and fit to estimate/predict the standard travel values subjected to 16 MPa pressure level for 900 seconds. Figure 6 presents the achievement of Hybrid-BA in comparison to the real-measured data and the values estimated by ANSYS software; the graph for Hybrid-BA is presented in the legend with “Neural Network Guess”. Evidently, the parallelism in all data types is apparent, where the blue red and green graphs present the measured data, results by Hybrid-BA and estimations by ANSYS, respectively. The overlaps of Hybrid-BA results and the measured data indicate the precision level achieved by Hybrid-BA, while ANSYS estimations remain a little bit distant as expected.

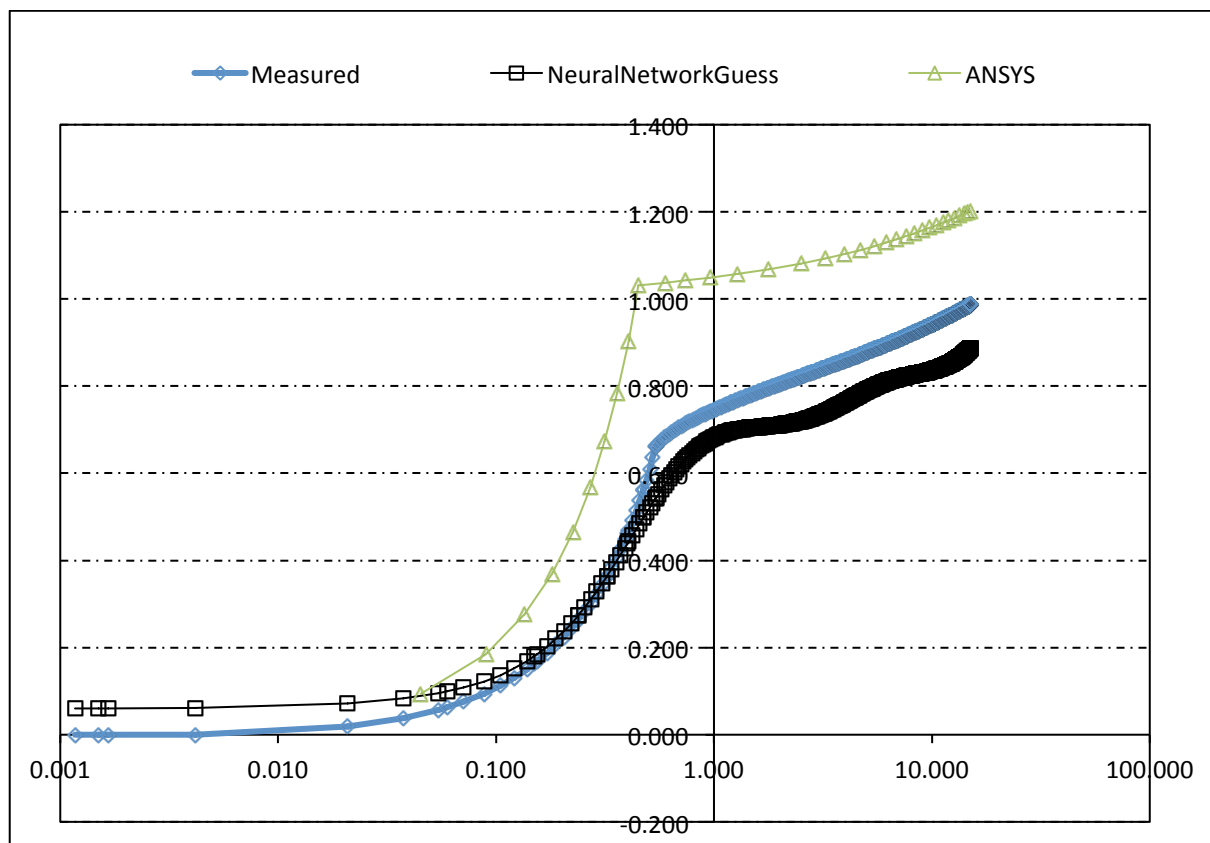


Figure 6; Experimental results for standard travel estimation by Hybrid-BA and ANSYS in comparison with measured data.

5. Conclusions

This study was conducted to investigate the viability of artificial neural networks trained by one of very recently developed metaheuristic algorithms, namely hybrid Bee algorithm, to achieve high precision in the estimation of expected results. The problem domain is based on a realistic design and engineering problem, which is known to be one of time-consuming data processing procedures. The creep behaviour of polypropylene materials in use of long term use purposes. The behavioural estimation of such cases is conveyed with FE method-based engineering software, ANSYS, which requires relatively long time to work out estimations. In this study, feed-forward ANN techniques are used to model the creep behaviours of polypropylene materials, where ANN model is trained with one of very recent metaheuristic approach for achieving high precision for the estimation as, in fact, it is well-known that FE-based estimations are not fine-grained. The experimental results suggest that ANN-BA estimations are better matching the real measured data.

References

- [1] ANSYS Version 16 Element Reference, 3.5.5.1. Implicit Creep Equations.
- [2] Bal, L., and Buyle-Bodin, F. (2014), "Artificial neural network for predicting creep of concrete", *Neural Computing and Applications*, 25(6), 1359-1367.
- [3] Bano, N., (2013), Neural Network Approach for Predicting the Failure of Turbine Components, PhD thesis, Department of Mechanical Engineering, Faculty of Engineering, University of Ottawa, Canada
- [4] Brinson H. F. and Brinson L. Catherine, (2008), *Polymer Engineering Science and Viscoelasticity: an Introduction*, Springer, New York, NY, 2008.
- [5] Chan, K. Y., Chan, K. W., Pong, G. T., Aydin, M. E., Fogarty, T. C., & Ling, S. H. (2009), "A statistics-based genetic algorithm for quality improvements of power supplies", *European Journal of Industrial Engineering*, 3(4), 468-492.
- [6] Dropik, M. J., Johnson, D. H., and Roth, D. E., (2002), "Developing an ANSYS Creep Model for Polypropylene from Experimental Data", International ANSYS Conference, 161, 2002.
- [7] Dugenci, M., (2015), "Honeybees-inspired heuristic algorithms for numerical optimisation", arXiv:1504.05766 [cs.NE].
- [8] El-Shafie, A., Abdelazim, T., and Noureldin, A. (2010), "Neural network modeling of time-dependent creep deformations in masonry structures", *Neural Computing and Applications*, 19(4), 583-594.
- [9] Furukawa T., Yagawa G., "Implicit Constitutive Modelling for Viscoplasticity Using Neural Networks", *Int. Journal for Numerical Methods in Engineering* 43 , pp. 195-219, 1998
- [10] Galli, P., & Vecellio, G. (2004). Polyolefins: The most promising large-volume materials for the 21st century. *Journal of Polymer Science Part A: Polymer Chemistry*, 42(3), 396-415.
- [11] Haykin, S. S. (2009). *Neural networks and learning machines* (Vol. 3). Upper Saddle River: Pearson Education.
- [12] Karaboga, D., B. Gorkemli, C. Ozturk, and N Karaboga. "A comprehensive survey: artificial bee colony (ABC) algorithm and applications." *Artificial Intelligence Review* 42, no. 1 (2014): 21-57.
- [13] Karaboga, D., and B. Akay. "A comparative study of artificial bee colony algorithm." *Applied Mathematics and Computation* 214 (2009): 108-132.
- [14] Keskin, T. E., M. Düğenci, F. Kaçaroğlu, (2014), "Prediction of water pollution sources using artificial neural networks in the study areas of Sivas, Karabük and Bartın (Turkey)", *Environmental Earth Sciences*, DOI: 10.1007/s12665-014-3784-6.
- [15] Kurt, H, Atik, K., Özkaymak, M., Recebli, Z., (2008), "Thermal performance parameters estimation of hot box type solar cooker by using artificial neural network", *Int. J. Therm. Sci.* 47(2),192-200.
- [16] Nutini, M., & Vitali, M., (2008), "Creep modelling of Polyolefins using artificial neural networks" In *Abaqus Users' Conference, Newport, Rhode Island, USA*.
<http://www.simulia.com/forms/world/pdf2008/NUTINI-AUC2008.pdf>

- [17] Pham, D. T., A. Ghanberzadeh, E. Koc, S. Otri, S. Rahim, and M. Zaidi. "The bees algorithm - Anovel tool for complex optimisation." *Intelligent Production Machines and Systems*, 2006.
- [18] Sarangi, P.P., A. Sahu, M. Panda, (2014), "Training a Feed-Forward Neural Network Using Artificial Bee Colony with Back-Propagation Algorithm", *Intelligent Computing, Networking, and Informatics: Advances in Intelligent Systems and Computing*, 243, 511-519
- [19] Sarkar, A., Sinha, S. K., Chakravartty, J. K., and Sinha, R. K. (2013), "Artificial Neural Network Modeling of In-Reactor Axial Elongation of Zr2. 5% Nb Pressure Tubes at RAPS 4 PHWR", *Nuclear Technology*, 181(3), 459-465.
- [20] Sen S., Twomey J.M., Ahmad J.Y.S., "Development of an Artificial Neural Network Constitutive Model for Aluminum 7075 Alloy", 2002 IERC Conference, May 19, 2002.
- [21] Şenyiğit, E., M Düğenci, M.E. Aydın, M. Zeydan, (2013), "Heuristic-based neural networks for stochastic dynamic lot sizing problem", *Applied Soft Computing*, 13(3), 1332–1339.
- [22] Yigit, K.S., Ertunc, H.M., (2006), "Prediction of the air temperature and humidity at the outlet of a cooling coil using neural networks", *Int. Commun. Heat Mass*, 33, 898–907.
- [23] Yang, I.H., Yeo, M.S., Kim, K.W., (2003), "Application of artificial neural network to predict the optimal start time for heating system in building", *Energy Convers. Manage.*, 44, 2791-2809.
- [24] Yuce, B., M. S. Packianather, E. Mastrocinque, D. T. Pham, and A. Lambiase. "Honey bees inspired optimization method: the Bees Algorithm." *Insects* 4, no. 4 (2013): 646-662.
- [25] Van Erp, T. B., (2012), *Structure Development and Mechanical Performance of Polypropylene*, PhD, Technische Universiteit Eindhoven, 2012.