LAUTECH Journal of Engineering and Technology 11 (1) 2017: 1-17

APPLICATIONS OF ARTIFICIAL NEURAL NETWORKS AND GENETIC ALGORITHMS IN DRYING OF FRUITS AND VEGETABLES : A REVIEW

Oke M.O.[1], Hussein J. B.[2], Olaniyan S.A.[1]* and Adeyanju J.A. [1]
[1]Department of Food Science and Engineering, Ladoke Akintola University of Technology, Ogbomoso, Oyo State, Nigeria
[2]Department of Food Science and Technology, Modibbo Adama University of Technology, Yola, Adamawa State, Nigeria
***Corresponding Author: saolaniyan18@lautech.edu.ng**

ABSTRACT

Fruits and vegetables play an important role in the diet of human beings and economic development of a country. They are cheapest and most available sources of important proteins, vitamins, minerals and essential amino acids. Considering the perishable nature of fruits and vegetables it is necessary to preserve them and drying is one such method to do it. The drying of fruits and vegetables is a complex operation that demands much energy and time. Due to this complexity, the use of drying mathematical models in estimating the drying kinetics, the behaviour and the energy needed in the drying of fruits and vegetables becomes indispensable. Numerous mathematical models, empirical and semi-empirical, have been proposed to estimate the drying characteristics of fruits and vegetables. But these models are generally solutions of simultaneous heat and mass transfer differential equations and the final result may be very complicated and difficult to use in actual drying systems. This article present a comprehensive review on the applications of artificial neural networks and genetic algorithms in drying of fruits and vegetables. The paper starts with the drying of fruits and vegetables, the introduction of basic theoretical knowledge of ANN and GA. Then summarize their application on modeling, predicting, and optimization of heat and mass transfer, thermodynamic performance parameters, and quality indicators as well as physicochemical properties of dried fruits and vegetables. Conclusively, opportunities and limitations of ANN and GA technique in are outlined to provide more ideas for research and development in this field.

Keywords: Artificial Neural Network, Genetic Algorithm, Drying, Fruits and Vegetables
Paper type Review paper

1. Introduction

Fruits and vegetables are the fresh agri produce having high moisture content and are perishable in nature. Being a high moisture commodity most of these get degraded due to microbial spoilage. Fruits and vegetables with their rich contents of minerals, vitamins, dietary fibre and antioxidants are the protective foods and considered as nature gifts for health and wellbeing of humans. Bacterial rotting by microbial respiration as well as physiological breakdown is seen in most of the fruits and vegetables. Sometimes moisture degradation in the quality of fruits and vegetables also starts immediately after the harvest leading to drying and shrivelling. Fruits and vegetables absorb environment gasses such as oxygen and produce carbon dioxide and ethylene. They also get infested easily with microorganisms like fungi, bacteria and insects affecting food safety. Drying is the only way to reduce these losses (Visavale, 2012). Its main objective aiming to improve the shelf life of fruits and vegetables by reducing the moisture content to such extent that microorganisms cannot grow and deterioration reactions are minimized.

Fruits and vegetables are dried to enhance storage stability, minimize packaging requirement and reduce transport weight. Drying of fresh fruits and vegetables is one of the most energy intensive processes in the food industry thus, good understanding of the drying processes plays a vital role in increasing the drying efficiency as well as in maintaining product quality resulting in significant reduction in postharvest losses (Visavale, 2012). The current degree of acceptance of dehydrated fruits and vegetables in the market can be further expanded with improvements in product quality and process

applications. Therefore, effective tools are needed to recognize the most appropriate drying method for a given foodstuff. Thus, help to minimize operating and energy costs, augment manufacturing rate, control and manage drying process according to the quality of product being dried, optimize design and operating parameters of commercial food dryers, and retrofit existing drying units (Aghbashlo, Hosseinpour and Mujumdar, 2015). Also consumers demand for healthy products requires the simulation and further optimization of the drying conditions to minimize detrimental quality changes that occur during processing (Di-Scala and Crapiste, 2008).

Drying of wet materials is a complex, dynamic, unsteady, highly nonlinear, strongly interactive, successively interconnected, and multivariable thermal process whose underlying mechanisms are not yet perfectly understood (Aghbashlo et al., 2015). The complexity of the drying process becomes more serious and problematic issue due to simultaneous transient coupled momentum, heat, and mass transfers, several phase transformations, time-varying physicochemical and structural changes of the product being dried, intensive chemical and biochemical reactions, irregular component migration, and abrupt surface hardening (Aghbashlo et al., 2015). Moreover, in a typical drying process, monitoring and control of food moisture content, nutritional value, sensorial attributes, and functional components are crucial aspects, which govern the quality of the finished product. Therefore, some key factors such as operational conditions of drying system and formulation or treatment of the product which govern the quality of end-product plays a key role in drying technology to assess the product dried for consumption or further processing.

In order to successfully transfer knowledge acquired experimentally from studies on food dehydration into industrial applications, mathematical modelling of drying kinetics is required (Hussein, Filli and Oke, 2016). A mathematical model is an important tool used to optimize operating parameters and to predict performance of a drying system (Babalis and Belessiotis, 2004). Numerous mathematical models, empirical and semi-empirical, have been proposed to estimate the drying characteristics of fruits and vegetables such as kiwifruit (Maskan, 2001), 2001), vegetables (Yaldiz and Ertekin, 2001), red pepper (Akpinar, Bicer and Yildiz, 2003), pear fruit (Lahsansi, Kouhila, Mahrouz and Jaouhari, 2004), rosehip (Erenturk, Gulaboglu and Gultekin, 2004; Erenturk, Gulaboglu and Gultekin, 2005), green bean (Doymaz, 2005), citrus aurantium leaves (Mohamed et al., 2005), tarragon (Arabhosseini, Huisman, Van Boxtel and Müller, 2008), potato (Aghbashlo, Kianmehr and Arabhosseini, 2009), mint leaves (Akpinar, 2010), carrot (Berruti, Klaas, Briens and Berruti, 2009), chilli pepper (Tunde-Akintunde, 2011), and tomato slice (Bagheri, Arabhosseini, Kianmehr and Chegini, 2013; Hussein et al., 2016; Kulanthaisami et al., 2010). Most of these mathematical models have presented physical and thermal properties of the drying material as a function of time for any air temperature and air velocity.

The empirical mathematical correlations usually give very accurate results for each specific experiment. But the equation is not valid for other conditions and there is no way to obtain a general equation for a range of drying parameters. This problem may be avoided by use of analytical drying models. But these models are generally solutions of simultaneous heat and mass transfer differential equations and the final result may be very complicated and difficult to use in actual drying systems (Movagharnejad and Nikzad, 2007). Nevertheless, heuristically soft computing approaches such as artificial neural networks (ANNs) have created a new and advancing frontier in food drying to deal with its nonlinearities and complexity. ANNs are computational structures inspired mathematically by the functional behaviour of the biological nervous system of the human brain, although much of the biological detail is ignored. They have been successfully used to model complex, dynamic, highly nonlinear, and ill-defined problems in various disciplines due to their favourable features such as efficiency, generalization, and simplicity (Aghbashlo et al., 2015).

The superb information processing features of the biological system such as nonlinearity, high parallelism, robustness, and failure tolerance enable ANNs as a powerful tool for explaining nonlinear relationship between variables even with limited, incomplete, non-integrated, uncertain, noisy, dynamic, multidimensional, and nonlinear data sources. ANNs have the ability to establish relationship between input and output data by learning from examples (i.e., data) through repetition without the need for a priori knowledge of the underlying phenomenological mechanisms and mathematical background governing the behavior of the system under consideration. For these reasons, ANNs are widely applied in food drying to solve problems of nonlinear function approximation, pattern detection, data interpretation, optimization, simulation, diagnosis, control, data sorting, clustering, and noise reduction (Aghbashlo et al., 2015).

ANNs has been successfully used to describe the drying characteristics of a variety of

agricultural materials like jackfruit bulbs and leather (Bala, Ashraf, Uddin and Janjai, 2005), strawberries (Menli, Kirmaci and Usta, 2009) and grapes (Kassem, Shokr, Aboukarima and Hamed, 2010). To predict quality changes during osmo-convective drying of blueberries (Chen, Ramaswamy and Alli, 2001). To predict food quality (Ni and Gunasekaran, 1998). To model temperature and moisture content in tomato slices undergoing microwave vacuum drying (Poonnoy, Tansakul and Chinnan, 2007a) and to estimate moisture ratio of a mushroom undergoing microwave vacuum drying (Poonnoy, Tansakul and Chinnan, 2007b).

On the other hand, in the drying process, there is need to reduce processing error, food wastage, analytical cost and time and lastly to model better operating conditions that can maximize the quality of dried products. Genetic algorithm (GA) is one of the search technique used to find good solutions to optimization and search problems. They belong to a particular class of evolutionary algorithms that use techniques inspired by evolutionary biology such as inheritance, mutation, selection and crossover (Shopova and Vaklieva-Bancheva, 2006). GA offers several advantages over the conventional optimization method such as less susceptibility to be stuck at local minima, requiring little knowledge of the process being optimized and capability to find the optimum conditions when the search space is very large (Fathi, Mohebbi and Razavi, 2011a).

Nevertheless, some major limitations are attributed to individual use of ANNs and GA. The ANNs are highly sensitive to parameters (Finnie, Witting and Desharnais, 1997; Zhang, Eddy and Hu, 1998) which can have a great influence on the ANNs performance. Zhang et al. (1998), Finnie et al. (1997) and Boozarjomehry and Svrcek (2001) reported that optimized ANNs are mostly determined by labour intensive trial and error techniques which include destructive and constructive neural network design. Fischer and Leung (1998) reported that ANNs techniques only search for a limited class of models and a significant amount of computational time is, thus, required. ANNs are highly liable to over-fitting and different types of neural network which are trained and tested on the same dataset can yield different results (Chiroma et al., 2017).

While GA performance is affected by population size, parent selection, crossover rate, mutation rate, and the number of generations (Nagata and Hoong, 2003). The selection of suitable GA parameter values is through cumbersome trial and error which takes a long time (Cheng and Li, 2008) since there is no specific systematic framework for choosing the optimal values of these parameters (Longhmanian, Jamaluddin, Ahmad, Yusof and Khalid, 2012). Similar to the selection of GA parameter values, Nagata and Hoong (2003) reported that the design of an ANNs is specific to the problem domain. The most valuable way to determine the initial GA parameters is to refer to the literature with a description of a similar problem and to adopt the parameter values of that problem (Coley, 1999; Mitchell, 1998).

All these irregularities are responsible for undermining the robustness of the ANNs and GA individually (Versace, Bhatt, Hinds and Shiffer, 2004). Hybridization of the two technique are considered the most reliable and promising way to eliminates all these constraints and leads to a better solution (Chiroma et al., 2017). (David, Darrell, Larry and Eshelman, 1992) and (Nagata and Hoong, 2003) reported that the ANNs and GA are the two computational intelligence techniques presently receiving attention from computer scientists and engineers. This attention is attributed to recent advancements in understanding the nature and dynamic behavior of these techniques (Chiroma et al., 2017). Furthermore, it is realized that hybridization of these techniques can be applied to solve complex and challenging problems (David et al., 1992). Therefore, optimizing ANNs using GA is ideal because the shortcomings attributed to neural network design will then be eliminated by making it more effective than using ANNs on their own.

The objective of this review article is to provide a preliminary understanding of ANNs and Gas. Answer why and when these computational tools are needed, how they are used and their possible applications in drying of fruits and vegetables. Furthermore, opportunities and advantages of the ANN and GA technology over the other available techniques, as well as its obvious limitations and disadvantages in various fields of drying technology, were discussed. The challenges of ANNs and GA technique and the viewpoint on the current situation and future which provide more ideas for research and development in this field were also discussed.

2. Artificial Neural Networks (ANNs)

Neural networks (NNs) or artificial neural networks (ANNs) are powerful and efficient tools to model a complex process, especially to represent a nonlinear relationship which is common in food processing (Meng and Ramaswamy, 2008). It is a collection of interconnecting computational elements simulated like neurons in biological systems. The foundation of artificial neural networks in a scientific sense begins with a biological neuron. The model was developed to mimic the function of the human brain (Funes, Allouche, Beltrán and Jiménez, 2015).

A brain contains billions of nerve cells (neurons) highly interconnected through synapses. A typical biological neuron (Figure 1) contains neuronal cell bodies (soma), dendrites, and axons. The cell body or soma holds the cell nucleus, various bio-chemical factories, and other components to support ongoing activity. The dendrites are a number of fibres branch out from the cell body and split into a bushy network surrounding the cell. A single stretched fibre called the axon and branches into a string for connecting to the dendrites and cell bodies of other neuron. The end of the axon is actually branch into multiple ends, called the boutons. Each neuron receives electrochemical inputs from other neurons at the dendrites. If the sum of these electrical inputs is sufficiently powerful to activate the neuron, it transmits an electrochemical signal along the axon and passes this signal to the other neurons whose dendrites are attached at any of the axon terminals. These attached neurons may then fire. It is important to note that a neuron fires only if the total signal received at the cell body exceeds a certain level (Meng and Ramaswamy, 2008).

The entire brain is composed of these interconnected electrochemical transmitting neurons. From a very large number of extremely simple processing units (each performing a weighted sum of its inputs and then firing a binary signal if the total input exceeds a certain level), the brain manages to perform extremely complex tasks (Meng and Ramaswamy, 2008). This is the principle on which ANNs models are based. ANNs consists of a series of entries which are equivalent to the dendrites where they receive information in the form of stimulation. The weights that exist in the synapses are equivalent in biological neuron to transmit mechanisms. The union of these values (entries and weights) equals the inhibitory and excitatory chemical signals that occur in the synapses and to induce neuron to change their behaviour. These values are the gateway of the network weighting function that converts these values in potential (Funes et al., 2015).

This potential is equivalent to the total number of signals that arrive on a neuron in the biological neuron by their dendrites. The weighting function is a weighted sum of the inputs and the synaptic weights. The output of weighting function comes from the activation function that transforms this value into another form that the outputs neurons can work. The output network is evaluated in the activation function that gives rise to this neuron signal output to another neighbour. The activation function will be given by the potential of resulting that can be of different ways (Funes et al., 2015). However, it should be noted that ANNs only represents extremely simplified brain models without actually attempting to model the biological system itself (Meng and Ramaswamy, 2008).

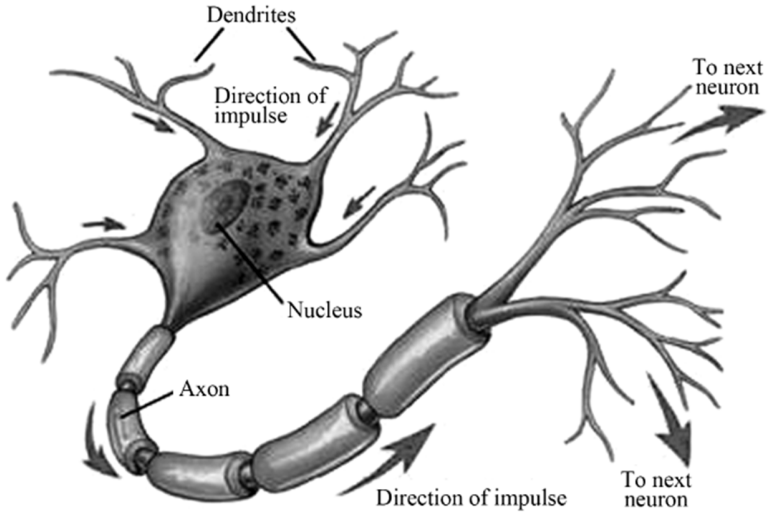

Figure 1: A Typical Biological Neuron
Source: Funes et al. (2015).

2.1. Neural Network Architecture

The artificial neuron is an element of simple processing that produces a single output from an

inputs vector. Funes et al. (2015) reported that when talking about architecture of an ANNs, must be taken into account the number of level or layers. The distribution of neurons in the neural network is done forming levels or layers of a determined number of nodes to each one. Neurons are arranged in layers; there is at least one input layer, one hidden layer, and one output layer in any ANNs. Each neuron in a layer is linked to neurons in other layers with varying connection weights (W) that represent the strengths of these connections.

The connection patterns depending on the links between the elements of the different layers. The ANNs can be classified as totally connected when all the outputs from a level get to all and each one of the nodes in the following level. In this case, there will be more connections than nodes. When some links in the network are lost, the network is partially connected (Funes et al., 2015). An ANNs can be viewed as a 'black box'. Input layer receives the information from an external source and passes this information to the network for processing. Hidden layer receives information from the input layer and processes the information. The entire process is hidden from view. Output layer receives processed information from the network and sends the results out to an external receptor. When the input layer receives information from an external source, it becomes ''activated'' and emits signals to the neurons in the hidden layer (Meng and Ramaswamy, 2008).

In general, the ANNs model can be classified into feed-forward neural network and feed-back neural network based on the type of neural network architecture and learning with teacher (supervision) style and learning without teacher (no supervision) style based on the learning style (Aghbashlo et al., 2015; Sun, Zhang and Mujumdar, 2018). Feed forward network refers to an ANN with unidirectional flow and information processing in which connections among the neurons do not form a directed cycle and is permitted only to obtain information from the previous neuron (Sun et al., 2018). It is organized in three or more layers, an input layer, an output layer, and one or more hidden layers. From the input layer to the output layer, the network is one-way connection. Only the two neurons in adjacent layers connect each other. There is no connection between the neurons at the same level and connections between the neurons do not form a directed cycle. So the received signals from the upper layer are only sent to the next layer of neurons and there is no feedback between the neurons (Fuangkhon, 2017). However, feedback network is an ANN with bidirectional information-processing procedure where each node obtains information from the previous and permits the feedback to the next layers (Sun et al., 2018). The output neurons have at least one feedback loop, and the signal can flow forward or reverse.

In term of learning style, we have supervised (learning with teacher) reinforced, and unsupervised (learning without teacher) modes. Learning is a process of updating the weights and bias levels of a network to produce a desired response to a specific input. In a supervised training mode, learning data set is provided with many pairs of input/output training patterns. The network connection intensity is constantly adjusted by the error between the desired output and the actual output until the satisfactory input-output relationship is reached. Under the guidance of teacher, learning neural network can adapt to the changes in the environment, but it is easy to forget the knowledge they have learned while learning new knowledge (Sun et al., 2018). In the reinforced learning, there are no targets given to regulate the weights. However, the algorithm is given a grade of the ANNs performance. In the unsupervised mode, training data set composes of input training patterns only without outside help to cluster different input patterns into different categories. The neural network of unsupervised learning automatically adjusts the weight of the connection according to the input data and classifies the data with similar features according to the statistics rule in the training process (Sun et al., 2018).

2.2. Artificial Neural Network Modeling Approach

There are many different kinds of ANNs approaches proposed for modeling various scientific and engineering problems. However, this section will cover the main ANNs approaches that have been employed in food drying researches. The following are the type of ANNs modeling approaches;

2.2.1. Single-layer feed forward

The simplest kind of layered networks is the single layer perceptron ANNs containing a single layer of output nodes. The inputs are directly transmitted to the outputs through a series of weights (Aghbashloa and Hosseinpoura, 2016). The output nodes use transfer functions to produce the outputs as shown in Figure 2.

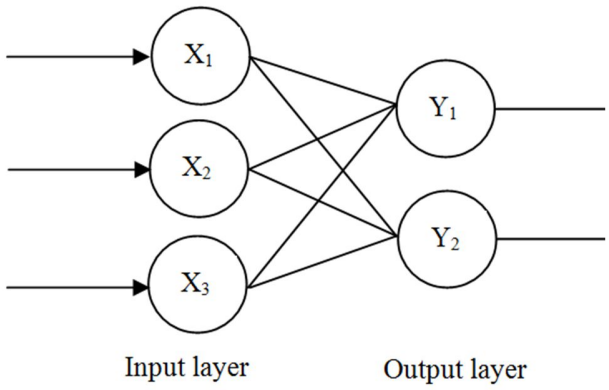

Figure 2: Single Layer Feed Forward Network
Source: Aghbashloa and Hosseinpoura (2016).

2.2.2. Multi-layer perceptron (MLP)

The MLP network is a typical feed forward networks which is mostly useful and common neural network architectures. It is appropriate for a variety of applications such as prediction and process modeling (Tohidi, Sadeghi, Mousavi and Mireei, 2012). A schematic illustration of this structure is shown in Figure 3. It has an input, an output, and one or more hidden layer(s). The units in each layer are connected to the units in the subsequent layer, so that the outputs of one layer are regarded as inputs to the next layer without connections between nodes in the same layer. The input nodes receive signals from the user. The first hidden layer receives signals through the connections from input layer. The output signal from the first hidden layer feed into the second hidden layer and so on. Finally, the signals are fed into the output layer to produce the desired output. The numbers of input and output nodes are determined by dimensions of input and output data, so that only the number of hidden layers and nodes is to be decided by users (Aghbashloa and Hosseinpoura, 2016).

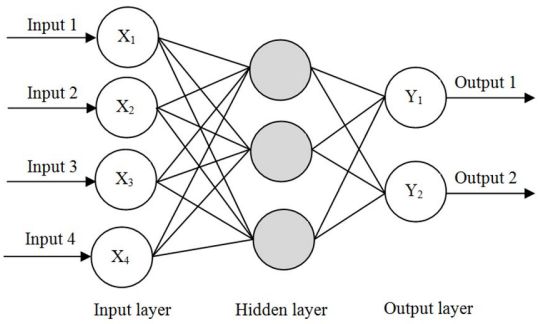

Figure 3: MLP basic topology
Source: Aghbashloa and Hosseinpoura (2016).

2.2.3. Back propagation network (BPN)

Back Propagation Network has been extensively studied, theoretically, and has been the most successful. The BPN is usually built from a three layered system consisting of input, hidden, and output layers (Figure 4). An equation in the hidden layers (transfer function) determines whether inputs are sufficient to produce an output (Huang, Kangas and Rasco, 2007). The BPN computation is derived using the chain rule of calculus until it can approximate function. There are several kinds of transfer functions, e.g. threshold or sigmoid functions. In training an ANN, the values predicted by the network are compared to experimental values using the delta rule, an equation which minimizes error between experimental values and network predicted values. The errors are then back propagated to hidden and input layers to adjust weights. This is repeated many times until errors between predicted and experimental values are minimized (Eberhart and Dobbins, 1990).

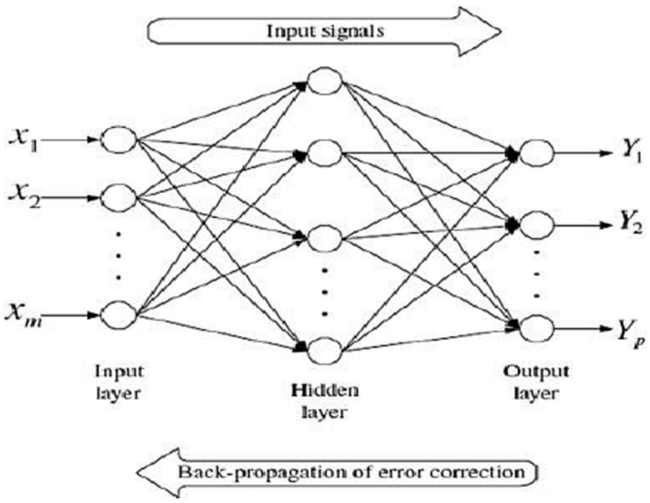

Figure 4: Back Propagation Network

2.2.4. Recurrent ANNs

This topology contains both feed forward and feedback connections between layers and nodes. Accordingly, the inputs to the nodes come from external input as well as from the internal nodes (Figure 5). This type of network can be found in single and multiple layer(s). It could be effectively applied for modeling, identification, and control of highly nonlinear dynamic systems due to the presence of feedback loops. Fully recurrent (Hopfield network and Boltzmann machine), simple recurrent, echo state, long short term memory, bi-directional, hierarchical, and stochastic neural networks are

different types of recurrent ANN topologies (Aghbashloa and Hosseinpoura, 2016).

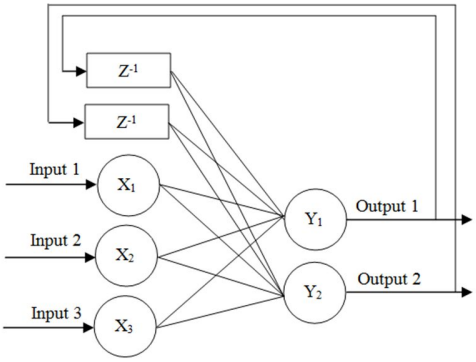

Figure 5: Typical Recurrent Network Structure

2.2.5. General regression neural network (GRNN)

General Regression Neural Network are memory based feed forward networks meaning that all the training samples are stored in the network. It possess a special property that they do not require iterative training. It provides estimates of continuous variables and converges to the underlying (linear or nonlinear) regression surface (Specht, 1991). This GRNN is a one-pass learning algorithm with a highly parallel structure. Specht (1991) reported that even with sparse data in a multidimensional measurement space, the algorithm provides smooth transitions from one observed value to another. The algorithmic form can be used for any regression problem in which an assumption of linearity is not justified.

2.2.6. Radial basis function network (RBF)

A radial basis function network is an artificial neural network that uses radial basis functions as activation functions. The output of the network is a linear combination of radial basis functions of the inputs and neuron parameters. Radial basis function networks have many uses, including function approximation, time series prediction, classification, and system control.

2.2.7. Self-organizing map (SOM)

The Self-Organizing Map is a neural network models that belongs to the category of competitive learning networks. The SOM is based on unsupervised learning, which means that no human intervention is needed during the learning and that little needs to be known about the characteristics of the input data (Firenze, Ricciardiello and Pagliano, 1994). We could, for example, use the SOM for clustering data without knowing the class memberships of the input data. The SOM can be used to detect features inherent to the problem and thus has also been called SOFM, the Self-Organizing Feature Map (Firenze et al., 1994).

2.3. Development of Artificial Neural Networks

In developing an ANNs models, the inputs and outputs are determined by the problem being investigated. ANNs can be used to simultaneously produce more than one output unlike traditional models where one model is required for each output (Meng and Ramaswamy, 2008). It has the capability of relating the input and output parameters without any prior knowledge of the relationship between them. Due to these reasons, ANNs are promising alternatives over mathematical, statistical, numerical, and analytical techniques for dealing with the nonlinearities and complexity of ill-defined food drying processes using past historical data without any prior regard about the nature of the relationships. Moreover, ANNs models can embed more than two input parameters to generate all desired outputs concurrently, which make it suitable for multivariable drying process (Aghbashloa and Hosseinpoura, 2016).

The optimization of ANNs configurations now depends on the determination of number of hidden layers, number of neurons in hidden layers, transfer function and learning rule, learning rate and learning runs (Meng and Ramaswamy, 2008). Trial and error is normally used to select those parameters. Meng and Ramaswamy (2008) reported that when selecting the number of hidden layers, number of neurons in hidden layers and learning runs, overtraining must be avoided. Generally the more the hidden layers and the neurons in hidden layers, the better the ANNs will perform (Nguyen and Cripps, 2001). Because with a large number of hidden layers and neurons, ANNs may memorize the input training samples (Rai, Majumdar, DasGupta and De, 2005) implying that the ANN is over trained. As a result, it is less likely to accurately predict new data, and its generalization ability is weakened.

Too many learning runs will possibly result in overtraining as well (Meng and Ramaswamy, 2008). Thus, it is always a good practice to keep the neural network as simple as possible to maintain its generalization potential. In the training or learning phase, a set of known input/output patterns is repeatedly presented to train the network. The weight factors between neurons are adjusted until the specified input yields the desired output. Through these adjustments, the ANNs learns the correct input/output response behaviour (Meng and Ramaswamy, 2008). This stage in the development of ANNs is typically the longest and most time consuming, and it is critical to the success of the

network. A well-trained neural network should have the capability to respond to previously unseen data sets as well, which is refereed as generalization (Meng and Ramaswamy, 2008). Therefore, after training, the testing of the trained network will proceed to assess its generalization ability.

All the available data sets should be divided into three categories for the training, validation, and testing the developed ANNs. The trained network will be subjected to input patterns not used in training, but whose outputs are known, and the network's performance will be evaluated. Goodness of fit of the developed ANNs to the experimental data can be evaluated using several statistical parameters. Linear correlation coefficient (R^2), root mean square (RSM), standard error of estimate (SSE), percentage relative error (PRE), mean absolute error (MAE), and mean absolute percentage error (MAPE) are the often used statistical index to evaluate the performance of ANNs (Sablani, Ramaswamy and Prasher, 1997).

2.4. Advantages of Artificial Neural Networks

Artificial Neural Networks have a number of advantages over other modeling techniques. It was summarized below by (Meng and Ramaswamy, 2008):

(i) Learning ability: A prior knowledge of the system is not required to construct a neural NN because it can learn the input/output relationship from the given data. This makes NN suitable to problems where the relationships are dynamic or nonlinear, which is difficult for other modeling techniques. Also due to ANN dynamic adaptive systems nature, they can adjust themselves and change depending on the new conditions that appear. In fact, a neural network can generate its own weight distribution by means of learning and even after this continue learning (Funes et al., 2015).

(ii) Self-organization: The neural networks employ their ability to adaptive learning for to autoorganize the information they receive during learning and/or the operation. In this way, neural networks are able to give an answer to a new situation, solve problems in which the input information is not very clear or incomplete (Funes et al., 2015).

(iii) Fault tolerance: The neural network is more tolerant of noisy and incomplete data because the information is distributed in the massive processing neurons and connections. These connections will have their values according to received stimuli, and an output pattern that represents the information stored that will generate. Partial destruction of the network will not degrade the overall performance significantly.

(iv) High computational speed: Because the NN is an inherently parallel architecture, the result comes from the collective behaviour of a large number of simple parallel processing units, which makes it suitable for the on-site modeling/controlling.

(v) Operation in real time: Neural computations can be performed in parallel. Most of the networks can operate in a real-time environment, the need for change in weights connections or training is minimal.

(vi) Easy insertion into existing technology: A network can be rapidly trained, tested, verified and transferred to a hardware implementation (Popescu, Popescu, Wilder and Karwe, 2001).

2.5. Limitations of Artificial Neural Networks

While ANNs have many advantages, they still have some limitations as follows (Meng and Ramaswamy, 2008):

(i) Requirement of large amount of training data: ANNs should not be considered to model a problem or process only with little training data since they rely heavily on such data. In addition, ANNs are not suitable in the situation where large but similar training data exist, which will cause the same problems as small training data sets. Thus, broad-based data sets are essential for training a neural network.

(ii) No guarantee of optimal results: Most training techniques are capable of ''tuning'' the network, but they do not guarantee that the network will operate properly. The training may ''bias'' the network making it accurate in some operating regions but inaccurate in others.

(iii) Requirement of good set of input variables: Selection of input variables that give the proper input/output mapping is often difficult but required for a good performing neural network. Since it is not always obvious which input variables will give the best result, some trial and error in selecting input variables is often required.

(iv) Black box instead of clear physical relationship: The individual relations between the input variables and the output variables are not developed by engineering judgment so that the model tends to be a black box or input/output table without analytical basis.

2.6. Genetic Algorithm (GA)

Genetic algorithm is one of the search technique used to find good solutions to optimization and search problems. They belong to a particular class of evolutionary algorithms that use techniques inspired by evolutionary biology such as inheritance, mutation, selection and crossover (Shopova and Vaklieva-Bancheva, 2006). The natural evolution theory states that a species will, after many generations, adapt to live better in its environment (Meng and Ramaswamy, 2008). For example, if a population of an animal lives mainly in a swampy area, they may eventually evolve with webbed feet. The reason is that the members of this population will die if they are poor swimmers because they cannot easily get food and live long enough to reproduce. The offsprings of the good swimmers will probably be good swimmers as well because they will usually carry genetic traits of their parents such as slight webbing between the toes (Meng and Ramaswamy, 2008).

GA solve an optimization problem in the same way as nature solves problems in evolution. They start with generating a set of possible solutions to the problem. Each candidate solution is called a chromosome, and the whole set of solutions is called a population. Each solution (chromosome) could consist of several decision variables (Meng and Ramaswamy, 2008). The problem to solve, usually a set of parameters to find, is coded as a vector, named an individual. The selection criterion is based on the fitness function. The fitness function plays the role of the environment to distinguish between good and bad solutions (Shopova and Vaklieva-Bancheva, 2006). The GA creates an initial population (a set of initial solutions). This population will change due to the reproduction of individuals in the successive iterations. Occasionally, with some low probability, mutations will make changes in some individuals. The best individuals are selected so that each population will be better than the previous one. For the GA setting, some configurations must be determined, such as the population size, selection, reproduction and mutation methods, probability of mutation, stopping criteria, etc. (Erenturk and Erenturk, 2007; Shopova and Vaklieva-Bancheva, 2006).

2.6.1. Principles of genetic algorithms

2.6.1.1. Encoding of chromosomes

Encoding of chromosomes is to transform the decision variables into a format that can be operated by GAs. Encoding is very important, which determines how well the algorithm performs on the problem. There are two main coding ways (Morimoto, 2006). The most traditional approach is to use a set of binary strings consisting of 0 and 1 to represent decision variables. Binary strings are commonly used since they are known to perform well with standard crossover and mutation operators (Meng and Ramaswamy, 2008). Each variable has one binary string. Each bit in this string can represent some characteristic of the solution, which is called a gene (Morimoto, 2006). The other coding way is to use integral numbers which might be more effective when the chromosome consists of many variables. There are many other ways of encoding depending on the problem. For example, one can encode directly real values in finding weights for an ANN (Meng and Ramaswamy, 2008).

2.6.1.2. Objective function

An objective function relates the decision variables of the problem and assigns a 'fitness' value to the solution (chromosome) that determines how good that solution is. Ideally this function will be as monotonic as possible (Keedwell and Narayanan, 2005), and it will vary consistently with decision variable values. The more adapted chromosomes will receive higher fitness values (Meng and Ramaswamy, 2008). Chromosomes with high fitness values are more likely to reproduce in each generation during the evolution process. When the GA is used for minimization, a transformation is necessary to derive a maximization problem (Meng and Ramaswamy, 2008).

2.6.1.3. Genetic operators

GAs are started with generating an initial population of chromosomes. This initialization is often achieved at random, but the population may be initialized by chromosomes which are already known to perform well (Meng and Ramaswamy, 2008). When random initialization of binary chromosomes is used, each bit of the chromosomes is randomly set to 0 or 1 according to a probability which is called the initialization probability. Once the first generation has been created, the genetic operators drive the population to find new, more optimal solutions (Meng and Ramaswamy, 2008). Selection, mutation, and crossover are the most important operators.

2.6.1.4. Selection

Selection is the process of picking out a suitable chromosome from the population in order to create a new generation. According to Darwin's evolution theory, suitable chromosomes are the ones with good fitness values. This operator is the implementation of the principle 'survival of the fittest' (Meng and Ramaswamy, 2008). However, to make sure that the GA does not converge on a set of solutions too quickly, a random element is usually introduced into the selection procedure (Keedwell

and Narayanan, 2005). There are many selection strategies.

i. Roulette wheel selection (proportional selection)

Roulette wheel selection is the most simple and fundamental selection approach. In this method, the probability for a chromosome to be selected is in proportion to its fitness value (Meng and Ramaswamy, 2008). The principle is just like a roulette wheel where each chromosome in the population occupies a slice of the wheel, the higher the fitness value, the larger the portion of wheel occupied by that chromosome. Therefore, the better the chromosomes are, the more chances they have of being selected (Meng and Ramaswamy, 2008). Roulette wheel selection gives more adapted chromosomes better chances to be kept in the next generation. However, there is still a chance for less adapted chromosomes to be selected since the selection procedure depends on a random number which will ensure that the diversity in the population is maintained (Meng and Ramaswamy, 2008).

ii. Tournament selection

In tournament selection, a number of chromosomes (normally 2) are randomly selected from the population and their fitness values are compared. The chromosome with the highest fitness is selected as a parent to generate the next generation (Meng and Ramaswamy, 2008). The random selection to the tournament gives a chance that two chromosomes with lower fitness values could be selected at once. In this situation, although the chromosomes are poor with respect to the rest of the population, the better chromosome of the two will be selected. Therefore, chromosomes with low fitness values can still be selected by the tournament selection which ensures that the diversity in the population could be kept (Meng and Ramaswamy, 2008).

iii. Elitist selection

In elitist selection, the best chromosome (or a few best chromosomes) is copied to new generation. The rest of the chromosomes in the new generation are created in a classical way. Since creating new population by crossover and mutation is easy to lose the best chromosome, elitist selection prevents losing the best found chromosomes (Meng and Ramaswamy, 2008).

2.6.1.5. Crossover

Crossover is the most important genetic operator. Crossover combines the information from two 'parent' to produce two new 'offspring' solutions which are different but related to the original solutions (Meng and Ramaswamy, 2008). Crossover is performed in the hope that the combination of two well-adapted chromosomes may give two new better adapted ones. There are a number of methods to achieve this purpose, and the most commonly used are single point, multipoint, and uniform crossover (Meng and Ramaswamy, 2008).

i. Single point crossover

In single point crossover, two chromosomes are involved and one crossover point is selected randomly. The two new offspring are created in such a way that the binary string from the beginning of the chromosome to the crossover point is copied from one parent, and the rest is copied from the second parent (Meng and Ramaswamy, 2008).

ii. Multipoint crossover

For multipoint crossover, n crossover positions are chosen randomly and sorted into ascending order. Then, the bits between successive crossover points are exchanged between the two parents to produce two new offspring (Meng and Ramaswamy, 2008).

iii. Uniform crossover

In uniform crossover, for each bit of the first offspring, a probability is calculated separately (mostly p = 0.5) that this bit should come from either the first parent or the second parent, and for the second offspring, the bit in the same position will take the value of the other parent (Meng and Ramaswamy, 2008). Uniform crossover combines the feature in each bit no matter where the bit is located, but in single point crossover, the bits towards the center of the chromosome are perturbed more often than those at the edges of the chromosome (Keedwell and Narayanan, 2005).

2.6.1.6. Mutation

After crossover, every offspring undergoes mutation. In binary chromosomes, mutation inverts one or more bits at random from 0 to 1 or vice versa. Mutation helps to keep the genetic diversity from one generation to the next and prevents premature convergence to a local optimum solution (Morimoto, 2006) by preventing the population of chromosomes from becoming too similar to each other. Without mutation, GA can only manipulate the genetic material that is present in parent population (Meng and Ramaswamy, 2008).

2.6.2. Parameters of genetic algorithms

There are two basic parameters in GAs: crossover probability and mutation probability (Meng and Ramaswamy, 2008). Crossover probability tells how often crossover will be performed. If crossover probability is 100%, then all offspring is made by crossover. If it is 0%, whole new generation is made from exact copies of chromosomes from old population. Normally, crossover probability is around 0.6 – 0.8 (Meng and Ramaswamy, 2008). Mutation probability tells how often mutation will be

conducted. If there is no mutation, offspring is taken after crossover without any change. But if mutation probability is 100%, whole chromosome is changed (Meng and Ramaswamy, 2008). Mutation probability is usually very small (0.01– 0.1). Some studies (Back, 1997; Mayer, Belward and Burrage, 1999) used high mutation probability (up to 0.4 or 0.6).

There are two other very important parameters GAs: population size and generation number (Meng and Ramaswamy, 2008). Population size tells the amount of chromosomes in a population. If there are too few chromosomes, genetic algorithm only has a small part of search space to explore. However, if there are too many chromosomes, GA will take more time to solve the problem (Meng and Ramaswamy, 2008). Research shows that after some limit (which depends mainly on encoding and the problem) it is not useful to increase population size (Meng and Ramaswamy, 2008). Maximum generation number must be pre-specified as a termination condition of GAs. If the algorithm reaches the maximum generation number, the evolution ends (Meng and Ramaswamy, 2008).

2.6.3. Types of genetic algorithm

There are two types of GAs, using different ways to carry chromosomes from one generation to the next. Generational genetic algorithms completely replace the chromosomes between generations (Meng and Ramaswamy, 2008). From the initial population generated, only two chromosomes are selected as parents for reproduction. The two parents create two new offspring by crossover and mutation. This process (selection, crossover, and mutation) is repeated until the number of offspring is the same as that in the parental generation. The produced offspring are then inserted into the population to replace their parents to form a new generation. The new generation will be the new parental generation for the next offspring generation. If the new generation contains a solution that produces an output that is close enough or equal to the desired answer then the problem has been solved. If this is not the case, then the new generation will go through the same process as their parents did. This will continue until a solution or the maximum number of generation is reached (Meng and Ramaswamy, 2008).

Steady-state genetic algorithms just partially replace the chromosomes between generations. From the initial population generated, two chromosomes are selected to create two new offspring by crossover and mutation replacing the two less adapted chromosomes, which have the two lowest fitness values in the population, by these two new chromosomes to form the new generation. The algorithm stops when the maximum number of generations is reached, or the optimum solution is found (Meng and Ramaswamy, 2008).

2.6.4. GA-ANNs optimization model

Once an ANN based model with good prediction accuracy is fitted, GA could be used to optimize its input space representing the medium. The block diagram of GA and ANN actions is shown in Figure 6. The developed ANN-GA approach consists of two parts: an ANN prediction and a GA part. First, an initial population is randomly generated, which contains a number of sets of initial process parameters. Then the strings stored in it are individually fed into an ANN-based prediction unit for the quality prediction of moulded parts (Abdalla, Elfaki and AlMurtadha, 2014). In a genetic algorithm, a population of strings (called chromosomes or the genotype of the genome), which encode candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem, evolves toward better solutions. The evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness), and modified (recombined and possibly randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm.

Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. If the algorithm has terminated due to a maximum number of generations, a satisfactory solution may or may not have been reached (Haupt and Haupt, 2004; Javadikia, Rafiee, Garavand and Keyhani, 2011). This cycle is repeated until desired convergence on optimal or near-optimal of the solutions are achieved. Abdalla et al. (2014) reported that the developed hybrid ANN-GA system have significantly reduced the time required to generate initial process parameters for injection moulding in comparison to the mould flow simulation for injection moulding, where it takes less than 2 minutes including the time for user input to obtain a set of initial process parameters corresponding to an input problem.

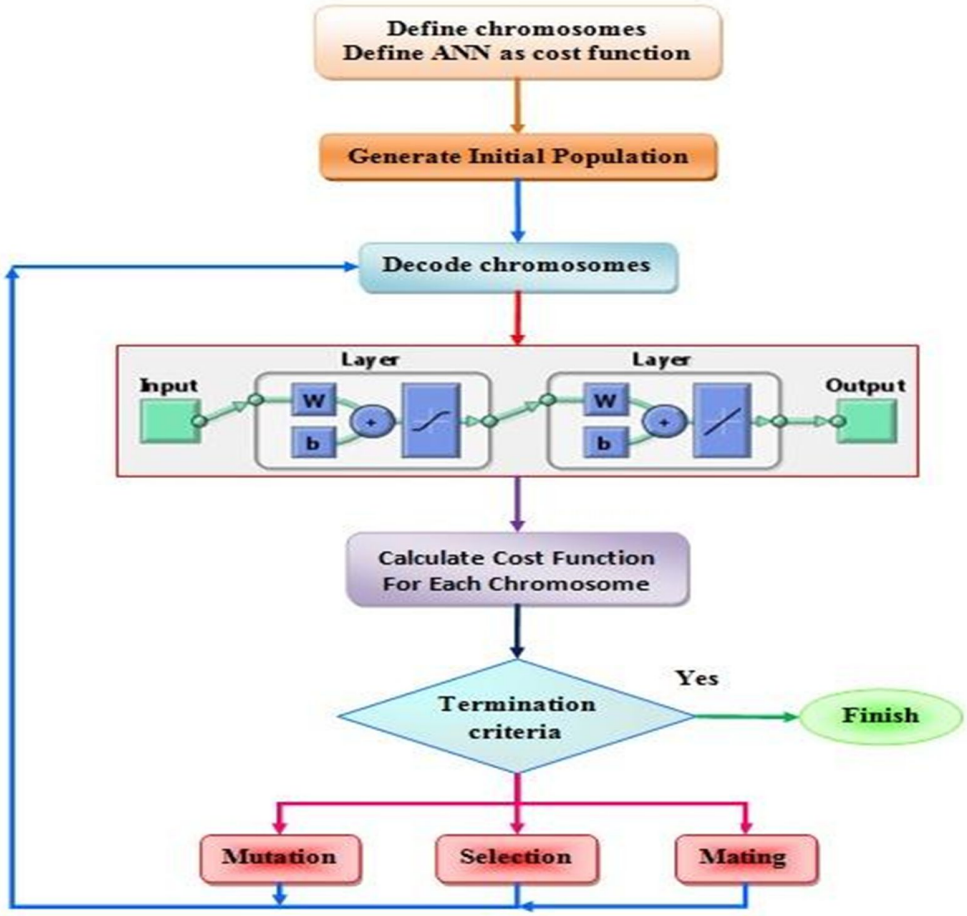

Figure 6: Block diagram of GA and ANN
Source: Javadikia et al. (2011).

2.6.5. Advantages of genetic algorithms over conventional optimization methods

Compared with conventional optimization methods, GA offers several advantages over the conventional optimization method such as less susceptibility to be stuck at local minima, requiring little knowledge of the process being optimized and capability to find the optimum conditions when the search space is very large (Fathi et al., 2011a). GAs search from a population of points not a single point. In many traditional optimization methods, we move gingerly from a single point in the decision space to the next using some transition rule to determine the next point. This point-to-point method is dangerous because it is a perfect prescription for locating false peaks in multimodal (many-peaked) search spaces (Meng and Ramaswamy, 2008). By contrast, GAs work from a rich database of points simultaneously (a population of strings) climbing many peaks in parallel, and the probability of finding a false peak is reduced over the methods going point to point (Meng and Ramaswamy, 2008).

GAs use payoff (objective function) information, not derivatives or other auxiliary knowledge. Many search techniques require much auxiliary information in order to work properly. For example, gradient techniques need derivatives (calculated analytically or numerically) in order to be able to climb the current peak and other local search procedures like the greedy techniques of combinatorial optimization (Goldberg, 1989) require access to most if not all tabular parameters. By contrast, GAs have no need for all this auxiliary information. To perform an effective search for better and better structures, they only require payoff values (objective function values) associated with individual strings. This characteristic makes the genetic algorithm a more canonical method than many search schemes (Meng and Ramaswamy, 2008).

Unlike many methods, GAs use probabilistic transition rules, not deterministic rules, to guide their search. The use of probability does not suggest that the method is some simple random search; it uses random choice as a tool to guide a search toward

regions of the search space with likely improvement. This makes GAs more robust than other traditional optimization methods (Meng and Ramaswamy, 2008).

2.7. Application of ANNs and GA in Drying of Fruits and Vegetables

Applications of ANNs in food processing have covered different areas: modeling, prediction, computation, classification, and optimization as well as process control. If the number of papers published were used to determine the order of relative popularity of application purposes, modeling or prediction was the most popular purpose for applications of ANNs in food processing areas; control was listed second; and then classification, optimization, and computation, respectively (Chen et al., 2001). ANNs have recently been applied in different fields of food dehydration. A comprehensive review paper which contains the numerous significant applications of ANN paradigm in drying technology was published by Aghbashlo et al. (2015). This shows that ANNs were found to be mostly used for modeling or prediction in food processing area. Because like other traditional modelling methods, ANNs cannot provide direct answers for optimization problems. In order to be used for optimization purposes, neural network models have to be combined with one of search techniques (Di-Scala et al., 2012). Genetic algorithms are a combinatorial optimization technique, searching for an optimal value of a complex objective function by simulation of the biological evolutionary process, based on crossover and mutation as in genetics. It has been found that the combination of ANN and GA models can become effective tools for optimization problems. Chen and Ramaswamy (2002) were the first to report on application of ANN and GA for the thermal processing optimization. ANN models were developed for predicting process time (PT), average quality retention (Qv), surface cook value (Fs), equivalent unit energy consumption (En), temperature difference (g) and ratio of F value from heating to total desired F value (ρ) under different processing conditions. These were then coupled with GA to search for the optimal quality retention and the corresponding retort temperature. The combined ANN-GA models were then used for investigating the effects of process variables on both optimal quality retention and retort temperature.

Coupling GAs to ANNs has also been investigated in many publications. Goni, Oddone, Segura, Mascheroni and Salvadori (2008) applied genetic algorithm search technique to obtain the initial training parameters of the neural network which improved its generalization capacity. Liu, Chen, Wu and Peng (2007) used GAs to determine the optimal number of neurons in the hidden layer of the NN. Morimoto, Purwanto, Suzuki and Hashimoto (1997) used a neural network to predict color change of tomatoes affected by heat treatment, and used a GA to search for the optimal heat treatment condition. Erenturk and Erenturk (2007) compared genetic algorithm and neural network approaches for the drying process of carrot. They reported that neural network approaches with (correlation coefficient = 0.999, chi-square = 1.89×10^{-5}, and sum square error = 2.039×10^{-3}) had the best agreement with experimental data. In addition, it was found that GA-ANN technique with 1 hidden layer including 7 neurons (topology of 3-7-1), predicts the nearest data to the experimental data (R^2 = 0.9936 and RMSE = 0.0220). More sensitivity of MR to slice thickness observed in this study demonstrates that this factor plays more significant role in hot-air drying process of papaw (Yousefi, 2017). The high performance of GA-ANN based model even reported in the case of prediction of amount of glucose release during *in vitro* gastrointestinal digestion of native and chemically modified starches (r = 0.984 – 0.993 and $RMSE$ = 0.338 – 0.588) (Yousefi and Razavi, 2017).

Fathi, Mohebbi and Razavi (2011b) used an intelligent system ANN and genetic algorithm (GA) for predicting shrinkage level of dried kiwi fruits based on Fractal theory. The image acquisitions were performed on different dried samples at a corresponding temperature and ImagJ software was used to determine fractal parameters of the fruit samples based on box counting method. Using as input parameters the moisture content (MC) with background interface line and level of samples shrinkage as an output of genetic algorithm, an optimization was achieved at seventeen neurons value and correlation coefficient (R^2) of 0.95. Mohebbi, Shahidi, Fathi, Ehtiati and Noshad (2011) proposed ANN and genetic algorithm (GA) model that can forecast moisture content (MC) of banana dried by means of ultrasound rays and osmotic dehydration. There are one output MC and 6- input parameters namely pretreatment, type of sugar, solution or osmotic concentration, drying temperature, drying time and pre-treatment time to the network model. The best performance using GA illustrate that the most sensitive input parameters are drying temperature and time with R-value of 0.94.

Fathi et al. (2011a) successfully predicted the physicochemical characteristics of the osmotically dehydrated kiwifruit by sucrose solution and finished with a convective dryer using a simple MLP ANN model optimized by GA. In another work, they adequately predicted the mass transfer kinetics

and total color difference of kiwifruit slices during osmotic dehydration using a simple ANN model. Similarly, the shrinkage of osmotically dehydrated and air-dried kiwifruit was predicted as a function of fractal dimension and moisture content using an MLP ANN topology optimized by GA (Fathi, Mohebbi and Razavi, 2011c). Mohebbi et al. (2011) successfully applied GA to optimize an MLP ANN topology for estimating the convective drying kinetics of the pre-osmosed and ultrasound-treated banana slices as a function of pretreatment and hot air drying parameters. In another work, Javadikia et al. (2011) the moisture content of thin layer drying of tomato is modeled via FFNN and GA used to optimized the FFNN to get best result, in other worlds, we could create the genetic-neural network to modeling it. The best model had only one hidden layer with 11 neurons and it had MSE, MSNE and MAE of 0.00006386, 0.00024223 and 0.005797 respectively. It had correlation coefficient of 0.999466 with significant of 0. This result is very excellent to modeling the drying of tomato. The GA simplified the structure of the MLP ANN model and reduced the computation time.

2.8. Future Research on Application of ANNs and GA in Drying of Fruits and Vegetables

Notwithstanding the ability of GA to extract rules from the ANN and enhance its interpretability, it is evident that research in that direction has not been given adequate attention in the literature. Very few reports on the application of GA to extract rules from ANNs have been encountered in the course of this review. More research is, therefore, required in this area of interpretability in order to finally eliminate the 'black box' nature of the NN.

The GA has been applied to the ANN model so as to get optimal ANN parameters of design and training. Neural network and genetic algorithm MATLAB toolboxes have been used to obtain the results which proofed that the new model can optimize ANN parameters precisely and effectively. As a future research direction, the GA should be applied to determine optimum ANN parameters within a wide search space i.e. find optimal ANN architecture from three hidden layers, optimal minimization algorithm from six different types, and study the impact of search space on training time, performance, and complexity.

3. Conclusions

Based on the literature review, the GA-ANNs had been a precise and appropriate instrument for modeling of drying process. GA is a significantly efficient method for optimizing the most important parameters of neural network structures that have significant influence on performance efficiency of ANNs such as hidden layers number, the processing elements number (PE), the learning rates and the momentum coefficients. The ANNs were trained using GA for determining network topology (neuron number of hidden layers, momentum and step size) in less time with acceptable performance. Further, for deducing prediction errors of ANN, GA optimize inputs with deleting negligible inputs in modeling outputs. The optimised ANNs-GA can potentially predict outputs with credible performance.

REFERENCES

Abdalla, O.A., Elfaki, A.O. and AlMurtadha, Y.M. (2014). Optimizing the Multilayer Feed-Forward Artificial Neural Networks Architecture and Training Parameters using Genetic Algorithm. *International Journal of Computer Applications, 96*(10), 42-48.

Aghbashlo, M., Hosseinpour, S. and Mujumdar, A.S. (2015). Application of Artificial Neural Networks (ANNs) in Drying Technology: A Comprehensive Review. *Drying Technology: An International Journal, 33*(12), 1397-1462. doi:10.1080/07373937.2015.1036288

Aghbashlo, M., Kianmehr, M. H. and Arabhosseini, A. (2009). Modeling of Thin-Layer Drying of Potato Slices in Length of Continuous Band Dryer. *Energy Conversion and Management, 50*(5), 1348 – 1355.

Aghbashloa, M. and Hosseinpoura, S. (2016). ANN Principles and Applications in Modeling and Control of Food Dryers. In P.K. Nema, B.P. Kaur, and A.S. Mujumdar (Eds.), *Drying Technologies for Foods : Fundamentals and Applications, Part II*. New Delhi: New India Publishing Agency

Akpinar, E. K. (2010). Drying of Mint Leaves in a Solar Dryer and Under Open Sun: Modelling, Performance Analyses. *Energy Conversion and Management, 51*(12), 2407 – 2418.

Akpinar, E. K., Bicer, Y. and Yildiz, C. (2003). Thin Layer Drying of Red Pepper. *Journal of Food Engineering, 59*(1), 99-104.

Arabhosseini, A., Huisman, W., Van Boxtel, A. and Müller, J. (2008). Modeling of Thin Layer Drying of Tarragon (Artemisia dracunculus L.). *Industrial Crops and Products, 28*(2), 53-59.

Babalis, S. and Belessiotis, V. (2004). Influence of Drying Conditions on the Drying Constants and Moisture Diffusivity during the Thin-Layer Drying of Figs. *Journal of Food Engineering, 65*(3), 449 – 458.

Back, T. (1997). *Mutation Parameters, Handbook of Evolutionary Computation* New York: Oxford University Press

Bagheri, H., Arabhosseini, A., Kianmehr, M. H. and Chegini, G. R. (2013). Mathematical Modeling of Thin Layer Solar Drying of Tomato Slices. *Agricultural Engineering International: CIGR Journal, 15*(1), 146 – 153.

Bala, B.K., Ashraf, M.A., Uddin, M.A. and Janjai, S. (2005). Experimental and Neural Network Prediction of the Performance of a Solar Tunnel Drier for Drying Jackfruit Bulbs and Leather. *Journal of Food Process Engineering, 28*, 552-566.

Berruti, F. M., Klaas, M., Briens, C. and Berruti, F. (2009). Model for Convective Drying of Carrots for Pyrolysis. *Journal of Food Engineering, 92*(2), 196 – 201.

Boozarjomehry, R.B. and Svrcek, W.Y. (2001). Automatic Design of Neural Network Structures. *Computer and Chemical Engineering, 25*, 1075-1088.

Chen, C.R., Ramaswamy, C.H. and Alli, I. (2001). Prediction of Quality Changes during Osmo-Convective Drying of Blueberries using Neural Network Models for Process Optimization. *Drying Technology, 19*(3-4), 507-523.

Chen, C.R. and Ramaswamy, H.S. (2002). Modeling and Optimization of Variable Retort Temperature (VRT) Thermal Processing using Coupled Neural Networks and Genetic Algorithms. *Journal of Food Engineering, 53*, 209.

Cheng, J. and Li, Q.S. (2008). Reliability Analysis of Structures using Artificial Neural Network Based Genetic Algorithms. *Computer Methods in Applied Mechanics and Engineering, 197*, 3742-3750.

Chiroma, H., Mohd-Noor, A.S., Abdulkareem, S., Abubakar, A.I., Hermawan, A., Qin, H., Herawan, T. (2017). Neural Networks Optimization through Genetic Algorithm Searches: A Review. *Applied Mathematics and Information Sciences, An International Journal, 11*(6), 1543-1564.

Coley, D.A. (1999). *An Introduction to Genetic Algorithms for Scientist and Engineers*. Singapore: World Scientific Publishing Co. Pte. Ltd.

David, J., Darrell, S., Larry, W. and Eshelman, J. (1992). *Combinations of Genetic Algorithms and Neural Networks: A Survey of the State of the Art.* Paper presented at the Proceedings of IEEE International Workshop on Communication, Networking and Broadcasting; Computing & Processing (Hardware/Software).

Di-Scala, K. and Crapiste, G. (2008). Drying Kinetics and Quality Changes during Drying of Red Pepper. *LWT- Food Science and Technology, 41*(5), 789–795.

Di-Scala, K., Meschino, G., Vega-Gálvez, A., Lemus-Mondaca, R., Roura, S. and Mascheroni, R. (2012). An Artificial Neural Network Model for Prediction of Quality Characteristics of Apples during Convective Dehydration. *Food Science and Technology*, 6.

Doymaz, I. (2005). Drying Behaviour of Green Beans. *Journal of Food Engineering, 69*(2), 161-165.

Eberhart, R.C. and Dobbins, R.W. (1990). Network Analysis. In R.C. Eberhart and R.W. Dobbins (Eds.), *In Neural Network PC Tools. A. Practical Guide*: Academic Press, San Diego, CA.

Erenturk, S. and Erenturk, K. (2007). Comparison of Genetic Algorithm and Neural Network Approaches for the Drying Process of Carrot. *Journal of Food Engineering, 78*, 905-912. doi:org/10.1016/j.jfoodeng.2005.11.031

Erenturk, S., Gulaboglu, M. and Gultekin, S. (2004). The thin-layer drying characteristics of rosehip. *Biosystems Engineering, 89*, 159-166.

Erenturk, S., Gulaboglu, M.S. and Gultekin, S. (2005). The Effects of Cutting and Drying Medium on the Vitamin C Content of Rosehip During Drying. *Journal of Food Engineering*, 513-518.

Fathi, M., Mohebbi, M. and Razavi, S. (2011a). Effect of Osmotic Dehydration and Air Drying on Physicochemical Properties of Dried Kiwifruit and Modeling of Dehydration Process Using Neural Network and Genetic Algorithm. *Food and Bioprocess Technology, 4*(8), 1519–1526.

Fathi, M., Mohebbi, M. and Razavi, S.M. (2011b). Application of Fractal Theory for Prediction of Shrinkage of Dried Kiwifruit using Artificial Neural Network and Genetic Algorithm. *Drying Technology, 29*(8), 918-925.

Fathi, M., Mohebbi, M. and Razavi, S.M.A. (2011c). Application of Image Analysis and Artificial Neural Network to Predict Mass Transfer Kinetics and Color Changes of Osmotically Dehydrated Kiwifruit. *Food and Bioprocess Technology, 4*(8), 1357-1366.

Finnie, G.R., Witting, G.E. and Desharnais, J.M. (1997). A Comparison of Software Effort Estimation Techniques: Using Function Points with Neural Networks, Case-Based Reasoning and Regression Models. *Journal of System Software, 39*(3), 281-289.

Firenze, F., Ricciardiello, L. and Pagliano, S. (1994). *Self-Organizing Networks: A Challenging Approach to Fault Diagnosis of Industrial Processes.* Paper presented at the Proceedings of ICANN'94 International Confrence on Artificial Neural Networks, London, UK, Springer.

Fischer, M.M. and Leung, Y. (1998). A Genetic-Algorithms Based Evolutionary Computational Neural Network for Modelling Spatial Interaction Data. *Annals of Regional Science, 32*, 437-458.

Fuangkhon, P. (2017). Parallel Distance-Based Instance Selection Algorithm for Feed-Forward Neural Network. *Journal of Intelligent Systems, 26*(2), 335–358. doi:https://doi.org/10.1515/jisys-2015-0039

Funes, E., Allouche, Y., Beltrán, G. and Jiménez, A. . (2015). A Review: Artificial Neural Networks as Tool for Control Food Industry Process. *Journal of Sensor Technology, 5*, 28-43. doi:http://dx.doi.org/10.4236/jst.2015.51004

Goldberg, D. (1989). Genetic Algorithms in Search, Optimization and Machine Learning [Press release]

Goni, S.M., Oddone, S., Segura, J.A., Mascheroni, R.H. and Salvadori, V.O. (2008). Prediction of Foods Freezing and Thawing Times: Artificial Neural Networks and Genetic Algorithm Approach. *Journal of Food Engineering, 84*(1), 164-178.

Haupt, R.L. and Haupt, S.E. (2004). *Practical Genetic Algorithms*. New Jersey, USA: John Wiley and Sons Inc. Hoboken.

Huang, Y., Kangas, L.J. and Rasco, B.A. (2007). Applications of Artificial Neural Networks (ANNs) in Food Science. *Critical Reviews in Food Science and Nutrition, 47*(2), 113-126.

Hussein, J.B., Filli, K.B. and Oke, M.O. (2016). Thin layer modelling of hybrid, solar and open sun drying of tomato slices. *Research Journal of Food Science and Nutrition, 1*, 15-27.

Javadikia, P., Rafiee, S., Garavand, A.T. and Keyhani, A. (2011). *Modeling of Moisture Content in Tomato Drying Procces by ANN-GA Technique.* Paper presented at the 2011 1st International eConference on Computer and Knowledge Engineering (ICCKE).

Kassem, A.S, Shokr, A.Z., Aboukarima, A.M. and Hamed, I.Y. (2010). Estimation of Moisture Ratio of Thompson Seedless Grapes Undergoing Microwave Drying Using Artificial Neural Network and Regression Models. *Journal of Saudi Social Agricultural Science, 9*(1), 23–47.

Keedwell, E. and Narayanan, A. (2005). *Intelligent Bioinformatics: The Application of Artificial Intelligence Techniques to Bioinformatics Problems*. New York: John Wiley & Sons Ltd.

Kulanthaisami, S., Rajkumar, P., Raghavan, G.S.V., Venkatachalam, P., Gariepy, Y., Subramanian, P. and Orsat, V. (2010). Drying Kinetics of Tomato Slices in Solar Cabinet Dryer Compared with Open Sun Drying. *Madras Agricultural Journal, 97*(7-9), 287 – 295.

Lahsansi, S., Kouhila, M., Mahrouz, M. and Jaouhari, J.T. (2004). Drying Kinetics of Prickly Pear Fruit (Opuntia Fiscus indica). *Journal of Food Engineering*, 173-179.

Liu, X., Chen, X., Wu, W. and Peng, G. (2007). A Neural Network for Predicting Moisture Content of Grain Drying Process using Genetic Algorithm. *Food Control, 18*, 928.

Longhmanian, S.M.R., Jamaluddin, H., Ahmad, R., Yusof, R. and Khalid, M. (2012). Structure Optimization of Neural Network for Dynamic System Modeling Using Multi-Objective Genetic Algorithm. *Neural Computing and Applications, 21*, 1281-1295.

Maskan, M. (2001). Drying Shrinkage and Rehydration Characteristic af Kiwifruit During Hot Air and Microwave Drying. *Journal of Food Engineering, 48*, 177-182.

Mayer, D.G., Belward, J.A. and Burrage, K. (1999). Survival of the Fittest-Genetic Algorithms Versus Evolution Strategies in the Optimization of Systems Models *Agricultural System, 60*, 113.

Meng, Y. and Ramaswamy, H. S. (2008). Neural Networks and Genetic Algorithms. In F. Erdogdu (Ed.), *Optimization in Food Engineering* (pp. 199-218). USA: CRC Press Taylor & Francis Group.

Menli, T., Kirmaci, V. and Usta, H. (2009). Modeling of Freeze Drying Behaviors of Strawberries by Using Artificial Neural Network. *Journal of Thermal Science and Technology, 29*(2), 11–21.

Mitchell, M. (1998). An Introduction to Genetic Algorithms [Press release]

Mohamed, A., Kouhila, M., Jamali, A., Lahsasni, S., Kechaou, N. and Mahrouz, M. (2005). Single Layer Solar Drying Behaviour of Citrus aurantium Leaves under Forced Convection. *Energy Conversion and Management, 46*(9-10), 1473 – 1483.

Mohebbi, M., Shahidi, F., Fathi, M., Ehtiati, A. and Noshad, M. (2011). Prediction of Moisture Content in Pre-Osmosed and Ultrasounded Dried Banana using Genetic Algorithm and Neural Network. *Food and Bioproducts Processing, 89*(4), 362-366.

Morimoto, T. (2006). Genetic Algorithms. In S.S. Sablani, A.K. Datta, M.S. Rahman, and A.S. Mujumdar (Eds.), *Handbook of Food and Bioprocess Modeling Techniques* (pp. 405). Boca Raton: CRC Press, Taylor & Francis.

Morimoto, T., Purwanto, W., Suzuki, J. and Hashimoto, Y. (1997). Optimization of Heat Treatment for Fruit During Storage using Neural Networks and Genetic Algorithms. *Computers and Electronics in Agriculture, 19*, 87-101.

Movagharnejad, K. and Nikzad, M. (2007). Modeling of Tomato Drying Using Artificial Neural Network. *Computers and Electronics in Agriculture, 59*, 78–85.

Nagata, Y. and Hoong, C.K. (2003). Optimization of a Fermentation Medium using Neural Networks and Genetic Algorithms. *Biotechnology Letters, 25*, 1837-1842.

Nguyen, N. and Cripps, A. (2001). Predicting Housing Value: A Comparison of Multiple Regression Analysis and Artificial Neural Networks *Journal of Real Estate Research (JRER), 22*, 313.

Ni, H. and Gunasekaran, S. (1998). Food Quality Prediction with Neural Networks. *Food Technology, 52*, 60-65.

Poonnoy, P., Tansakul, A. and Chinnan, M.S. (2007a). Artificial Neural Network Modeling for Temperature and Moisture Content Prediction in Tomato Slices Undergoing Microwave-Vacuum Drying. *Journal of Food Science, 72*(1), 41–47.

Poonnoy, P., Tansakul, A. and Chinnan, M.S. (2007b). Estimation of Moisture Ratio of a Mushroom Undergoing Microwave-Vacuum Drying Using Artificial Neural Network and Regression Models. *Chemical Product and Process Modeling, 2*(3), 1–13.

Popescu, O., Popescu, D., Wilder, J. and Karwe, M. (2001). A New Approach to Modelling and Control of a Food Extrusion Process Using Artificial Neural Network and an Expert System. *Journal of Food Process Engineering, 24*, 17-36. doi:http://dx.doi.org/10.1111/j.1745-4530.2001.tb00529.x

Rai, P., Majumdar, G.C., DasGupta, S. and De, S. (2005). Prediction of the Viscosity of Clarified Fruit Juice using Artificial Neural Network: A Combined Effect of Concentration and Temperature. *Journal of Food Engineering, 68*, 527-533.

Sablani, S.S., Ramaswamy, H.S. and Prasher, S.O. (1997). A Neural Network Modeling of Heat Transfer to Liquid Particle Mixtures in Cans Subjected to End-Over-End Applications. *Food Research International, 30*(2), 105-116.

Shopova, E. and Vaklieva-Bancheva, N. (2006). Basic-A Genetic Algorithm for Engineering Problems Solution. *Computers and Chemical Engineering, 30*, 1293–1309.

Specht, D.F. (1991). A General Regression Neural Network. *IEEE Transactions on Neural Networks, 2*(6), 568-576.

Sun, Q., Zhang, M. and Mujumdar, A.S. (2018). Recent Developments of Artificial Intelligence in Drying of Fresh Food. *A review, Critical Reviews in Food Science and Nutrition*. doi:10.1080/10408398.2018.1446900

Tohidi, M., Sadeghi, M., Mousavi, S.R. and Mireei, S.A. (2012). Artificial Neural Network Modeling of Process and Product Indices in Deep Bed Drying of Rough Rice. *Turkish Journal of Agriculture, 36*, 738-748. doi:10.3906/tar-1106-44

Tunde-Akintunde, T. Y. (2011). Mathematical Modeling of Sun and Solar Drying of Chilli Pepper. *Renewable Energy, 36*(8), 2139 – 2145.

Versace, M., Bhatt, R., Hinds, O. and Shiffer, M. (2004). Predicting the Exchange Traded Fund DIA with a Combination of Genetic Algorithms and Neural Networks. *Expert Systems with Applications, 27*, 417-425.

Visavale, G.L. (2012). Principles, Classification and Selection of Solar Dryers. In C.L. Hii, S.P. Ong, S.V. Jangam, and A.S. Mujumdar (Eds.), *Fundamentals, Applications and Innovations* (pp. 1-50). Singapore.

Yaldiz, O. and Ertekin, C. (2001). Thin Layer Solar Drying of Some Vegetables. *Drying Technology, 19*, 583–596.

Yousefi, A. and Razavi, S.M. (2017). Modeling of Glucose Release from Native and Modified Wheat Starch Gels during in Vitro

Gastrointestinal Digestion using Artificial Intelligence Methods. *International Journal of Biological Macromolecules, 97*, 752-760

Yousefi, A.R. (2017). Estimation of Papaw (*Carica papaw* L.) Moisture Content using Adaptive Neuro-Fuzzy Inference System (ANFIS) and Genetic Algorithm-Artificial Neural Network (GA-ANN). *Iranian Food Science and Technology Research Journal, 12*(6), 766-778 doi:10.22067/ifstrj.v12i6.62521

Zhang, G., Eddy, B.P. and Hu, M.Y. (1998). Forecasting with Artificial Neural Networks: The State of the Art. *International Journal of Forecasting, 14*, 35-62.