



Research Article

## BLEVE risk effect estimation using the Levenberg-Marquardt algorithm in an artificial neural network model

Tolga BARIŞIK<sup>1\*</sup>, Ali Fuat GÜNERİ<sup>2</sup>

<sup>1</sup>Department of Occupational Health and Safety, Istanbul Yeni Yüzyıl University, Istanbul, 34020, Türkiye

<sup>2</sup>Department of Industrial Engineering, Yıldız Technical University, Istanbul, 34020, Türkiye

### ARTICLE INFO

#### Article history

Received: 11 November 2021

Revised: 10 December 2021

Accepted: 19 February 2022

#### Keywords:

Artificial Neural Networks;  
Levenberg-Marquardt  
Algorithm; BLEVE; Modelling

### ABSTRACT

With the advancement of sciences such as machine learning, deep learning, and artificial intelligence, various algorithms are designed and developed. Learning and application models based on data types such as sensor data and databases of computers are made. The BLEVE effects, one of the most common types of fire in industries, are predicted using the Levenberg-Marquardt algorithm, which has become increasingly popular in recent years. Here, the equations in the TNO (Toegepast Natuurwetenschappelijk Onderzoek-Netherlands Applied Scientific Research Organization) (YellowBook Static) model of BLEVE are used.

The aim of this research is to use artificial neural networks to predict the risk size of the BLEVE event. All the results from the TNO model of BLEVE effects were estimated using an artificial neural network model. Without utilizing equations, outcomes that are close to true results could be estimated in this method. Furthermore, findings were acquired fast, with linear outcomes in many settings. For this reason, the study's necessity and significance have shifted in this direction. Sixteen TNO model equations from BLEVE are applied, and heat flux values are calculated as a result.

As a result of the studies, the BLEVE effects were predicted by the artificial neural network model created using the Levenberg-Marquardt algorithm. It is seen that the estimated results and the actual results calculated are close to each other. The statistical values between the predicted results of the artificial neural network model created by the Levenberg-Marquardt algorithm and the actual results were examined. The average relative error between the last stage of the BLEVE model created with ANN and the actual values was 3.34%. In addition, it has been observed that when the training iteration is increased within the algorithm, it gets closer to the real results, and statistical values such as standard error decrease even more. In other words, the computation was done by increasing the iteration sequence processing cycle of the result estimations and repeated with the number of training iterations in order to better evaluate the network's performance. It has been observed that as the number of iterations increases, closer and more realistic results emerge.

\*Corresponding author.

\*E-mail address: [tolga.barisik@yeniuyuzuil.edu.tr](mailto:tolga.barisik@yeniuyuzuil.edu.tr)

*This paper was recommended for publication in revised form by  
Eyup Debik*



**Cite this article as:** Barışık T, Güneri A F. BLEVE risk effect estimation using the Levenberg-Marquardt algorithm in an artificial neural network model. Sigma J Eng Nat Sci 2022;40(4):877–893.

## INTRODUCTION

Artificial neural networks have recently been seen as a highly sought-after field within the artificial intelligence discipline. Especially in the field of technology (mobile phones, computers, various smart devices, etc.), this artificial neural network system is used. Artificial neural networks are a form of decision-making system that learns the relationships between events from samples and then applies what they have learned to invisible examples [1]. Learning, generalization, classification, correlation, feature determination, and optimization are some of the functions of artificial neural networks [2].

Artificial neural networks use different learning types and techniques. Teacher learning, positive learning, non-teacher learning, and mixed methods are learning styles [3]. Artificial neural network models for teacher learning (multi-layer perceptron-MCA) and teacher-less learning will be used in this analysis (adaptive resonance theory-ART). MCA model: the learning structure needs an instructor to learn the situation. The teacher explains the situation that should be studied as input and output into the method. It is ensured that both inputs and outputs corresponding to these inputs are visible in the system for each sample. The method, on the other hand, maps these inputs to the outputs that the instructor specifies. As a result, the relationships between the inputs and outputs are discovered. In the ART model, there is no teacher who can support the method learn. In the method, only input values are specified. The device is supposed to figure out how the parameters in the examples relate to one another on its own. However, after understanding the method, the user must create conditions that illustrate what the outputs would mean [1,2].

When some current studies are examined, applications are seen in different fields. Sabdani 2020 examined strong resonance bifurcations with reflection symmetry. Numerical analysis of these bifurcations and modeling of their results with economic and artificial neural networks were made [4]. In another study is numerical treatment of a dynamic nonlinear hepatitis-B model: an evolutionary approach. Farman 2020 investigated mathematical models of Hepatitis-B [5]. Tian 2009 determined new delay-dependent asymptotic stability criteria for neural networks with time-varying delays. For neural networks (NNs) with time-varying delays, the problem of delay-dependent asymptotic stability criteria is explored. [6]. Naik et al. 2020 has researched chaotic dynamics of the fractional order HIV-1 model involving AIDS-associated cancer cells. The HIV-1 model in fractional order, which includes AIDS-associated

cancer cells, is introduced. Stability analysis of the proposed model is made. Numerical simulations using the two-stage Adam-Bashforth method have been performed to support our results. The results show that, among other properties, fractional models are better predictors [7].

BLEVE; It is an abbreviation of the initials of the words boiling liquid expanding vapor cloud explosion. The BLEVE case, which will be used in this study to test the artificial neural network, is one of the fire events that can occur in many industrial facilities. With the passage of time, manufacturing facilities have grown and advanced. As an example, energy and chemical facilities can be listed. Despite the fact that there are numerous studies in the field of facility technology, various processes and safety are still needed. Major industrial accidents over the years, particularly in recent years, demonstrate the critical importance of the safety sector. In terms of the risk description, it includes the probability of any damage, failure, or other negative effects as a result of an accident, as well as the possibility of causes such as pollution [8].

Significant losses can occur in major industrial accidents as a result of death and injury caused by fire, explosion, and gas spread, as well as occupational accidents and diseases. One of the most common forms of incidents is fire. In all facilities, there is a high risk of burning. BLEVE fires, which are fireball-shaped, are among these types of fires.

The most dangerous incident in liquefied flammable gas tanks, such as LPG, is known as BLEVE. The temperature inside the tank increases rapidly as a result of a fire that begins outside the tank; moreover, the transition from the liquid to the gas phase increases [9]. The pressure relief valve of the tank acts as the internal pressure of the tank rises, releasing the excess pressure inside; however, due to the speed at which the tank transitions to the gas phase, the evacuation mechanism is unable to avoid the pressure rise [10,11]. The tank ruptures when the internal pressure of the tank reaches the maximum pressure of the tank [12,13]. At this point, the fragments that broke away from the tank and the tank itself are scattered around like shrapnel, often violently enough to fly hundreds of meters [12,14]. The warm, pressurized gas inside the tank bursts through the breach stage, forming a huge fireball that slowly rises [15].

BLEVE; In a brief, it's the reaction that happens when a substance that has been overheated under pressure is suddenly released into the atmosphere [16]. This occurrence releases massive quantities of thermal radiation over long distances, and fatal burns to people in the area [17].

In this study, the heat flux values of the BLEVE event were estimated with the artificial neural network model. In this way, thermal radiation effects will be examined. In order to construct a model in a neural network, the Levenberg-Marquardt algorithm was used. A network model was developed using this algorithm, and predictions about the BLEVE event were made.

When looking at the algorithms used in the artificial neural network, the Levenberg-Marquardt algorithm is widely used and gives good results in terms of performance from many algorithms. It is good in terms of speed and memory performance than algorithm mechanisms such as Gradient Descent, Newton Method, Conjugate Gradient and Quasi-Newton [18].

Many studies have looked into the significance and role of the Levenberg-Marquardt algorithm in a variety of applications. Artificial intelligence plays an important role in the development of artificial neural network modelling, according to these studies.

In this study, it is aimed to design a modeling and calculation mechanism of artificial neural networks and results, which can be used in responding to risks such as fire and explosion, which can occur at the same time, through the example scenario of BLEVE effects, which are among the major industrial accident types.

The BLEVE effects were predicted by an artificial neural network model developed using the Levenberg-Marquardt algorithm as a result of the studies. It can be seen that the estimated and calculated results are very close to each other.

In real-life applications, especially when the decision support mechanism will be designed, the desired goal will be achieved by using these algorithms by giving faster and more accurate results. In this way, since there will be more training sets in the artificial neural network model, it will show the results directly in any scenario.

The goal of this research is to expand it and contribute to other domains. It is possible to model fire, explosion, and gas dispersion events, which are common in significant industrial accidents like BLEVE. It will also assist in reducing the harmful environmental consequences of these events.

## THEORY

In this study, the artificial neural network was modelled with the input data determined using the Levenberg-Marquardt algorithm, which is a multi-layer perceptron (MLP) (teacher learning) type of modelling method in the literature. The outputs produced by the network are compared with the actual values calculated for BLEVE. The feasibility of performance determination is investigated by using the curves to be created from the obtained values and the artificial neural network.

The scenario was determined and calculated for the modelling of the fireball explosion (BLEVE). In the

scenario, the gas in the tankers was chosen as LPG. Its molecular weight is 53.06 gr/mol. Artificial neural network results were developed for the outputs of the data on LPG gas explosions in tankers. For the BLEVE model, inputs are given to the network. What is expected from the network is the thermal radiation heat flux ( $W/m^2$ ) value as the output value corresponding to these inputs.

An artificial neural network model and a calculator were created to explosion of LPG gas in tanks of different mass and volumes. The equations for BLEVE are given below chapter. It is given in the heading number. All equations in the BLEVE model are processed sequentially. Each equation was processed systematically, and its results were obtained in the artificial neural network.

## BLEVE

Some models have been developed for calculating the results of BLEVE thermal radiation effects (heat flux). One of the most used modelling is the TNO (YellowBook Static) method. In this study, artificial neural networks were trained and evaluated according to the data in the TNO model [17; 19]. The equations for BLEVE are given below [20; 21]:

Fireball radius equation, [m (in meters)]:

$$r_{fb} = c_9 \times m^{0.325} \quad (1)$$

Where m; represents the mass in kilograms.  $c_9$  is the constant coefficient, and its value is 3.24.

Fireball formation time equation, t [s (in seconds)]:

$$t = c_{10} \times m^{0.26} \quad (2)$$

Where m; represents the mass.  $c_{10}$  is the constant coefficient, and its value is 0.852

The equation for the lift-off height of the fireball from the ground,  $H_{bleve}$ , [m (in meters)]:

$$H_{bleve} = 2 \times r_{fb} \quad (3)$$

The equation for distance between fireball center and object, X [m (in meters)]:

$$X = \sqrt{(x_{bleve}^2 + H_{bleve}^2)} \quad (4)$$

Here is  $x_{bleve}$ ; It is the measurement distance between the area where the heat flux is felt and the fireball.

The image (shape) factor equation affecting the fire,  $F_{shape}$  [without units]:

$$F_{shape} = \left( \frac{r_{fb}}{x} \right)^2 \quad (5)$$

The fraction equation for heat generated from the fire-ball, heat release rate,  $F_s$  [without units]:

$$F_s = c_6 \times (P_{sv})^{0.32} \quad (6)$$

Here  $P_{sv}$  is the rupture pressure in bar and  $c_6$  is also constant. Its value is 0.00325.

The temperature difference equation,  $(\Delta T)$  [K (in Kelvin)]

$$\Delta T = T_f - T_a \quad (7)$$

Here  $T_f$  (BLEVE temperature) is constant, and its value is 1500 K.  $T_a$  is determined as the air temperature.

The equation for net heat dissipation,  $\Delta H$  [j / kg]:

$$\Delta H = \Delta H_c - \Delta H_v - C_p \Delta T \quad (8)$$

Some values in the equation are fixed. The heat of combustion ( $\Delta H_c$ ) is 49752800 J / kg. The value of  $\Delta H_v$  is the ratio of heat of vaporization to molecular weight.  $C_p$  value is the ratio of specific heat to molecular weight.

Surface emission power equation, SEP [W / m<sup>2</sup> (in Watt / square meter)]:

$$SEP = \Delta H \times m \times \frac{F_s}{4 \times \pi \times rfb^2 \times t} \quad (9)$$

Radiation true path length equation,  $x$  [m]:

$$x = X - rfb \quad (10)$$

The vapor pressure equation of water formed in the air after fire,  $P_{wx}$  [Pa (in Pascal)]:

$$P_{wx} = \left( \frac{H_{ua}}{100} \right) \times (P_{vW}) \times (T_a) \times (x) \quad (11)$$

Here  $H_{ua}$ ; It is the air humidity rate in %.

Water vapor absorption coefficient equation,  $\alpha x$  [without unit]:

$$\alpha x = (0,0428) \times \ln(P_{wx}) - (0,2893) \quad (12)$$

Carbon dioxide partial pressure equation in air,  $P_{vCO_{2x}}$  [Pa]:

$$P_{vCO_{2x}} = (P_{vCO_2}) \times (x) \quad (13)$$

The  $P_{vCO_2}$  value in this equation is 0.03% (0.0003) of atmospheric pressure (101325 Pa) and is constant. Hence, it is 30.3975 Pa.

The carbon dioxide absorption coefficient equation,  $\alpha_c$  [without units]:

$$\alpha c = (0,0428) \times (P_{vCO_{2x}})^{0,4004} \quad (14)$$

The equation for atmospheric permeability value,  $\tau_a$  [without units], affects the radiation propagation:

$$\tau_a = 2.02 \times (P_{wx})^{-0.09} \quad (15)$$

Calculated heat flux / Thermal radiation value equation detected by the receptor,  $q''$  [W / m<sup>2</sup>]:

$$q'' = SEP \times F_{shape} \times \tau_a \quad (16)$$

## ARTIFICIAL NEURAL NETWORK MODEL

The artificial neural network created is a network model that learns with the Levenberg-Marquardt algorithm, creating input and output levels and uses consultancy learning.

## LEVENBERG-MARQUARDT ALGORITHM

One of the minimum search algorithms is the Levenberg-Marquardt (LM) algorithm. It approaches the error surface parabolically in each iteration (number of training repetitions), and the minimum of the parabola becomes the most suitable solution for that iteration.

Many software applications use LM to solve common curve fitting issues. The LM, like many other fitting algorithms, only finds a local minimum, which is not always the global minimum. LM interpolates the Gauss-Newton algorithm (GNA) and the gradient descent procedure [22, 23]. Because LM is more resilient than GNA, it will usually find a solution even if it begins far from the absolute minimum. LM is slower than GNA for well-behaved functions and rational starting parameters. Using a trust zone method, the LM can also be considered as Gauss-Newton [24, 25].

Newton's method would be as follows if there is a function  $E(x)$  to look for the minimum and this function is reduced according to the  $x$  parameter [26].

$$\Delta x = - [\nabla^2 E(x)]^{-1} \nabla E(x) \quad (17)$$

The expression  $\nabla^2 E(x)$  is the Hessian matrix.  $\nabla E(x)$  is the slope. Since the function  $E(x)$  is the sum of squares of network errors, this relation can be represented as:

$$E(x) = \sum_{i=1}^N e_i^2(x) \quad (18)$$

Expanding the expression  $\nabla E(x)$  results in the following two equations:

$$\nabla E(x) = J^T(x)e(x) \quad (19)$$

$$\nabla E(x) = J^T(x)J^T(x) + s(x) \quad (20)$$

Here  $J(x)$ ; It is the Jacobian matrix.

$$j(x) \begin{bmatrix} \frac{\partial e_1(x)}{\partial x_1} & \frac{\partial e_1(x)}{\partial x_2} & \dots & \frac{\partial e_1(x)}{\partial x_n} \\ \frac{\partial e_1(x)}{\partial x_1} & \frac{\partial e_1(x)}{\partial x_2} & \dots & \frac{\partial e_2(x)}{\partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial e_N(x)}{\partial x_1} & \frac{\partial e_N(x)}{\partial x_2} & \dots & \frac{\partial e_N(x)}{\partial x_2} \end{bmatrix} \quad (21)$$

This function is accepted as 0 in Gauss-Newton method and the equation is given below.

$$S(x) = \sum_{i=1}^N e_i(x) \nabla^2 e_i(x) \quad (22)$$

$$\Delta x = [J^T(x)J(x)]^{-1}J^T(x)e(x) \quad (23)$$

With the Levenberg-Marquardt update, the Gauss-Newton approach becomes:

$$\Delta x = [J^T(x)J(x) + \mu I]^{-1}J^T(x)e(x) \quad (24)$$

The letter I represents the unit matrix in the equation. The Marquardt parameter is a number between 0 and 1. A scalar number is a parameter. When E (x) rises, the parameter is multiplied by a factor (β), and when it falls, the parameter is divided by the β factor. The procedure becomes a small-step gradient reduction if the parameter is a large number and the Gauss-Newton method is a small number [27].

To solve the first part in relation 24, it is necessary to use the Hessian matrix. In the Levenberg-Marquardt algorithm, the approximate value of the Hessian matrix is used. The Hessian matrix's approximate value is as follows:

$$H = [J^T(x)J(x) + \mu I]^{-1} \quad (25)$$

To calculate the Jacobian matrix in the Hessian matrix, all of the network's training inputs must be trained, yielding the following error vector equation:

$$v^T = [v_1, v_2, \dots, v_N] = [e_{1,1}, e_{2,1}, \dots, e_{S^M} + 1, e_{1,2}, \dots, e_{S^M,Q}] \quad (26)$$

All of the weights and bias values in the artificial neural network are combined to form the vector in the following equation:

$$x^T = [x_1, x_2, \dots, x_N] = [w^1_{1,1}, w^1_{1,2}, \dots, w^I_{S^I,R}, b^1, \dots, b^I_{S^I}, w^1_{1,1}, \dots, b^M_{S^M}] \quad (27)$$

The output layer's local gradients are calculated using equations 26, and the hidden layer's local gradients are calculated using equations 27. Two matrices are generated as long as the input, and hidden layers' gradient are distinct.

$$S_q^{-m} = -F^m n_q^m \quad (28)$$

$$S_q^{-m} = -F^m(n_q^m) \cdot (W^{m+1})^T \cdot S_q^{-m+1} \quad (29)$$

The weights and bias values of Jacobian matrix elements are computed using equations 28 and 29:

$$[J_{h,1}] = \frac{\partial v_h}{\partial x_i} = \frac{\partial e_{k,q}}{\partial w_{i,j}^m} = \frac{\partial e_{k,q}}{\partial n_{i,q}^m} x \frac{\partial n_{i,q}^m}{\partial w_{i,j}^m} = S_{i,h}^{-m} x \frac{\partial n_{i,q}^m}{\partial w_{i,j}^m} = S_{i,h}^{-m} x a_{jq}^{m-1} \quad (30)$$

$$[J_{h,1}] = \frac{\partial v_h}{\partial x_i} = \frac{\partial e_{k,q}}{\partial b^m} = \frac{\partial e_{k,q}}{\partial n_{i,q}^m} x \frac{\partial n_{i,q}^m}{\partial b_i^m} = S_{i,h}^{-m} x \frac{\partial n_{i,q}^m}{\partial b_i^m} = S_{i,h}^{-m} \quad (31)$$

The Levenberg-Marquardt algorithm changes whatever connection weight in the network has the greatest impact on the outcome. As a result, it gets to the solution quickly. However, as a defect, a large amount of memory is required [28].

## METHOD STAGES AND APPLICATION

The number of inputs, output, neuron and intermediate layer numbers were determined to obtain optimum results using the application's trial and error technique. In this context, it was decided to use two hidden layers for each equation. In each model of the applications, one more than the total input and output values were determined as the number of neurons and the number of hidden layers was kept constant as 2.

In the created artificial neural network model, the hyperbolic tangent (sigmoid) is used in the input and intermediate layers. The purelin function, a linear transfer function, is used for the output layer. Back propagation logic is used in this algorithm because it has an easy and understandable difference.

Artificial neural network training was created with different data sets from the performance and time perspective and the results were examined. For example, 100 training data were given as input and 100 uneducated data results were calculated within 200 different values. To measure the performance of the artificial neural network in calculating data outside the range of training data, some values are given outside of the training range.

The artificial neural network created initially for the applications was created to calculate one output value for one input value, and then an artificial neural network model was created, which was trained with the Levenberg-Marquardt algorithm, using consultancy learning by increasing the number of inputs. In all of the applications, 100 data were used for each of the inputs and outputs, and the training and calculation were calculated over these 100 data sets.

## STAGES OF APPLICATION

As previously mentioned, all equations in the BLEVE model were processed in order in application. Each equation was processed systematically in the artificial neural network, and the results were obtained. Every stage's phase and operation are as follows:

- For equation (1), the LPG mass was determined as the input value. The model was created and applied as one input, one output, two hidden layers and three neurons in each hidden layer. The test data in the data set were computed using values that differed from the training data set. In this application, the results of the fireball radius were calculated using the mass values.
- For equation (2), the LPG mass was determined as the input value. The model was created and applied as one input, one output, two hidden layers and three neurons in each hidden layer. The test data in the data set were computed using values that differed from the training data set. In this application, the results of the fireball formation time were calculated using mass values.
- For equation (3), the output values of the fireball radius calculated with the artificial neural network in equation (1) were determined as the input value. The model was created and applied as one input, one output, two hidden layers and three neurons in each hidden layer. The test data in the data set used were calculated by giving different values than the training data set. In this application, the results of the rising distance of the fireball were calculated.
- For the equation (4), two input values will be created. The first input value is the output values of the rising distance of the ball of flame calculated with artificial neural network in equation (3). The other input is determined as the measurement distance. The model was created and applied with two inputs, one output, two hidden layers and four neurons in each hidden layer. The test data in the data set used were calculated by giving different values than the training data set. In this application, the results of the distance from the centre of the fireball to the measurement point were calculated.
- For the equation numbered (5), two input values were created. The first input value is the output values of the fireball radius calculated with the artificial neural network in equation (1). The other input is the output values of the distance from the fireball centre to the measurement point calculated with the artificial neural network in the equation (4). The model was created and applied with two inputs, one output, two hidden layers and four neurons in each hidden layer. The test data in the data set used were calculated by giving different values than the training data set. In this application, the results of the image (shape) factor are calculated.
- For equation (6), the rupture pressure was determined as the input value. The model was created and applied as one input, one output, two hidden layers and three neurons in each hidden layer. The test data in the data set used were calculated by giving different values than the training data set. In this application, the results of the heat release rate were calculated using mass values.
- For equation (7), the air temperature was determined as the input value. The model was created and applied as one input, one output, two hidden layers and three neurons in each hidden layer. The test data in the data set used were calculated by giving different values than the training data set. In this application, the results of the temperature difference were calculated using mass values.
- In the equation (8), the output values of the temperature difference calculated with the artificial neural network in the equation (7) are determined as the input value. The model was created and applied as one input, one output, two hidden layers and three neurons in each hidden layer. The test data in the data set used were calculated by giving different values than the training data set. In this application, the results of the net heat dissipation were calculated using mass values.
- For the equation numbered (9), five input values were created. The first input value is the output values of the net heat dissipation calculated with artificial neural network in equation (8). The 2nd input value is the value of the LPG mass. The 3rd input value is the output values of the heat emission ratio calculated with the artificial neural network in the equation (6). The 4th input value is the output values of the fireball radius calculated with the artificial neural network in equation (1). The last input value is the output value of the fireball formation time calculated with the artificial neural network in the equation (2). The model was created and applied with five inputs, one output, two hidden layers and seven neurons in each hidden layer. The test data in the data set used were calculated by giving different values than the training data set. In this application, the results of the surface emission power (SEP) are calculated.
- For the equation numbered (10), two input values were created. The first input value is the output value of the fireball centre to the measurement point calculated with the artificial neural network in the equation (4). The other input value is the output values of the fireball radius calculated with the artificial neural network in equation (2). The model was created and applied with two inputs, one output, two hidden layers and three neurons in each hidden layer. The test

- data in the data set used were calculated by giving different values than the training data set. In this application, the results of the distance from the fireball surface to the measurement point are calculated.
- For the equation numbered (11), two input values were created. The first input value is the air humidity (Hua) ratio values in percentage. The other input value is the output values of the distance from the fireball surface to the measurement point calculated by artificial neural network in the equation numbered (10). The model was created and applied with two inputs, one output, two hidden layers and three neurons in each hidden layer. The test data in the data set used were calculated by giving different values than the training data set. In this application, the results of the water vapor pressure are calculated.
  - In equation (12), the input value is they are the output values of the water vapor pressure calculated with the artificial neural network in equation (11). The model was created and applied as one input, one output, two hidden layers and three neurons in each hidden layer. The test data in the data set used were calculated by giving different values than the training data set. In this application, the results of the water vapor absorption coefficient were calculated using mass values.
  - For the equation numbered (13), the input value is the output value of the distance from the fireball surface to the measurement point calculated with the artificial neural network in equation (10). The model was created and applied as one input, one output, two hidden layers and three neurons in each hidden layer. The test data in the data set used were calculated by giving different values than the training data set. In this application, the results of the partial carbon dioxide pressure in the air are calculated.
  - For the equation numbered (14), the input value is the output value of the carbon dioxide partial pressure calculated with the artificial neural network in equation (13). The model was created and applied as one input, one output, two hidden layers and three neurons in each hidden layer. The test data in the data set used were calculated by giving different values than the training data set. In this application, the results of the carbon dioxide absorption coefficient are calculated.
  - For the equation numbered (15), the input value is the output value of the water vapor pressure calculated with the artificial neural network in equation (11). The model was created and applied as one input, one output, two hidden layers and three neurons in each hidden layer. The test data in the data set used were calculated by giving different values than the training data set. In this application, the results of the atmospheric permeability are calculated.
  - For the equation numbered (16), three input values were created. The first input value is the output values of the surface emission power (SEP) calculated with artificial neural network in equation (9). The 2nd input value is the output values of the image (shape) factor calculated with artificial neural network in equation (5). The last input value is the output values of atmospheric permeability calculated with artificial neural network in equation (15). The model was created and applied with three inputs, one output, two hidden layers and five neurons in each hidden layer. The test data in the data set used were calculated by giving different values than the training data set. In this application, the results of the heat flux are calculated.

## RESULTS AND DISCUSSION

### Results

#### Graphical Results

Based on the implementation of the steps specified in the previous chapter. title, the graphs comparing the test inputs used for training with the ANN results, the ANN results and the actual results of the tested inputs are as follows:

The results of the trained and test data of the mass (ANN and actual results) values of LPG and the fireball radius for Equation (1) are shown in Figure 1:

The results of the trained and test data of the mass (ANN and actual results) values of LPG and the fireball formation time for Equation (2) are shown in Figure 2:

The results of the trained and test data of the fireball radius (ANN and actual results) values and the fireball rising distance for Equation (3) are shown in Figure 3:

The ANN and actual results of the data of the fireball radius values and the fireball rising distance for Equation (4) are shown in Figure 4:

The ANN and actual results data of the image (shape) factor values for Equation (5) are shown in Figure 5:

The results of the trained and test data of image (shape) factor (ANN and actual results) values and heat release rate from flame surface for Equation (6) are shown in Figure 6:

The results of the trained and test data of air temperature (ANN and actual results) values and temperature difference for Equation (7) are shown in Figure 7:

The results of the trained and test data of air temperature difference (ANN and actual results) values and net heat dissipation for Equation (8) are shown in Figure 8:

The ANN and actual results data of surface emission power values for Equation (9) are shown in Figure 9:

The ANN and actual results data of distance from the fireball surface to the measurement point values for Equation (10) are shown in Figure 10:

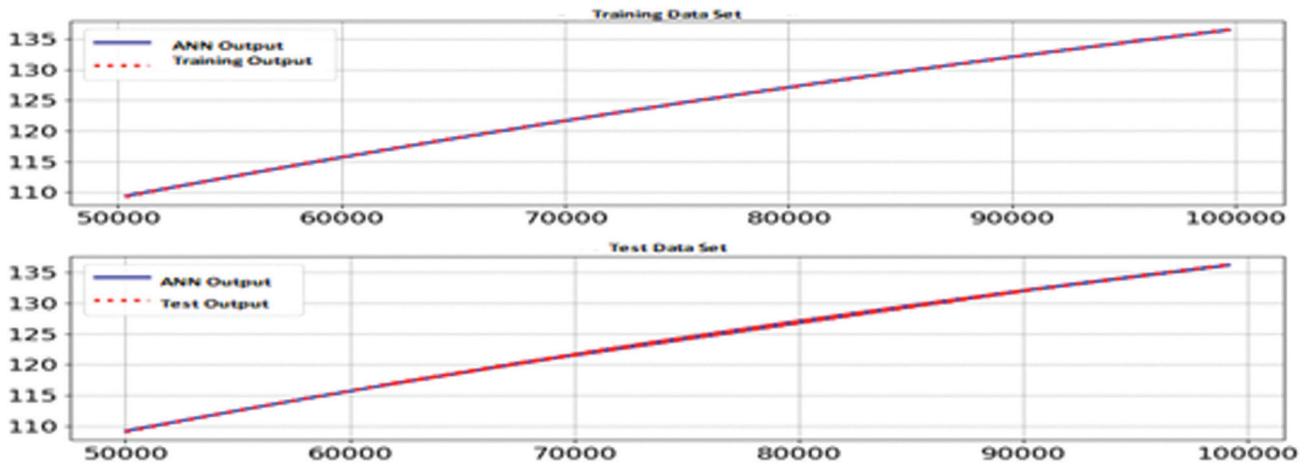


Figure 1. Comparison of ANN and actual results for fireball radius.

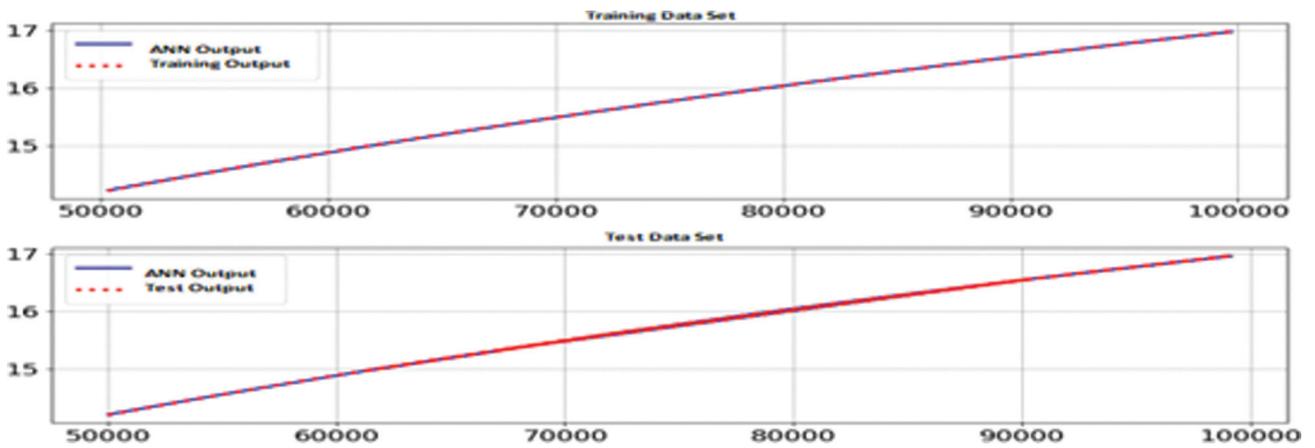


Figure 2. Comparison of ANN and actual results for fireball formation time.

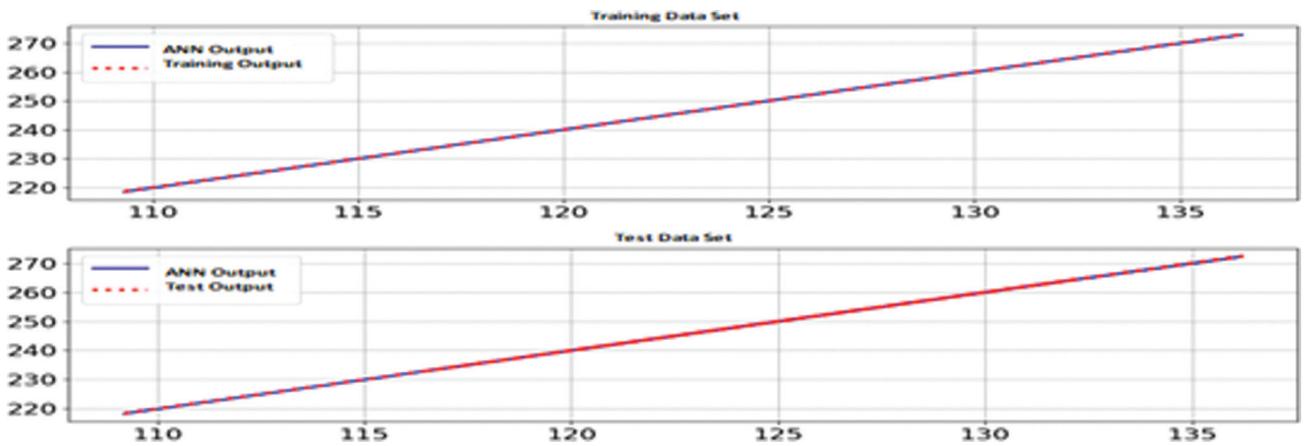


Figure 3. Comparison of ANN and actual results for fireball rising distance.

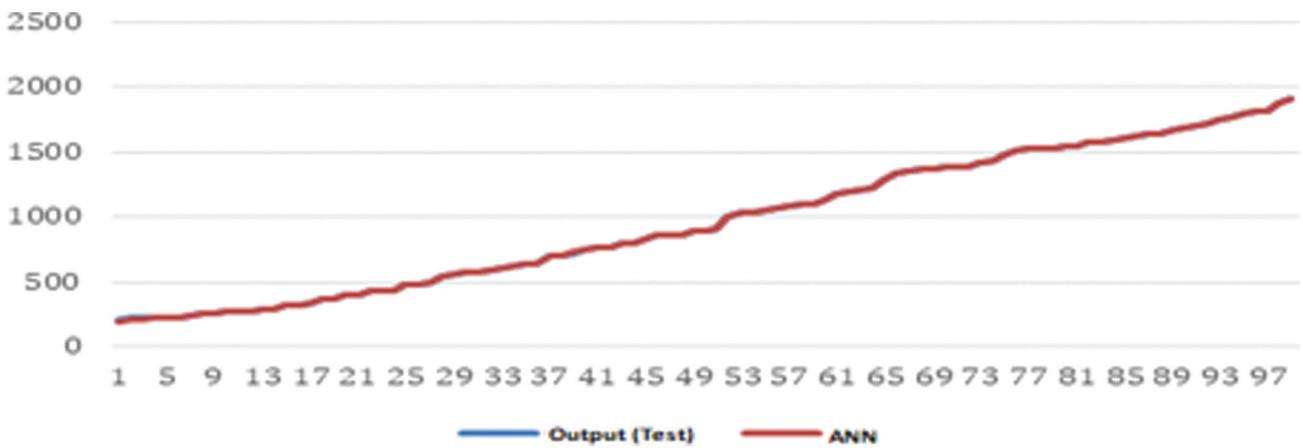


Figure 4. Comparison of ANN and actual results for distance from fireball center to measurement point.

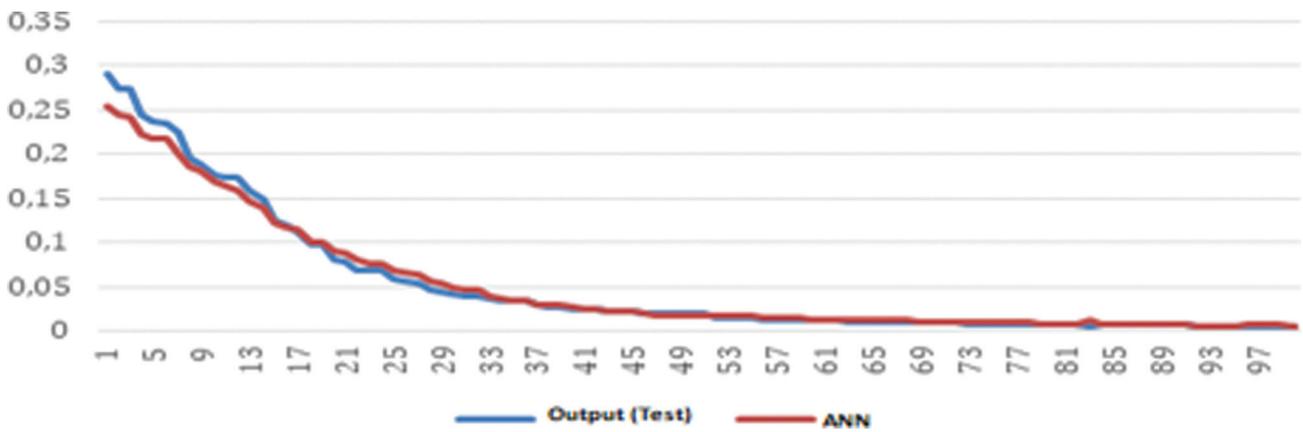


Figure 5. Comparison of ANN and actual results for image (shape) factor.

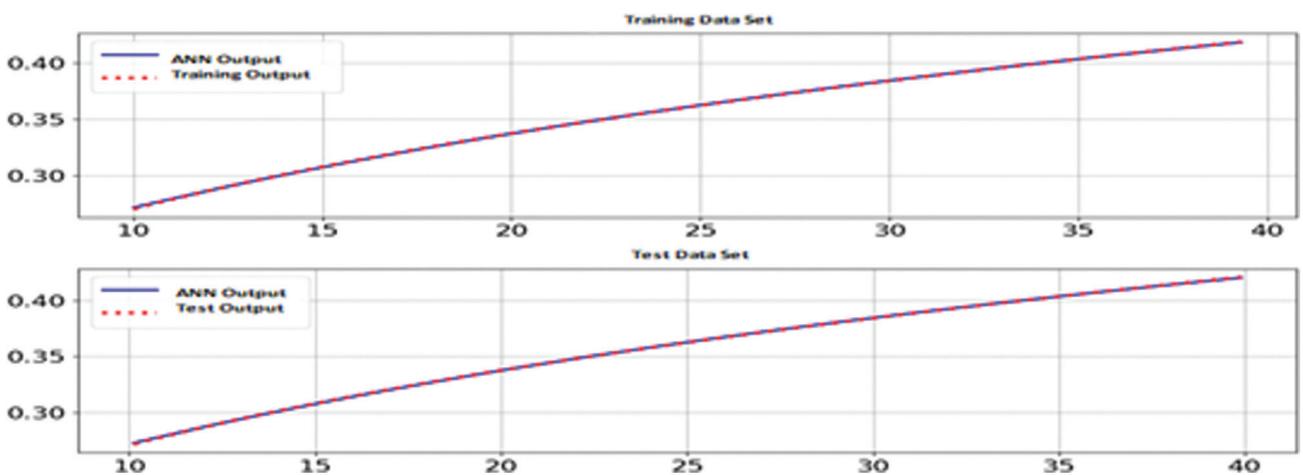


Figure 6. Comparison of ANN and actual results for heat release rate from flame surface.

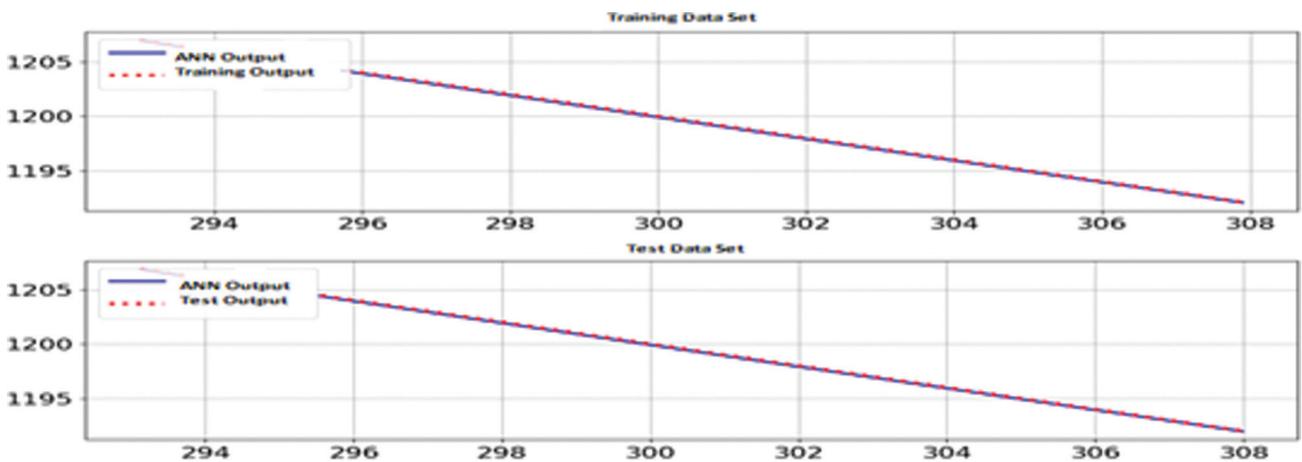


Figure 7. Comparison of ANN and actual results for temperature difference.

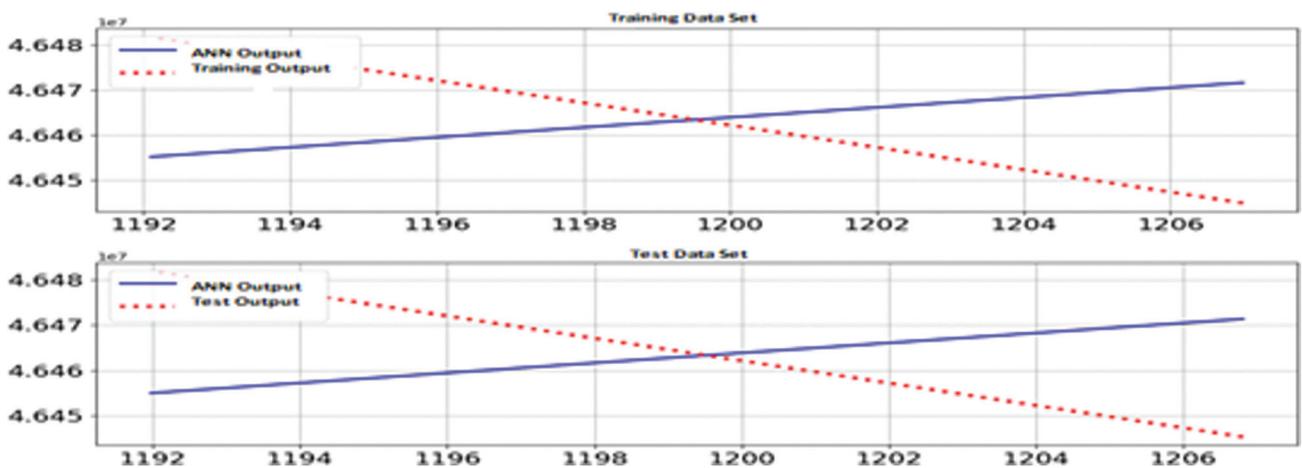


Figure 8. Comparison of ANN and actual results for net heat dissipation.

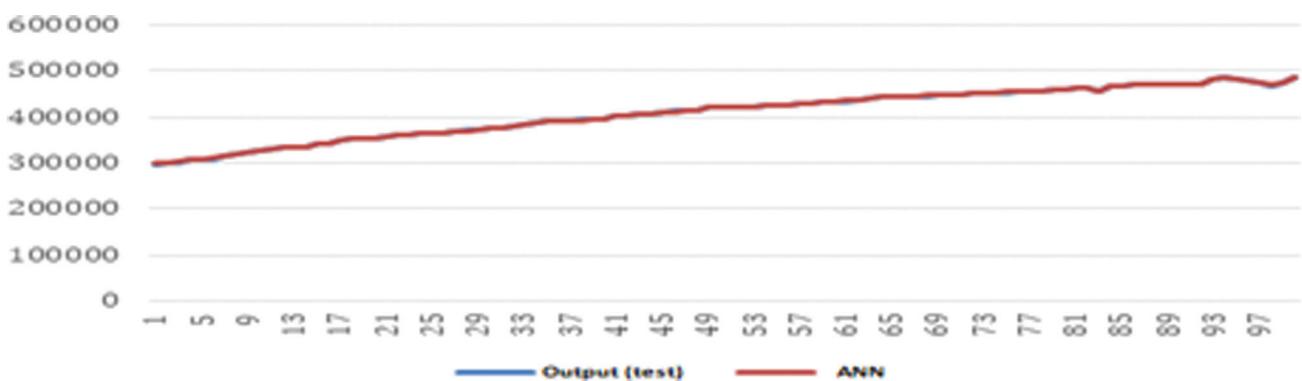


Figure 9. Comparison of ANN and actual results for surface emission power.

The ANN and actual results data of water vapor pressure in the air values for Equation (11) are shown in Figure 11:

The results of the trained and test data of water vapor absorption coefficient (ANN and actual results) values and water vapor pressure for Equation (12) are shown in Figure 12:

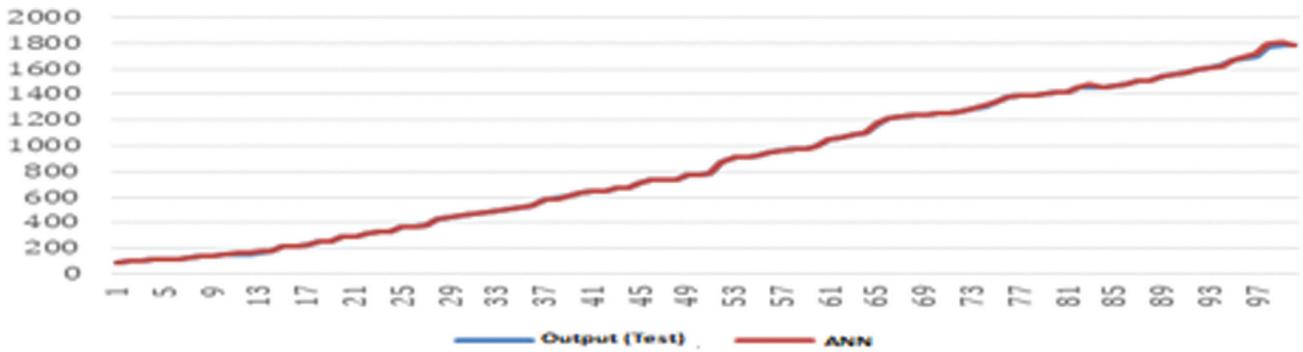


Figure 10. Comparison of ANN and actual results for distance from the fireball surface to the measurement point.

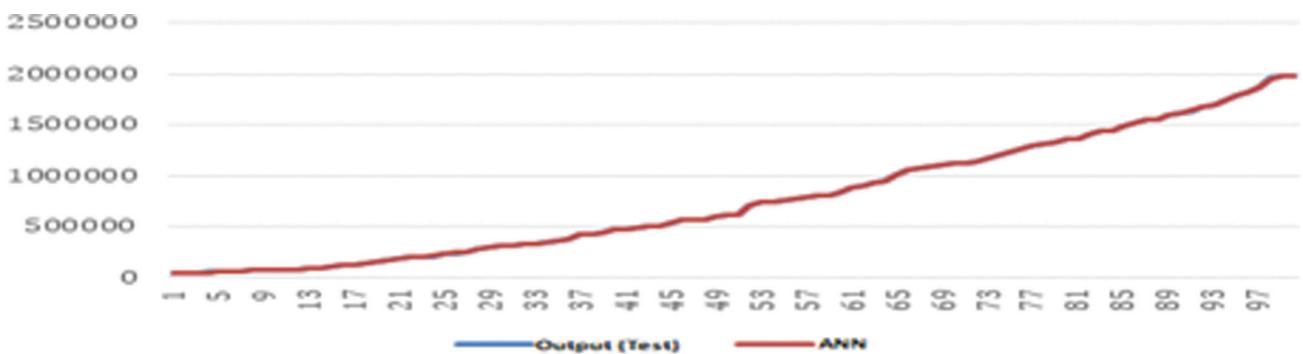


Figure 11. Comparison of ANN and actual results for water vapor pressure in the air.

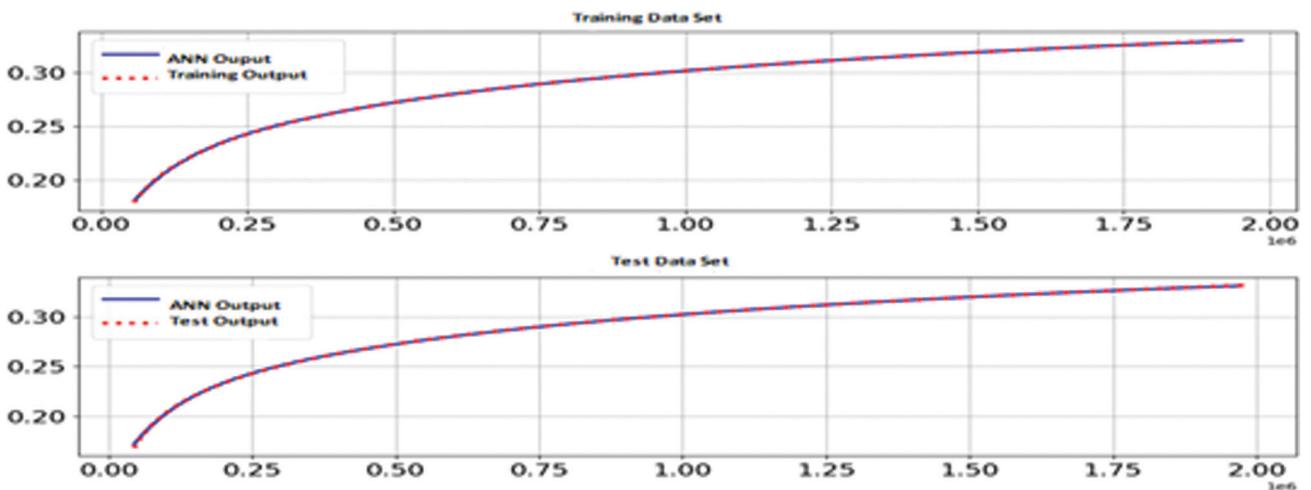


Figure 12. Comparison of ANN and actual results for water vapor absorption coefficient.

The results of the trained and test data of partial pressure of carbon dioxide in the air (ANN and actual results) values and distance from the fireball surface to the measurement point for Equation (13) are shown in Figure 13:

The results of the trained and test data of carbon dioxide absorption coefficient (ANN and actual results) values and

carbon dioxide partial pressure for Equation (14) are shown in Figure 14:

The results of the trained and test data of atmospheric permeability (ANN and actual results) values and water vapor pressure for Equation (15) are shown in Figure 15:

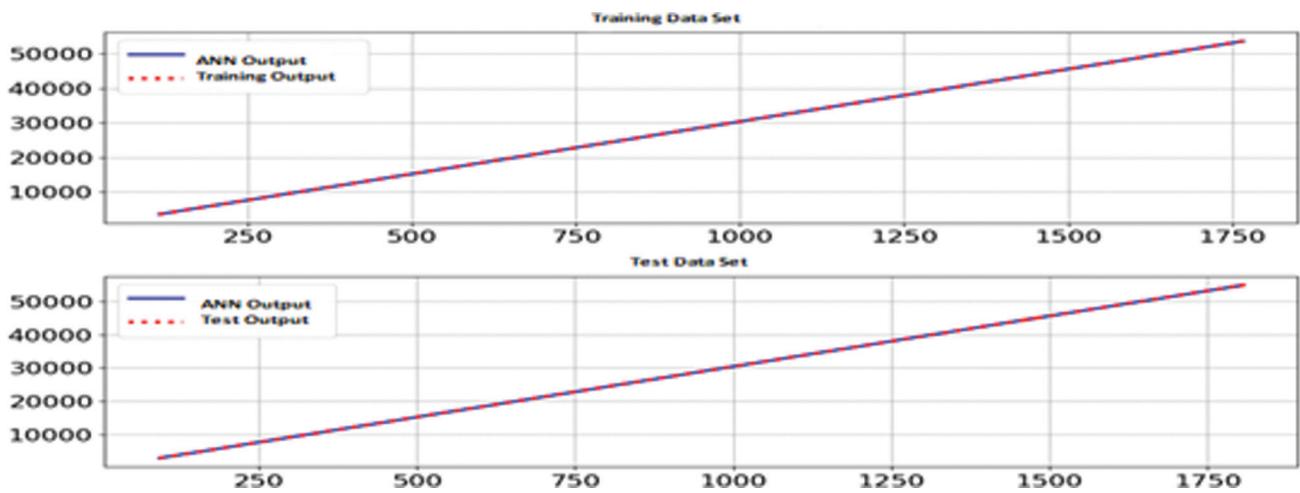


Figure 13. Comparison of ANN and actual results for the partial pressure of carbon dioxide in the air.

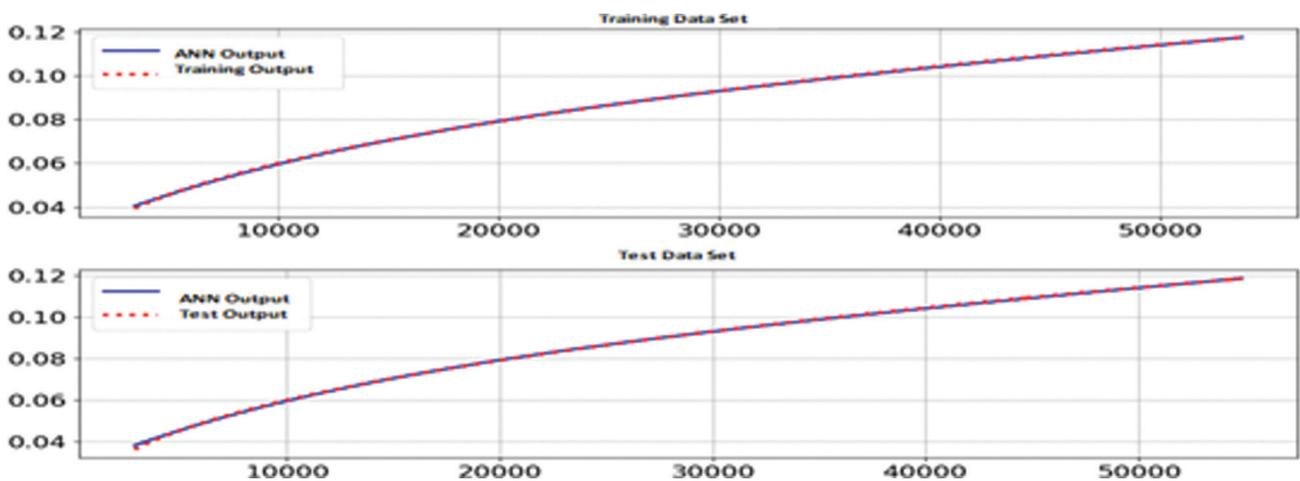


Figure 14. Comparison of ANN and actual results for carbon dioxide absorption coefficient.

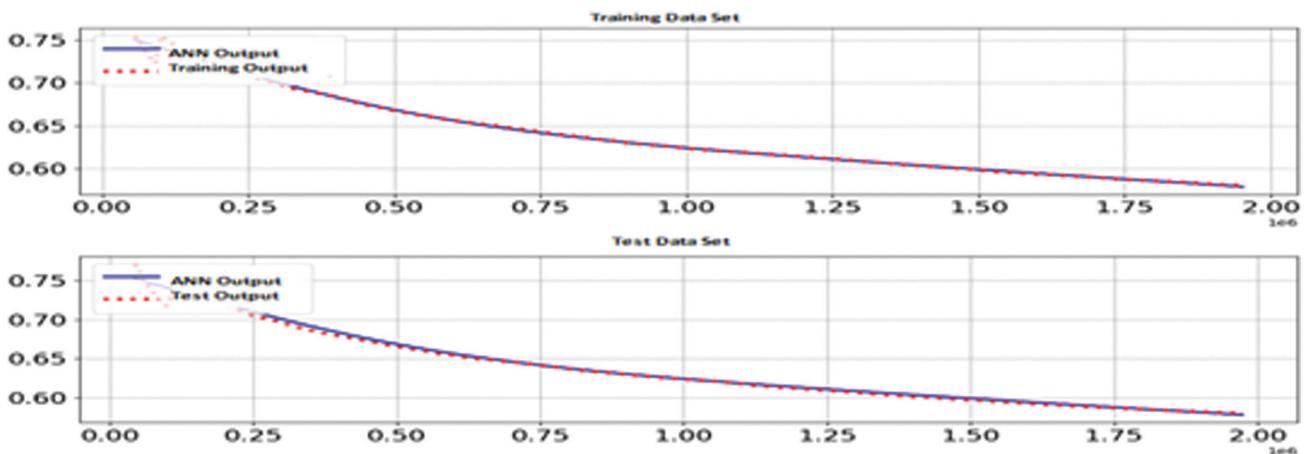


Figure 15. Comparison of ANN and actual results for atmospheric permeability.

The ANN and actual results data of heat flux values for Equation (16) are shown in Figure 16:

Based on these results, when the appropriate network model and learning algorithm are used, values that are very close to the expected output are obtained at the values that the network is not trained. It is understood that these values obtained can be used in making estimates close to fair results. Although some test value ranges were given outside the training range, it was observed that the deviation in the results was small and within an acceptable range.

**STATISTICAL RESULTS**

Based on the above results, the average RMSE (Root Mean Square Error), R<sup>2</sup>, Standard Error and Percent

Average Relative Error values of the actual results of the tested inputs with ANN results are given in the Table 1.

The artificial neural network's working output is based on the values in the tables, and it operates in such a way that the values are very similar to reality. In all BLEVE equation models, the 100 iteration sequence operation cycle was repeated with the maximum number of training iterations, and training data were learned. In some step models, it has been observed that learning occurs in shorter iterations without reaching 100 iterations, and values close to correct results are calculated. As mentioned before, some test data sets consist of values that are not included in the range of training data set. Although the ANN model is given values from the data range in which it is not trained, it is still found close to the correct result and in calculations where the deviations are low. In addition, even when ANN

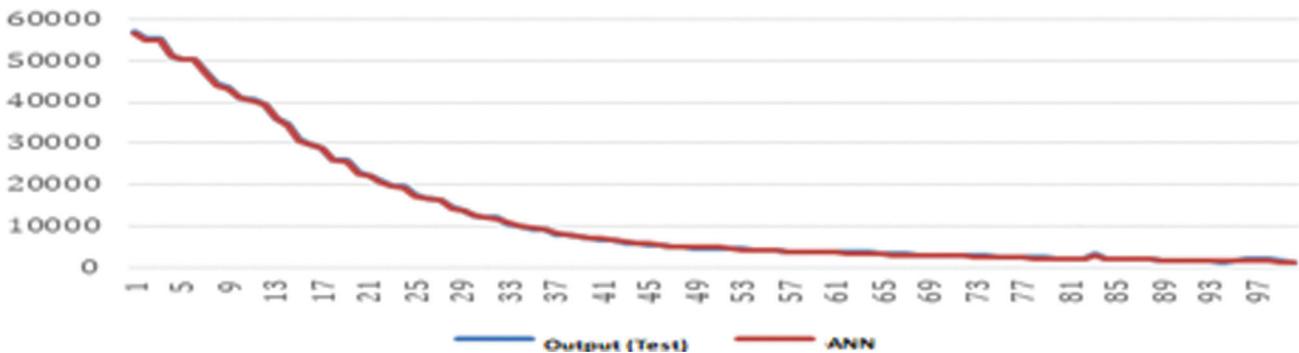


Figure 16. Comparison of ANN and actual results for heat flux.

Table 1. Statistics of BLEVE equations results

Equation Number of Model (BLEVE)	Number of TNO	Average RMSE	R <sup>2</sup>	Standard Error	Percent Average Relative Error
1		0,039163982	0,999971492895458	0,0411235632242887	0,031552896
2		0,002967298	0,99998083143435	0,0034308086321631	0,020005816
3		0,049857745	0,999990130334786	0,0484581557814802	0,019327986
4		0,713700999	0,999983528850211	2,17971187713268	0,285665103
5		0,002740433	0,994289429937661	0,0050259314252044	15,10272771
6		0,000283308	0,999945452141368	0,0003097671049413	0,075136241
7		0,063944124	0,999990602163493	0,0140819367989871	0,006120764
8		18972,94767	0,999997670104328	7,75013081087416	0,030182387
9		477,0849746	0,999962948591004	327,236987951567	0,079021082
10		7,715861269	0,999944204413832	3,97345474637233	0,190054267
11		1470,495478	0,999996430009117	1103,8849423202	0,267410229
12		0,000184794	0,999836466254009	0,0005979460889715	0,152032927
13		15,99473124	0,99999940532668	12,4670475992749	0,098403715
14		0,000247295	0,999767304056404	0,0003770205207641	0,505381419
15		0,002265918	0,990362242820547	0,0055103075234544	0,518838194
16		221,6802217	0,999885031745213	165,667346684379	3,343618219

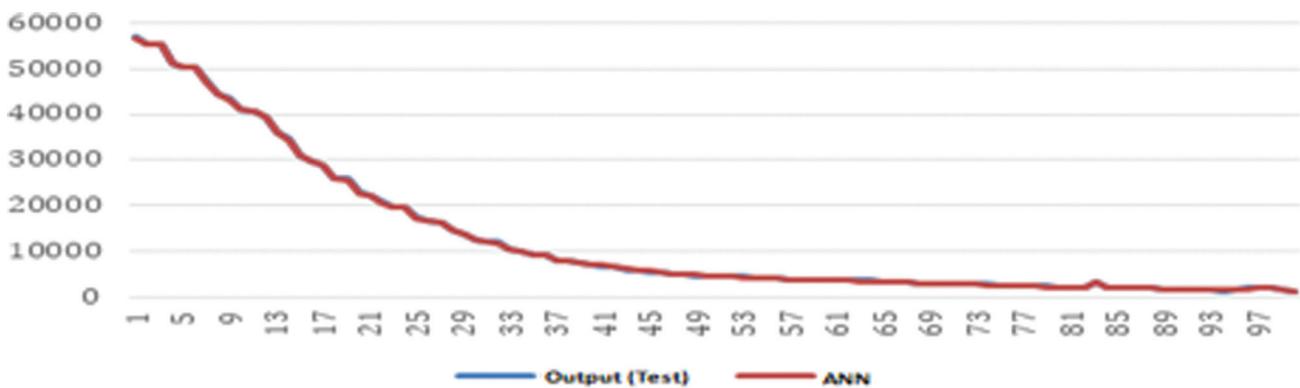


Figure 17. Comparison of ANN and actual results for 122 iterated heat flux.

Table 2. Comparative statistics of two heat flux results

Heat Flux Values	Average RMSE	R <sup>2</sup>	Standard Error	Percent Average Relative Error
100 Iterations	221,6802217	0,999885031745213	165,667346684379	3,343618219
122 Iterations	98,83887135	0,99997311320879	80,2869685205154	1,504089919

results are reprocessed as input, it has been observed that the results calculate values that are close to reality and with fewer deviations than expected.

## DISCUSSION

The iteration number of the heat flux values measured in the last step was consistently set to 100 in a consistent manner, as was the case in every step, and the results were analysed. However, in order to see the relationship and proportion of the results with the number of iterations, calculations were made with 250 and 500 iterations, respectively, in the last step. As a result of these additional calculations, it is seen that increasing the number of iterations increases the accuracy of the results up to a certain point, but it has been observed that it has no effect after the threshold value is exceeded. At this step, the threshold value was determined as 122. Giving the iteration number as 250 or 500 does not affect the result, but the ANN training and calculation reaches a conclusion in the 122nd iteration. In this context, increasing the iteration does not always increase the increase in accuracy but proportionally up to a specific threshold value.

In the Figure 17, there is a comparison of the data calculated with the 122 iterated heat flux artificial neural network.

The comparative statistical results of two heat flux values calculated with 100 iterations and calculated with 122 iterations are given in the Table 2:

As can be seen from the table, as the number of iterations increases, the value of R<sup>2</sup> gets closer to 1. The average RMSE value gets smaller, the standard error and the

percentage average standard relative error values are also decreasing. It gets closer to the real results.

When other studies are examined, it is not made to interpret all equations of a modelling such as BLEVE by processing them sequentially.

Barisik 2021 has prepared the artificial neural network modelling infrastructure and decided on input and output values for BLEVE calculations [29]. In this study, the measurement distance value was determined as input and thermal radiation calculations as output, and experiments were made in modelling. In other words, one input, one output, three hidden layers and three neurons in each layer were designed and the model was created. In the literature, Hemmatian et al. 2020 has his work in conjunction with a related analysis. The aim of this research is to use an artificial neural network to estimate BLEVE mechanical energy [30].

## CONCLUSION

After applying all of the equations in the BLEVE TNO model one by one, the final value, the heat flux value, was found using the Levenberg-Marquardt algorithm. The outputs produced in almost every step (processing the equations in order) were used as input to the next step. At each stage, the values found by the artificial neural network are used as test data in the input of the next equation without correction. As a result, the results produced by all ANN models are continuously processed, and the next stage has begun. In this way, the deviation rate and interaction between the values within the network itself have been controlled.

**Table 2.** The different thermal harm categories [34]

Thermal density (Kw/m <sup>2</sup> )	Thermal damage to the environment	Thermal harm to humans	Classification
37.5	Very serious damage to facilities	100% mortality in 1 minute and 50% mortality in 20 seconds	FATAL
25	Minimum energy to ignite wood	50% mortality in 1 minute	HEAVY LOSS
12.5	Plastic materials melting	1% death in 1 minute and Grade 1 burn in 10 seconds	MEDIUM DEGREE
4.0	Damage to PVC insulation materials	Pain occurrences of 20 seconds or more	LIGHT
1.6	No damage to equipment, buildings, etc.	Minimum pain threshold	HARMLESS

The deviation rate in each step is transferred to the next step in the same way. The deviation rates in each step from the first step to the last step of the ANN were also observed, the standard deviation in the last step was calculated, and the total deviation in its algorithm and the deviations between the steps were calculated.

As stated in the above section, calculation and training data learning was carried out by repeating 100 iterations of the results of almost all equations with the sequence operation cycle, that is, with the maximum number of training iterations. It has been observed that in some step models, learning takes place in shorter iterations before reaching 100 iterations, and values close to the correct results are calculated. Afterwards, higher iterations were applied. ANN models found close to the correct result in calculations with low deviations. In addition, it has been observed that even when ANN results are reprocessed as input, the results calculate values that are close to reality and have less deviations than expected.

As a result of the simulation studies, it is understood that these values obtained with the Levenberg-Marquardt (LM) algorithm can be used to make estimations close to realistic results. It has been observed that the deviation in the results is small and in the acceptable range.

The performance of the network can be affected by factors such as the network model used; the training algorithms prepared, the number of intermediate layers and neurons, and the number of samples [31, 32]. LM algorithm can be combined with some algorithms such as the genetic algorithm (GA) to achieve optimization suitability, higher success, calculate results in a shorter time, and high convergence. Realistic results can be predicted in fewer iterations [33].

Interpretation can be made about the effect degrees of the estimated heat flux value in the application made with ANN. In this way, it can be possible to take measures against adverse effects. According to the ranges of the heat flux values calculated in the TNO model, the degree of impact on the environment and people is given in the Table 3:

When the heat flux values in Table 3 are examined, for the BLEVE TNO YellowBook Static model, the values of

the artificial neural network modelled with the Levenberg-Marquardt algorithm are more than 37.5 kW / m<sup>2</sup>. Therefore, it is a condition that can have very serious and fatal consequences. However, factors such as increased measurement distance, gas outflow, decreased atmospheric permeability, and increased surface emission power cause the heat flux value to decrease. (Fig.16, Fig.17). Therefore, when the values in Table 3 are examined, the heat flux value of the modelling fell below 1.6 kW / m<sup>2</sup> after a certain period of time. Therefore, it is a harmless situation.

This study is aimed to be developed further and to contribute to other fields. Modelling of fire, explosion and gas dispersion events, which are major industrial accidents such as BLEVE, can be made. It will also contribute to reducing the negative environmental impacts of these accidents.

The desired goal will be reached in application areas, especially when the decision support mechanism is built, by applying these algorithms, which will provide faster and more accurate outcomes. Because the artificial neural network model will have more training sets, the results will be displayed directly in any scenario.

## NOMENCLATURE

$q''$	Heat flux, W/m <sup>2</sup>
$P_{wx}$	Vapor pressure of water formed in the air after fire, Pa
SEP	Surface emission power, W/m <sup>2</sup>
$F_{shape}$	Shape factor
$\Delta H$	Heat dissipation, j/kg
$F_s$	Heat release rate
$r_{fb}$	Radius ball shape of fire in BLEVE event, m
$H_{bleve}$	Height of the fireball from the ground, m
$c_9$	The constant coefficient, 3.24
$c_{10}$	The constant coefficient, 0.852
$X$	Distance between fireball center and object, m
$x_{bleve}$	The measurement distance between the area where the heat flux, m
$T_f$	BLEVE temperature, K
$T_a$	Air temperature, K

$H_{ua}$	Air humidity rate, %
$P_{wx}$	Vapor pressure, Pa
$P_{vCO_{2x}}$	Carbon dioxide partial pressure, Pa
$J(x)$	Jacobian matrix
$v^T$	Error vector
$w$	Weight
$R^2$	Coefficient of determination (Correlation)
Greek symbols	
$\alpha_x$	Water vapor absorption coefficient
$\alpha_c$	carbon dioxide absorption coefficient
$\tau_a$	Atmospheric permability
$\nabla^2 E$	Hessian matrix
$\nabla E$	Slope
$\beta$	Threshold weight
Subscripts	
$t$	Refers to time
$m$	Refers to mass

## AUTHORSHIP CONTRIBUTIONS

Authors equally contributed to this work.

## DATA AVAILABILITY STATEMENT

The authors confirm that the data that supports the findings of this study are available within the article. Raw data that support the finding of this study are available from the corresponding author, upon reasonable request.

## CONFLICT OF INTEREST

The author declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

## ETHICS

There are no ethical issues with the publication of this manuscript.

## REFERENCES

- [1] Oztemel E. Yapay Sinir Ağları. 4th ed. Istanbul: Papatya Bilim Yayıncılık; 2020. [Turkish]
- [2] Gurney K. An Introduction to Neural Networks. 1st ed. Florida: CRC Press; 1997. [CrossRef]
- [3] Fausett LV. Fundamentals of Neural Networks: Architectures, Algorithms and Applications. 1st ed. New Jersey: Prentice Hall; 1994.
- [4] Mazrooei-Sebdani R, Eskandari Z. Numerical detection and analysis of strong resonance bifurcations with a reflection symmetry and some applications in economics and neural networks. *Int J Bifurcat Chaos* 2020;30:2050100. [CrossRef]
- [5] Farman M, Tabassum MF, Naik PA, Akram S. Numerical treatment of a nonlinear dynamical Hepatitis-B model: an evolutionary approach. *Eur Phys J Plus* 2020;135:941. [CrossRef]
- [6] Tian J, Xu D, Zu J. Novel delay-dependent asymptotic stability criteria for neural networks with time-varying delays. *Comput Appl Math* 2009;228:133–138. [CrossRef]
- [7] Naik PA, Owolabi M, Yavuz M, Zu J. Chaotic dynamics of a fractional order HIV-1 model involving AIDS-related cancer cells. *Chaos Solit Fractals* 2020;140:110272. [CrossRef]
- [8] Ministry of Family, Labour and Social Services, Ministry of Environment and Urbanization and Ministry of Interior. Regulation on prevention and mitigation of major industrial accidents. Available at: <https://www.resmigazete.gov.tr/eskiler/2019/03/20190302-1.htm>. Accessed on Nov 20, 2020.
- [9] Roberts AF. Thermal radiation hazards from releases of LPG from pressurised storage. *Fire Saf J* 1981;4:197–212. [CrossRef]
- [10] Chamberlain GA. Developments in design methods for predicting thermal radiation from flares. *Chem Eng Res Des* 1987;65:299–309.
- [11] Crocker WP, Napier DH. Assessment of mathematical models for fire and explosion hazards of liquefied petroleum gases. *J Hazard Mater* 1988;20:109–135. [CrossRef]
- [12] Baker WE, Cox PA, Kulesz JJ, Strehlow RA, Westine PS. *Explosion Hazards and Evaluation*. 1st ed. New York: Elsevier Science Publishing; 1983.
- [13] Roberts T, Gosse A, Hawksworth S. Thermal radiation from fireballs on failure of liquefied petroleum gas storage vessels. *Process Saf Environ Prot* 2000;78:184–192. [CrossRef]
- [14] Prugh RW. Quantitative evaluation of “Bleve” hazards. *J Fire Prot Eng* 1991;3:9–24. [CrossRef]
- [15] Weterings RAPM, Duijm NJ, Bakkum EA, Mercx WPM, Van Den Berg AC, Engelhard WFJM, et al. *Methods for The Calculation of Physical Effects Due To Releases of Hazardous Materials (Liquids and Gases) - 3rd ed*. London: Yellow Book; 2005.
- [16] Center for Chemical Process Safety. *Guidelines for Consequence Analysis of Chemical Releases*. 1st ed. New York: Wiley-AIChE; 1999.
- [17] Birk AM, Cunningham MH. The boiling liquid expanding vapour explosion. *J Loss Prev Process Ind* 1994;7:474–480. [CrossRef]
- [18] Quesada A. 5 algorithms to train a neural network. Available at: [https://www.neuraldesigner.com/blog/5\\_algorithms\\_to\\_train\\_a\\_neural\\_network](https://www.neuraldesigner.com/blog/5_algorithms_to_train_a_neural_network). Accessed on Nov 20, 2022.

- [19] Rashid ZA, Alias AB, Hamid KHK, Bani MS, El Harbawi M. Analysis the effect of explosion efficiency in the TNT equivalent blast explosion model. In: Hashim MA, editor. International Conference on Global Sustainability and Chemical Engineering; 2014 Aug 20-22; Kuala Lumpur, Malaysia: Springer; 2014. pp. 381–390. [\[CrossRef\]](#)
- [20] Mishra KB. Multiple BLEVE's and fireballs of gas bottles: Case of a Russian road carrier accident. *J Loss Prev Process Ind* 2016;41:60–67. [\[CrossRef\]](#)
- [21] Crowl DA, Louvar JF. *Chemical Process Safety: Fundamentals with Applications*. 2nd ed. London: Pearson College Div; 2001.
- [22] Prashanth P, Krishnapillai S. A hybrid neural network strategy for identification of structural parameters. *Struct Infrastruct Eng* 2010;6:379–391. [\[CrossRef\]](#)
- [23] Litta AJ, Idicula SM, Mohanty UC. Artificial neural network model in prediction of meteorological parameters during premonsoon thunderstorms. *Int J Atmos Sci* 2013;525383. [\[CrossRef\]](#)
- [24] El-Hamid ASA, Eissa AH, Radwan AM. Levenberg – Marquardt's algorithm used for PID controller parameters optimization. *Int J Eng Res* 2015;6:286–290. [\[CrossRef\]](#)
- [25] Todorov Y, Terziyska M, Ahmed S, Petrov M. Fuzzy-neural predictive control using Levenberg-Marquardt optimization approach. In: IEEE International Symposium on Innovations in Intelligent Systems and Applications, IEEE INISTA; 2013 June 19-21; Albena, Bulgaria: IEEE; 2013. pp. 19–21. [\[CrossRef\]](#)
- [26] Ardic S. Determination of performance of centrifugal pumps by using artificial neural networks. Master Thesis. Eskisehir: Eskisehir Osmangazi University-Institute of Science; 2014.
- [27] Hashemipoor SS, Suratgar AA, Fard A, Hashemipoor SM. Designing a robust MEMS AC voltage reference source using artificial neural network. In: Li W, Zhou J, editors. 2009 2nd IEEE International Conference on Computer Science and Information Technology; 2009 Aug 8-11; Beijing, China: IEEE; 2009. pp. 184–187. [\[CrossRef\]](#)
- [28] Cavuslu MA, Becerikli Y, Karakuzu C. Hardware implementation of neural network training with Levenberg-Marquardt algorithm. *Bilgisayar Bilimleri ve Mühendisliği Dergisi* 2012;5:31–38. [Turkish]
- [29] Barisik T, Guneri AF. Calculation the environmental impacts of BLEVE using artificial neural networks. *Fresenius Environmental Bulletin* 2021;30:9611–9616.
- [30] Hemmatian B, Casal J, Planas E, Hemmatian B, Rashtchian D. Prediction of BLEVE mechanical energy by implementation of artificial neural network. *J Loss Prev Process Ind* 2020;63:104021. [\[CrossRef\]](#)
- [31] Tavana M, Abtahi AR, Di Caprio D, Poortarigh M. An artificial neural network and Bayesian network model for liquidity risk assessment in banking. *Neurocomputing* 2018;275:2525–2554. [\[CrossRef\]](#)
- [32] Efkolidis N, Angelos Markopoulos A, Karkalos N, Hernández CG, Talón JLH, Kyratsis P. Optimizing models for sustainable drilling operations using genetic algorithm for the optimum ANN. *Appl Artif Intell* 2019;33:881–901. [\[CrossRef\]](#)
- [33] Sakshi S, Kumar R. A neuro-genetic technique for pruning and optimization of ANN weights. *Appl Artif Intell* 2019;33:1–26. [\[CrossRef\]](#)
- [34] Wang KX, Qian X, He Y, Shi T, Zhang X. Failure analysis integrated with prediction model for LNG transport trailer and thermal hazards induced by an accidental VCE: A case study. *Eng Fail Anal* 2020;108:104350. [\[CrossRef\]](#)