

# Underwater localization system based on visible-light communications using neural networks

ALZAHRAA M. GHONIM,<sup>1,\*</sup> WESSAM M. SALAMA,<sup>2</sup> ABD EL-RAHMAN A. EL-FIKKY,<sup>3</sup>  
ASHRAF A. M. KHALAF,<sup>1</sup> AND HOSSAM M. H. SHALABY<sup>4,5</sup> 

<sup>1</sup>Department of Electrical Engineering, Faculty of Engineering, Minia University, Minia 61111, Egypt

<sup>2</sup>Department of Basic Science, Faculty of Engineering, Pharos University, Alexandria, Egypt

<sup>3</sup>College of Engineering and Technology, Arab Academy for Science, Technology and Maritime Transport, Alexandria 1029, Egypt

<sup>4</sup>Electrical Engineering Department, Faculty of Engineering, Alexandria University, Alexandria 21544, Egypt

<sup>5</sup>Department of Electronics and Communications Engineering, Egypt-Japan University of Science and Technology (E-JUST), Alexandria 21934, Egypt

\*Corresponding author: alzahraaghonim80@gmail.com

Received 13 January 2021; revised 27 March 2021; accepted 6 April 2021; posted 7 April 2021 (Doc. ID 419494); published 29 April 2021

Underwater localization using visible-light communications is proposed based on neural networks (NNs) estimation of received signal strength (RSS). Our proposed work comprises two steps: data collection and NN training. First, data are collected with the aid of Zemax OpticStudio Monte Carlo ray tracing software, where we configure 40,000 receivers in a 100 m × 100 m area in order to measure the channel gain for each detector in seawater. The channel gains represent the input data set to the NN, while the output of the NN is the coordinates of each detector based on the RSS intensity technique. Next, an NN system is built and trained with the aid of Orange data mining software. Several trials for NN implementations are performed, and the best training algorithms, activation functions, and number of neurons are determined. In addition, several performance measures are considered in order to evaluate the robustness of the proposed network. Specifically, we evaluate the following parameters: classification accuracy (CA), area under the curve (AUC), training time, testing time, F1, precision, recall, logloss, and specificity. The corresponding measures are as follows: 99.1% for AUC and 98.7% for CA, F1, precision, and recall. Further, the performance results of logloss and specificity are 7.3% and 99.3% respectively. © 2021 Optical Society of America

<https://doi.org/10.1364/AO.419494>

## 1. INTRODUCTION

Over the past years, the demands for high-speed underwater communication links have expanded due to the development of human applications in underwater communications. Having an accurate location definition is considered a vital approach in the communication field. In the same context, underwater exploration, offshore oil field discovery, deep submarine communication, disaster prevention through environmental monitoring, and military systems are considered among its significant applications.

Besides, there are colossal challenges in underwater wire communication installation due to high implementation costs and the nonexistence of flexibility for many underwater applications. Accordingly, there is an increasing interest in underwater wireless communication, which transmits data using wireless carriers.

Radio frequency (RF), acoustic, and optical communications are three famous techniques for underwater wireless communications. However, RF exhibits high attenuation in underwater

transmission. Consequently, it is suitable only for very short distances and data rates less than 1 Mbps. On the other hand, acoustic waves support ranges up to kilometers but with limited data rates up to kbps. Furthermore, acoustic waves consume power in a range of tens of watts, whereas RF power consumption is contingent on the distance and varies from milliwatts to hundreds of watts [1].

Underwater optical wireless communications (UOWC) use optical waves as transmission carriers. The optical link power consumption is in range from milliwatts to tens of watts, depending on the transmitter type. It can support high data rates up to Gbps. It is of low cost and considered the safest technology for marine life. However, UOWC suffers from severe absorption and scattering, which need to be addressed to lessen this effect. A lot of researchers are attempting to increase the current average link distance, which is just a few tens of meters up to 100 m [2]. Moreover, comprehensive surveys of UOWC have been introduced in many works to cover the technical background, e.g., [1].

Recently, localization is becoming granular to commercial and scientific communities. It involves many applications related to monitoring, surveillance, and tracking [3]. In addition, underwater localization is of special importance nowadays. Indeed, it is useful for underwater surveillance systems, tracking and studying oceanic animals, following climate changes, assisted navigation, industrial applications (offshore exploration), etc. [4].

Accordingly, in this paper, we focus on localization in underwater environments, specifically, to localize a diver or any object. The proposed system is characterized by the ability to find the exact coordinates of any object under the effect of absorption and scattering inside seawater. This system is featured with high precision, cheap cost, and low computational complexity, which enables the system's hardware feasibility.

In general, localization may be classified as range-based and range-free techniques [4]. Typically, the range-based schemes achieve higher accuracy than that of range-free schemes. Several technologies utilize range-based localization, e.g., time of arrival (TOA), angle of arrival (AOA), and received signal strength (RSS) techniques. Both TOA and AOA techniques provide high accuracy at the expense of high cost and complexity. On the other hand, the RSS technique provides medium accuracy but has low cost [3]. Consequently, RSS is utilized in this paper as an inexpensive technology for underwater localization.

Previous efforts on RSS localization included different environments. An underwater acoustic positioning system has been one of the forefront techniques. Specifically, in [5], Zhang *et al.* proposed an RSS-based underwater acoustic sensor network localization algorithm with stratification compensation. In [6], by analyzing the mobility patterns of water near the seashore, Zhang *et al.* proposed a localization method for underwater acoustic wireless sensor networks based on a mobility prediction and a particle swarm optimization algorithm. On the other hand, in [7], Saeed *et al.* proposed RSS-based localization technique for underwater optical wireless sensor networks.

Researchers have used Zemax OpticStudio as an MCRT simulator. Specifically, in [2], the authors performed different underwater channel scenarios using Zemax OpticStudio, and the results have shown to be consistent with experimental results. In [8], the authors used Zemax OpticStudio, combined with Zemax programming language (ZPL), for underwater dynamic channel modeling for single-input multiple-user scenarios. In addition, the statistical distributions have been obtained, which are required for bit error rate and outage probability analyses.

Recently, NNs have been successfully utilized in several spectra of applications with data-intensive robotics, tracking, navigation, object recognition, medical diagnoses, image processing, and other applications [9]. In a related context, many studies have taken advantage of NN hybrid with VLC systems to improve system performance [10–13]. In [10], Chaleshtori *et al.* studied the effect of training algorithms in an artificial neural network (ANN) equalizer in VLC systems using an organic light source. In [11], Ma *et al.* investigated the design and implementation of machine-learning-based demodulation methods in the physical layer of VLC systems. In [12], Irshad *et al.* developed a decision tree algorithm and examined it with different machine learning classifiers for indoor and underwater

localization in VLC networks. In [13], Alonso-González *et al.* proposed a fingerprinting indoor positioning estimation system based on NN to predict device position in a 3D environment.

In this paper, we perform an MCRT simulation using ZPL and Zemax by configuring 40,000 detectors, so that we measure the channel gains for each detector in seawater. Realistic configurations are taken into consideration by adding commercial transmitters and detectors as well as wavelength-dependent reflection coefficients and objects. Furthermore, new techniques with different strategies for underwater localization are proposed based on NNs. Specifically, the NN is performed as a predictive technique to estimate a 2D positioning system. That is, in order to estimate the Cartesian coordinates ( $x, z$ ) of a mobile device, a grid of RSSs is processed in the proposed system. The power-received signal is utilized as NN input. In order to acquire a superb proposed positioning system, four group of trails are presented to define the best training algorithm and the best activation function. First, an identity-activation function is processed with several training algorithms, specifically, stochastic gradient descent (SGD), adaptive moment estimation (Adam), and limited memory Broyden-Fletcher-Goldfarb-Shanno bound (L-BFGS-B) constraints. Next, a logistic-activation function and an ReLu activation function are executed, respectively, with the previously mentioned training algorithms. Finally, the previously mentioned training algorithm is performed using a tanh activation function. It turns out that the proposed positioning system features low complexity; therefore, it is appropriate to be integrated into mobile devices.

The rest of this paper is organized as follows: In Section 2, the simulation model used to generate the data set is introduced. Section 3 is devoted to the description of NN construction, activation function, and training algorithms used. Orange data mining and methodology are presented in Section 4. In Section 5, the results are discussed in order to evaluate the performance and robustness of the networks as well as nominate the best technique. Finally, concluding remarks are given in Section 6.

## 2. CONCEPT OF SIMULATION MODEL

Underwater visible light communications (UVLC) refer to data transmission in an aquatic environment using optical signals [14]. In this section, we specify a channel model for the geometry of an underwater environment. Specifically, a 3D simulation environment is created with determination of the dimensions and shape of the working environment. In addition, the properties and locations of the transmitter LEDs and receiver photodiodes are specified. In our channel modeling, we utilize Zemax OpticStudio combined with ZPL. The channel gains of 40,000 detectors in seawater of a 100 m<sup>2</sup> area are evaluated at a 7 m depth. Zemax OpticStudio determines the channel gains by tracing and collecting the energies of the detected rays. In our proposed technique, four CREE white LEDs filtered by a cyan color (490 nm) are placed on the sea surface. Each CREE lamp features a rectangular array of 10 × 10 LED chips [15]. Therefore, in order to simulate the light ray, an array of four lamps, each with 100 chips, is utilized; each chip also emits 500,000 rays. In addition, the seawater is modeled as a wavelength-dependent medium. The cyan color is utilized here,

**Table 1. Simulation Parameters Used in Zemax Solver**

Transmitter specifications	Source: Rectangular array with 4 LEDs Cree white LED filtered in cyan color [8,15] Power: 1 W Viewing angle: 120° [8] Number of rays per LED chip: 500,000 rays [8]
Receiver specifications	Total area: 100 m <sup>2</sup> Aperture area: 5 cm <sup>2</sup> [8] Field of view: 180° [8]
Type of water	Seawater

as it exhibits the minimum extinction coefficient in seawater [8]. The channel impulse response used in the Zemax OpticStudio solver is given by [2,16]

$$b(t) = \sum_{i=1}^{N_r} p_i \delta(t - \tau_i), \quad (1)$$

where  $p_i$  and  $\tau_i$  are the power and delay for the  $i$ th ray, respectively, and  $N_r$  is the total number of rays emitted by the LEDs. Accordingly, the DC channel gain  $H_0$  is defined as [2]

$$H_0 = \int_{-\infty}^{\infty} b(t) dt. \quad (2)$$

The total received power at the receiver is defined as [17]

$$P_r = P_t H_d(0) + \int_{\text{wall}} P_t H_{\text{ref}}(0), \quad (3)$$

where  $H_d(0)$  and  $H_{\text{ref}}(0)$  are the DC channel gains of the direct and reflected paths, respectively, and  $P_t$  is the total optical transmitted power by LEDs. In our study, we configure five reflections for all objects.

In order to estimate and detect the powers and path lengths from each source to each detector, the nonsequential ray tracing feature of Zemax OpticStudio is utilized. With the aid of ZPL, the received channel gain for each pixel is extracted. ZPL gives the possibility to divide the total detection area (10 m × 10 m) into smaller areas, each of dimension 50 mm × 50 mm. That is, it converts one large detector to tiny detectors. Consequently, we are able to measure the channel gains of 40,000 detectors. Besides, each power received in each detector enables us to obtain the localizations of 40,000 coordinates. Table 1 represents the simulation parameters utilized in Zemax OpticStudio.

On the other hand, UWOC scattering in seawater, which is due to the random nature of molecular motions, has a non-negligible effect [18]. Therefore, each detector area has a different channel gain because of various fading effects caused by the attenuation and scattering variations in UVLC. In addition, the reflection characteristics from both sea surface and sea bed have significant impacts on the channel gain. Therefore, all these effects are taken into consideration in our simulation. The total extinction coefficient  $C(\lambda)$  of seawater in Zemax OpticStudio is configured as

$$C(\lambda) = a(\lambda) + b(\lambda), \quad (4)$$

where  $a(\lambda)$  and  $b(\lambda)$  represent the absorption and scattering contributions, respectively, and  $\lambda$  is the transmission wavelength. Their utilized values are according to [19]. Furthermore, both shot and thermal noises are taken into considered at the receiver side [20].

### 3. NEURAL NETWORKS

Recently, there have been several machine learning methods that vary based on accuracy and computational requirements. NNs have become among the most efficient and robust methodologies for solving classification problems and pattern-recognition method preferable than many others, for instance, Enhanced J48 tree, ICT-Net, and scaled conjugate gradient (SCG) [21]. NNs have become one of the most efficient and robust methodologies for solving classification problems and pattern recognition. An NN predicts an output pattern after training it using input data samples. The NN consists of several layers of nodes. Specifically, an input layer (of  $n$  nodes),  $K$  hidden layers (with  $K \geq 1$ ), and an output layer. There are  $m_k$  nodes in the  $k$ th hidden layer,  $k \in \{1, 2, \dots, K\}$ . In addition, a set of biases  $\{b_i\}_{i=1}^K$  exists in the network structure. The nodes are connected through a set of links, each has weight  $w_{ji}$ ,  $j \in \{1, 2, 3, \dots, m_{k-1}\}$ ,  $i \in \{1, 2, 3, \dots, m_k\}$ . Here,  $m_0 \stackrel{\text{def}}{=} n$ , representing the number of nodes in the input layer. The weights are adjusted dynamically through the training process. In each node, the estimated output  $\hat{Y}_i$  is evaluated as

$$\hat{Y}_i = f \left( \sum_j w_{ji} \cdot x_j + b_i \right), \quad (5)$$

where  $f(\cdot)$  is an activation function. It should be noted that the outputs of one hidden layer are the inputs of the next hidden layer. Figure 1(a) shows the case of one hidden layer ( $K = 1$ ) and one output. The structure of a node is shown in Fig. 1(b).

There are several kinds of activation functions, e.g., Identity, Logistic, Tanh, and ReLU, which will be discussed later.

#### A. Training Stage

Training stage or learning process is the main step in NNs. It is an iterative operation based on minimizing a cost function  $J(\cdot)$  on the difference between the estimated output  $\hat{Y}$  and target output  $Y$ . Basically, it depends on adjusting both sets of weights  $w = \{w_{ji}\}$  and biases  $b = \{b_i\}$ :

$$J(\hat{Y}, Y) = f_{\text{cost}}(w, b). \quad (6)$$

When the cost function is minimum or the training process reaches a predefined number of iterations (epochs), the training stage is terminated.

#### B. Training Algorithms

The following training algorithms are utilized in our NN implementations. Stochastic gradient descent (SGD) [22], adaptive moment estimation (Adam) [23], and limited memory Broyden-Fletcher-Goldfarb-Shanno bound (L-BFGS-B) [24].

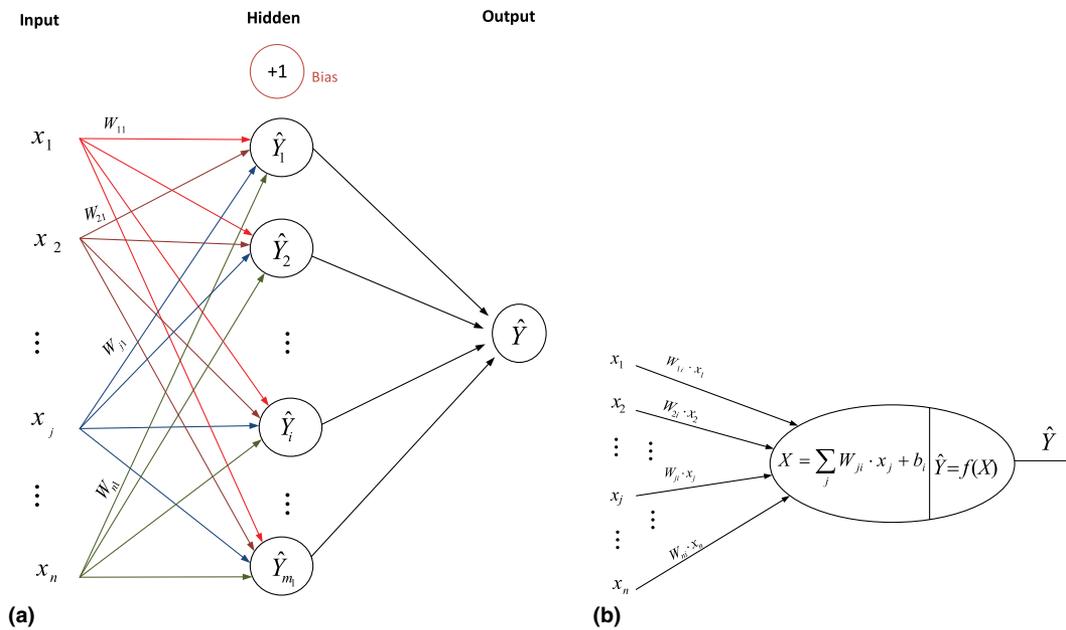


Fig. 1. (a) Neural network construction. (b) Basic structure of a node.

L-BFGS-B is considered an extension to the limited memory algorithm (L-BFGS). It has been utilized for optimizing large nonlinear problems; besides, it has the ability to deal with a variable's bounds. It relies on minimizing a nonlinear function  $G$  with  $n$  variable

$$\min U(G), \quad (7)$$

subject to simple bounds

$$l \leq G \leq u, \quad (8)$$

where  $U$  is a nonlinear function with available gradient  $g$ , and  $l$  and  $u$  vectors are the upper and lower variable bounds, respectively. Details of the algorithm and its mechanism are described in [25]. Specifically, at each iteration, a limited-memory BFGS approximation is updated to the Hessian. Moreover, this limited memory matrix is used to identify a quadratic model of the objective function  $U$ . Next, the search direction is estimated using two phases: first, defining the active variables set that utilize the gradient projection method. Subsequently, the quadratic model is minimized with reference to the free variables. The search direction is defined as the vector leading from current iteration to the approximate minimizer.

### C. Activation Functions

The identity activation function has a linear output:

$$f(x) = x. \quad (9)$$

Its range extends from  $-\infty$  to  $\infty$ . One of the main disadvantages of identity function is that it cannot be used in complicated models with several parameters.

To overcome the problem of linearity, there are many activation functions that execute nonlinearity transformations, e.g., Logistic, Tanh, and ReLU.

The logistic activation function, also called the sigmoid function, is given by

$$f(x) = \frac{1}{1 + e^{-x}}. \quad (10)$$

It is a suitable solution in the case of a probability prediction output model because its range is between 0 and 1. However, its output is not zero-centered.

At the first glance, the tanh activation function looks like a sigmoid:

$$\tanh(x) = \frac{e^{2x} - 1}{e^{2x} + 1}. \quad (11)$$

However, its performance is superior to a sigmoid because of its ability to map values between  $-1$  and  $1$ . Accordingly, it is the best choice in the case of classification between two classes. On the other hand, its output saturation is still a problem as in a sigmoid.

The ReLU activation is given by [26]

$$\text{ReLu}(x) = \max(0, x). \quad (12)$$

It plays an important role in overcoming the saturation issues (through the positive region only). In addition, its range extends from 0 to  $\infty$ .

## 4. ORANGE DATA MINING

Data mining is becoming a vital approach in machine learning due to its advantages and widespread applications in business, economics, health, science, and engineering. In this paper, we utilize Orange as our data-mining tool. Orange is an open-source data-mining tool in machine learning [27]. It is used as data analysis and data visualization with various toolboxes. In addition, Orange comprises diverse add-ons to extend functionality [28].

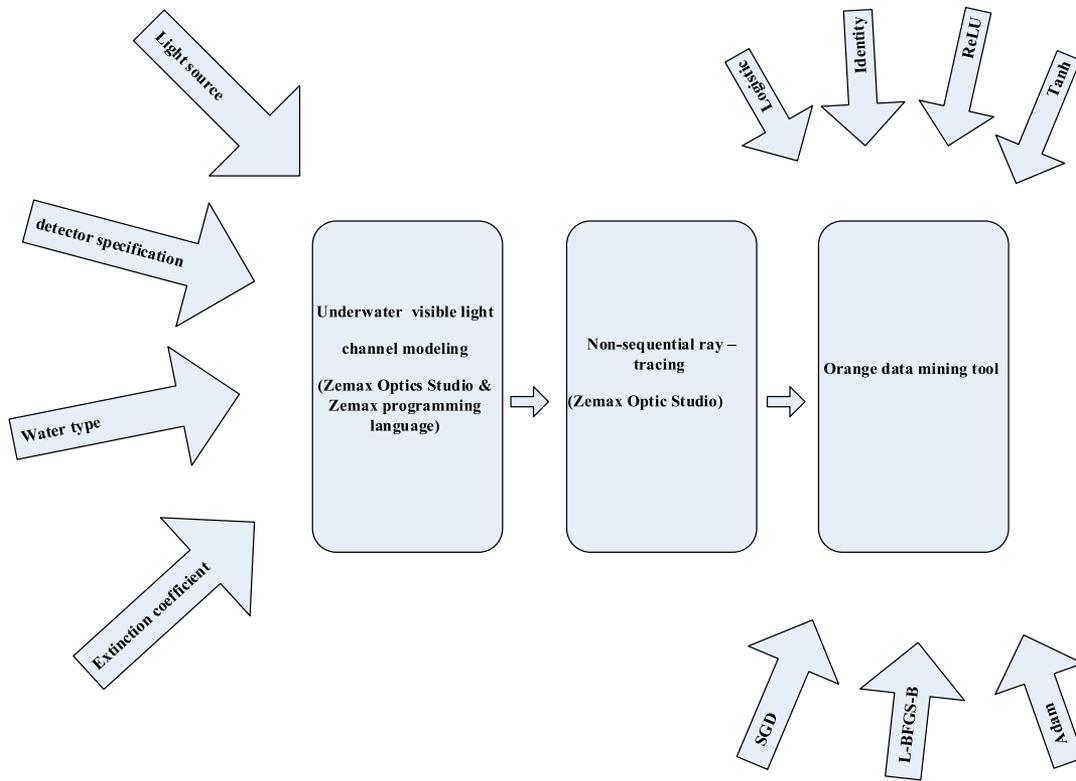


Fig. 2. Summary of major steps of proposed system.

**A. Methodology**

In this section, we present our methodology for obtaining the best NN configurations for target localization. It is split into two phases. The first phase is data set collection using Zemax software as discussed. The second phase is training an NN using the estimated data set through the Orange data-mining toolbox.

In our proposed technique, the data set is determined in a  $10 \times 10 \text{ m}^2$  underwater area in pure seawater at a depth of 7 m. The data set collection stage passes through many levels, as illustrated in Fig. 2, which provides an overall summary of the major steps followed in the adopted channel modeling methodology. Specifically, in the first step, we create a 3D simulation environment, where we specify the geometry of the underwater environment, the objects within it, the reflection characteristics of the surface materials, and the aspects of the light sources and detectors. In the second step, we apply the nonsequential ray tracing feature of Zemax to calculate the detected power and path lengths from source to detector for each ray. In the third step, we import this data to model the simulation environment, where we need to specify the dimensions and shape of the environment as well as the properties and locations of the transmitter (LED) and receiver photodiode. The source CREE LED is chosen with a cyan color to provide minimum extinction coefficient. The output 40,000 records of Zemax are used as the Orange data set input. The CAD model of seawater volume is obtained from [29].

Figure 3 explains the steps of our proposed technique in the stage of using the Orange toolbox.

After importing our data set from Zemax, a normalization technique, as a preprocessing step, is applied to RSS data so as

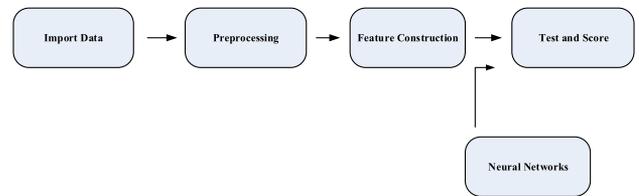


Fig. 3. Block diagram of proposed work utilizing Orange data mining.

to be centered to a mean  $\mu$  in order to improve it and reduce its redundancy. Next, to convert the training data from multivariable targets to single-variable targets, a new feature is formed using feature constructor widgets. This feature is built according to

$$X_1 = x^2 + z^2, \tag{13}$$

where it is utilized to convert the training data from points in the  $x-z$  plane to a line. Therefore,  $X_1$  is considered as the estimated target. Finally, NN widgets with several training algorithms and activation functions are utilized for training the data. Finally, NN widgets with several training algorithms and activation functions are utilized for training the data. Test and score as well as confusion matrix widgets are used as our evaluation metrics.

As aforementioned, a multilayer perceptron (MLP) neural network with one hidden layer is constructed. Our experiments include studying the effect of changing the number of neurons in the hidden layer. The number of neurons is changed from one to 10 neurons to acquire the best number. Several types of activation functions (Identity, Logistic, Tanh, and ReLU) are

**Table 2. SGD Learning Algorithm with Identity Activation Function**

No. of Neurons	Training Time [s]	Test Time [s]	AUC	CA	F1	Precision	Recall	Logloss	Specificity
1	34.073	0.044	0.989	0.987	0.987	0.987	0.987	0.149	0.993
2	23.972	0.052	0.991	0.987	0.987	0.987	0.987	0.073	0.993
3	24.078	0.036	0.991	0.987	0.987	0.987	0.987	0.073	0.993
4	16.497	0.036	0.991	0.987	0.987	0.987	0.987	0.072	0.993
5	17.387	0.035	0.991	0.987	0.987	0.987	0.987	0.072	0.993
...	...	...	...	...	...	...	...	...	...
10	16.844	0.036	0.991	0.987	0.987	0.987	0.987	0.073	0.993

**Table 3. Adam Learning Algorithm with Identity Activation Function**

No. of Neurons	Training Time [s]	Test Time [s]	AUC	CA	F1	Precision	Recall	Logloss	Specificity
1	31.495	0.04	0.991	0.987	0.987	0.987	0.987	0.071	0.993
2	25.247	0.034	0.991	0.987	0.987	0.987	0.987	0.071	0.993
3	25.885	0.034	0.991	0.987	0.987	0.987	0.987	0.073	0.993
4	21.267	0.039	0.991	0.987	0.987	0.987	0.987	0.073	0.993
5	18.731	0.045	0.991	0.987	0.987	0.987	0.987	0.073	0.993
...	...	...	...	...	...	...	...	...	...
10	19.654	0.039	0.991	0.987	0.987	0.987	0.987	0.073	0.993

performed. SGD, Adam, and L-BFGS-B are utilized as our solvers. We used 200 iterations in our experiments. In order to reap the advantage of using all the data sets for training and validation, stratified tenfold cross validation is used as our sampling technique in test and score widget [30]. In addition, our results are averaged over a set of three classes.

In order to validate the robustness of underwater localization application using NNs, different metrics are used, specifically, the training time, test time, AUC, CA, F1, precision, recall, logloss, specificity [31,32], and receiver operating characteristic (ROC) curve [33]. Furthermore, a confusion matrix is used to represent both proportions of prediction and number of instances for the neural network, besides computing all the used metrics through it. This is a list of rates that is often computed from a confusion matrix. Therefore, discrete widgets are used to divide the declares into three equal classes.

## 5. RESULTS

In order to obtain the superb robustness of our proposed technique, several trials, including varying training algorithms, activation functions, and number of neurons in the hidden layer, are performed in this section, and the results are presented.

### A. Identity Activation Function

In this subsection, we evaluate the performance when using NNs with an identity activation function and various training algorithms.

#### 1. SGD Training Algorithm

First, we adopt an NN with an SGD training algorithm and identity activation function. This NN is evaluated by changing the number of neurons from one to 10 neurons, and the results

are presented in Table 2, which shows that AUC, CA, F1, precision, and recall give high accuracy around 0.987 with a training time of 34.073 and one neuron in a hidden layer. However, the best results are obtained when using two neurons, where the AUC increases to 0.991. The values of both logloss and specificity are 0.073 and 0.993, respectively. The performance does not improve when increasing the number of neurons above two; however, the training time decreases a bit.

#### 2. Adam Training Algorithm

The same procedure is repeated using identity as the activation function, but Adam algorithm is used instead of SGD. The results are shown in Table 3. It is clear that the best results are obtained when using one neuron with a training time of 31.495 s and test time of 0.04 s. The other metrics are similar to the best results when using identity and SGD.

#### 3. L-BFGS-B Training Algorithm

Combining the identity activation function with the L-BFGS-B algorithm gives the same result but with one difference, i.e., a small training time (Table 4). Specifically, the training times are 2.137 and 3.057 s when using one and 10 neurons, respectively. Good results are obtained using two neurons in the hidden layer.

From the last three tables, we can determine the best training algorithm used with the identity activation function. Our selection metrics are based on number of neurons and training time. As can be seen, identity with SGD gives the best results with two neurons in the hidden layer at a training time of 23.972 s. While identity with Adam gives best results with one neuron in the hidden layer at a training time of 31.495 s. When combining identity with L-BFGS-B, the best results are obtained for two neurons in the hidden layer at a training time of 3.184 s.

**Table 4. L-BFGS-B Learning Algorithm with Identity Activation Function**

No. of Neurons	Training Time [s]	Test Time [s]	AUC	CA	F1	Precision	Recall	Logloss	Specificity
1	2.137	0.047	0.989	0.987	0.987	0.987	0.987	0.149	0.993
2	3.184	0.035	0.991	0.987	0.987	0.987	0.987	0.071	0.993
3	2.603	0.033	0.991	0.987	0.987	0.987	0.987	0.071	0.993
4	2.624	0.04	0.991	0.987	0.987	0.987	0.987	0.071	0.993
5	3.116	0.058	0.991	0.987	0.987	0.987	0.987	0.071	0.993
...	...	...	...	...	...	...	...	...	...
10	3.057	0.037	0.991	0.987	0.987	0.987	0.987	0.071	0.993

**Table 5. Adam Learning Algorithm with Logistic Activation Function**

No. of Neurons	Training Time [s]	Test Time [s]	AUC	CA	F1	Precision	Recall	Logloss	Specificity
1	125.467	0.045	0.992	0.987	0.987	0.987	0.987	0.075	0.994
2	58.662	0.06	0.992	0.987	0.987	0.987	0.987	0.073	0.994
3	53.329	0.067	0.991	0.987	0.987	0.987	0.987	0.072	0.994
4	40.75	0.058	0.991	0.987	0.987	0.987	0.987	0.072	0.994
5	34.917	0.061	0.991	0.987	0.987	0.987	0.987	0.073	0.994
...	...	...	...	...	...	...	...	...	...
10	31.466	0.077	0.991	0.987	0.987	0.987	0.987	0.072	0.994

**Table 6. SGD Learning Algorithm with Logistic Activation Function**

No. of Neurons	Training Time [s]	Test Time [s]	AUC	CA	F1	Precision	Recall	Logloss	Specificity
1	160.016	0.043	0.988	0.987	0.987	0.987	0.987	0.209	0.993
2	111.396	0.042	0.991	0.987	0.987	0.987	0.987	0.078	0.993
3	109.075	0.047	0.991	0.987	0.987	0.987	0.987	0.078	0.993
4	82.641	0.052	0.991	0.987	0.987	0.987	0.987	0.075	0.993
5	76.674	0.048	0.991	0.987	0.987	0.987	0.987	0.076	0.993
...	...	...	...	...	...	...	...	...	...
10	66.228	0.065	0.991	0.987	0.987	0.987	0.987	0.074	0.993

Consequently, the L-BFGS-B algorithm with the identity activation function performs superbly when compared with other algorithms with the same activation function.

## B. Logistic Activation Function

In this subsection, the three training algorithms are tested with the logistic activation function.

### 1. Adam Training Algorithm

First, the same previous procedure is performed using the Adam algorithm (the results are illustrated in Table 5). One of its distinguishing features is the high training and testing times compared with the other combinations, whereas the other results are almost same. The best results are acquired when using the first neurons. In the same context, according to Table 25 in [34], it is observed that the training time varies from 132.82 to 37.99 s, utilizing the logistic activation function, while our results obtain the training time from 125.467 to 31.466 s with more time stability. Note that the more neurons, the less the training time.

### 2. SGD Training Algorithm

Next, adding the logistic activation function to the SGD algorithm gives the results shown in Table 6. The results obtained are almost the same as the previous results, except for one difference. It has a high training time of 160.016 s, which is acquired with one neuron. However, the training time decreases with increasing the number of neurons and reaches 66.228 s for 10 neurons. The best results can be obtained when using two neurons with a training time of 111.396 s and test time of 0.042 s.

### 3. L-BFGS-B Training Algorithm

In order to exploit the merit of low computational time, the logistic activation function is used with the L-BFGS-B algorithm. Table 7 shows the corresponding results. It behaves similarly, despite the small training time, which value increases with increasing number of neurons and reaches 5.415 s for 10 neurons. The best results are gained when using two neurons with training and test times of 3.609 and 0.055 s, respectively.

The last three tables show that, with the logistic activation function, the best training time is 125.467 s with one neuron, 111.396 s with two neurons, and 3.609 s with two neurons

**Table 7. L-BFGS-B Learning Algorithm with Logistic Activation Function**

No. of Neurons	Training Time [s]	Test Time [s]	AUC	CA	F1	Precision	Recall	Logloss	Specificity
1	3.178	0.041	0.989	0.987	0.987	0.987	0.987	0.149	0.993
2	3.609	0.055	0.991	0.987	0.987	0.987	0.987	0.071	0.993
3	3.6	0.042	0.991	0.987	0.987	0.987	0.987	0.071	0.993
4	2.444	0.04	0.991	0.987	0.987	0.987	0.987	0.071	0.993
5	3.8	0.059	0.991	0.987	0.987	0.987	0.987	0.071	0.993
...	...	...	...	...	...	...	...	...	...
10	5.415	0.053	0.991	0.987	0.987	0.987	0.987	0.071	0.993

**Table 8. Adam Learning Algorithm with Tanh Activation Function**

No. of Neurons	Training Time [s]	Test Time [s]	AUC	CA	F1	Precision	Recall	Logloss	Specificity
1	79.98	0.045	0.991	0.987	0.987	0.987	0.987	0.072	0.993
2	37.067	0.057	0.991	0.987	0.987	0.987	0.987	0.072	0.993
3	39.432	0.051	0.991	0.987	0.987	0.987	0.987	0.072	0.993
4	23.947	0.037	0.991	0.987	0.987	0.987	0.987	0.072	0.993
5	27.558	0.039	0.991	0.987	0.987	0.987	0.987	0.072	0.993
...	...	...	...	...	...	...	...	...	...
10	24.009	0.039	0.991	0.987	0.987	0.987	0.987	0.072	0.993

**Table 9. L-BFGS-B Learning Algorithm with Tanh Activation Function**

No. of Neurons	Training Time [s]	Test Time [s]	AUC	CA	F1	Precision	Recall	Logloss	Specificity
1	2.888	0.026	0.989	0.987	0.987	0.987	0.987	0.149	0.993
2	3.393	0.055	0.991	0.987	0.987	0.987	0.987	0.071	0.993
3	2.512	0.055	0.991	0.987	0.987	0.987	0.987	0.071	0.993
4	2.673	0.038	0.991	0.987	0.987	0.987	0.987	0.071	0.993
5	3.031	0.053	0.991	0.987	0.987	0.987	0.987	0.071	0.993
...	...	...	...	...	...	...	...	...	...
10	4.229	0.061	0.991	0.987	0.987	0.987	0.987	0.071	0.993

for the Adam, SGD, and L-BFGS-B algorithms, respectively. Accordingly, the L-BFGS-B algorithm with the logistic activation function performs superbly when compared with the other algorithms with the same activation function.

### C. Tanh Activation Function

About the tanh activation function, the same mechanism is used in order to obtain the best training algorithm.

#### 1. Adam Training Algorithm

First, Table 8 shows that the preferable results are acquired when using one neuron with the Adam training algorithm. The NN behaves similarly but with a training time of 79.98 s and a test time of 0.045 s.

#### 2. L-BFGS-B Training Algorithm

As known, L-BFGS-B is famous for its low training time. Table 9 shows the results of it with the tanh activation function. It is clear that the best results are gained when using two neurons with a low training time of 3.393 s and a test time of 0.055 s.

It is clear from the table that, as the number of neurons increases, the training time increases.

#### 3. SGD Training Algorithm

Regarding the SGD training algorithm, the results are illustrated in Table 10. Note that superior results are obtained when using two neurons with training and test times of 44.525 s and 0.059 s, respectively. The training time is directly proportional to the number of neurons, as illustrated in the table.

By comparing the superior results of the tanh activation function with the three training algorithms, it is clear that the L-BFGS-B algorithm performs superbly when compared with the other algorithms with the least training time.

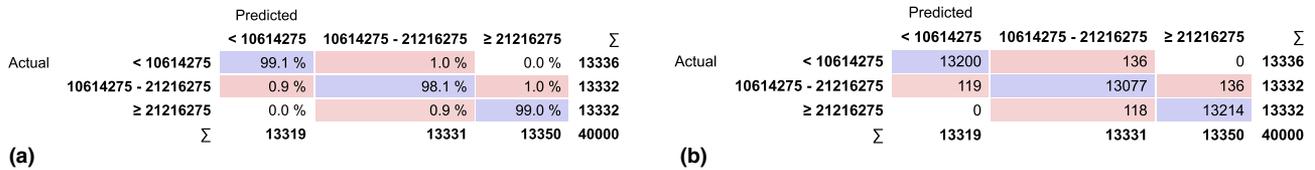
### D. Confusion Matrices

In order to show proportion of prediction and number of instances for NNs consisting of the identity and logistic activation functions with all mentioned learning algorithms, confusion matrices are utilized.

The results shown in Fig. 4(a) present these values as prediction proportions over three classes. The actual prediction ratios are 99%, 98.1%, and 99.1%, respectively. It should be noted

**Table 10. SGD Learning Algorithm with Tanh Activation Function**

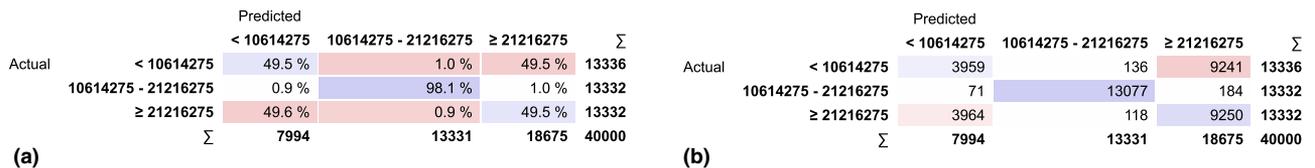
No. of Neurons	Training Time [s]	Test Time [s]	AUC	CA	F1	Precision	Recall	Logloss	Specificity
1	117.86	0.045	0.989	0.987	0.987	0.987	0.987	0.154	0.993
2	44.525	0.059	0.991	0.987	0.987	0.987	0.987	0.073	0.993
3	42.86	0.046	0.991	0.987	0.987	0.987	0.987	0.073	0.993
4	31.303	0.052	0.991	0.987	0.987	0.987	0.987	0.073	0.993
5	34.427	0.085	0.991	0.987	0.987	0.987	0.987	0.073	0.993
...	...	...	...	...	...	...	...	...	...
10	25.94	0.041	0.991	0.987	0.987	0.987	0.987	0.073	0.993



**Fig. 4.** Confusion matrices for NNS adopting identity, logistic, or tanh activation functions. (a) Proportion of prediction. (b) Number of instances.

**Table 11. Adam Learning Algorithm with ReLU Activation Function**

No. of Neurons	Training Time [s]	Test Time [s]	AUC	CA	F1	Precision	Recall	Logloss	Specificity
1	30.214	0.035	0.823	0.657	0.643	0.657	0.657	0.529	0.828
2	43.687	0.052	0.828	0.662	0.629	0.661	0.662	0.497	0.831
3	42.731	0.045	0.991	0.987	0.987	0.987	0.987	0.072	0.993
4	34.239	0.055	0.991	0.987	0.987	0.987	0.987	0.072	0.993
5	22.885	0.04	0.991	0.987	0.987	0.987	0.987	0.072	0.993
...	...	...	...	...	...	...	...	...	...
10	23.675	0.045	0.992	0.987	0.987	0.987	0.987	0.072	0.994



**Fig. 5.** Confusion matrices for NN adopting ReLU activation function and Adam algorithm, when using one neuron. (a) Proportion of prediction. (b) Number of instances.

that these values coincide for the three activation functions: identity, logistic, and tanh.

Figure 4(b) indicates the number of instances of the three classes corresponding to prediction proportion values.

**E. ReLU Activation Function**

In this subsection, the three training algorithms are tested with the ReLU activation function.

**1. Adam Training Algorithm**

The results of ReLU with the Adam algorithm are shown in Table 11. Using one or two neurons gives bad results, while the best results are acquired when using three neurons at training and test times of 42.731 s and 0.045 s, respectively. Increasing the number of neurons further does not improve the performance.

Unlike the previous states, the confusion matrix differs with every neuron number.

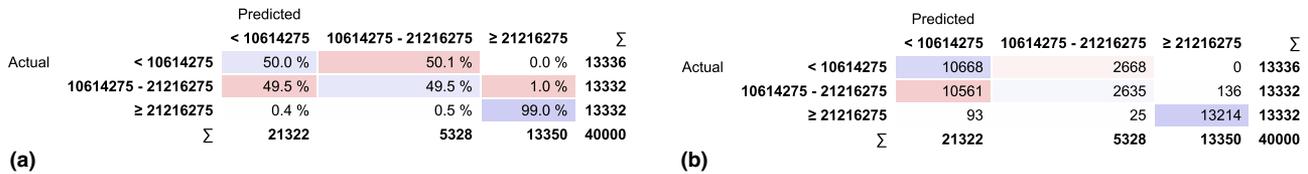
The actual prediction proportions acquired when using one neuron are 49.5%, 98.1%, and 49.5% for the three classes of data sets as shown in Fig. 5(a). Figure 5(b) illustrates the number of instances of the corresponding percentages.

The prediction proportions of processing with two neurons are 99%, 49.5%, and 50% for the three classes, respectively, as presented in Fig. 6(a). The corresponding number of instances are given in Fig. 6(b).

On the other hand, when using three neurons, the prediction proportions and corresponding number of instances similar to that are given in Figs. 4(a) and 4(b), respectively.

**2. L-BFGS-B Training Algorithm**

Next, the same procedure is carried out with the ReLU activation function and L-BFGS-B training algorithm. Table 12



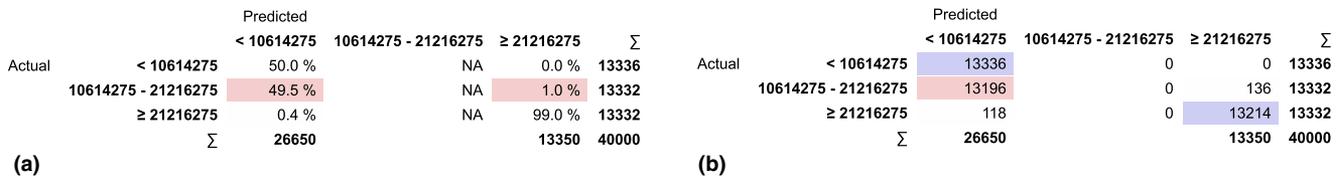
**Fig. 6.** Confusion matrices for NN adopting ReLU activation function and Adam algorithm, when using two neurons. (a) Proportion of prediction. (b) Number of instances.

**Table 12.** L-BFGS-B Learning Algorithm with ReLU Activation Function

No. of Neurons	Training Time [s]	Test Time [s]	AUC	CA	F1	Precision	Recall	Logloss	Specificity
1	1.752	0.034	0.822	0.657	0.547	0.492	0.657	0.529	0.828
2	2.644	0.04	0.827	0.663	0.552	0.496	0.663	0.497	0.831
3	3.97	0.057	0.991	0.987	0.987	0.987	0.987	0.071	0.993
4	3.96	0.039	0.991	0.987	0.987	0.987	0.987	0.071	0.993
5	3.691	0.044	0.991	0.987	0.987	0.987	0.987	0.071	0.993
...	...	...	...	...	...	...	...	...	...
10	4.385	0.04	0.991	0.987	0.987	0.987	0.987	0.071	0.993



**Fig. 7.** Confusion matrices for NN adopting ReLU activation function and L-BFGS-B (or SGD) algorithm, when using one neuron. (a) Proportion of prediction. (b) Number of instances.



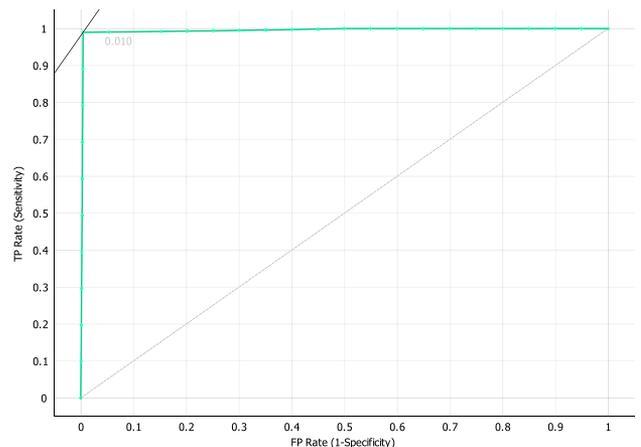
**Fig. 8.** Confusion matrices for NN adopting ReLU activation function and L-BFGS-B (or SGD) algorithm, when using two neurons. (a) Proportion of prediction. (b) Number of instances.

presents the results. The optimum results for this stage of work are acquired when using three neurons with training and test times of 3.9 and 0.057 s, respectively. The same conclusion associated with L-BFGS-B algorithm is extracted, i.e., a small training time that increases with increasing number of neurons.

Figure 7(a) shows the confusion matrices when adopting one neuron in the hidden layer, while the corresponding numbers of instances are illustrated in Fig. 7(b). Specifically, the prediction proportions are 49.5%, 98.1%, and NA for the three classes. Here, NA is due to the ReLU dying problem [35].

Figures 8(a) and 8(b) show the prediction proportions and corresponding numbers of instances, respectively, with two neurons in the hidden layer. We have 99%, NA, and 50% prediction proportions of the three classes.

On the other hand, for the iterations with three neurons, Figs. 4(a) and 4(b) give the results.



**Fig. 9.** ROC curve of L-BFGS-B learning algorithm with the identity activation function.

**Table 13. SGD Learning Algorithm with ReLU Activation Function**

No. of Neurons	Training Time [s]	Test Time [s]	AUC	CA	F1	Precision	Recall	Logloss	Specificity
1	42.72	0.043	0.822	0.657	0.547	0.492	0.657	0.53	0.828
2	53.042	0.049	0.828	0.663	0.552	0.496	0.663	0.499	0.831
3	66.112	0.047	0.989	0.987	0.987	0.987	0.987	0.076	0.993
4	46.973	0.051	0.991	0.987	0.987	0.987	0.987	0.073	0.993
5	21.926	0.04	0.991	0.987	0.987	0.987	0.987	0.073	0.993
...	...	...	...	...	...	...	...	...	...
10	29.786	0.039	0.991	0.987	0.987	0.987	0.987	0.073	0.993

**Table 14. Superior Results of L-BFGS-B Algorithm with Different Activation Functions**

Activation Function	Number of Neurons	Training Time [s]
Identity	2	3.184
Logistic	2	3.609
ReLU	3	3.97
Tanh	2	3.393

### 3. SGD Training Algorithm

The performance of the ReLU activation function with the SGD learning algorithm is shown in Table 13. The results show that the iterations when using three neurons perform superbly when compared with other neurons' iterations. The corresponding training and test times are 66.112 and 0.047 s, respectively.

Confusion matrices of runs using one and two neurons are given in Figs. 7(a) and 7(b) and Figs. 8(a) and 8(b), respectively. On the other hand, Figs. 4(a) and 4(b) illustrate the outputs of confusion matrices for the case of three neurons in the hidden layer.

Comparing the best results from all training algorithms with ReLU activation function, note that the training times (with three neurons) are 42.731, 3.97, and 66.112 s for Adam, L-BFGS-B, and SGD, respectively. Based on this, L-BFGS-B algorithm performs superbly compared with the others.

## F. Comparing Final Results

From the previous results, L-BFGS-B can be considered the best training algorithm because of its very low training time. To find the superior activation function, Table 14 shows the preferable results of the L-BFGS-B algorithm with different activation functions. It is clear that the identity activation function performs superbly when used with two neurons, where the training time is 3.184 s.

In a related context, the ROC curve is utilized to evaluate the best acquired results performance. Figure 9 indicates good separation of the L-BFGS-B algorithm.

Finally, a comparative study is done with other related works in view of the robustness of the training algorithm. This is indicated in the percentage accuracy shown in Table 15. Our result outperforms that of other referred works except for [13]. However, the latter suffers from an enormous computational time, which is 880.84 s for a data set of size around 762,432. Meanwhile, our acquired computational time is around 3.184 s for a data set of size 40,000.

**Table 15. Comparative Study with Related Works in Terms of Percentage Accuracy**

Reference	Training Algorithm	Percentage Accuracy
This work	Neural networks	98.7%
[12]	Enhanced J48 tree	98.15%
[13]	SCG	99.47%
[36]	ICT-Net	97.14%

## 6. CONCLUSION

In this paper, an underwater localization technique based on utilizing neural networks (NNs) and RSS values from LED lamps of a VLC network has been proposed. Our proposed technique is performed with the aid of Zemax ray tracing and Orange software through several trials, specifically, varying learning algorithms, activation functions, and number of neurons in one hidden layer. It has been observed that the L-BFGS-B algorithm always achieves the best performance with two or three neurons in the hidden layer and a high accuracy of about 98.7%. Moreover, the identity activation function performs superbly when combined with L-BFGS-B at a small training time of about 3.184 s with two hidden layers only. It turns out that the proposed positioning system features low complexity; therefore, it is appropriate to be integrated into mobile devices.

**Disclosures.** The authors declare no conflicts of interest.

**Data Availability.** Data underlying the results presented in this paper are not publicly available at this time but may be obtained from the authors upon reasonable request.

## REFERENCES

1. N. Saeed, A. Celik, T. Y. Al-Naffouri, and M.-S. Alouini, "Underwater optical wireless communications, networking, and localization: A survey," *Ad Hoc Netw.* **94**, 101935 (2019).
2. F. Miramirkhani and M. Uysal, "Visible light communication channel modeling for underwater environments with blocking and shadowing," *IEEE Access* **6**, 1082–1090 (2017).
3. J. Luo, L. Fan, and H. Li, "Indoor positioning systems based on visible light communication: state of the art," *IEEE Commun. Surveys Tuts.* **19**, 2871–2893 (2017).
4. H.-P. Tan, R. Diamant, W. K. Seah, and M. Waldmeyer, "A survey of techniques and challenges in underwater localization," *Ocean Eng.* **38**, 1663–1676 (2011).
5. B. Zhang, H. Wang, T. Xu, L. Zheng, and Q. Yang, "Received signal strength-based underwater acoustic localization considering stratification effect," in *OCEANS 2016-Shanghai* (IEEE, 2016), pp. 1–8.
6. Y. Zhang, J. Liang, S. Jiang, and W. Chen, "A localization method for underwater wireless sensor networks based on mobility prediction and particle swarm optimization algorithms," *Sensors* **16**, C1–C17 (2016).

7. N. Saeed, A. Celik, T. Y. Al-Naffouri, and M.-S. Alouini, "Underwater optical sensor networks localization with limited connectivity," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE, 2018), pp. 3804–3808.
8. A. E.-R. A. El-Fikky, M. E. Eldin, H. A. Fayed, A. A. E. Aziz, H. M. H. Shalaby, and M. H. Aly, "NLOS underwater VLC system performance: static and dynamic channel modeling," *Appl. Opt.* **58**, 8272–8281 (2019).
9. B. A. Erol, A. Majumdar, J. Lwowski, P. Benavidez, P. Rad, and M. Jamshidi, "Improved deep neural network object tracking system for applications in home robotics," in *Computational Intelligence for Pattern Recognition* (Springer, 2018), pp. 369–395.
10. Z. N. Chaleshtori, P. A. Haigh, P. Chvojka, S. Zvanovec, and Z. Ghassemlooy, "Performance evaluation of various training algorithms for an equalization in visible light communications with an organic led," in *2nd West Asian Colloquium on Optical Wireless Communications (WACOWC)* (IEEE, 2019), pp. 11–15.
11. S. Ma, J. Dai, S. Lu, H. Li, H. Zhang, C. Du, and S. Li, "Signal demodulation with machine learning methods for physical layer visible light communications: Prototype platform, open dataset, and algorithms," *IEEE Access* **7**, 30588–30598 (2019).
12. M. Irshad, W. Liu, L. Wang, and M. U. R. Khalil, "Cogent machine learning algorithm for indoor and underwater localization using visible light spectrum," *Wirel. Pers. Commun.* **116**, 993–1008 (2019).
13. I. Alonso-González, D. Sánchez-Rodríguez, C. Ley-Bosch, and M. A. Quintana-Suárez, "Discrete indoor three-dimensional localization system based on neural networks using visible light communication," *Sensors* **18**, 1040 (2018).
14. M. V. Jamali, P. Nabavi, and J. A. Salehi, "MIMO underwater visible light communications: Comprehensive channel study, performance analysis, and multiple-symbol detection," *IEEE Trans. Veh. Technol.* **67**, 8223–8237 (2018).
15. "Cree LEDs," <http://www.cree.com>. Accessed: January, 2018.
16. A. E.-R. A. El-Fikky, A. S. Ghazy, H. S. Khallaf, E. M. Mohamed, H. M. H. Shalaby, and M. H. Aly, "On the performance of adaptive hybrid MQAM-MPPM scheme over Nakagami and log-normal dynamic visible light communication channels," *Appl. Opt.* **59**, 1896–1906 (2020).
17. P. Chvojka, S. Zvanovec, P. A. Haigh, and Z. Ghassemlooy, "Channel characteristics of visible light communications within dynamic indoor environment," *J. Lightwave Technol.* **33**, 1719–1725 (2015).
18. C. D. Mobley, B. Gentili, H. R. Gordon, Z. Jin, G. W. Kattawar, A. Morel, P. Reinersman, K. Stamnes, and R. H. Stavn, "Comparison of numerical models for computing underwater light fields," *Appl. Opt.* **32**, 7484–7504 (1993).
19. G. Schirripa Spagnolo, L. Cozzella, and F. Leccese, "Underwater optical wireless communications: Overview," *Sensors* **20**, 2261 (2020).
20. A.-M. Cailean, B. Cagneau, L. Chassagne, V. Popa, and M. Dimian, "Evaluation of the noise effects on visible light communications using Manchester and Miller coding," in *International Conference on Development and Application Systems (DAS)* (IEEE, 2014), pp. 85–89.
21. G. Ou and Y. L. Murphey, "Multi-class pattern classification using neural networks," *Pattern Recogn.* **40**, 4–18 (2007).
22. L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of Computational Statistics (COMPSTAT)* (Springer, 2010), pp. 177–186.
23. D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, San Diego, California, 2015, pp. 1–15.
24. M. Martinez-Pabon, T. Eveleigh, and B. Tanju, "Optimizing residential energy management using an autonomous scheduler system," *Expert Syst. Appl.* **96**, 373–387 (2018).
25. C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, "Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization," *ACM Trans. Math. Softw.* **23**, 550–560 (1997).
26. R. Zaheer and H. Shaziya, "GPU-based empirical evaluation of activation functions in convolutional neural networks," in *2nd International Conference on Inventive Systems and Control (ICISC)* (IEEE, 2018), pp. 769–773.
27. A. H. Wahbeh, Q. A. Al-Radaideh, M. N. Al-Kabi, and E. M. Al-Shawakfa, "A comparison study between data mining tools over some classification methods," *Int. J. Adv. Comput. Sci. Appl.* **8**, 18–26 (2011).
28. J. Demšar, T. Curk, A. Erjavec, Č. Gorup, T. Hočevar, M. Milutinovič, M. Možina, M. Polajnar, M. Toplak, A. Starič, M. Štajdohar, L. Umek, L. Žagar, J. Žbontar, M. Žitnik, and B. Zupan, "Orange: data mining toolbox in Python," *J. Mach. Learn. Res.* **14**, 2349–2353 (2013).
29. "Grab CAD models," <https://grabcad.com>. Accessed: January, 2018.
30. S. Purushotham and B. Tripathy, "Evaluation of classifier models using stratified tenfold cross validation techniques," in *International Conference on Computing and Communication Systems* (Springer, 2011), pp. 680–690.
31. M. Daoud and M. Mayo, "A survey of neural network-based cancer prediction models from microarray data," *Artif. Intell. Med.* **97**, 204–214 (2019).
32. P. Ssekidde, O. Steven Eyobu, D. S. Han, and T. J. Oyana, "Augmented cwt features for deep learning-based indoor localization using WiFi RSSI data," *Appl. Sci.* **11**, 1806 (2021).
33. V. Mingote, A. Miguel, A. Ortega, and E. Lleida, "Optimization of the area under the roc curve using neural network supervectors for text-dependent speaker verification," *Comput. Speech Lang.* **63**, 101078 (2020).
34. I. Lorencin, N. Andelić, J. Španjol, and Z. Car, "Using multi-layer perceptron with Laplacian edge detector for bladder cancer diagnosis," *Artif. Intell. Med.* **102**, 101746 (2020).
35. L. Lu, Y. Shin, Y. Su, and G. E. Karniadakis, "Dying ReLU and initialization: Theory and numerical examples," *Commun. Comput. Phys.* **28**, 1671–1706 (2020).
36. B. Chatterjee and C. Poullis, "On building classification from remote sensor imagery using deep neural networks and the relation between classification and reconstruction accuracy using border localization as proxy," in *16th Conference on Computer and Robot Vision (CRV)* (IEEE, 2019), pp. 41–48.