

**Please cite the Published Version**

Zabala-Blanco, David , Azurdia-Meza, Cesar A., Lobos Soto, Benjamín, Soto, Ismael, Játiva, Pablo Palacios, Ahumada-García, Roberto and Ijaz, Muhammad  (2025) Extreme Learning Machine Models for Classifying the LED Source in a 2D Visible Light Positioning Database. IET Optoelectronics. ISSN 1751-8776

**DOI:** <https://doi.org/10.1049/ote2.70004>

**Publisher:** The Institution of Engineering and Technology Wiley

**Version:** Published Version

**Downloaded from:** <https://e-space.mmu.ac.uk/639685/>

**Usage rights:**  [Creative Commons: Attribution 4.0](https://creativecommons.org/licenses/by/4.0/)

**Additional Information:** This is an Open Access article published in IET Optoelectronics by the Institution of Engineering and Technology /Wiley

**Data Access Statement:** Data available on request from the authors.

**Enquiries:**

If you have questions about this document, contact [openresearch@mmu.ac.uk](mailto:openresearch@mmu.ac.uk). Please include the URL of the record in e-space. If you believe that your, or a third party's rights have been compromised through this document please see our Take Down policy (available from <https://www.mmu.ac.uk/library/using-the-library/policies-and-guidelines>)

ORIGINAL RESEARCH OPEN ACCESS

# Extreme Learning Machine Models for Classifying the LED Source in a 2D Visible Light Positioning Database

David Zabala-Blanco<sup>1</sup>  | Cesar A. Azurdia-Meza<sup>2</sup> | Benjamín Lobos Soto<sup>1</sup> | Ismael Soto<sup>3</sup> | Pablo Palacios Játiva<sup>4</sup> | Roberto Ahumada-García<sup>1</sup> | Muhammad Ijaz<sup>5</sup>

<sup>1</sup>Department of Computing and Industries, Universidad Católica del Maule, Talca, Chile | <sup>2</sup>Department of Electrical Engineering, Universidad de Chile, Santiago, Chile | <sup>3</sup>Department of Electrical Engineering, Universidad de Santiago de Chile, Santiago, Chile | <sup>4</sup>School of Informatics and Telecommunications, Universidad Diego Portales, Santiago, Chile | <sup>5</sup>School of Engineering, Manchester Metropolitan University, Manchester, UK

**Correspondence:** David Zabala-Blanco ([dzabala@ucm.cl](mailto:dzabala@ucm.cl)) | Cesar A. Azurdia-Meza ([cazurdia@ing.uchile.cl](mailto:cazurdia@ing.uchile.cl))

**Received:** 2 December 2024 | **Revised:** 11 April 2025 | **Accepted:** 16 April 2025

**Handling Editor:** Joaquin Perez (GE)

**Funding:** This work was supported by Project ANID Vinculación Internacional (FOVI240009).

**Keywords:** cameras | optical communication | terahertz waves

## ABSTRACT

In recent years, there has been a surge in interest in indoor positioning systems that use visible light communication (VLC) technology combined with light-emitting diodes (LEDs). These systems have gained attention because of their ability to offer high bandwidth, precise localisation, and potential for wireless communication to extend into the visible light spectrum in the future, making VLC a notable candidate. Furthermore, the visible light spectrum proves advantageous in the industrial internet of things setting, as it does not offer electromagnetic interference as in radio frequency (RF) spectrum. This paper analyses a database made up of approximately 356 image samples obtained from a CMOS sensor. The database encompasses eight distinct classes, each demonstrating frequency (bit rate) variations ranging from 1 to 4.5 kHz in 500 Hz increments. The aim is to implement this database for classification applications as a first stage with several neural networks based on extreme learning machines (ELM) in various forms: (1) standard ELM, (2) regularised ELM, (3) weighted ELM in two configurations, and (4) multilayer ELM with 2 and 3 hidden layers. The findings of this study reveal that standard ELM is particularly promising, achieving more than 99% in accuracy and G-mean, while maintaining low computational complexity (measured in tenths of seconds) when compared to convolutional neural networks and multilayer perceptrons, which offer superior performance, however at the cost of significant computational demands.

## 1 | Introduction

The increasing technology-driven requirement for high-speed indoor data transmission has become essential. Visible light communication (VLC) is a specific type of wireless optical communication technology that utilises the visible light spectrum of 400–800 THz to facilitate data transfer [1]. Known for its rapid speed and effectiveness, VLC offers a promising approach to meet the rising need for indoor connectivity [2].

An important use of VLC is visible light positioning (VLP), which leverages the current lighting systems to both convey data and accurately establish the position of devices within an indoor space [3–5]. This technique could be applied in a wide range of sectors, such as tracking assets in warehouses or identifying machinery locations in subterranean mines among others [6].

Nowadays, there is a growing trend to use machine learning in conjunction with evolutionary models [7]. For example, in 2D/

This is an open access article under the terms of the [Creative Commons Attribution](https://creativecommons.org/licenses/by/4.0/) License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2025 The Author(s). *IET Optoelectronics* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology.

3D location estimation within indoor VLP systems that utilise imaging multiple-input multiple-output (MIMO) [8]. Meanwhile, neural networks show promising results as equalisers in VLC [9]. In the realm of ELM, one example involves hybrid location algorithms that employ extreme learning machine (ELM) and clustering for positioning with a single light emitter diode (LED) and a single rotatable photodetector [10].

Currently, VLP shows potential in multiple areas, such as public places, industry, vehicle localisation, and hospital navigation [11]. In public places, such as museums and markets, navigation systems can be implemented to deliver proper information to visitors [12]. At the same time, a hybrid VLC and radio frequency (RF)-based parking automation system has also been introduced [13]. In industry, VLP can be applied in warehouses for product monitoring, worker localisation, and robot navigation; a received signal strength (RSS)-based VLP estimator for industrial internet of things has likewise been presented [14]. Vehicle localisation has been suggested for intelligent mobility via vehicle-to-vehicle and vehicle-to-infrastructure communication (using optical cameras [15]) [16], as well as for inter-vehicle tracking with CMOS cameras [17]. In electromagnetically sensitive environments, it has been adopted for hospital navigation [18].

Taxonomy of VLP systems is generally categorised into two parts: (1) software algorithms and (2) hardware components. In software algorithms, there are specialised lighting systems, which may consist of a single LED or a matrix of LEDs. For instance, with a single LED, fingerprinting can be employed to create an environmental snapshot that enables proximity estimation through triangulation and signal strength measurement. LED matrices can also be utilised forming robust systems that leverage not only signal intensity but also integrate QR code-based algorithms and computer vision techniques. For lights not modified in their surroundings, costs are reduced but only a limited number of algorithms, such as communication networks and computer vision analytics, are available. Concerning hardware, analysis involves the transmitter, modem, and receiver. White LEDs, fluorescent lights, and infrared lights commonly serve as transmitters. The modem evaluation includes investigating the device and its optical parts, such as lenses. The evaluation of the receiver centres around its photodiodes (PDs) and image sensors (cameras) [19].

One challenge faced by VLP systems is effective processing of incoming information while minimising uncertainty. To tackle this, machine learning-based approaches have been explored, including (1) multi-layer perceptron (MLP) and (2) Gaussian processes (GP), as well as the simulation of noise conditions, such as dust particles using wood shavings, which affects the signal uncertainty due to the position of the transmitters (TX). The work presented in ref. [20] highlights the potential applicability of MLP and GP techniques for VLP systems.

Within the previous framework, the current article examines the incorporation of machine learning via the ELM method, using data from the VLP database obtained from a master's thesis [21], which explored convolutional networks (CNN). Various forms of ELMs, including standard, regularised, weighted versions, and multilayer ELMs with 2 and 3 hidden neurons, are identified as

promising tools for efficiently handling the complex data of VLP systems. In assessing ELM's efficacy within VLP, the utilised database comprised 2840 samples alongside 8 distinct LED frequency types (spanning 1 to 4.5 kbps). These were derived from an experimental setup detailed in ref. [22], featuring an LED transmitter, an optical receiver, and an oscilloscope. In this work, for the moment, the task is to classify the samples by LED frequencies (bit-rates) in the indoor VLP setting (image classification and feature identification), thus not only identifying the light source (that can be useful for visible light communication and positioning system, where multiplexing is a mandatory task) but also simplifying the digital signal processing behind the VLP receiver and increasing the precision in regression task (i.e., considered for future research line).

On the one hand, the main objective is to evaluate and determine which of these variants better adapt to the presented scenario, offering a practical orientation on the most effective selection method to face the complexities associated with the VLP data. On the other hand, the main contributions of this article are the following: (1) The introduction of ELM approaches to classify and identify the source of the light (LED) in a VLP system (a classification application). (2) A metaheuristic optimisation procedure to find the hyperparameters that maximises the accuracy and G-mean by avoiding the increment of the training time for each ELM model. (3) The detailed comparison in terms of efficiency and complexity of ELM variants: standard, regularised, weighted, and multilayer. In this sense, the superior ELM approach is presented for the VLP scenario by allowing a very fast learning phase using a commercial computer. (4) Comparative results with CNN and MLP approaches are realised in terms of efficiency and complexity by adding confusion matrices to observe the type of error of classifiers. Once again, the superiority of the ELM technique is evident for all metrics.

The main novelty of this article is that it serves as an extension and enhancement of the work that the authors presented in ref. [23] following an invitation to this journal's special issue. It offers a more detailed explanation of the hyperparameter search for an ELM applied to multi-signal VLP frequency classification, where the inclusion of multilayer ELM is done, comparing its performance with conventional methods, such as CNNs and MLPs, and experimentally demonstrates the feasibility of ELM in a multiple-input single-output (MISO) system. At the same time, fundamentals, methodology, and optimisation of the ELM hyperparameters are carefully explained in the new version of the article in order not to leave the reader with frequently asked questions. This is especially significant for the scientific community, as it relies on a publicly available dataset, enabling continuous improvements by various research groups and eventual practical applications in emerging technologies—particularly in countries, such as Chile and elsewhere in Latin America, where most of this paper's authors are based.

Finally, the study confirms the viability of using a standard ELM (the simplest ELM model) that achieves high performance and low computational cost. This approach addresses a frequent limitation of machine learning models—namely complexity and resource usage—while maintaining cost-effectiveness and robust classification of VLP signals via ELM.

This document has the following organisation. Section 2 outlines the description of each ELM and the mathematical basis of the algorithms used in this work. Section 3 shows the methods and metrics used to evaluate the performance and complexity of the algorithms, as well as the database used in this article. Section 4 presents the optimisation of ELM hyperparameters, a detailed comparison of the performance of the different ELM-based methods, and benchmarking machine learning approaches. Section 5 ends with the main conclusions and future works.

## 2 | Extreme Learning Machine

An ELM is a machine learning algorithm used in the field of artificial neural networks. A distinguishing feature of an ELM is its focus on one hidden layer in neural networks [24]. Unlike traditional training methods that adjust hidden-layer weights through an iterative process, ELM initialises the weights randomly and then fixes them. Then it uses a single training process to calculate the weights of the network output layer. This means that the training time is significantly reduced since the hidden layer acts as a set of random characteristics and the output layer is efficiently trained using linear regression techniques. According to literature review, ELM has the following advantages over traditional methods based on the descent of gradients: (1) ELM can provide the best performance for generalisation in some cases and can learn faster than the popular learning algorithms used for feed-forward neural networks [25]. (2) ELM does not need to adjust the input weights or the parameters of the activation functions, which simplifies the learning process and reduces the risk of falling into local minimums. Next, the article explains the operation of the ELMs to be used: standard ELM, regularised ELM, weighted ELM in two versions, and multilayer ELM with 2 and 3 hidden layers. By considering electrical/telecommunication engineers cannot be experts in artificial intelligence area, this section will be carefully developed. The specialised literature (machine learning community) could skip reading this section.

### 2.1 | Standard ELM

Standard ELM is a learning algorithm for single-layer neural feedforward networks that randomly select input weights and analytically determine hidden layer output weights. The weights between the output layer and the hidden layer are calculated analytically with the generalised Moore–Penrose inverse [26]. The most basic type of neural network can be represented by the following equation:

$$f_L(x_j) = \sum_{i=1}^{\tilde{N}} \beta_i g(w_i x_j + b_i) = t_j, j = 1, \dots, N, \quad (1)$$

where  $N$  represents the set of random samples and  $\tilde{N}$  represents the number of hidden neurons;  $w_i = [w_{i1}, w_{i2}, \dots, w_{in}]^T$  is the weight matrix that connects the  $i$  –  $th$  hidden node with the  $j$  –  $th$  training example  $x_j$ ;  $b_i$  are the biases of the hidden layer;  $g(\cdot)$  symbolises the activation function;  $\beta_i = [\beta_{i1}, \beta_{i2}, \dots, \beta_{in}]^T$  are the weights of the output layer; and  $t_j$  represents labels

corresponding to  $x_j$ . The previous equation can also be written as follows:

$$H\beta = T, \quad (2)$$

where  $H$  is the output matrix of the hidden layer given by the following equation:

$$H = \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \dots & g(w_N \cdot x_1 + b_N) \\ \vdots & \ddots & \vdots \\ g(w_1 \cdot x_N + b_1) & \dots & g(w_N \cdot x_N + b_N) \end{bmatrix}, \quad (3)$$

$\beta = [\beta_1^T, \dots, \beta_N^T]^T$  is the matrix of the output weights and  $T = [t_1^T, \dots, t_N^T]^T$  is the matrix of the output results of the output layer.

It has been theoretically proved that single hidden layer feed-forward neuronal networks (SLFNs) are random hidden nodes that have a universal approximation capacity and that the hidden nodes can be generated randomly regardless of the training data [27]. Therefore, proportioning the training data is not necessary to intervene the  $H$  out matrix. Consequently, the ELM training is equivalent to finding the solution of a linear system of the  $\beta$ . The Barlett theory [28] describes that for obtaining a better performance of generalisation, the ELM must maximise the training mistake  $\|\epsilon\|$ , where  $\epsilon$  corresponds with the following equation:

$$\epsilon = H\beta - T. \quad (4)$$

To find  $\beta$ , the least square method can be used, which is a procedure to find the best parameters to minimise the difference between the desired output and the real outputs of the model, obtaining the following expression:

$$\beta = H^\dagger T, \quad (5)$$

where  $H^\dagger$  represents the generalised Moore–Penrose inverse of the output matrix in hidden layer  $H$ , which is possible to obtain through orthogonal projection [29] and can be obtained through the following expression:

$$H^\dagger = (H^T H)^{-1} H^T, \quad (6)$$

this expression is valid when the number of samples is greater than the number of hidden nodes ( $N > \tilde{N}$ ), which is the common situation by considering the vast number of samples in databases. However, when the number of samples is less than the number of hidden neurons ( $N < \tilde{N}$ ), the connections between the hidden and output nodes can be found in the following form:

$$\beta = H^T (H H^T)^{-1} T. \quad (7)$$

Equation (6), known as the standard ELM, is the version used in this work where the number of hidden neurons must be less than the number of learning samples, which is the most common situation by simplicity architecture and numerous datasets.

Finally, Figure 1 provides a visual representation of the ELM network for better compression purposes where inputs  $X$  with  $n$  attributes, number of neurons  $\tilde{N}$ , weights between the input and

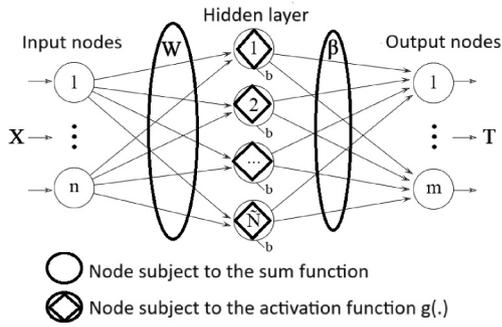


FIGURE 1 | ELM model along with its variables and constants.

hidden nodes  $W$ , biases of hidden neurons  $b$ , activation function  $g(\cdot)$ , weights between the hidden and output nodes  $\beta$ , and outputs  $T$  with  $m$  classes are also illustrated. As mentioned, in the training process (when the hyperparameters, the number of hidden neurons and the mapping function, must be fixed),  $W$  and  $b$  are generated arbitrary by following a uniform distribution for instance, and  $\beta$  is calculated based on the generalised Moore–Penrose inverse of the output matrix in hidden layer  $H$  (expression (6)). Namely, in the ELM algorithm there is no iterative process for establishing  $W$ ,  $b$ , and  $\beta$  as it is in traditional machine learning approaches.

## 2.2 | Regularised ELM

The regularised ELM [30] is a variant of the standard ELM that integrates a regularisation term to mitigate model complexity and prevent overfitting. Its primary aim is to achieve a balance between empirical risk, indicated by training error, and structural risk, which seeks to enhance class separation. The mathematical formulation of R-ELM allows for the adjustment of the trade-off between these risks through a regularisation parameter  $C$ . By optimising the associated Lagrangian, optimality conditions are derived, enabling efficient computation of the adjustment parameter  $\beta$ . As the value of  $C$  increases, R-ELM simplifies and approaches the standard ELM, which primarily focuses on minimising training error.

By developing the idea, ELM algorithm is based on the principle of minimising empirical risk, which implies reducing the real risk of an algorithm due to lack of knowledge of the distribution of data. Consequently, it focuses on evaluating the performance of a known set of training databases [31]. To solve the problem of overfitting, an algorithm with good generalisability is used, achieving an optimum balance between empirical risk and structural risk [32].

In this approach, the real risk is represented by the weighting of the sum of these two types of risk. The proportion between them can be regularised by introducing a weight factor, where the empirical risk is presented as the sum of the mean squared error and the structural error is defined through the derivative that maximises the distance of the separation margin between classes. Finally, to obtain a robust estimate that minimises interference between atypical values, the error variable is weighted [32]. Then, the following result is obtained:

$$\text{Minimizing : } \frac{1}{2}\|\beta\|^2 + \frac{1}{2}C\|D\varepsilon\|^2,$$

$$\text{according to : } \sum_{i=1}^{\tilde{N}} \beta_i g(w_i x_j + b_i) - t_j = \varepsilon_j, \quad (8)$$

$$j = 1, 2, \dots, \tilde{N},$$

where  $\varepsilon_j$  is the training error that regularises parameter  $C$ , which can be adjusted to the ratio between the empirical risk and the structural risk, where the optimal compensation between the two risks means an improvement in the model's performance, represented by  $\varepsilon[\varepsilon_1, \varepsilon_2, \dots, \varepsilon_{\tilde{N}}]$  and  $D = \text{diag}[v_1, v_2, \dots, v_{\tilde{N}}]$ . The Lagrangian for Equation (8) takes the form of:

$$L(\beta, \varepsilon, \alpha) = \frac{C}{2}\|D\varepsilon\|^2 + \frac{1}{2}\|\beta\|^2 - \alpha(H\beta - T - \varepsilon), \quad (9)$$

where  $\alpha_j \in R(j = 1, 2, \dots, \tilde{N})$  represents the lagrangian multiplier. Equation (10) is adjusted to balance empirical and structural risk through a scalar function (lagrangian). This adjustment is achieved by establishing the gradients for  $\beta$ ,  $\varepsilon$  y  $\alpha$  when matched to 0. The result of this process satisfies the optimal conditions posed by the Karush–Kuhn–Tucker theorem, which tackles the conditions of the optimisation problem by the restrictions of equality and inequality [32, 33]. Then, we have the following equation:

$$\begin{cases} \frac{\partial L}{\partial \beta} \rightarrow \beta^T = \alpha H, \\ \frac{\partial L}{\partial \varepsilon} \rightarrow C\varepsilon^T D^2 + \alpha = 0, \\ \frac{\partial L}{\partial \alpha} \rightarrow H\beta - T - \varepsilon = 0. \end{cases} \quad (10)$$

When solving the systems of equations that describe the connections between the hidden layer and the output layer, we get the following expression:

$$\beta = \left( \frac{I}{C} + H^T D^2 H \right)^{-1} H^T D^2 T, \quad (11)$$

where  $I$  is the identity matrix sized  $\tilde{N} \times \tilde{N}$ , where  $\tilde{N}$  is the number of hidden nodes as previously was established. In the scenario where  $D = \text{diag}(v_1, v_2, \dots, v_{\tilde{N}})$  is a unitary matrix,  $\beta$  is simplified to the following:

$$\beta = \left( \frac{I}{C} + H^T H \right)^{-1} H^T T, \quad (12)$$

This equation represents the regularised ELM (ELM-R) [32], where  $C$  is known as the regularisation parameter and must be a positive real number. Note that in the case of big  $C$ s, the R-ELM tends to be the standard ELM. For the regularised ELM, the architecture can be also represented by Figure 1. Compared to the explanation of the standard ELM, we must only considered that the weights of the output layer ( $\beta$ ) are given by Equation (12), where an additional degree ( $C$ ) of freedom exists (the regularisation parameter) and should be setting at the begging of the learning phase like the rest of the hyper-parameters.

### 2.3 | Weighted ELM

The Weighted ELM (ELM-W) [34] is a variant of the ELM that incorporates weights in the hidden layer to address class imbalance in datasets. Unlike the standard ELM, which employs fixed weights, the ELM-W utilises an adjusted weight vector that prioritises samples with higher training errors. Its formulation is like that of ELM-R but includes a weighting matrix  $W$ . When calculating the adjustment parameter  $\beta$ , this matrix is considered to enhance the model's ability to manage data disproportion, proposing formulae to define  $W$  based on the number of samples in each class. As can be seen below, the weighted ELM models differ in their methods for establishing these weights, with the former achieving balance through sample quantity and the latter applying principles of the golden ratio.

By reinforcing the explanation, ELM algorithm faces challenges when operating with unbalanced data, where the quantity of samples by class varies, leading to lower precision when working with these types of data sets. To address this problem, an ELM-W is introduced for the learning of unbalanced data, which was initially proposed for binary classification problems [34]. A diagonal matrix  $U$  is defined as the reciprocal of the samples, associating the weight with each sample in training. In general, if the sample comes from a minority class, the associated weight  $U_{ii}$  is relatively higher than the other method. To maximise the marginal distance and minimise the accumulated error weighted with respect to each sample, the following optimisation problem is proposed:

$$\text{Minimizing : } L_{PELM} = \frac{1}{2} \|\beta\|^2 + CU \frac{1}{2} \sum_{i=1}^{\tilde{N}} \|\varepsilon_i\|^2, \quad (13)$$

$$\text{according to : } h(x_i)\beta = t_i^T - \varepsilon_i^T, \quad i = 1, \dots, \tilde{N},$$

where  $\varepsilon_i$  is the training error of sample  $x_i$ , the result of the difference between hoped output  $t_i$  and the actual output  $h(x_i)\beta$  [34]. According to Karus-Kush-Tucker theory [33], the equivalent dual optimisation problem is expressed as follows:

$$L_{DELM} = \frac{1}{2} \|\beta\|^2 + CU \frac{1}{2} \sum_{i=1}^{\tilde{N}} \varepsilon_i^2 - \sum_{i=1}^{\tilde{N}} \alpha_i (h(x_i)\beta - t_i + \varepsilon_i), \quad (14)$$

where the Lagrange multiplier  $\alpha_i$  is the constant factor of sample  $x_i$  in the linear combination to form the final decision function. By deriving in regard to variables  $(\beta, \varepsilon_i, \alpha_i)$  and setting equal to 0, you get the following:

$$\begin{aligned} \beta &= H^T \alpha, \\ \alpha_i &= CU \varepsilon_i, \\ h(x_i)\beta &= t_i + \varepsilon_i, \end{aligned} \quad (15)$$

from these equations two versions are derived from  $\beta$ , depending on the number of samples and the number of hidden nodes [34]. As previous ELM, in the common situation where the number of samples is greater than the number of hidden neurons, the right pseudo-inverse is applied, obtaining:

$$\beta = \left( \frac{I}{C} + H^T U H \right)^{-1} H^T U T. \quad (16)$$

It is crucial to highlight that the weighted ELM depends on the definition of the elements in the matrix  $U$ . Therefore, Weighted ELM 1 (ELM-W1) and Weighted ELM 2 (ELM-W2) are introduced as the most widely used solutions. In these cases, it is necessary to define  $U$ , determining which type of ELM will be used;  $U_1$  represents weighted 1 and  $U_2$  weighted 2. In the first case, the weighted scheme is defined as follows:

$$W1: U_{ii} = \frac{1}{(at_i)}, \quad (17)$$

where  $at_i$  is the number of samples that belong to  $t_i$ ,  $i = 1, \dots, m$ . This type of ELM is ideal when the database is mostly slanted towards the border of the majority class. When the database is mostly slanted towards the minority class border, ELM-W2 is used, whose diagonal matrix takes the shape of the following:

$$W2: \begin{cases} U_{ii} = \frac{0,618}{a(t_i)} & \text{si } t_i > \text{mean}(t_i), \\ U_{ii} = \frac{1}{a(t_i)} & \text{in another case,} \end{cases} \quad (18)$$

minimising the balance step in a 0.618:1 ratio. Note that it takes the value of the golden ratio, since it represents perfection in nature, between the minority and the majority classes [34]. The architecture of the weighted ELM can be identified by Figure 1. As happens with the regularised ELM, the change to be noted with respect to the standard ELM comes to be the determination of  $\beta$ . In this occasion, expression (16) must be taking into account, where the diagonal matrix  $U$  appears and it is given by Equations (17) and (18) according to the version of the weighted ELM. Based on previous explanations, the ELM algorithms in their learning phase are presented below (Algorithm 1):

---

#### ALGORITHM 1 | ELM Training Algorithm.

---

Given the training set  $\phi = \{(x_i, t_i) | i = 1, \dots, M\}$  where  $M$  is the number of samples, set the hyper-parameters of the ELM model, including the activation function  $g(\cdot)$ , the number of hidden neurons  $N$ , the regularisation parameter  $c$ , among others.

1: Originate the input weights  $w_i$  and biases  $b_i$  by following a random distribution.

2: Compute the output of the hidden layer matrix  $H$  by exploiting input information  $x_j$ , see Equation (3).

3: Calculate the output weights  $\beta$ , via expressions (6), (12), and (16) for the standard, regularised, and weighted ELMs, respectively. Here, the output information  $t_j$  is also used.

---

The testing phase is only given by probe the ELM model for new database (a testing set), all connections among the nodes remain the same as those configured in training stage.

### 2.4 | Multilayer ELM

The multilayer extreme learning machine (ELM-M) [35] is an advanced version of the standard ELM, specifically designed to handle larger and more complex datasets through a multi-layer neural network architecture. This model employs ELM autoencoders (ELM-AE) [35] to achieve unsupervised learning in a

layer-wise manner, effectively compressing, dispersing, or maintaining the dimensionality of input features. The ELM-AE utilises orthogonal random weights and biases to project input data into a transformed space, calculating its output matrix via least squares operations. Unlike traditional deep neural networks that necessitate comprehensive training and weight tuning, the ELM-M initialises the hidden layer using an ELM-AE and directly optimises its output, employing activation functions that are contingent on the relationship between the number of nodes in successive layers. This approach not only accelerates the training process but also enhances the ELM-Ms ability to learn and represent complex patterns within the data effectively.

In other words, the ELM-M comes to be an evolution of ELM, designed to create neural networks that process large and complex data sets by using stepwise unsupervised learning, similar to deep neural networks [36]. It works based on the ELM-AE, which uses singular value features to structure the multilayer network.

ELM-AE may represent input features in three ways: compressed, sparse, and equal. The compressed representation reduces the dimension of the data, the sparse one increases it, and the equal one keeps it constant. For unsupervised learning, it uses the input data as output, choosing orthogonal weights and random biases for the hidden nodes, optimising the performance of the model in the following form [37]:

$$\begin{aligned} h &= g(w \cdot x + b), \\ w^T w &= I, b^T b = 1, \end{aligned} \quad (19)$$

where  $w = [w_1, \dots, w_L]$  are the orthogonal random weights, and  $b = [b_1, \dots, b_L]$  are the orthogonal random biases between the input and hidden nodes. This projection is based on the Johnson-Lindenstrauss principle, which allows estimating points into a lower-dimensional space without losing significant information about their distances. In an ELM-AE, the output weight matrix  $\beta$  learns the transformation from the feature space to the input data. If the data is sparse or compressed, the matrix  $\beta$  is determined as follows:

$$\beta = \left( \frac{I}{C} + H^T H \right)^{-1} H^T X, \quad (20)$$

where  $H$  represents the output matrix of the hidden layer of the ELM-AE, and  $X$  are the input data and the target output results of the output layer. Expression (20) holds as long as the number of samples ( $N$ ) is greater than the number of neurons ( $\tilde{N}$ ), which is very common in the context of machine learning and even more so in deep learning applications. For simplicity purposes and for visualising the optimisation procedure (refer to Section 4), we set  $C$  to infinity. In other words the ELM-AE considers an ELM standard.

As mentioned, ELM-M differs from conventional deep neural networks by using ELM-AE for weight initialisation, eliminating the need for exhaustive fine-tuning [36]. In an ELM-M, the hidden layer activation function can be either linear or piecewise nonlinear, depending on whether the number of nodes in the current layer  $\tilde{N}^k$  is equal to that in the previous

layer  $\tilde{N}^{(k-1)}$ ; if they are equal, a linear function is used, and if they are different, a nonlinear function such as sigmoid is chosen, thus allowing it to adapt to various data structures. The ELM-M equation can be defined as follows [38]:

$$H^k = g\left(\left(\beta^k\right)^T H^{k-1}\right). \quad (21)$$

The output matrix of the  $k$ -th hidden layer is denoted as  $H^k$ , the input layer  $X$  can be regarded as the 0th hidden layer, where  $k = 0$ . The result of the connections between the last hidden layer and the output node  $t$  (the target) is calculated analytically using the regularised least squares method, that is, the method proposed by ELM standard.

### 3 | Materials and Methods

Here, the experimental set-up used for performing statistically reliable measurements and, consequently, to execute the results and discussion will be presented.

#### 3.1 | Database

The methodology (experimental setup for image sampling and the resulting database) used for the experimental sampling can be found in ref. [22]. The database is created for VLP. This represents of photographs of the ceiling with the camera facing up and diverse roll angles. The images are considered over a grid with 5 regularly spaced reference points on the floor area. The receiver corresponds to a Sony IMX219 CMOS sensor model, which obtained an average of 356 images by LED class. During the dataset acquisition, the image sensor was positioned parallel to the floor at a height of 25.6 cm facing upwards. The camera is rotated 5° between photographs and 65 images were acquired at points (1.623,1.260,0) m, (1.873,1.260,0) m, (2.123,1.260,0) m, (2.373,1.259,0) m, and (2.623,1.259,0) m on the floor surface. The relationship between the 5 degrees and 65 images comes to represent the amount of images by class ( $5 \times 65 = 325$ ). Its rotation is very relevant in VLP experiments, since this phenomenon will happen in the reality due to random movements of the receiver simply. In this form, the results obtained and discussion given can be considered valid. Note that the VLP dataset belongs to a bidimensional space since it discards dimension of the height. The database always takes into account that the receptor in the floor, by considering that the ceiling of the experiment surface is 2.71 m. To conclude, we here present the most representative information provided by the database builders. For more details regarding the experiment space and the positions of the transmitter and the receiver with its movement, one could write direct to authors of refs. [22, 39]. In fact, in these, there are figures that represent the VLP system with details, which are not presented here by copyright reasons. To evaluate ELM, we used the database available for download in ref. [39], which contains 2839 samples, where all images are  $150 \times 150$  pixels. The configuration includes eight transmitters modulated by On-Off Keying (OOK), frequencies (data-rates) that go from 1 to 4.5 kHz (1–4.5 kbps), and an increase between 500 Hz

(0.5 kbps) LEDs, which will be the type of class (8 in total), as we can see in Table 1. For clarification purposes, the VLP signal is composed by an illumination signal, which is a DC voltage that can turn on the LED (according to its requirements) and a communication signal, which is an OOK signal characterised by a data-rate. In order to note each LED and, therefore, set up a logical experiment for VLP, each communication signal has a particular bit-rate as can be seen in Table 1. It is worth to note that this range of data-rate are destined for low internet of things or industry applications, which demands precise location and real-time tracking in indoor environments, such as mobile warehouse robots [22].

As can be seen, the dataset can be considered balanced since all classes almost have the same number of samples. In this sense, it is expected that weighted ELMs do not present superior performances without computational cost. These results could be verified in the next section. Due to their differentiating feature, each LED light presents a different pattern in the image obtained by the receiver. The images were processed with Matlab, rescaled to 50% (a factor of 2) in order to minimise the computational cost without loss the performance of the proposal, vectored to a row vector for processing purposes with the ELM algorithms, and normalised from  $-1$  to  $1$  to ensure and optimise the ELM learning as recommend its creator [24]. In Figure 2, a random sample of each class is displayed in order to

TABLE 1 | Luminary classes and their modulation bit-rates.

Tx	Bitrate (kbps)	Class	Number of samples
Tx13	1	7	353
Tx7	1.5	3	354
Tx11	2	5	356
Tx4	2.5	2	356
Tx14	3	8	351
Tx8	3.5	4	356
Tx12	4	6	355
Tx3	4.5	1	358

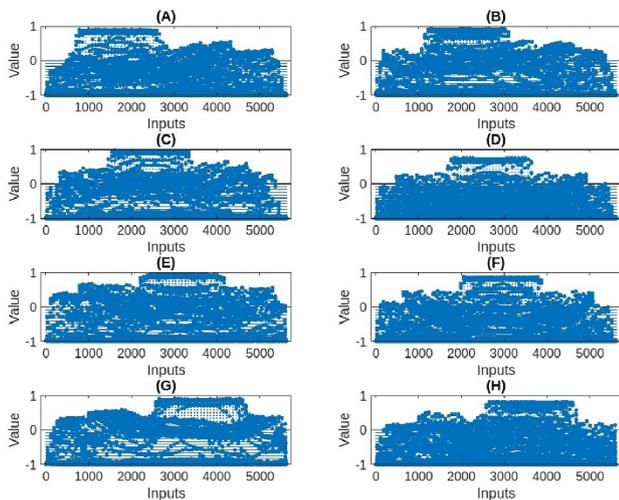


FIGURE 2 | ELM inputs for (A) 4.5, (B) 2.5, (C) 1.5, (D) 3.5, (E) 2, (F) 4, (G) 1, and (H) 3 kHz signals.

evidence inputs and values of the ELMs. According to the digital signal processing, the  $y$ -axis represents the input value to the ELM algorithm, which is normalised from  $-1$  to  $1$  to ensure learning capacity, and the  $x$ -axis the number of the inputs ( $150/2^2 = 5625$  attributes) to the ELM model, which come from the rescaling and vectoring processes. It can be seen that with the digital image processing a class can not be visually distinguished; nevertheless, this will not be the case for ELM methods (see Section 4). In summary, the output nodes represent the source of illumination (the LED class as can be seen in Table 1), while the input to the ELM models are the sampled images as can be seen in Figure 2, which are processed in order to simplify the ELM architecture by maintaining its performance, that come from the image sensors.

### 3.2 | Metrics

ELM methods are evaluated using the VLP database and considering geometric median (G-mean), accuracy, and training time metrics [40]. Obviously, a model is considered efficient for classifying when both the G-mean and the accuracy are close to 1 (100%), and is considered simpler when the training time is shorter.

The G-mean, also called the geometric median, uses the convention of calling classes positive and negative in binary classification problems [41]. In this metric, the successes in the positive and negative classes are considered equally important. To begin,  $TP$  represents the number of positive elements that were correctly identified,  $FP$  are false positives (number of negative elements incorrectly identified as positives),  $TN$  indicates the number of negative elements that were correctly identified, and  $FN$  are false negatives (number of positive elements incorrectly identified as negatives). In Equation (22),  $FP$  (false positives) is the number of LEDs from other classes that were conveyed as LEDs in classes of interest, and  $FN$  (false negatives) are the number of LEDs in classes of interest that were conveyed as LEDs from other classes.

$$G - \text{mean} = \sqrt{\left(\frac{TP}{TP + FN}\right) \times \left(\frac{TN}{TN + FP}\right)}. \quad (22)$$

Nevertheless, the classification application in our work has eight classes, one class for each led. In this situation, the G-mean is given by the following [42]:

$$G - \text{mean} = \left(\prod_{j=1}^{cl} \text{Accuracy}_j\right)^{1/cl}, \quad (23)$$

where  $cl$  denotes the number of classes and  $\text{Accuracy}_j$  comes to be the accuracy in the class  $j$ . In this sense, accuracy refers to the ability to approach to the value of the real wanted magnitude [43], determined by the following equation:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}. \quad (24)$$

This expression represents the ratio between the elements that were correctly identified and the total. It is important to note

that this metric is a general indicator of performance, which means that the success of classifying each class has the same weight. Notice that in our work, each LED is identified by one class, since a LED can be characterised by an OOK communication signal with a particular oscillation frequency or bit-rate (see previous subsection).

Finally, to evaluate the complexity of the neural network, it is essential to measure the training time (TT) in the different versions of the ELM algorithm [44]. Taking into account the underlying theory of ELM, complexity is mainly found when determining the connections between the hidden layer and the output layer. This measure depends on the hardware and software utilised for the classification application. This study used Matlab software to simulate each ELM and the rest of the machine learning approaches to be evaluated. Model training and calculations were performed on a personal computer with the following characteristics: Intel Core i5-10300h 2.5 GHz (8 CPUs) processor, 24 GB of RAM, and Nvidia GeForce GTX 1650Ti.

### 3.3 | Validation Schema

This study used five-fold cross-validation to evaluate system performance (accuracy and G-mean) [45]. Using five-fold cross-validation is a methodology that provides a precise and unbiased evaluation of the classifier performance. The database is divided into five equal parts, each with 20% of the samples. For each division, the artificial neural network is trained with 80% of the data and validations are performed with the remaining 20%. The overall results are presented as the mean of the metrics of the execution of five independent validations of the dataset.

The Monte Carlo simulation technique is also used to analyse the complexity (time training) of the classifier [46]. In this context, a Monte Carlo simulation means repeating the experiment many times by fixing the training and testing sets in order to know statistically reliable results of learning speeds (means and variances). In this form, the mean of the sampling distribution with a minimum deviation standard sampling could be more representative of the time training results. Notice that this metric depends on the hardware and software used in the work, which were presented in previous subsection.

## 4 | Results and Discussion

In this section, the hyperparameters that maximised the accuracy and G-mean metrics without excessive computational cost are initially found. After that, the optimised ELM models in terms of performance and complexity are compared to each other. Finally, the best ELM model is called into question with classical and benchmarking neural networks where confusion matrices are also included and analysed.

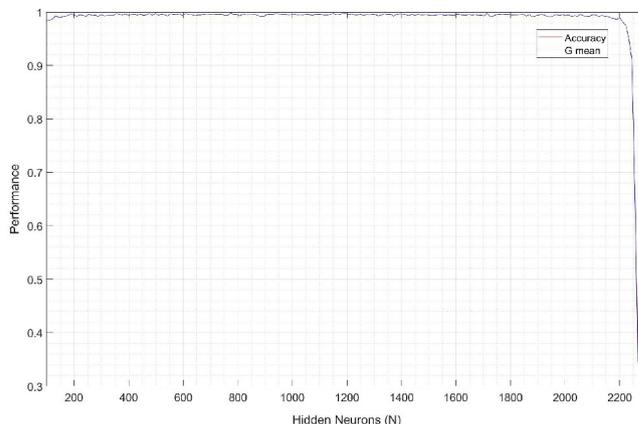
### 4.1 | Hyperparameter Optimisation

It is important to mention that the ELM variants used were implemented with the sigmoid function  $g(x) = 1/[1 - \exp(x)]$

as the mapping function. This choice is based on the proven generalisation of the sigmoid function [47]. The most important results obtained are analysed and divided into six model categories: (1) standard ELM (ELM); (2) regularised ELM (ELM-R); and weighted ELM divided into two parts: (3) weighted ELM 1 (ELM-W1), (4) weighted ELM 2 (ELM-W2), (5) multilayer ELM with 2 hidden layers (ELM-M2), and (6) multilayer ELM with 3 hidden layers (ELM-M3). Note that the optimisation procedure is realised by force brute, namely by proving the system performance in all possible combinations of the hyper-parameters of each ELM model (based on representative vectors) and, then, selecting the best point of the performance (accuracy and G-mean) taking into account the minimisation of the architecture network (training time). Of course, to estimate the optimal hyper-parameters of the ELM approaches, we used the results of the validation sets. Consequently, we could appropriately compare our results with the results reported in the state of the art and determine with models is the best for diverse metrics. For simplicity purposes, the hidden neurons will be identified by only  $N$  in the rest of the paper.

#### 4.1.1 | Standard ELM

This model is where the exclusive evaluation of hidden neurons ( $N$ ) is performed, as shown in Figure 3. The results show optimal performance in the range of 100 to 2300 neurons, exceeding 90% of performance. When the number of hidden neurons match to the number of samples (2270), the performance decreases significantly, overfitting issue always occurs [24]. In this number, the generalised Moore-Penrose becomes simply an inverse matrix. To work with this database, a value of 200 hidden neurons ( $N$ ) is selected, as it is a value that ensures convergence on a stable ELM result (with precision and G-mean greater than 99%), which means that it is not overly complex with respect to the number of neurons (learning speed). On the one hand, we do not choose a value below 200 hidden neurons to avoid the risk of leaving the optimisation zone. On the other hand, we exclude numbers greater than 200 hidden nodes in order to do not increase the computational efforts for specially training phase in the ELM models. Note that for all ELM models, the number of hidden neurons corresponds to a value



**FIGURE 3** | Standard ELM performance: Accuracy and G-mean in terms of the hidden neurons of the standard ELM. The red line corresponds to Accuracy, and the blue line is the G-Mean.

greater than the number of samples in the training stage, since for this the maximum result should occur according to the ELM theory [24].

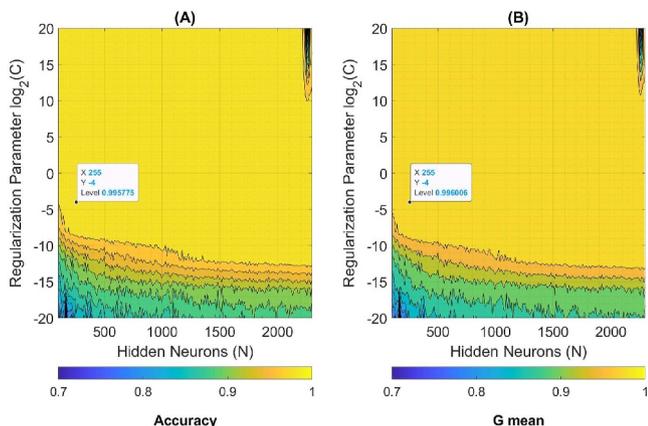
#### 4.1.2 | Regularised ELM

This model requires the evaluation of two hyperparameters: the number of neurons in the hidden layer ( $N$ ) and the regularisation parameter ( $C$ ). In the following, the values of  $C$  will be shown in logarithm scale for better visualisation of the optimal zones. To efficiently visualise performance, the most efficient areas are coloured yellow in the contour graph (refer to Figure 4). If an  $N = 250$  is selected with a  $C = 2^{-4}$ , it is possible to see performances (accuracy and G-mean) near to 99%, as can be identified by the markers, by minimising the computational costs (number of hidden neurons). It is evident that this metric is not affected by the regularisation parameter, see expression (12). For smaller  $C$ s than  $2^{-12}$ , both performance metrics tend to have lower values.

Notice that by comparing the results of ELM-R given by  $C = 2^{20}$  with standard ELM, we numerically confirm that ELM-R has the same behaviour that the standard ELM when  $C$  acquires a big value, see Section 2. In the rest of the contour plots, we adopt a colour bar from 0.7 to 1 in order to visually (easily) compare these.

#### 4.1.3 | Weighted ELM 1

The database used [22] can be considered a balanced one according to Table 1, with a maximum difference of units in over 350 samples by class, therefore, this ELM model (ELM-W1), as well as weighted 2 (ELM-W2), does not show a big difference or improvement in performance when compared to the standard ELM and ELM-R. With the same criteria as above, the chosen values for the ELM 1 imbalance hyperparameters are  $N$  equal to 300 and  $C$  having a value equal to  $2^4$ , refer to the markers of Figure 5. It can be seen that a superior performance demand a  $C$  greater than  $2^5$ . Namely, the regularisation parameter is more



**FIGURE 4** | Performance of regularised ELM (ELM-R). (A) Accuracy and (B) G-mean in terms of the regularisation parameter and number of hidden neurons.

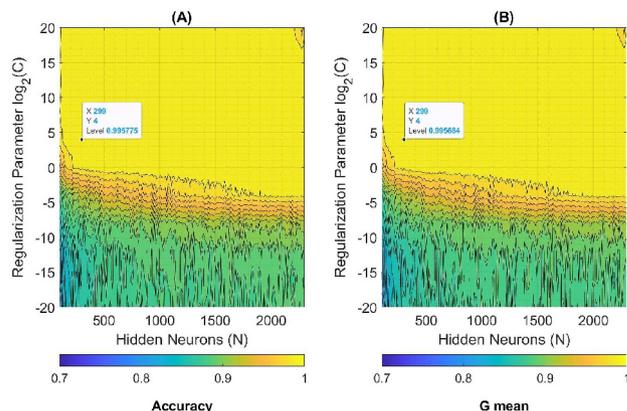
important (the optimal zone is smaller) in the ELM-W1 than in the ELM-R when dataset is balanced.

#### 4.1.4 | Weighted ELM 2

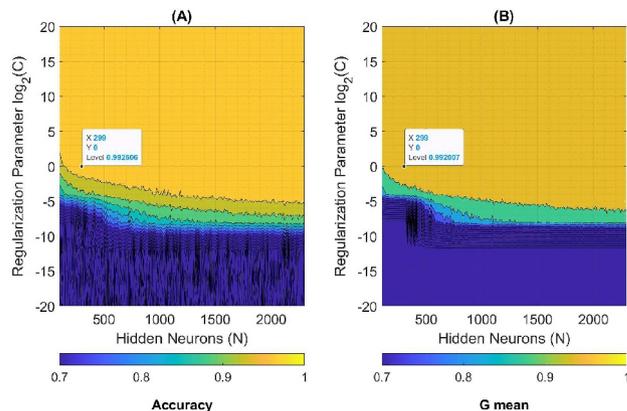
As previously indicated, the database is relatively balanced. Figure 6 with their markers shows that ELM-W2 does not improve its performance. On the contrary, it makes them worse, specially for lower  $C$ s. For comparison purposes in the next section, the selected values for the hyperparameters are  $N = 300$  and  $C = 2^0$ , identified performances by markers once again.

#### 4.1.5 | Multilayer ELM With 2 Hidden Layers

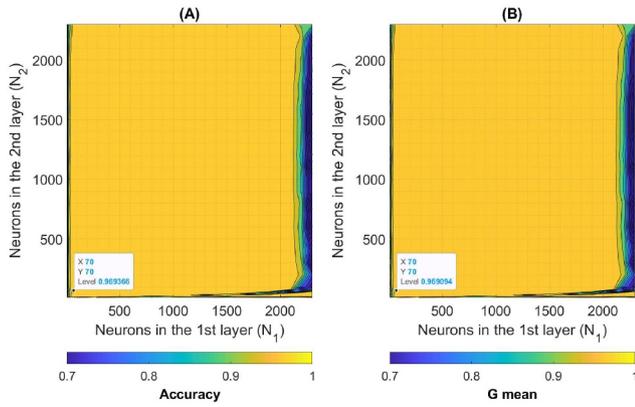
Figure 7 displays the contour plots of (A) accuracy and (B) G-mean metrics in terms of the first and second hidden layers for the ELM-M2. It can be seen that in most combinations the performance tends to be 0.97. In the cases, when the number of



**FIGURE 5** | Performance of weighted ELM 1 (ELM-W1). (A) Accuracy and (B) G-mean as a function of the regularisation parameter and the number of hidden nodes. Here, the yellow area represents the best values for hyperparameters  $N$  and  $C$ , and the blue zones show inferior performances.



**FIGURE 6** | Performance of weighted ELM 2 (ELM-W2). (A) Accuracy and (B) G-mean as a function of the regularisation parameter and the number of hidden nodes.

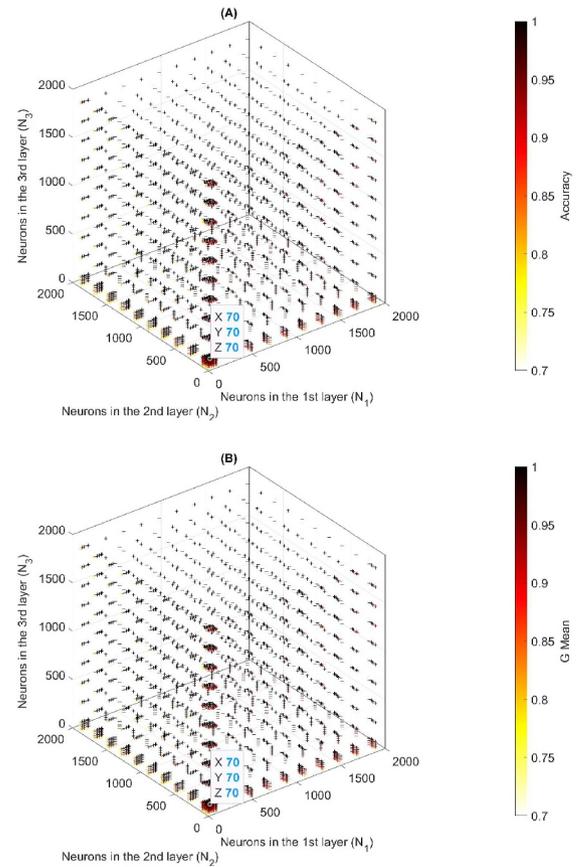


**FIGURE 7** | Performance of multilayer ELM with 2 hidden layers (ELM-M2). (A) Accuracy and (B) G-mean in terms of the number of nodes in the first and second hidden layers.

hidden nodes in the first layer is near to units or bigger than the number of training samples, both metrics exhibit detrimental results. As the accuracy and G-mean are almost 1 in most number of neurons of the first and second layers, subfigures (A) and (B) seem identical, but for low values of these metrics, differences are notorious. With the marker the number of hidden neurons ( $N_1 = 70$  and  $N_2 = 70$ ) that will be considered as optimised hyper-parameters with a minimum computational complexity are indicated.

#### 4.1.6 | Multilayer ELM With 3 Hidden Layers

Finally, (A) accuracy and (B) G-mean of multilayer ELM with 3 hidden layers (ELM-M3) in terms of the number of nodes in the first ( $x$ -axis), second ( $y$ -axis), and third ( $z$ -axis) hidden layers are shown in Figure 8 using a 3-D scatter plot, where performance can be evaluated using a colour scale (best (1) and worst (0.7) results are identified with black and clear colours, respectively). Namely, Figure 8A,B displays the accuracy and G-mean as a function of the of the hidden number of nodes of the multilayer ELM, respectively. It is a graphic (visual) form for representing the optimisation points by considering three parameters at the same time, which is the case of the ELM-M3, where colour to represent the fourth dimension. Consequently, each point comes to be the performance metric in terms of the number neurons in each hidden layer. Note that in the situation of 4 hyper-parameters (for instance a multilayer ELM with 4 hidden layers), a graphical representation of optimisation procedure is not possible to the best author knowledge. While the yellow marks are the worst results, the black crosses come to be the superior trends. Configuration with regular results (near to 0.9) can be identified with red colours. In ELM-M3 occur again the following observations of ELM-M2, superior behaviour (accuracy and G-mean) on most hyper-parameters and, consequently, similar performance of both metrics. For this reason and simplicity purposes (in order to obtain a negligible training time), we adopt the number of three hidden nodes equal to  $N_1 = N_2 = N_3 = 70$ , point identified with a box in subfigures 8 for visualisation purposes. Notice that given a value of  $N_3$  (a cut on the  $z$ -axis), we could represent the optimisation of the



**FIGURE 8** | Performance of multilayer ELM with 3 hidden layers (ELM-M3). (A) Accuracy and (B) G-mean in terms of the number of nodes in the first, second, and third hidden layers.

multilayer ELM with a contour plot. Namely, in the same way that regularised and weighted ELMs. Finally, it can be observed that as the number of degrees of freedom increases (the ELM model has more hyper-parameters), the optimisation task demands more computational efforts. In this sense, there are generally lines, zones (areas), and volumes in the standard ELM, regularised/weighted 1/weighted 2/multilayer with 2 hidden layers ELM, and multilayer with 3 hidden layers ELM, respectively, when the optimisation tests are executed.

## 4.2 | Comparison Among ELM Models

After adjusting the key parameters of the algorithms based on ELM, an exploration of different configurations was made in terms of performance and complexity as illustrated in Table 2. Performance evaluation is performed using general measures such as accuracy and the G-mean. In terms of complexity, training time (expressed in seconds) is used as a metric to understand how efficient the algorithm is during the learning phase and consequently its feasibility in a commercial notebook. All these results are reported for hyper-parameters that maximise accuracy and G-mean and minimise the training time (based on previous section), and are exposed along with their standard deviations in order to see the confidence interval (to choose the best ELM method more objectively).

**TABLE 2** | Performance and complexity (training time, TT) comparison of ELM-based algorithms.

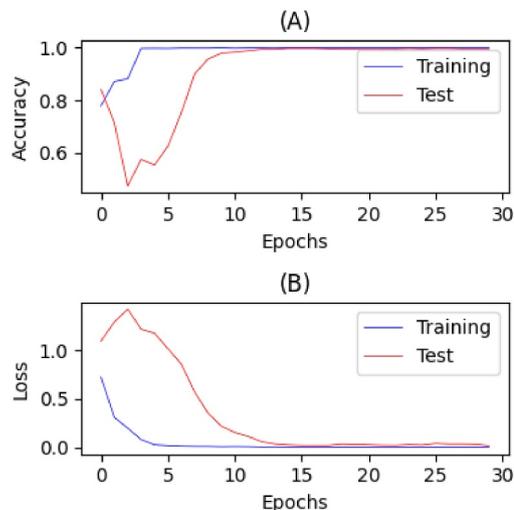
ELM type	Hyperparameters	Accuracy	G-mean	TT
ELM	$N = 200, C =$ does not apply	$0.9938 \pm 0.0037$	$0.9938 \pm 0.0034$	$0.1562 \pm 0.0482$
ELM-R	$N = 250, C = 2^{-4}$	$0.9944 \pm 0.0023$	$0.9943 \pm 0.0022$	$0.2125 \pm 0.0178$
ELM-W1	$N = 300, C = 2^4$	$0.9951 \pm 0.0026$	$0.9950 \pm 0.0024$	$0.3173 \pm 0.0965$
ELM-W2	$N = 300, C = 2^0$	$0.9951 \pm 0.0026$	$0.9950 \pm 0.0024$	$0.4250 \pm 0.0639$
ELM-M2	$N_1 = N_2 = 70$	$0.9725 \pm 0.0048$	$0.9718 \pm 0.0046$	$0.1977 \pm 0.0090$
ELM-M3	$N_1 = N_2 = N_3 = 70$	$0.9708 \pm 0.0057$	$0.9704 \pm 0.0059$	$0.2019 \pm 0.0073$

Table 2 shows that all ELM perform well in terms of both performance metrics (almost 100%) with a negligible deviation standard. The multilayer ELM approaches do not enhance the accuracy or G-mean, namely not necessarily a deep learning approach is better than a machine learning approach. However, the standard ELM shows better results compared to the rest, in terms of shorter training times, compared to ELM-W1 and ELM-W2, which have a much longer training time. Note that for all ELM models, the learning speed is in the order of decimal seconds, which means that these can be considered as machine learning approaches in real-time. At the same time, ELM learning stability (see low deviation standards) is close to 0, which is a natural goal for reliability reasons. As can be seen, in this section we compared diverse architectures of the ELM algorithm by showing their benefits and problems in terms of performance and complexity. In this form, the best model can be chosen in order to contrast it with the reported approaches in the state of the art (discussion of results).

### 4.3 | Discussion of Results

This article focuses on using ELM to tackle the VLP problem. The key parameters of the ELM algorithms were adjusted to achieve a balance between performance and complexity. As mentioned, Table 2 shows a detailed comparison of the different variants of the ELM in terms of accuracy, G-mean, and training time. The standard ELM has the same results as other variants in terms of accuracy and G mean, but with significantly shorter training time.

The results show the standard ELM as the less complex option for a classification task, while the article [22] takes an approach based on CNN for the VLP problem. This research has also performed a CNN simulation in Python according to the parameters described in ref. [22]. The CNN architecture used in this research comprises multiple layers designed for image classification. The network starts with a 2D convolutional layer featuring 32 filters of size  $3 \times 3$ . This layer applies the ReLU activation function and accepts input images of  $150 \times 150$  pixels with three colour channels. It is followed by batch normalisation to stabilise learning and accelerate convergence. Next, a MaxPooling layer of size  $2 \times 2$  reduces spatial dimensionality, and a Dropout layer with a rate of 0.25 mitigates overfitting. The output is then flattened and sent to a fully connected dense layer with 256 units and ReLU activation. This is followed by another batch normalisation and a Dropout of 0.5 to enhance the model’s robustness. Finally, the network includes two additional dense layers: one with 8 units and ReLU activation, and the last



**FIGURE 9** | CNN Results. (A) Accuracy, (B) Loss function. The red line corresponds to testing, and the blue line is training.

with 8 units and softmax activation. This final layer produces the probabilities for each of the classes. The training hyperparameters used correspond to a learning rate of 0.001 (default). This work employs the Adam optimiser. The model trains for 30 epochs with a batch size of 64 images. The result of simulated CNN achieves an accuracy greater than 99%, with 15 training periods as seen in Figure 9A. This is confirmed by the evaluation performed using the sparse categorical crossentropy loss function. This function is suitable for multi-class classification problems where the target labels are integers representing different classes. This loss function measures the discrepancy between the predicted probabilities and the actual class labels. During training, the model iteratively adjusts its weights to minimise this loss. By minimising sparse categorical crossentropy over the specified number of epochs, the model gradually improves its ability to predict the correct class labels for new, unseen data. Figure 9B. As can be noted, ELM and CNN present promising results, but choosing between the two depends on several factors, including the nature of the data and the specific requirements of the application. Standard ELM stands out in terms of accuracy with a low computational cost in training, while CNN also presents very good accuracy results, but has a much more complex and extensive architecture as can be explained and high training time (as can be seen below). Note that the code implemented to obtain the CNN results is available at <https://github.com/RahumadaG/VLP> (accessed on 11 October 2024).

Additionally, a predefined model for MLP was established in MATLAB known as the Scaled Conjugate Gradient (SCG) method [48], which only requires the loading of the dataset for its operation. The SCG method, developed by Møller, is an optimisation technique that minimises objective functions using conjugate directions, eliminating the line search at each iteration, and reducing computational costs. Its adaptability, robustness to noise and outliers, and ability to automatically adjust the step size make it effective for optimising complex functions, including nonlinear and multimodal ones. MLP is composed by two-layer feedforward network with sigmoid hidden neurons and softmax output neurons, suitable for classification tasks. This implementation is done in order to validate and test the ELM-based results with respect to a benchmarking machine learning approach with the same architecture. In this sense, the only hyperparameter is fixed to 200 hidden neurons (identical to the standard ELM). Figure 10 displays the cross entropy according to epochs in train and testing stages. It can be seen that the initial and stopped values correspond to 1.03 and  $5.77 \times 10^{-8}$  in training phase by considering 50 epochs and, hence, for 44 the best validation performance corresponds to 0.0026 where six validation checks were executed. The MLP behaviour achieves testing results equals to 99.6% accuracy (this metric along with G-mean plus/minus standard deviations will be presented in the following paragraph).

To conclude, Table 3 illustrates the comparison in terms of performance and complexity for the superior proposal (ELM standard) and the reported in literature (CNN) [22] and benchmarking ML technique (MLP). It can be seen that all models get a performance near to 1 for accuracy as well as G-mean metrics. Nevertheless, the MLP results exhibit larger standard deviations contrasted with ELM and CNN methods, which means that MLP performances metrics are not so reliable. In terms of training time, the advantage of the ELM standard against the CNN and MLP approaches becomes unquestionable, since it consumes tenths of seconds while the CNN and MLP are

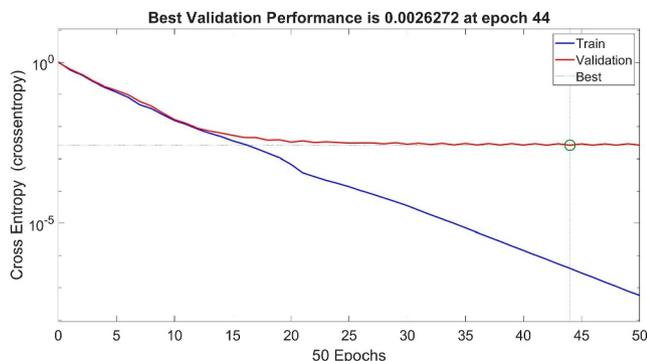


FIGURE 10 | MLP results by showing the cross entropy according to epochs.

TABLE 3 | Comparison in terms of performance and complexity.

Model	Accuracy	G-mean	TT
ELM standard	$0.9938 \pm 0.0037$	$0.9938 \pm 0.0034$	$0.1562 \pm 0.0482$
CNN	$0.9961 \pm 0.0030$	$0.9961 \pm 0.0030$	$1228.2570 \pm 5.5676$
MLP	$0.9951 \pm 0.0290$	$0.9883 \pm 0.0560$	$0.2555 \pm 0.0820$

in order in minutes and seconds, respectively. In other words, the training time by adopting the ELM standard can be considered as negligible. To sum up, the CNN is slightly superior to the standard ELM only in terms of performance (accuracy and G-mean). However, taking into account a complexity analysis, the standard ELM outperforms the CNN by a factor of approximately 1000. While the time training (the metric used to measure the complexity architecture of the algorithm) is 0.1562 s for the ELM, the CNN is characterised by a time training of 1228.2570 s. In VLP practical applications, the training time and, therefore, architecture complexity, is a very important parameter since the optical communications must be in real-time by considering time-variant and/or frequency selective channels [9].

As a complement, the confusion matrices can confirm and see beyond the good performance of ELM standard, CNN, and MLP methods. These are shown in Figures 11–13, respectively, and for simplicity, do not consider the cross-validation scheme that was executed 5 times for the purpose of simplicity. As can be observed, all of the confusion matrices have a good performance (general and by class) of over 99%, with the chosen hyperparameter values. In particular, it can be observed that the class 8 is the most confused by class 4, which could be solved by further separating the frequencies (bit-rates) of the communication part in the VLP system. This is the reason why the VP of the class 8 presents the lowest values among the rest of the classes.

As can be seen in this section, the comparison of the superior ELM approach (standard ELM) with the benchmarking

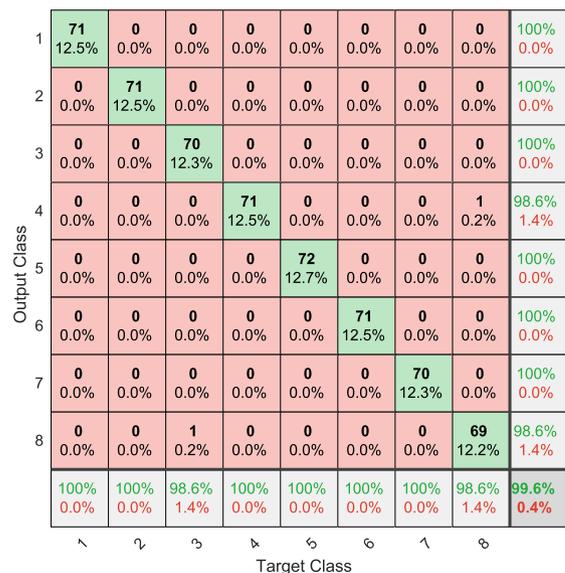


FIGURE 11 | Confusion matrix of the standard ELM given by  $N = 200$ . The diagonal corresponds to the correctly classified examples.

Output Class	1	71 12.5%	0 0.0%	100% 0.0%							
	2	0 0.0%	71 12.5%	0 0.0%	100% 0.0%						
	3	0 0.0%	0 0.0%	71 12.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	4	0 0.0%	0 0.0%	0 0.0%	70 12.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.2%	98.6% 1.4%
	5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	72 12.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	72 12.7%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	70 12.3%	0 0.0%	0 0.0%	100% 0.0%
	8	0 0.0%	70 12.3%	0 0.0%	100% 0.0%						
			100% 0.0%	98.6% 1.4%	99.8% 0.2%						
		1	2	3	4	5	6	7	8		
		Target Class									

**FIGURE 12** | Confusion matrix of the CNN. The diagonal corresponds to the correctly classified examples.

Output Class	1	67 11.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	2	0 0.0%	70 12.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	3	0 0.0%	0 0.0%	64 11.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	4	0 0.0%	0 0.0%	0 0.0%	66 11.6%	1 0.2%	0 0.0%	0 0.0%	0 0.0%	1 0.2%	97.1% 2.9%
	5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	80 14.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	80 14.1%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	73 12.9%	0 0.0%	0 0.0%	100% 0.0%
	8	1 0.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	65 11.4%	0 0.0%	98.5% 1.5%
			98.5% 1.5%	100% 0.0%	100% 0.0%	100% 0.0%	98.8% 1.2%	100% 0.0%	100% 0.0%	98.5% 1.5%	99.5% 0.5%
		1	2	3	4	5	6	7	8		
		Target Class									

**FIGURE 13** | Confusion matrix of the MLP with  $N = 200$ . The diagonal corresponds to the correctly classified examples.

methodologies (CNN and MLP) is carefully executed, by showing not only performance metrics (accuracy and G-mean) but also complexity behaviour (training time). At the same time, confusion matrices are reported by exposing the reasons behind the ML and DL outcomes. As a future research line, this comparison should be considered in the context of positioning issue, by determining the root mean square error in terms of distance from the centre of the experiment space for the diverse VLP methods. Note that the current work is limited to the classification stage.

## 5 | Conclusion and Future Works

This paper has examined various forms of Extreme Learning Machines, including: (1) standard ELM, (2) regularised ELM,

(3) two variations of weighted ELM, and (4) Multilayer ELM with either 2 or 3 hidden layers, as applied to the VLP dataset from [22]. The findings suggest that standard ELM is an excellent option, achieving a performance rate exceeding 99% with a shorter training duration than alternatives such as CNN and MLP. An in-depth analysis of the ELM variants, compared to their outcomes on the VLP dataset, underlines the models' effectiveness, especially in scenarios where class distribution is balanced. The key contribution of this study is to offer a practical framework for selecting the most efficient approach in environments such as indoor VLP, where dealing with unbalanced samples and outliers presents notable challenges. These insights provide important strategies for navigating the complex issues associated with challenging datasets in the VLP domain.

As future works, we considered applied the algorithms based on ELM not only for classification purposes but also for positioning issues for each visible LED luminaire and considering 3D positions for a single optical receiver. Taking into account more datasets is also expected in order to test and validate the exploitation of machine learning approaches in harsh environments such as including industrial facilities, subterranean mines and local greenhouses where scattering and shadowing effects, non-line-of-sight links, among others impairments gain importance. Simultaneously achieving a robust VLC with high spectral efficiency and, finally, a VLP in the order of centimetres could be explored to build a multifunctional 6G wireless network and internet of things applications.

### Author Contributions

All authors contributed to the conception and design of the study. The conceptualisation, methodology, and formal analysis were performed by David Zabala-Blanco, Cesar A. Azurdia-Meza, Benjamin Lobos Soto, and Roberto Ahumada-Garcia. The first draft of the manuscript was written by David Zabala-Blanco, Benjamin Lobos Soto, and Roberto Ahumada-Garcia, and the review and editing of the draft version were carried out by Cesar A. Azurdia-Meza, Ismael Soto, Pablo Palacios Játiva, and Muhammmad Ijaz. Funding was acquired by David Zabala-Blanco. All authors read and approved the final manuscript.

### Acknowledgements

Project FONDECYT Regular 1211132; ANID Vinculación Internacional FOVI240009; Project FONDECYT Iniciación 11240799; STIC-AMSUD AMSUD220026; ANID PFCHA/Beca de Doctorado Nacional/2019 21190489; Escuela de Informática y Telecomunicaciones, Universidad Diego Portales; UDLA Telecommunications Engineering Degree FICA, UDLA; CYTED AgIoT Project (520rt0011); CORFO CoTH2O 'Consorcio de Gestión de Recursos Hídricos para la Macrozona Centro-Sur' (20CTECGH-145896); ANID-Basal AC3E FB0008; and ANID-Subdirección de Capital Humano/Doctorado Nacional/2024-21241043. DICYT Regular 062413SG. ANID Sub-Directorate of Human Capital under the National Doctorate Program, 2024-21241043.

### Conflicts of Interest

The authors declare no conflicts of interest.

### Data Availability Statement

Data available on request from the authors.

## References

1. Y. Almadani, D. Plets, S. Bastiaens, et al., "Visible Light Communications for Industrial Applications—Challenges and Potentials," *Electronics* 9, no. 12 (2020): 2157, <https://doi.org/10.3390/electronics9122157>.
2. P. Palacios Játiva, C. A. Azurdia-Meza, I. Sánchez, et al., "An Enhanced VLC Channel Model for Underground Mining Environments Considering a 3D Dust Particle Distribution Model," *Sensors* 22, no. 7 (2022): 2483, <https://www.mdpi.com/1424-8220/22/7/2483>.
3. P. P. Játiva, C. A. Azurdia-Meza, M. R. Cañizares, I. Sánchez, and D. Iturralde, "On the Performance of Visible Light Communications in Underground Mines," in *2020 IEEE Latin-American Conference on Communications (LATINCOM)* (IEEE, 2020), 1–6.
4. O. I. Younus, N. Chaudhary, Z. Ghassemlooy, et al., "A Unilateral 3D Indoor Positioning System Employing Optical Camera Communications," *IET Optoelectronics* 17, no. 4 (2023): 110–119, <https://doi.org/10.1049/ote2.12094>.
5. M. Rêgo, J. Perez, P. Fonseca, and L. N. Alves, "Frame Recovery Using Multiple Images in Rolling Shutter Based Systems," *IET Optoelectronics* 17, no. 4 (2023): 162–174, <https://doi.org/10.1049/ote2.12100>.
6. D. Iturralde, F. Seguel, I. Soto, C. Azurdia, and S. Khan, "A New VLC System for Localization in Underground Mining Tunnels," *IEEE Latin America Transactions* 15, no. 4 (2017): 581–587, <https://doi.org/10.1109/TLA.2017.7896341>.
7. V. P. Rekkas, L. A. Iliadis, S. P. Sotiroudis, et al., "Artificial Intelligence in Visible Light Positioning for Indoor IoT: A Methodological Review," *IEEE Open Journal of the Communications Society* 4 (2023): 2838–2869, <https://doi.org/10.1109/ojcoms.2023.3327211>.
8. J. A. Apolo, I. N. Osahon, B. Ortega, V. Almenar, J. Tang, and S. Rajbhandari, "Experimental Demonstrations of High-Accuracy 3D/2D Indoor Visible Light Positioning Using Imaging Multiple-Input Multiple-Output Receivers and Artificial Neural Networks," *IET Optoelectronics* 19, no. 1 (2025): e70000, <https://doi.org/10.1049/ote2.70000>.
9. J. Shi, O. Huang, Y. Ha, et al., "Neural Network Equalizer in Visible Light Communication: State of the Art and Future Trends," *Frontiers in Communications and Networks* 3 (2022): 824593, <https://doi.org/10.3389/frcmn.2022.824593>.
10. R. Liu, Z. Liang, K. Yang, and W. Li, "Machine Learning Based Visible Light Indoor Positioning With Single-LED and Single Rotatable Photo Detector," *IEEE Photonics Journal* 14, no. 3 (2022): 1–11, <https://doi.org/10.1109/jphot.2022.3163415>.
11. Z. Zhu, Y. Yang, M. Chen, C. Guo, J. Cheng, and S. Cui, "A Survey on Indoor Visible Light Positioning Systems: Fundamentals, Applications, and Challenges," *IEEE Communications Surveys & Tutorials* (2024): 1, <https://doi.org/10.1109/comst.2024.3471950>.
12. L. He, L. Shi, X. Li, and M. Sun, "Supermarket Commodity Guide System Based on Indoor Visible Light Positioning," in *2022 5th World Symposium on Communication Engineering (WSCE)* (IEEE, 2022), 33–37.
13. A. Madahian, P. A. Ardakani, J. Abouei, A. Mirvakili, A. Mohammadi, and V. Koomson, "A Hybrid VLC/RF Parking Automation System," *IEEE Access* 11 (2023): 66960–66978, <https://doi.org/10.1109/access.2023.3290847>.
14. P. Du, S. Zhang, A. Alphones, and C. Chen, "Faster Deployment for Indoor Visible Light Positioning Using Xgboost Algorithms in Industrial Internet-Of-Things," in *IECON 2021–47th Annual Conference of the IEEE Industrial Electronics Society* (IEEE, 2021), 1–7.
15. P. Singh, H. Jeon, S. Yun, B. W. Kim, and S. Y. Jung, "Vehicle Positioning Based on Optical Camera Communication in V2I Environments," *Computers, Materials & Continua* 72, no. 2 (2022): 2927–2945, <https://doi.org/10.32604/cmc.2022.024180>.
16. R. M. Marè, C. E. Cugnasca, C. L. Marte, and G. Gentile, "Intelligent Transport Systems and Visible Light Communication Applications: An Overview," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)* (IEEE, 2016), 2101–2106.
17. T. H. Do and M. Yoo, "Visible Light Communication-Based Vehicle-To-Vehicle Tracking Using CMOS Camera," *IEEE Access* 7 (2019): 7218–7227, <https://doi.org/10.1109/access.2018.2890435>.
18. T. Lang, Z. Pan, N. Ortiz, et al., "Integrated Design of Low Complexity RSS Based Visible Light Indoor Positioning and Power-Line Communication System for Smart Hospitals," in *2021 IEEE International Conference on Consumer Electronics (ICCE)* (IEEE, 2021), 1–5.
19. P. Chen, M. Pang, D. Che, Y. Yin, D. Hu, and S. Gao, "A Survey on Visible Light Positioning From Software Algorithms to Hardware," *Wireless Communications and Mobile Computing* 2021 (2021): 1–20, <https://doi.org/10.1155/2021/9739577>.
20. W. Raes, N. Knudde, J. De Bruycker, T. Dhaene, and N. Stevens, "Experimental Evaluation of Machine Learning Methods for Robust Received Signal Strength-Based Visible Light Positioning," *Sensors* 20, no. 21 (2020): 6109, <https://doi.org/10.3390/s20216109>.
21. C. D. S. Pereira, *Optical Camera Communications and Machine Learning for Indoor Visible Light Positioning (Master's Thesis)* (Universidade de Aveiro, 2021), <http://hdl.handle.net/10773/33645>.
22. C. Pereira, "Optical Camera Communications and Machine Learning for Indoor Visible Light Positioning," *U Porto Journal of Engineering* 9, no. 4 (2023): 125–143, [https://doi.org/10.24840/2183-6493\\_009-004\\_001955](https://doi.org/10.24840/2183-6493_009-004_001955).
23. B. L. Soto, C. A. Azurdia-Meza, D. Zabala-Blanco, I. Soto, P. P. Játiva, and R. Ahumada-García, "Analysis of a Visible Light Positioning Database in Extreme Learning Machines Applications," in *2024 14th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*, 2024), 330–335.
24. G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme Learning Machine: Theory and Applications," *Neurocomputing* 70, no. 1–3 (2006): 489–501, <https://doi.org/10.1016/j.neucom.2005.12.126>.
25. L. Tobar-Valenzuela, M. Mora, F. Silva-Pavez, I. Torres-Gonzalez, and P. Barría-Valdebenito, "Fast Tuning of Extreme Learning Machine Neural Networks Based With Simple Optimization Algorithms," in *2022 IEEE International Conference on Automation/XXV Congress of the Chilean Association of Automatic Control (ICA-ACCA)* (IEEE, 2022), 1–5.
26. S. Lu, X. Wang, G. Zhang, and X. Zhou, "Effective Algorithms of the Moore-Penrose Inverse Matrices for Extreme Learning Machine," *Intelligent Data Analysis* 19, no. 4 (2015): 743–760, <https://doi.org/10.3233/ida-150743>.
27. G.-B. Huang and H. A. Babri, "Upper Bounds on the Number of Hidden Neurons in Feedforward Networks With Arbitrary Bounded Nonlinear Activation Functions," *IEEE Transactions on Neural Networks* 9, no. 1 (1998): 224–229, <https://doi.org/10.1109/72.655045>.
28. P. Bartlett, "For Valid Generalization the Size of the Weights Is More Important Than the Size of the Network," *Advances in Neural Information Processing Systems* 9 (1996).
29. Penrose R., "A Generalized Inverse for Matrices," in *Mathematical Proceedings of the Cambridge Philosophical Society*, . vol. 51(3). Cambridge University Press; 1955. p. 406–413, <https://doi.org/10.1017/s0305004100030401>.
30. W. Deng, Q. Zheng, and L. Chen, "Regularized Extreme Learning Machine," in *2009 IEEE Symposium on Computational Intelligence and Data Mining* (IEEE, 2009), 389–395, <https://doi.org/10.1109/CIDM.2009.4938676>.
31. Y. X. Wang, J. Lei, and S. E. Fienberg, "Learning With Differential Privacy: Stability, Learnability and the Sufficiency and Necessity of ERM Principle," *Journal of Machine Learning Research* 17, no. 1 (2016): 6353–6392.

32. W. Deng, Q. Zheng, and L. Chen, "Regularized Extreme Learning Machine," in *2009 IEEE Symposium on Computational Intelligence and Data Mining* (IEEE, 2009), 389–395.
33. A. Dreves, F. Facchinei, C. Kanzow, and S. Sagratella, "On the Solution of the KKT Conditions of Generalized Nash Equilibrium Problems," *SIAM Journal on Optimization* 21, no. 3 (2011): 1082–1108, <https://doi.org/10.1137/100817000>.
34. W. Zong, G. Huang, and Y. Chen, "Weighted Extreme Learning Machine for Imbalance Learning," *Neurocomputing* 101 (2013): 229–242, <https://www.sciencedirect.com/science/article/pii/S0925231212006479>.
35. E. Cambria, G. B. Huang, C. Kasun, et al., "Extreme Learning Machines [Trends & Controversies]," *IEEE Intelligent Systems* 28, no. 6 (2013): 30–59, <https://doi.org/10.1109/MIS.2013.140>.
36. J. Tang, C. Deng, and G. B. Huang, "Extreme Learning Machine for Multilayer Perceptron," *IEEE Transactions on Neural Networks and Learning Systems* 27, no. 4 (2016): 809–821, <https://doi.org/10.1109/tnnls.2015.2424995>.
37. D. Zabala-Blanco, M. Mora, C. A. Azurdia-Meza, A. Dehghan Firoozabadi, P. J. Palacios Játiva, and S. Montejo-Sánchez, "Multilayer Extreme Learning Machine as Equalizer in OFDM-Based Radio-Over-Fiber Systems," *IEEE Latin America Transactions* 19, no. 10 (2021): 1790–1797, <https://doi.org/10.1109/tla.2021.9477280>.
38. D. F. Carrera, C. Vargas-Rosales, D. Zabala-Blanco, et al., "Novel Multilayer Extreme Learning Machine as a Massive MIMO Receiver for Millimeter Wave Communications," *IEEE Access* 10 (2022): 58965–58981, <https://doi.org/10.1109/access.2022.3178709>.
39. C. Pereira, P. Fonseca, and L. Alves, "Image Sensor-Based Visible Light Positioning Dataset," *IEEE Dataport* (2023), <https://doi.org/10.21227/9zxs-bm77>.
40. Z. Fayyaz, M. Ebrahimian, D. Nawara, A. Ibrahim, and R. Kashef, "Recommendation Systems: Algorithms, Challenges, Metrics, and Business Opportunities," *Applied Sciences* 10, no. 21 (2020): 7748, <https://doi.org/10.3390/app10217748>.
41. R. P. Espíndola and N. F. Ebecken, "On Extending F-Measure and G-Mean Metrics to Multi-Class Problems," *WIT Transactions on Information and Communication Technologies* 35 (2005).
42. D. Zabala-Blanco, R. Hernández-García, and R. J. Barrientos, "SoftVein-WELM: A Weighted Extreme Learning Machine Model for Soft Biometrics on Palm Vein Images," *Electronics* 12, no. 17 (2023): 3608, <https://www.mdpi.com/2079-9292/12/17/3608>.
43. D. Zabala-Blanco, D. Martínez-Pereira, M. Flores-Calero, J. Datta, and A. D. Firoozabadi, "A-Survey: Identification and Classification of Fingerprints via the Extreme Learning Machine Algorithm," *Inteligencia Artificial* 26, no. 71 (2023): 75–113, <https://doi.org/10.4114/intartif.vol26iss71pp75-113>.
44. K. B. Lassen and W. M. van der Aalst, "Complexity Metrics for Workflow Nets," *Information and Software Technology* 51, no. 3 (2009): 610–626, <https://doi.org/10.1016/j.infsof.2008.08.005>.
45. J. G. Moreno-Torres, J. A. Sáez, and F. Herrera, "Study on the Impact of Partition-Induced Dataset Shift on K-fold Cross-Validation," *IEEE Transactions on Neural Networks and Learning Systems* 23, no. 8 (2012): 1304–1312.
46. D. Luengo, L. Martino, M. Bugallo, V. Elvira, and S. Särkkä, "A Survey of Monte Carlo Methods for Parameter Estimation," *EURASIP Journal on Advances in Signal Processing* 2020, no. 1 (2020): 1–62, <https://doi.org/10.1186/s13634-020-00675-6>.
47. A. D. Rasamoelina, F. Adjailia, and P. Sinčák, "A Review of Activation Function for Artificial Neural Network," in *2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMII)* (IEEE, 2020), 281–286.
48. M. F. Møller, "A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning," *Neural Networks* 6, no. 4 (1993): 525–533, [https://doi.org/10.1016/s0893-6080\(05\)80056-5](https://doi.org/10.1016/s0893-6080(05)80056-5).